

Savitribai Phule Pune University



*A
Project Report*

On

”E-HEALTH CARE FACILITIES”

Submitted by

Ms. AANCHAL OSWAL	B80724239
Ms. VACHANA SHETTY	B80724254
Mr. MUSTAFA BADSHAH	B80724209
Mr. ROHIT PITRE	B80724245

Under the Guidance of

Prof.(Mrs.) MANALI VASHI Mr. SAURABH VAIDYA



K J's Educational Institutes
K J College of Engineering and Management Research
Department of Computer Engineering
2014-2015

K J's Educational Institutes
K J College of Engineering and Management Research
Department of Computer Engineering



CERTIFICATE

This is to certify that Ms. AANCHAL OSWAL, Ms. VACHANA SHETTY, Mr. MUSTAFA BADSHAH and Mr. ROHIT PITRE , of K J College of Engineering and Management Research has submitted the **Project Report** entitled

"E-HEALTH CARE FACILITIES"

They have satisfactorily completed and submitted **Project Report** as prescribed by **Savitribai Phule Pune University** for Fourth Year Engineering (Computer Engineering) for the Academic Year 2014-2015.

Place : Pune

Date :

Prof. Suhas Patil Prof.(Mrs.) Manali Vashi Mr. Saurabh Vaidya
[Project Co-ordinator] [Project Guide] [External Guide]

Prof. Deepak C. Mehetre Dr. Sanjeev J. Wagh
[Head of Department] [Principal]

[External Examiner]

Date :

Acknowledgment

First and foremost, we would like to thank our guides, **Prof.(Mrs.) Manali Vashi** and **Mr. Saurabh Vaidya**, for their guidance and support. We will forever remain grateful for the constant support and guidance extended by our guides, in making this report. Through our many discussions, they helped us to form and solidify ideas. The invaluable discussions we had with them, the penetrating questions they put to us and the constant motivation, has all led to the development of this project. We are also thankful to our family members for encouragement and support.

We wish to express our sincere thanks to the Head of department, **Prof. Deepak C. Mehetre**, as well as our principal **Dr. Sanjeev J. Wagh**, also grateful to our departmental staff members for their support. We would like to thank our friends for listening to our ideas and providing feedback and suggestions for improving those ideas.

Ms. Aanchal Oswal
Ms. Vachana Shetty
Mr. Mustafa Badshah
Mr. Rohit Pitre

K J College of Engineering and Management Research, Pune
May 2015

Abstract

The problems we face these days with our convention systems are missing on the doctor due to lack of an appointment, hampering our work when we lose or damage important medical records, missing out on a crucial dose of our medicine while we are busy in our work, etc. These are few of the major problems each of us face in our hectic and busy life.

This project deals with the development and implementation of a smart-phone application that lets you make a pre-appointment with your doctor, digitally stores all our medical records on a secure cloud server and gives our loved ones peace of mind by reminding us to take our medicine on time. Users can easily make appointments with a doctor of their preference which is suitable to their convenience of time and date. As well as, users can input the symptoms that he/she is suffering from, and will be able to get an idea of the disease what he/she is suffering from. Medical records of the specific user will be uploaded by the doctor or hospital which can be easily accessed and downloaded by the user whenever needed. Using our phones facilities the application can enable reminders to take the specific medicine at the specific time according to our medical prescription which is also uploaded by our doctors. Additional features of calling emergency services, giving location based details of hospitals and medical care facilities and general tips for health care are also incorporated in the application.

Contents

1	Introduction	1
1.1	Overview	1
1.2	Brief Description	2
1.3	Problem Definition, Project Objectives, Goal	2
1.4	Applying software engineering approach	3
2	Company Profile	5
2.1	The Genesis	5
2.2	The Beginning	5
2.3	The Development	6
2.4	The Future Plan	7
2.5	Product File	7
3	Literature Survey	9
3.1	Data mining for disease diagnosis	10
3.1.1	Diabetes	10
3.1.2	Heart disease	10
3.1.3	Cancer	10
3.1.4	Multiple diseases	10
3.2	Feasibility Study	11
3.2.1	Economical Feasibility	11
3.2.2	Legal Feasibility	11
3.2.3	Time Feasibility	11
3.2.4	S/W and H/W Availability	11
4	Project Plan	13
4.1	Project Plan, Schedule and Team Structure	13
5	Software Requirements Specification	17
5.1	Introduction	17
5.1.1	Purpose	17
5.1.2	Project Scope	18
5.1.3	Product Features	18

5.1.4	Use Classes and Characteristics	19
5.1.5	Operating Environment	22
5.1.6	Design and Implementation Constraints	24
5.1.7	Assumptions and Dependencies	24
5.2	System Features	25
5.2.1	System Feature	25
5.3	External Interface Requirements	27
5.3.1	Hardware Interfaces	27
5.3.2	Software Interfaces	27
5.3.3	Communication Interfaces	28
5.4	Non-functional Requirements	29
5.4.1	Performance and Scalability Requirements	29
5.4.2	Safety Requirements	29
5.4.3	Software Quality Attributes	29
5.5	Other Requirements	31
5.5.1	System Hardware Requirements	31
5.5.2	System Software Requirements	31
5.6	Performance Expectations	32
5.7	User and human factors	32
5.8	Physical Environment	32
5.9	Interface Requirements	32
5.10	Security Requirements	32
5.11	Quality Requirements	33
5.11.1	Application Quality (Non-Functional) Requirements	33
5.11.2	Availability	34
5.12	System Analysis Models	35
6	System Design	39
6.1	System Architecture and Business logic	39
6.2	Algorithm	40
6.3	UML Diagrams	42
7	Technical Specifications	45
7.1	Technology details to be used in the project	45
7.2	References to Technology	45
8	Snapshots of Project	46
8.1	Mobile Application	46
8.2	Website	56
9	Conclusion	63

10 References	64
10.1 Papers	64
10.2 Websites	65
Appendices	66
 Appendix A Mathematical Models	66
A.1 Mathematical Model	66
A.1.1 Set Theory	66
A.2 Functional Decomposition of the system	66
A.2.1 Venn Diagram	67
A.2.2 Functional Mapping-1	67
A.2.3 Functional Mapping-2	68
 Appendix B Testing/Reliability of Design	69
B.1 Test Plan for Design	69
B.2 Details of Design Testing	70
B.3 Test Cases	72
 Appendix C Project Management and Project Status	74
C.1 Project Management Approach	74
C.2 Status of Project Work done	74
 Appendix D Papers Published	75
 Appendix E User Manual	76
E.1 Preface	76
E.1.1 About This Guide	76
E.1.2 Audience	76
E.2 System Overview	76
E.3 System Summary	76
E.3.1 User Access Levels	77
E.3.2 Contingencies	77
E.4 Getting Started	77
E.4.1 Installation	77
E.4.2 Signing In	78
E.4.3 Signing Up	78
E.5 System Menu	79
E.5.1 Take Appointment	79
E.5.2 Check Appointment	80
E.5.3 Check Prescription	80
E.5.4 Diagnosis	81
E.5.5 Update Information	82

E.6 Exit System	83
---------------------------	----

List of Figures

1.1	<i>Waterfall Model</i>	4
4.1	<i>Gantt Chart</i>	16
5.1	<i>Class Diagram</i>	20
5.2	<i>Use Case Diagram</i>	21
5.3	<i>Operating Environment 1</i>	22
5.4	<i>Operating Environment 2</i>	23
5.5	<i>Data Flow Diagram - Level 0</i>	35
5.6	<i>Data Flow Diagram - Level 1</i>	36
5.7	<i>Entity-Relationship Diagram</i>	37
5.8	<i>State Transition Diagram</i>	38
6.1	<i>System architecture</i>	39
6.2	<i>Interaction Diagram (Sequence Diagram)</i>	42
6.3	<i>Package Diagram</i>	43
6.4	<i>Component Diagram</i>	44
8.1	<i>Login/Sign up Page</i>	46
8.2	<i>User/Patient Registration Page 1</i>	47
8.3	<i>User/Patient Registration Page 2</i>	47
8.4	<i>Registration Confirmed</i>	48
8.5	<i>Signing In</i>	48
8.6	<i>Startup Page</i>	49
8.7	<i>Select the Speciality/Doctor/Date/Time of your choice</i>	49
8.8	<i>Verify the details before fixing an appointment</i>	50
8.9	<i>Request for the appointment sent</i>	50
8.10	<i>Current Appointment Status</i>	51
8.11	<i>Appointment Status after Confirmation from doctors side</i>	51
8.12	<i>Select Symptoms for Disease Diagnosis</i>	52
8.13	<i>Probable Diseases Diagnosed</i>	52
8.14	<i>Recent Trends</i>	53
8.15	<i>Medicines to be taken; when Prescribed</i>	53

8.16	<i>Time when the Medicines have to be taken</i>	54
8.17	<i>Alert/Reminder for the Medicines to be taken</i>	54
8.18	<i>Update Information</i>	55
8.19	<i>Information Updated</i>	55
8.20	<i>Login/Sign up Page</i>	56
8.21	<i>Sign Up Page</i>	56
8.22	<i>Logging in</i>	57
8.23	<i>Home Page</i>	57
8.24	<i>Appointment Schedule</i>	58
8.25	<i>Appointments Page</i>	58
8.26	<i>Patient Records</i>	59
8.27	<i>Update Information</i>	59
8.28	<i>Appointments List</i>	60
8.29	<i>Confirm/Reject Appointments</i>	60
8.30	<i>Confirmed Appointments</i>	61
8.31	<i>List of Appointments Scheduled</i>	61
8.32	<i>Patient Details</i>	62
8.33	<i>Edit Patient Details/Prescribe Medicines</i>	62
E.1	<i>Sign in</i>	78
E.2	<i>Menu Page</i>	79
E.3	<i>Take Appointment</i>	79
E.4	<i>Status of Appointment</i>	80
E.5	<i>Check Prescription</i>	80
E.6	<i>Select Symptoms</i>	81
E.7	<i>Diagnosis</i>	81
E.8	<i>Recent Trends</i>	82
E.9	<i>Update Information</i>	82
E.10	<i>Information Updated</i>	83
E.11	<i>Menu Page</i>	83

List of Tables

3.1	Comparison of disease diagnosis algorithms	12
4.1	Project Plan and Schedule	13
4.2	Project Management	14
4.3	Task Sheet	14
4.4	Task Sheet(Development)	15
5.1	System Feature: View Hospital Details	25
5.2	System Feature: Take Appointment	25
5.3	System Feature: View Medical History	26
5.4	System Feature: View Prescription	26
5.5	System Feature: Alert/Reminder	26
5.6	Priority	34
C.1	Project Status	74

Chapter 1

Introduction

The healthcare environment is information rich but knowledge poor. Our project deals with the development and implementation of a smart-phone application for electronic health care facilities, which is simple and easy to use. A user(patient) can get a brief idea about what disease is he/she suffering from, just by inputting the symptoms using our smart-phone application. Even users can make a pre-appointment with your doctor, can digitally store all our medical records on a secure cloud server, reminding them to take medicine on time. Users can easily make appointments with a doctor of their preference which is suitable to their convenience of time and date.

1.1 Overview

We propose android mobile application E-health care facilities for various things, such as taking pre-appointment with the doctors of users' choice. User will enter out the symptoms which he is suffering from, and using Data Mining techniques, he will come to know that possibly which disease he has caught. Also there is facility of storing the patient records digitally, so that no tension of losing the paper type records. This will lead to reduce the use of paper; ultimately will lead to save the trees. There is reminder facility for users or patients to take their medicines on time, so that no missing of medicine will cause any fatal problems to the patients. Again there are chances to extend the system, which will provide emergency service for accidental cases, by locating nearest hospitals from the place where incidence is done, even more, nearest hospitals will come to know where accident has occurred and they can provide possible help as soon as possible.

1.2 Brief Description

Basically there are some existing systems which remind us to do anything like in our mobile phones, there is facility to add reminder for any type of work. Our smart-phone android application has somewhat same facility as it reminds patients to take medicines on time, as per the schedule given by respective doctor. No patient is required to add any reminder as like we do in our regular mobile systems, automatic reminding is possible in our system. Again, it will remind the user (patient) for doctors next appointment if any there.

As per the doctors time available, this application will help user (patient) to get appointed to any particular doctor according to users choice. It will use some scheduling algorithms for giving appointments to users; so that it will be convenient for users as well as for doctors. With this android smart-phone application, user is allowed to enter the symptoms what he/she is suffering from, and according to inputted symptoms, the disease what he/she has will be predicted as a result. Though this predicted result will be the actual one, but that will be closer to the actual disease. This facility will help the doctors to predict the actual disease in one go only. For prediction of disease, data mining techniques will be used, as the dataset for this particular function is very vast. Using data mining techniques

Another facility of this application may extend to some emergency cases. In case of accidents, user will be able to locate the nearest hospital or after giving indication by user, nearest hospital will come to know about accident, so that user in hazardous conditions will not be helpless. This facility will be using GPS positioning system for finding the nearest hospital from the place of accident.

1.3 Problem Definition, Project Objectives, Goal

Our Project aims:

1. To develop an android based hospital management system to deal with disease prediction, pre-appointments, reminding and scheduling system.
2. To ease the process of taking appointment.
3. Provide an easy interface to users.
4. Store information securely in data base.

5. To maintain distinct the accounts which can be used by multiple users and would be independent of the devices. The user can log in through any mobile device having the app installed.

1.4 Applying software engineering approach

We are using waterfall model in project. Its details are as follows:

- This is the most crucial phase for the whole project here project team along with the customer makes a detailed list of user requirements. The project team chalks out the functionality and limitations of the software they are developing, in detail.
 - The document which contains all this information is called SRS, and it clearly and unambiguously indicates the requirements. A small amount of top-level analysis and design is also documented. This document is verified and endorsed by the customer before starting the project. SRS serves as the input for further phases.
1. **System Design and Software Design:** Using SRS as input, system design is done. System design included designing of software and hardware i.e. functionality of hardware and software is separated-out. After separation design of software modules is done. The design process translates requirements into representation of the software that can be assessed for quality before generation of code begins. At the same time test plan is prepared, test plan describes the various tests which will be carried out on the system after completion of development.
 2. **Implementation and Unit Testing:** Now that we have system design, code generation begins. Code generation is conversion of design into machine-readable form. If designing of software and system is done well, code generation can be done easily. Software modules are now further divided into units. A unit is a logically separable part of the software. Testing of units can be done separately. In this phase unit testing is done by the developer itself, to ensure that there are no defects.
 3. **Integration and System testing:** Now the units of the software are integrated together and a system is built. So we have complete software at hand which is tested to check if it meets the functional and performance requirements of the customer. Testing is done, as per the steps defined in the test plan, to ensure defined input produces actual results which agree with the required results. A test report is generated which contains test results.

4. **Operation and Maintenance:** Now that we have completed the tested software, we deliver it to the client. His feed-backs are taken and any changes, if required, are made in this phase. This phase goes on till the software is retired

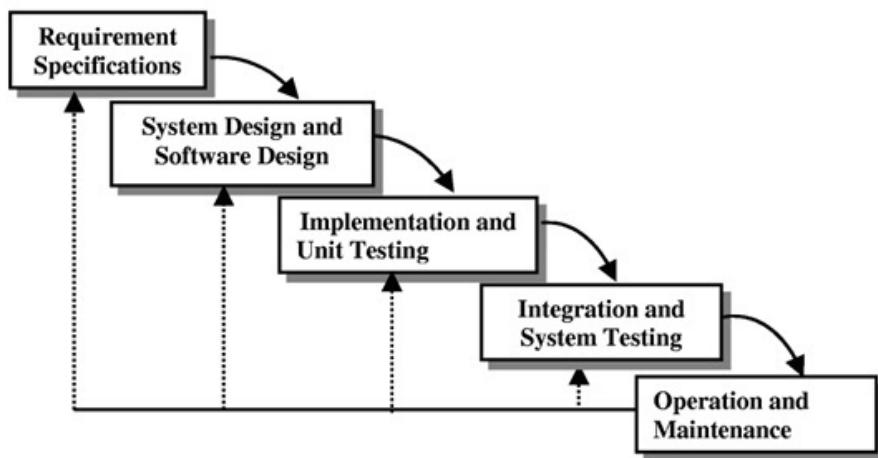


Figure 1.1: *Waterfall Model*

Chapter 2

Company Profile

2.1 The Genesis

For more than two decades, Persistent has been an innovation partner for the world's largest technology brands, leading enterprises and pioneering start-ups. With a global team of more than 6,000 employees, Persistent has 300 customers spread across North America, Europe, and Asia. Today, Persistent focuses on developing best-in-class solutions in four key next-generation technology areas: Computing, Mobility, Analytics and Collaboration, for telecommunications, life sciences, consumer packaged goods, banking and financial services and healthcare verticals.

2.2 The Beginning

Persistent is a global company specializing in software product development and technology services. For more than 20 years, it is partnered closely with pioneering start-ups, leading enterprises and the world's largest technology brands helping them accelerate innovation, time to market and enterprise transformation. Our fine-tuned product development and award winning engineering processes have delivered over 3000 best-in-class solutions for more than 300 technology, telecommunication, life science, healthcare, banking and financial services customers around the world – in just the past five years alone. Anand Deshpande is the Founder, Chairman, and Managing Director of Persistent Systems. Persistent is founded in 1990. Anand is responsible for the overall leadership, strategy, and management of the company. Anand has been the driving force in growing Persistent from start-up to the publicly-traded global company of today. In India, the Company has its Registered Office in Mumbai with branches located at New Delhi, Ahmedabad, Baroda, Pune and Navi Mumbai. **Motto:- "Atithi Devo Bhav."**

2.3 The Development

Product engineering is core to Persistent and has been the foundation of the organization since it was founded in 1990. This coupled with Persistent's work over the years with many leading companies have made Persistent well-equipped to help clients with challenges such as:

- Developing a product on time and within budget.
- Adapt to constantly changing technology and business models.
- Meeting end user expectation by providing the right functionality.
- To discover, develop, manufacture, market products and services to fulfill health care needs of the country.
- To be a leader in the country in the field of Antibiotics.
- To progressively internationalize its operation.

Persistent's product engineering services include

- Development
- Testing & QA
- Porting
- Documentation
- Training
- Sales Support
- Deployment

Persistent's product engineering service provides the following benefits to clients

- Faster time-to-market
- Lower development costs
- Access to experts in the latest technologies
- Access to solutions that integrates next generation technologies to end products

2.4 The Future Plan

To be a well-respected technology company focused on delivering best in class innovative solutions to its customers and partners.

Vision

Global Vision is the world leader in the design and delivery of automated proofreading solutions. Its complete suite of advanced solutions featuring text-based, pixel-based and Braille inspection technologies are designed to eliminate printed artwork, text verification, and text comparison related errors, providing end-to-end security at every stage of the workflow process.

Mission

- Be the dominant Solutions Player in the SMAC space
- Be a company that all stakeholders (customers, shareholders, employees, competitors, society) respect
- Create a Global Brand
- Create a scalable and efficient innovation engine that helps accelerate growth in business

2.5 Product File

Software product companies are facing increased competition while prices for software are decreasing. As a result, companies are having to innovate faster putting pressure on the process of idea creation. Accelerating idea creation process can many times allow product companies to gain a early mover competitive advantage. Persistent helps its customers from the product idea stage through the rapid development of proof of concepts

Clients benefit in the following ways:

- Get access to cutting edge technology and computer scientists
- Speed up development of IP
- Leverage usability engineering as a strategic differentiator
- Reduce risks by ensuring viability of product concept

Persistent Systems, has partnered with Dassault Systems (DS) to deliver best-in-class product development software applications enabling 3D design, modeling, simulation, manufacturing as well as data and process management for better collaboration. As one of the largest Value-Added Resellers (VARs) for DS products in North America, Persistent combines expertise in product development with next-generation technology areas such as , Mobility, BI/Analytics and Social to provide a full portfolio of solutions and services that will help companies compete more effectively.

Persistent Offerings

- Expertise in all stages of product lifecycle under one roof including design, engineering and validation, prototyping and tooling as well as process management
- Leading-edge technical knowledge in addition to unsurpassed industry and product expertise
- Proven work performance that includes fast and effective problem resolution for uninterrupted business operations and increased customer success Greater Return on Investment (ROI) and an increased competitive advantage for our customers
- Lower Total Cost of Ownership (TCO) due to reduced spending and speedy innovation

Chapter 3

Literature Survey

Medical data mining has been a great potential for exploring hidden patterns in data sets of medical domain. In healthcare, though data mining is not widely used but it has become increasingly popular. Data mining can improve decision-making by discovering patterns and trends in large amounts of complex data. There are two primary goals of data mining-prediction and description. Prediction involves some variables or fields in the data set to predict unknown or future values of other variables of interest. On the other hand Description focuses on finding patterns describing the data that can be interpreted by humans.

With the rapid increase in population the number of diseases occurring is also increasing. Many diseases have closely related symptoms which make it difficult for the doctors to predict the exact disease in one go. This is where data mining comes into assistance; it helps in predicting the disease which is nearly accurate. Even though the prediction is not very accurate it at least gives the doctor a brief idea what the disease could be. Thus, in no way disease prediction using data mining is a substitute to doctors whereas; its a compliment to the doctors.

We have done a survey on disease diagnosis research, there are two types of systems are available for this purpose-

- Domain specific which is for doing diagnosis of particular diseases such as system involved with diagnosis of heart diseases only it wont identify any other diseases.
- Second types of system which concentrate on multiple diseases diagnosis.

3.1 Data mining for disease diagnosis

3.1.1 Diabetes

In [1], there is prediction of diabetes using amalgam K-NN with K-means. The inconsistent data from Pima Indian Diabetes Database (PIDD) is identified and corrected using K-means clustering algorithm thereafter K-NN classification algorithm is used to classify the data.

3.1.2 Heart disease

Diagnosis of heart disease is a significant and tedious task in medicine. [3] gives us a simple technique to predict risk of heart attacks. The data classification is based on MAFIA algorithms which result in accuracy. C4.5 algorithm is used as the training algorithm to show rank of heart attack with the decision tree. Finally, the heart disease database is clustered using the K-means clustering algorithm. The results showed that the system is capable of predicting the heart attack successfully.

3.1.3 Cancer

In paper [5] the author has used J48 classification algorithm for detecting breast cancer using a two level diagnosis. In the first level the diagnosis is done on the basis of Wisconsin Breast Cancer Dataset (WBCD); the result obtained from the WBCD is classified into malignant and benign classes. At the second level diagnosis is based on the pathological and physiological parameters of malignant breast cancer dataset and then classified into various types.

3.1.4 Multiple diseases

In [7], for a small set, iterative search is applied, and if the larger data set is there, K-nn algorithm is applied. As complexity of k-nn algorithm is more for Euclidean distance method, LAMSTAR algorithm is used for assigning weights to the elements in dataset to reduce K-nn algorithm complexity.

3.2 Feasibility Study

The feasibility study is major factor which contributes to analysis of system. In earlier stages of S/W development, it is necessary to check whether system is feasible or not. There are 4 aspects of checking feasibility. Detail study was carried out to check workability of proposed system, so the feasibility study is system proposal regarding to its workability, impact on organization, ability to meet user requirements and effective use of resources thus when application progresses, it normally goes through a feasibility study and risk analysis.

Four key considerations are involved in feasibility analysis :

1. Economic Feasibility
2. Legal Feasibility
3. Time Feasibility
4. S/W and H/W Availability

3.2.1 Economical Feasibility

System has been developed using resource S/W. Cost involved in deploying system, such that there should be private network is required. Benefit of this product will definitely outweigh its cost once system has been setup, we will get the reliability function.

3.2.2 Legal Feasibility

We shall be using licensed version of various development tools and software required. Thus this project is legally feasible.

3.2.3 Time Feasibility

There is a lot of learning curve to all the components involved in this project. The project mainly deals with software. The time that will be required to do this project is decided to be 6 months including the planning taken place). The project is timed to be deployable in the last week of March. The total project is feasible in terms of time.

3.2.4 S/W and H/W Availability

The project requires Microsoft Windows based operating system and some software tools that are easily available in the market for sale and it's possible to request it from the college as well.

Ref. No.	Paper	Author names	Algorithms	Limitations	Various measures used
1	An amalgam KNN to predict Diabetes Mellitus	Nirmala Devi.M, Appavu Balamurugan, Swathi U.V	Amalgam of K-means and K-NN	It is lazy learner. There is no thumb rule to determine value of parameter k (Number of nearest neighbours).	Accuracy, sensitivity and specificity
2	Decision Tree Discovery for the Diagnosis of Type II Diabetes	Asma A. Al-Jarullah	J48 Decision Tree Algorithm and C4.5	There are other risk factors that the data collection does not consider. Dataset contains data of only female patients.	Accuracy
3	Predicting the Analysis of Heart Disease Symptoms Using Medicinal Data Mining Methods	V. Manikantan and S. Latha	K-Mean based MAFIA with ID3 and C4.5	Prediction of heart attack using patient prescription is not included.	Accuracy.
4	Classification of Heart Disease Using K- Nearest Neighbor and Genetic Algorithm	M.Akhil jabbar, B.L Deekshatulu, Priti Chandra	KNN and genetic algorithm	(KNN+GA) was not successful for breast cancer and primary tumour. As the k value goes on increasing accuracy of data sets is decreasing.	Accuracy
5	Two level Diagnosis of Breast Cancer using data mining	Rajkumar Gaur Grewal, Babita Pandey	J48 classification algorithm	J48 is not feasible when larger dataset is used, as small change in dataset reflects in larger modification in decision tree.	Sensitivity.
6	Masses Detection Using SVM Classifier Based on Textures Analysis	Fatima Eddoudi, Fakhita Regragui, Abdelhak Mahmoudi and Najib Lamouri	SVM classification based on Haralick vector, Algorithm developed by S.M. Kwok and R. Chandrasekhar	The displacement and the orientation used for the calculation of co-occurrence matrix significantly affect the results. The mixing of two approaches (co-occurrence and contours) just gave satisfactory results	95% of classification rate can be achieved by using pre-segmented mammograms by maxima thresholding.

Table 3.1: Comparison of disease diagnosis algorithms

Chapter 4

Project Plan

4.1 Project Plan, Schedule and Team Structure

Activity	Start Date	Due Date	Milestone	Priority
Architecture & Planning	25/07/2014	15/10/2014	SRS & Prototype	2
Design Phase	01/11/2014	25/12/2014	PSD file	1
Development Phase	20/12/2014	20/03/2015	First version of working app	3
Testing Phase	25/03/2015	15/04/2015	Test Plan & Test case document	4
Deployment Phase	16/04/2015	20/04/2015	Project Completion Doc	5

Table 4.1: Project Plan and Schedule

TASK	Conducted By	Assisted By	Remark By Guide
1) Preparation of abstract and other related documents	Aanchal Vachana	Mustafa Rohit	Good
2) Collection of research papers related to topic.	Aanchal Vachana	Mustafa Rohit	Good
3) On field survey for requirement gathering.	Aanchal Vachana	Mustafa Rohit	Good
4) Preparation of critique on collected research papers.	Aanchal Vachana	Mustafa Rohit	Good
5) Technical research on different technologies and protocols.	Aanchal Rohit	Mustafa Vachana	Good
6) Block diagram for the required system.	Aanchal Rohit	Mustafa Vachana	Good
7) UML diagram development.	Aanchal	Rohit	Good

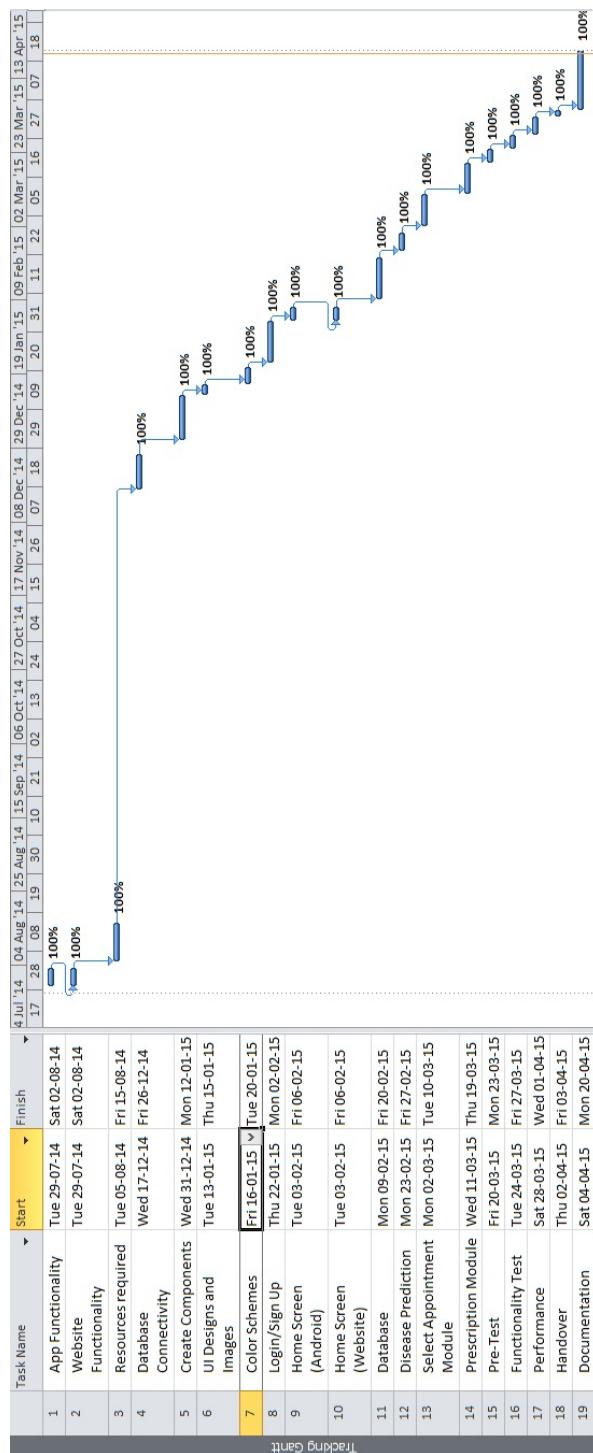
Table 4.2: Project Management

TASK	Start Date	Due Date	Priority
1) Pre-project discussion regarding modules to be developed and concepts to be implemented in planned system.	1/07/2014	11/07/2014	High
2) Detailed discussion on scope of the project and feasibility issues regarding project.	14/07/2014	20/07/2014	Medium
3) Submission of abstract and finalization of topic.	21/07/2014	22/07/2014	High
4) Research on previously developed system based on same technology and collection of research papers.	30/07/2014	10/08/2014	High
5) Collection of some more research papers and study of different technologies that can be implemented.	20/08/2014	5/09/2014	Medium
6) Preparation of critique on different research papers that are being referred.	08/09/2014	10/09/2014	Low
7) Technical research on different protocols and working of different technologies.			High
9) Block diagram and UML diagrams for the system to be developed.	18/10/2014	20/10/2014	High

Table 4.3: Task Sheet

TASK	Start Date	Due Date	Assigned to
App functionality	29-07-14	02-08-14	Rohit
Website Functionality	29-07-14	02-08-14	Aanchal
Resources required	05-08-14	15-08-14	Vachana, Mustafa
Database Connectivity	17-12-14	26-12-14	Rohit, Aanchal
Create Components	31-12-14	12-01-15	Rohit, Aanchal
UI Designs and Images	13-01-15	15-01-15	Vachana, Mustafa
Color Schemes	16-01-15	20-01-15	Aanchal, Vachana
Login/Sign Up	22-01-15	02-02-15	Rohit, Aanchal
Home Screen (Android)	03-02-15	06-02-15	Rohit
Home Screen (Website)	03-02-15	06-02-15	Aanchal
Database	09-02-15	20-02-15	Aanchal
Disease Prediction	23-02-15	27-02-15	Rohit
Select Appointment Module	02-03-15	10-03-15	Rohit
Prescription Module	11-03-15	19-03-15	Aanchal
Pre-Test	20-03-15	23-03-15	Mustafa
Functionality Test	24-03-15	27-03-15	Rohit, Aanchal
Performance	28-03-15	01-04-15	Mustafa
Handover	02-04-15	03-04-15	Aanchal
Documentation	04-04-15	20-04-15	Vachana

Table 4.4: Task Sheet(Development)

Figure 4.1: *Gantt Chart*

Chapter 5

Software Requirements Specification

5.1 Introduction

This project deals with the development and implementation of a smart-phone application that lets you make a pre-appointment with your doctor, digitally stores all our medical records on a secure cloud server and gives our loved ones peace of mind by reminding us to take our medicine on time. Users can easily make appointments with a doctor of their preference which is suitable to their convenience of time and date. As well as, users can input the symptoms that he/she is suffering from, and will be able to get an idea of the disease what he/she is suffering from. Medical records of the specific user will be uploaded by the doctor or hospital which can be easily accessed and downloaded by the user whenever needed. Using our phones facilities the application can enable reminders to take the specific medicine at the specific time according to our medical prescription which is also uploaded by our doctors. Additional features of calling emergency services, giving location based details of hospitals and medical care facilities and general tips for health care are also incorporated in the application.

5.1.1 Purpose

The main aim of our project is:

- To provide easiest way to deal with disease prediction, pre-appointments, reminder and scheduling system.
- To maintain distinct the accounts which can be used by multiple users and would be independent of the devices. The user can log in through any mobile device having the app installed.

- To ease the process of taking appointment.
- Store information securely in data base.
- The core concept for this project is to avoid waiting in the queue to take appointment.

5.1.2 Project Scope

- Proposed application can be used by all and is user friendly.
- This application will allow the user to take appointment, cancel appointment with the doctor of his choice.
- This application will be innovative in its own aspect as it will accommodate the whole process of adding reminders for taking medicines and will minimize the dependency of user on other available options.
- The application will also consist of facility to predict the disease just by entering the symptoms.
- This app would combine a number of functionalities into one, so the user need not download a number of applications for performing different tasks.
- The future scope would be to use the GPS system to deal with some emergency services in case of accidents.

5.1.3 Product Features

- The idea of our app differs as it can be used by multiple users. For example, our app would require the user to create his account which he would access from any android device.
- The data provided by the user in this app during signup would be saved in the database and would be independent of the devices.
- Our app would work like a social networker where the user would be able to create an account and use this app from any device which supports android OS, to book the ticket.

5.1.4 Use Classes and Characteristics

1. List of actors and their details

Actors in this case are the users/patients which would interact with the system to book the tickets. Also the doctor to confirm the Appointment, update schedule, and upload patient history and the hospital just registers the user/patient.

2. Use case description

The main purpose of the use case diagram is to make the user understand the basic functionalities of the particular project. The use case describes how the user uses the application to fix an appointment with the doctor or to check his medical history, etc. For instance, the user should sign up first that is user registration should be done and that information of user will be stored in the database. After user registration, user can perform following operations like fix an appointment with the doctor, check his/her medical history, view the prescription to be taken, set a reminder for his medicines. The doctor here confirms the appointment, updates the daily schedule, uploads the patient history and uploads the prescription to be taken by the patient. With all these we are also providing the delete, update and validate operations on the server side

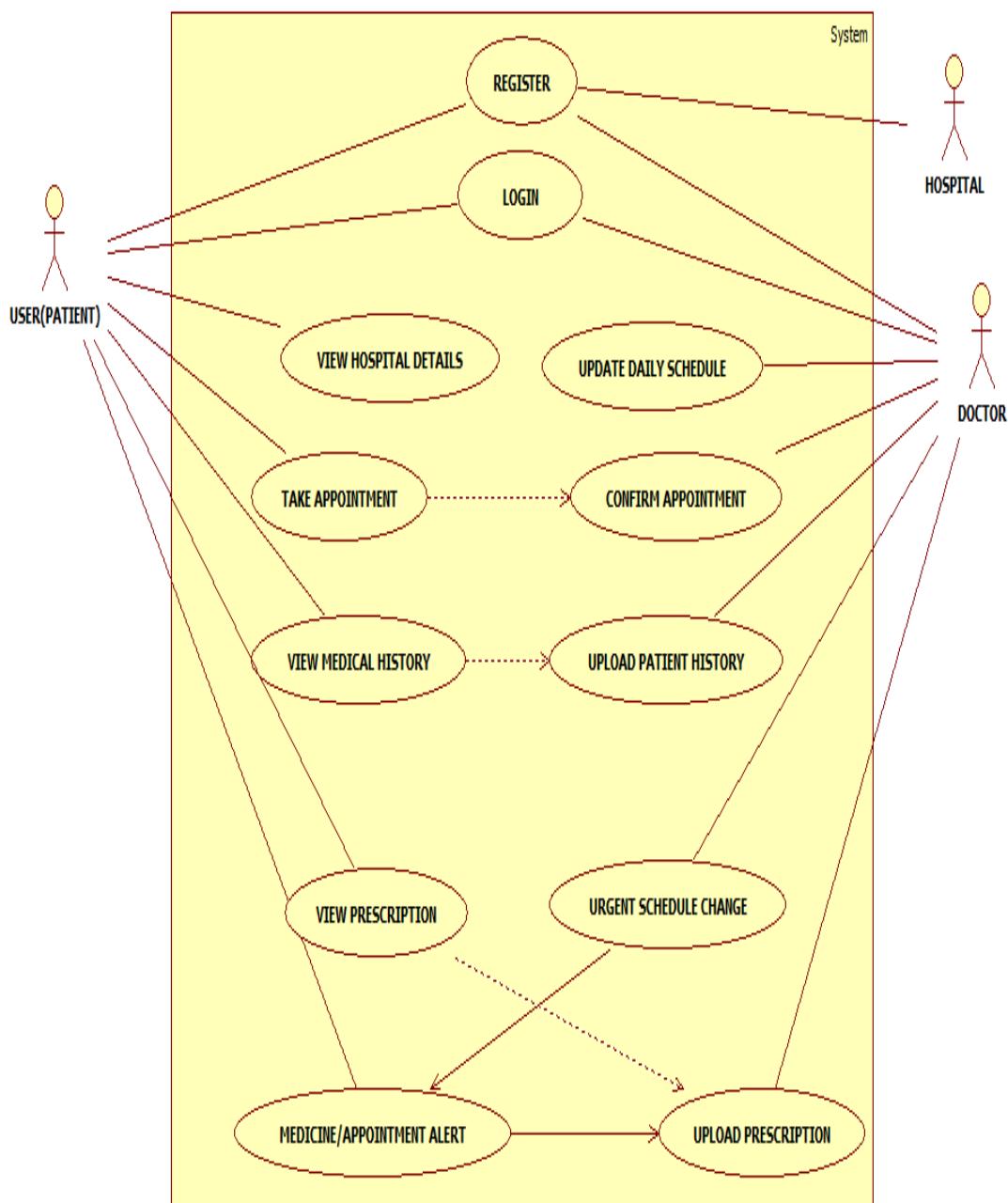
3. Preconditions

- There should be Internet connectivity available.
- The user should have android compatible smart device.
- The application should be installed by the user.
- Servers should be available at the time of request.

• Use Case Suite

1. User To-Do list management

- User registers with the system.
- User logs into the system.
- User views the hospital details.
- User takes an appointment.
- User views his medical history.
- User views the prescription.
- User can set an alert/reminder.
- User can cancel the appointment.
- User logs out from the system.

Figure 5.1: *Class Diagram*

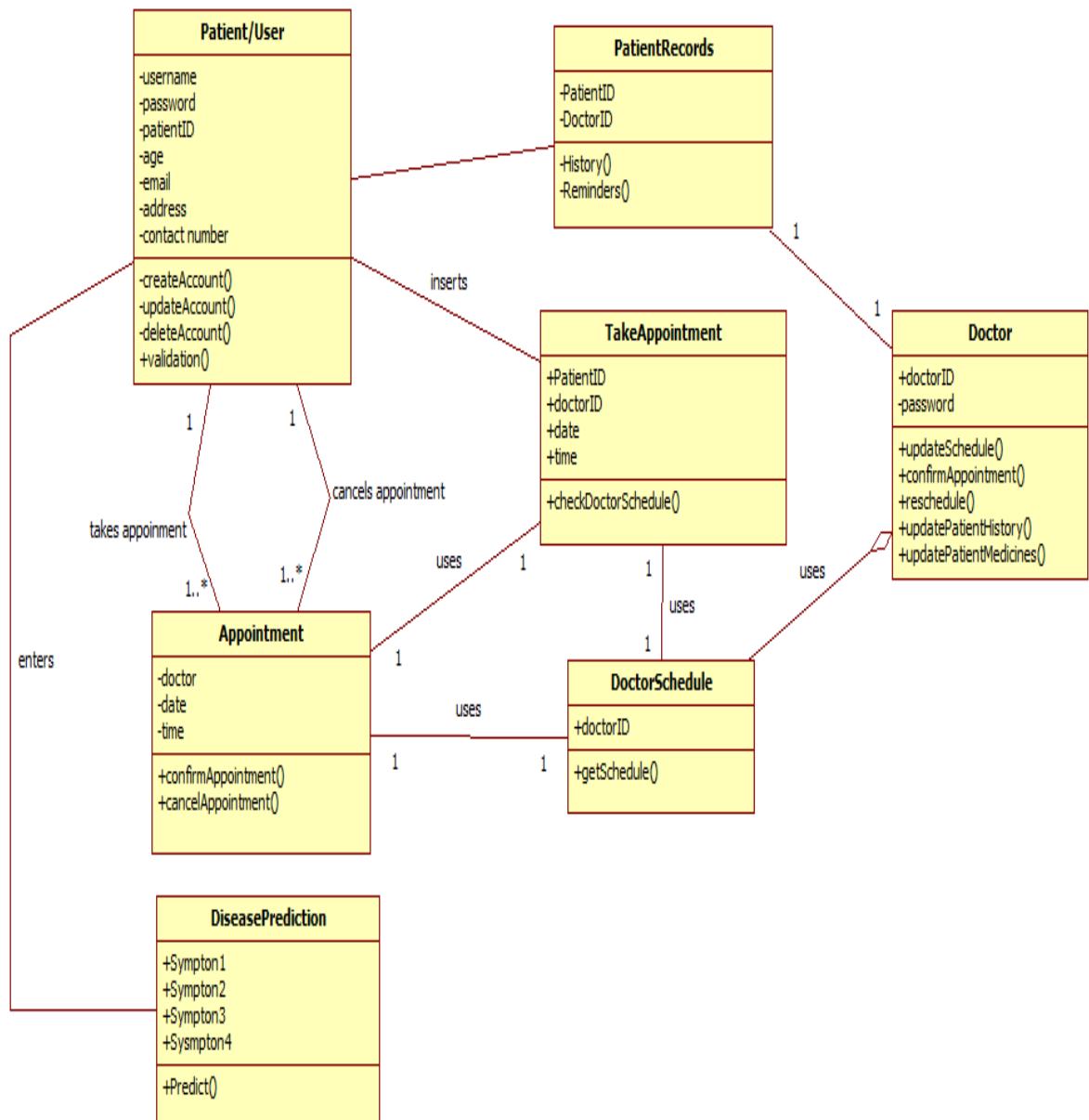


Figure 5.2: Use Case Diagram

2. Hospital To-Do list management

- Registers the patient.

3. Doctor To-Do list management

- Doctor confirms the appointment.
- Doctor updates daily schedule.
- Doctor uploads patient history.
- Doctor makes urgent schedule changes.
- Doctor uploads prescription.

• User Story

User enters his username and password to log in to the system. User fixes an appointment with the doctor. Doctor confirms the appointment and updates the schedule. User sets an alert of this appointment. User logs out of the system.

5.1.5 Operating Environment

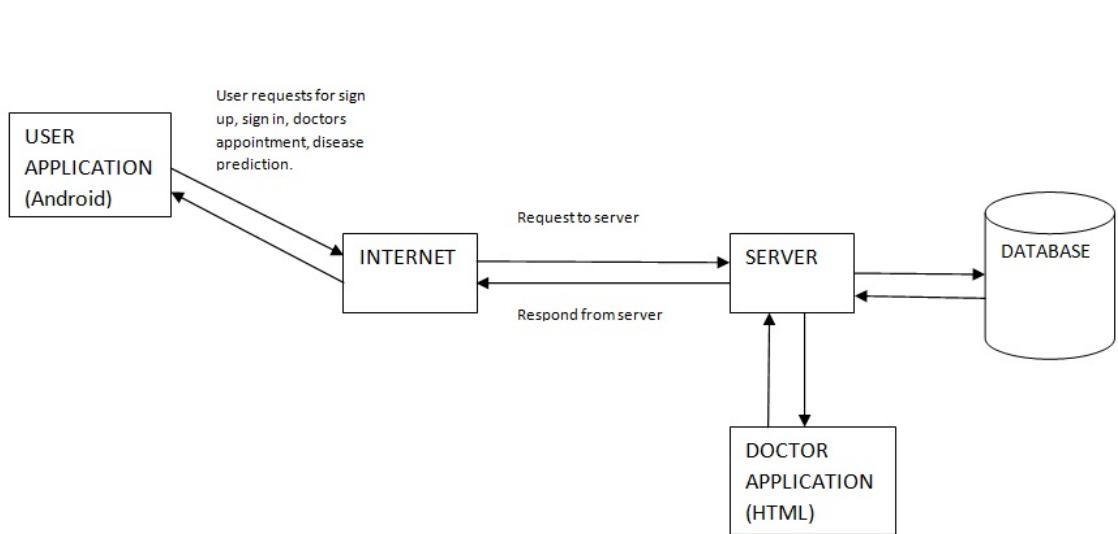
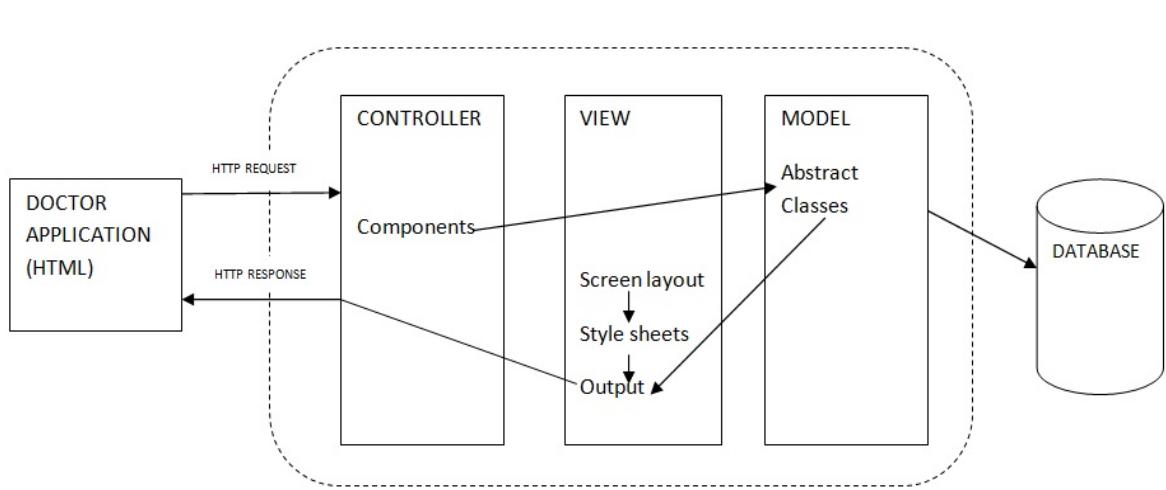


Figure 5.3: *Operating Environment 1*

Figure 5.4: *Operating Environment 2*

Step 1: The work here starts during the first time installation of our application where the user has to sign up. During sign up the patients personal and basic details need to be entered like the first name, last name, date of birth, any genetic disease, or any other health ailments. These details will be gathered and will be stored into the database. Once the user has an account he can sign in directly. Thus the user can use different android phones or the webpage to log in and will not be restricted to only his phone.

Step 2: Once the user registers, he can sign in anytime. The user can check the hospital details and perform various actions. The user can fix an appointment with the doctor of his choice by checking the doctors daily updated schedule. With this schedule he can check for the doctors availability.

Step 3: If the doctor is available, then he acknowledges the request by the user and confirms the appointment. The user gets a confirmation and can set a reminder/alert as to when the appointment is scheduled.

Step 4: The doctor can make urgent schedule changes also which will lead to cancellation or confirmation of the appointment. The doctor can also upload the prescription of the medicines to be taken by the user.

5.1.6 Design and Implementation Constraints

- Regulatory Policies: There are no regulatory policies.
- Hardware limitations: There are no hardware limitations.
- Interfaces to other applications: There shall be no interfaces.
- Parallel operations: There are no parallel operations.
- Audit Functions: There shall be no audit functions.
- Control Functions: There shall be no control functions.
- Higher order Language Functions: SQL shall be used for the database information.
- Signal handshake protocols: There are no signal handshake protocols.
- Reliability Requirements: Total number of bugs in the system shall not exceed 1
- Criticality of application: The server application shall be available 365 days.
- Safety and Security issues: The password and a valid user name are the security issues. Data protection shall be satisfied by the backup process at the server side.

5.1.7 Assumptions and Dependencies

1. The user must have the ability to use the Internet.
2. The user must be connected to the Internet to use the system.
3. The users mobile phone must be equipped with the Android Operating System.
4. The accuracy of information of the users is the responsibility of all users.

5.2 System Features

It consists of the features which includes the following:

1. View Hospital Details
2. Take Appointment
3. View Medical History
4. View Prescription
5. Alert/Reminder

5.2.1 System Feature

ID	<i>View Hospital Details</i>
Description	<i>This includes information related to the hospital and the various doctors practising in that hospital.</i>
Actors	<i>The user will be the actor.</i>
Preconditions	<i>To view the hospital details the user must be registered on the application only then he will be able to use this service.</i>
Basic Steps	<i>The user will select the view hospital details option. The database checks for the details related to the selected hospital and provide the user with the information.</i>
Exceptions	<i>Exception which could arise would be the user selects a hospital which is not available in the systems database.</i>

Table 5.1: System Feature: View Hospital Details

ID	<i>Take Appointment</i>
Description	<i>Here the user takes an appointment with the doctor of his choice after checking that doctors schedule for availability.</i>
Actors	<i>The user will be the actor.</i>
Preconditions	<i>To take an appointment, the user must be registered on the application only then he will be able to use this service. Also the doctor should be available to confirm the appointment.</i>
Basic Steps	<i>The user checks the doctors schedule, checks for availability and then takes an appointment with that doctor if he is available.</i>
Exceptions	<i>Exception occurs if the doctor hasn't updated his schedule then it would lead to clashes with the appointment timings.</i>

Table 5.2: System Feature: Take Appointment

ID	<i>View Medical History</i>
Description	<i>This includes the medical history of the patient which will be uploaded by the doctor.</i>
Actors	<i>The user will be the actor.</i>
Preconditions	<i>To view the medical history, the user must be registered on the application only then he will be able to use this service. Also the medical history has to be uploaded by the doctor.</i>
Basic Steps	<i>After logging in to the application the user can view his medical history which is uploaded by his doctor.</i>
Exceptions	<i>It wouldnt display the medical history if it is not uploaded in the database by the doctor.</i>

Table 5.3: System Feature: View Medical History

ID	<i>View Prescription</i>
Description	<i>The medicines prescribed by the users doctor are displayed here.</i>
Actors	<i>The user will be the actor.</i>
Preconditions	<i>To view the prescription the user must be registered on the application only then he will be able to use this service. Also the doctor should have prescribed some medicines to the patient.</i>
Basic Steps	<i>The user selects the view prescription option and checks what medicines the doctor has prescribed for him.</i>
Exceptions	<i>If the doctor has not prescribed any medicines for the user then it will not display anything.</i>

Table 5.4: System Feature: View Prescription

ID	<i>Alert/Reminder</i>
Description	<i>The user can set a reminder/alert as to when his next appointment is or what time he has to take his medicines.</i>
Actors	<i>The user or will be the actor.</i>
Preconditions	<i>To set an alert/reminder the user must be registered on the application only then he will be able to use this service. Also either the user must have an appointment scheduled or some medicines prescribed to set an alert for the above.</i>
Basic Steps	<i>Once the users appointment is confirmed he can set a reminder to ensure that he doesnt miss it. Similarly he can set an alert for his medicine ingestion.</i>
Exceptions	<i>If the user has no appointment scheduled or no medicines prescribed he cannot use this service.</i>

Table 5.5: System Feature: Alert/Reminder

5.3 External Interface Requirements

E-health care facility system provides a group of works with interface environments. Also there will be a database which will keep all the records of the user while visiting the page.

5.3.1 Hardware Interfaces

There is no need of any hardware interface for this System.

5.3.2 Software Interfaces

1. Two product options for viewing

A. Name: Android device

Version number: Android GingerBread or later versions.

Source: Google inc.

Purpose: The operating system specified above is required as the container of the client software at the client site in order to execute the client site of User interface.

Definition of the interface: Android is a smart phone OS which provides an interface to work on any smartphone.

B. Name: BlueStacks

Version Number: - BlueStacks version 0.7.7.813

Source: - Opensource Application developed by Silicon Valley-based software company BlueStacks

Purpose: The software specified above enables users to use this app on desktops and computers that are not equipped with Android.

Definition of the interface: BlueStacks is a software that simulates an Android OS.

2. Name: Apache HTTP Server

Version Number: 2.0.5.5

Source: The Apache Software Foundation.

Purpose: In order to execute the client site part, the web server specified above is required as the provider of the client software at the server site.

Definition of the Interface: The Apache Server Project is an effort to develop and maintain an open source server for modern OS. The goal of this project is to provide a secure, efficient and extensible server that provides services in sync with the current standards.

3. Name: MySQL

Version Number: 5.0

Source: Oracle Corporation.

Purpose: Required as database server.

Definition of the interface: MySQL is the worlds most popular open source DB software. With superior speed, and ease of use, MySQL has become the preferred choice of corporate IT managers because it eliminates the major problems associated with downtime, maintenance, administration and support.

5.3.3 Communication Interfaces

The default communication protocol for data transmission between the server and the client is the Transmission Control Protocol/ Internet Protocol (TCP/IP). At the upper level, Hyper Text Transfer Protocol (HTTP, default port=80, default of Apache port=8080) will be used for communication between the web server and the client.

5.4 Non-functional Requirements

5.4.1 Performance and Scalability Requirements

The application can sustain long periods of continuous usage by one user or multiple users. The database that will be accessible by the server will hold tables, tables that will contain information about the users, detailed information about locations and fares. Authentication information like username and password needs to be stored. The database will grow as number of users increase. The scalability requirements of the system are another important issue as well as the performance requirements. The application should work efficiently with 1 thousand users approximately using the application simultaneously. The system will have ability to provide all users with efficient support, which will not be broken down.

5.4.2 Safety Requirements

Any safety problem will not take place throughout the lifecycle of the software system. Every data can be accessed and seen just after data entrance. Safety factors will be supplied through:

- Physical server security.
- Disaster recovery plan.
- Back up of data.

5.4.3 Software Quality Attributes

1. Usability

This project deals with the development and implementation of a smart-phone application for E-Health care facilities. To use this application we need to use a smart phone with android as operating system.

2. Maintainability and Upgradeability

Making changes or upgradeability in the system will not be that much difficult. By having some knowledge of programming, some features of the system might be converted to a new version. According to the needs of upgrade, system requirements might change such as change in operating system or not.

3. Supportability and Operateability

Supportability will be provided over the whole product life of the system. System will be quite easy to use but educational support will be given if needed. The application is a multi-user web based app, can

be run on every smart phone (Android) and Internet connection has to be established before using this application. The user is expected to be comfortable using android OS and have basic knowledge of English.

4. Business Lifecycle

The application is designed for everyone. It can be used by business professionals or students. Hence, this system is feasible for a range group of business and great number of people in any sector. Some innovations in the system may be performed and can have a greater range of business life.

5.5 Other Requirements

5.5.1 System Hardware Requirements

- Smart Phone (Android).
- A server to process all the functions having high power and multiple core processors.

5.5.2 System Software Requirements

1. Language: Java J2SE and JDK

J2SE (Java 2 Standard Edition) Java would be the required language for development of the project. JDK is the development kit used to compile java programs.

2. IDE: Eclipse, SDK

Just like visual studio provides development environment for VB and .Net, Eclipse provides an integrated development environment (IDE) for Java.

3. Database, Data Library

Serialized Objects / Serialization - Database in Java In case the project needs database this is how it is handled in java.

- First step is to use data structures like Vectors and Lists. These come under Java Collections API.
- Secondly we declare our own classes using these data structures. E.g. a class Student to hold all the student information. Now these classes need to be pre-compiled and called within Java application as libraries. This is called as a Java Class Library
- Now class objects cannot be saved to hard drive directly. We need to convert these objects to bytes so that they can be saved to hard drive. To do this we must use a concept called as Serialization. Basically it is a concept where in objects are converted to byte streams so that they can be saved to hard drive or sent via internet and vice versa. The reverse process is called as deSerialization.
- Finally to save these bytes to hard drive or to send them via network we need Java I/O.

4. GUI

AWT and SWING are used for GUI design.

5. Computing Architecture

In order to implement a architecture or a Software As A Service (SaaS architecture) we need Web Service we need to implement a web

service. GlassFish Server to host web service SOAP API to be able to call web service at client side we need to use SOAP API or even XML.

6. Rich Client Side Applications

When implementing client-server applications or even based applications, the client side applications can be implemented using architecture of java called as (JWS) Java Web Start. This allows us to create applications with rich GUIs which are also called as Rich Internet Applications (RIA). These are smarter than implementing conventional web pages.

5.6 Performance Expectations

The application can sustain long periods of continuous usage by one user or multiple users. It will load the home page quickly within 2 seconds. The database that will be accessible by the server will hold tables, tables that will contain information about the users, detailed information about locations and fares. The database will grow as number of users increase.

5.7 User and human factors

The user is expected to be comfortable using a smart phone with android OS, Internet and have basic knowledge of English

5.8 Physical Environment

The application will be a multi user application. The user can use this application once downloaded from hosted site. The Internet connection should be established before using this application.

5.9 Interface Requirements

No, we do not need an interface with any other system for it to function properly.

5.10 Security Requirements

Security breaches on these kinds of application and are of a major concern because it can involve both enterprise information and private

customer data. Security includes processes for authentication, authorization and information handling. Building security into the app from the beginning can be more effective and less disruptive in the long run. In our application, security is required for maintaining privacy and preventing illegal use so that the authentic user should be able to access his own account. This is provided by authenticating username and password. If the user enters the wrong password three times then his account will be kept on hold and he won't be able to access it for 24hrs.

5.11 Quality Requirements

5.11.1 Application Quality (Non-Functional) Requirements

The application will be user friendly so the user will be able to use it efficiently. Also it will work efficiently on all android platforms. The system is expected to sustain long periods of continuous usage by one user or multiple users. The app will provide good security to the user so that the user feels secure using this app and providing information. The database will grow as number of users increase. It will load the home page quickly within 2 seconds.

5.11.2 Availability

The application can sustain long period of continuous usage by one or multiple users.

Priority

Quality Parameter	Priority	Statement of Req.	Yes/No
Compatibility	1	Should be executed in expected time	Yes
Completeness	1	Expected i/p should get relative and complete o/p	Yes
Correctness	2	Conversion should be efficient	Yes
Cost of ownership	1	Should be below 1000	Yes
Environmental	1	Should not contribute to excess use of environmental harming	Yes
Extensibility	1	Advancement acceptable	Yes
Installation Complexity	1	Easy installable in local machine	Yes
Parallel Processing	3	Concept of multitasking is utilized	Yes
Performance	1	90per. of appl. should be converted	Yes
Portability	1	Should be installable in various common platform	Yes
Regulatory	1	Supervise proper seq. execution	Yes
Reusability	1	Main purpose is reusability	Yes
Scalability	1	Should be expandable in features	Yes
Security	1	Not accessible to external world so is safe	Yes
Time to Market	1	Should be published in Apple store or android store by completion of this year	Yes
Training Complexity	1	Should be easy to learn and maintain	Yes
Usability	1	Handy to use and execute	Yes

Table 5.6: Priority

5.12 System Analysis Models

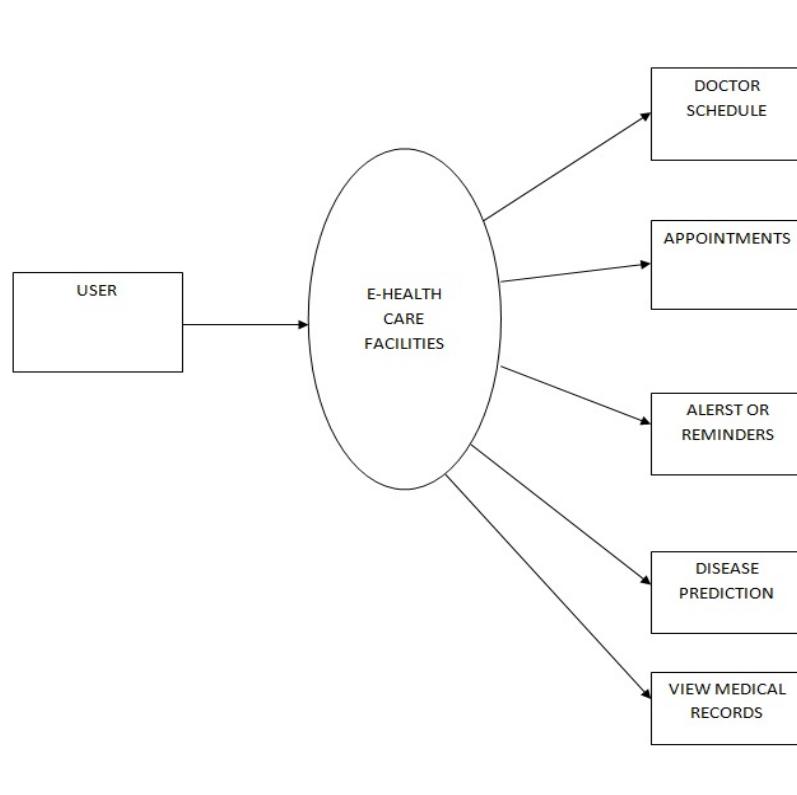


Figure 5.5: *Data Flow Diagram - Level 0*

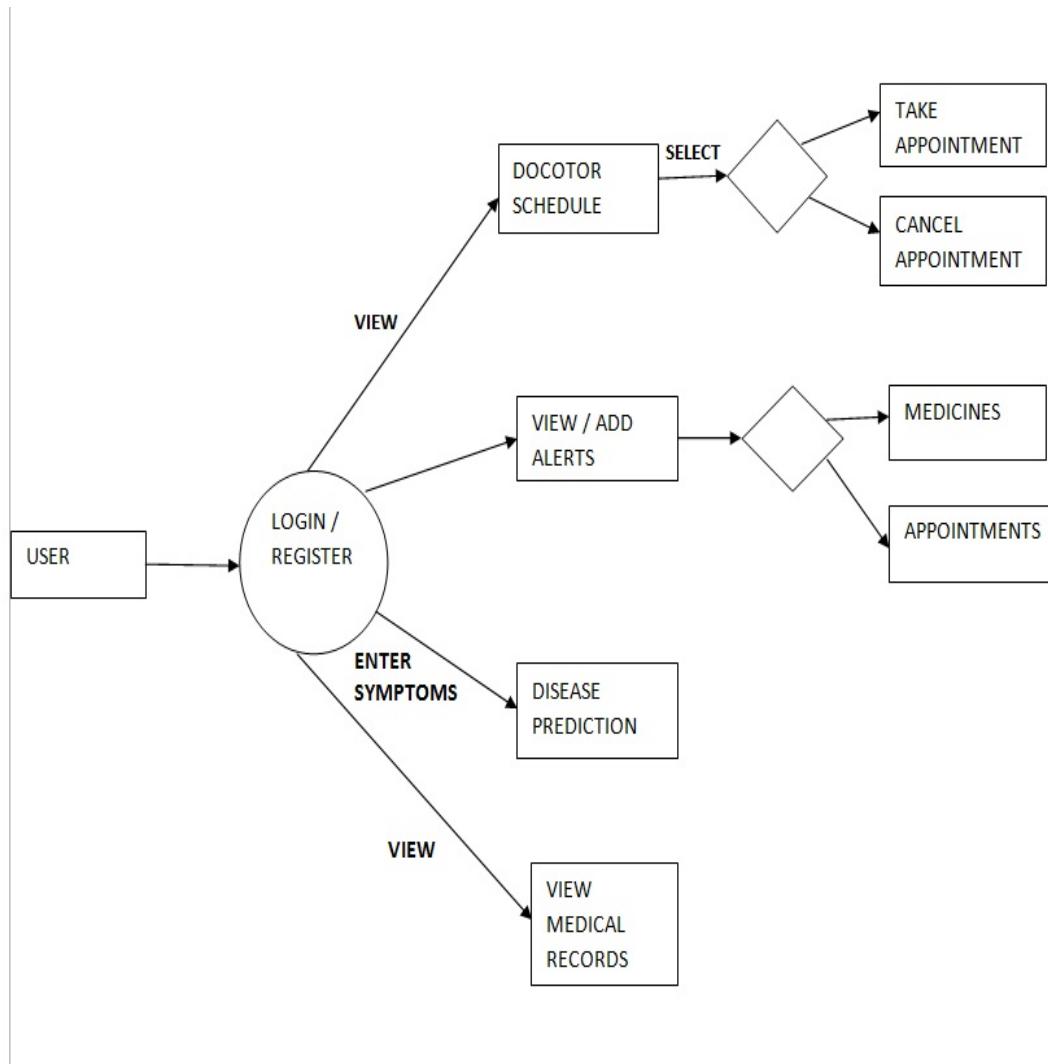


Figure 5.6: Data Flow Diagram - Level 1

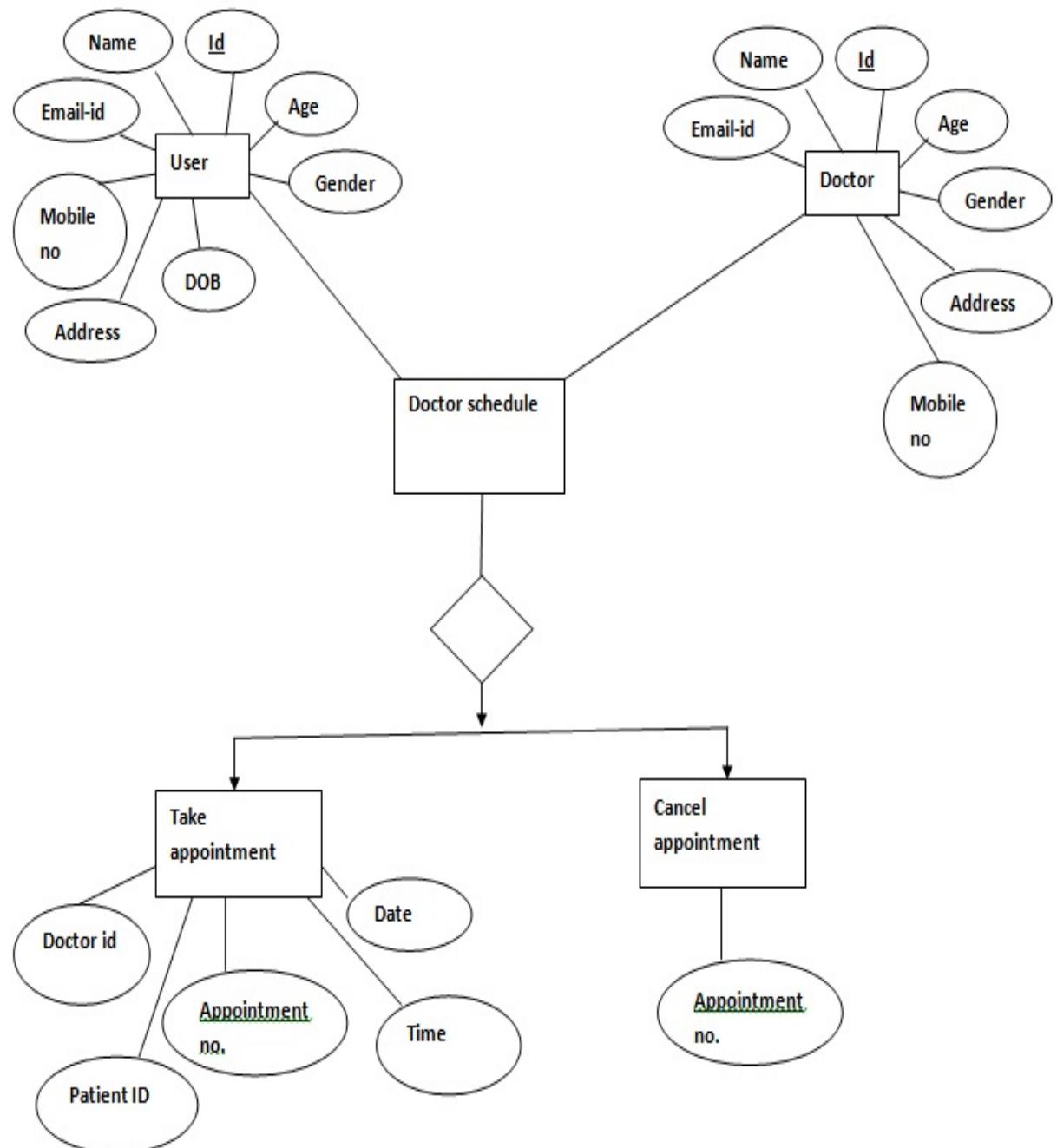
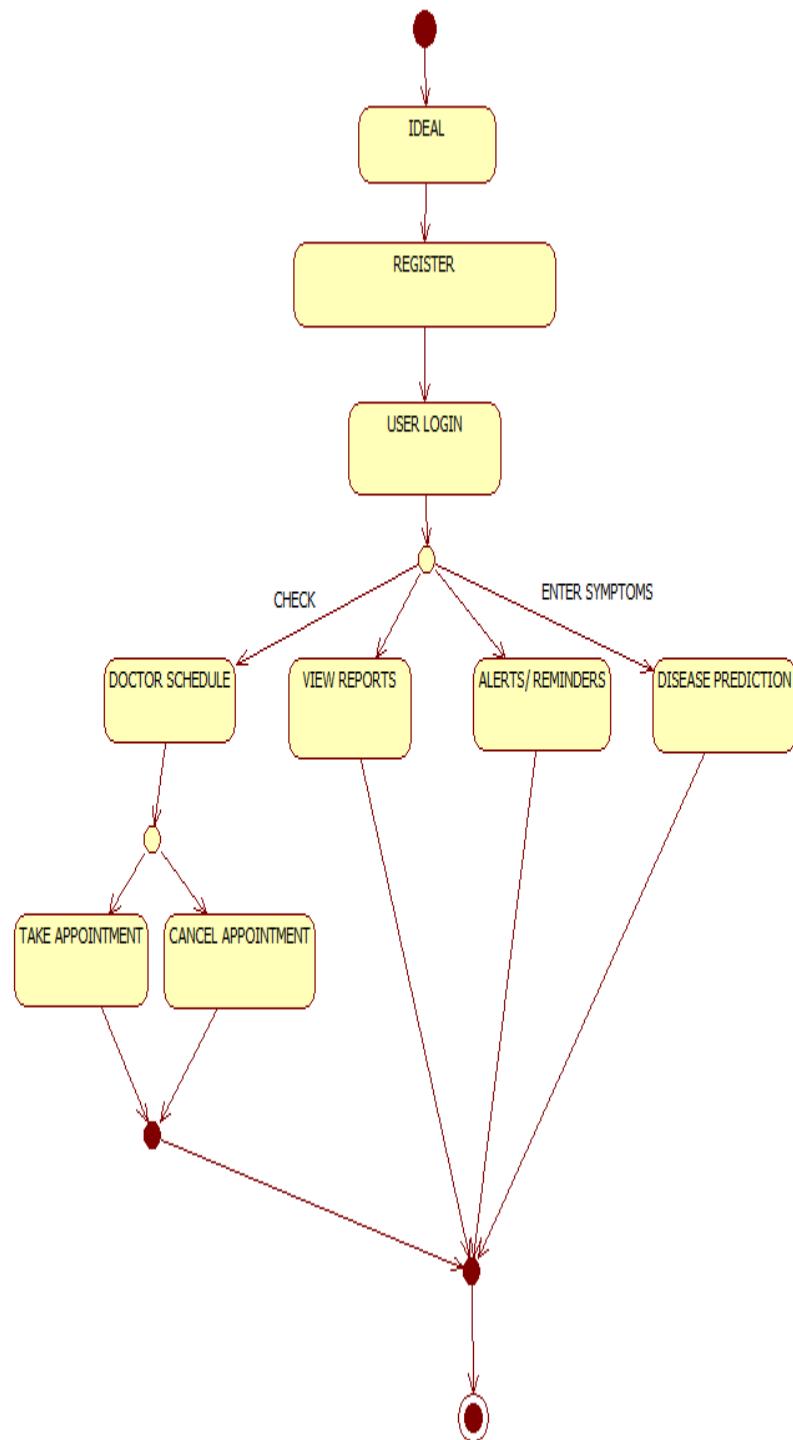


Figure 5.7: Entity-Relationship Diagram

Figure 5.8: *State Transition Diagram*

Chapter 6

System Design

6.1 System Architecture and Business logic

The structure of system divided into two components:

1. The user application which considers User requests for sign up, sign in, doctors appointment, disease prediction, personal information gathering,etc.
2. The doctor application which considers users' request for appointment, scheduling the appointments according to doctor's convinience, storing doctors' personal data along with hospital information, etc.

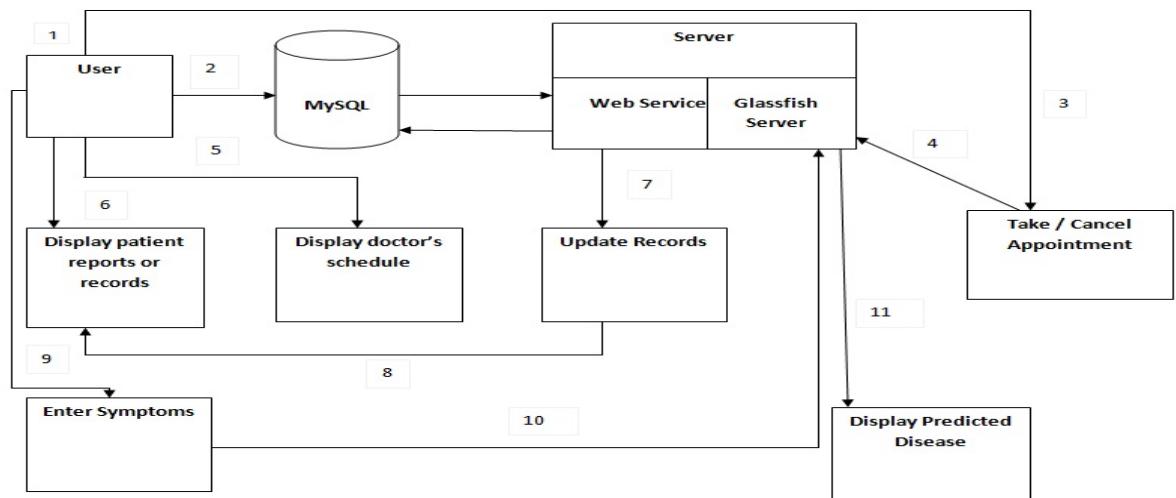


Figure 6.1: *System architecture*

6.2 Algorithm

This algorithm gives the probability for various diseases as per the symptoms entered by the user. The probability of each disease is calculated using the number of symptoms entered by the user and the total number of symptoms for the corresponding disease.

ALGORITHM

Input: Multiple Symptoms from the user (S_1, S_2, \dots, S_n)

Output: Probability of various diseases.

Steps:

1. Start
2. $\text{probability}[] = 0$
3. $\text{counter}[] = 0$
4. $\text{disease}[] = 0$
5. If symptoms == NULL the goto step 16
6. Map all the disease ids to $D[]$ if Symptom (S_i) is in D_i
7. Find the max value of $\text{disease}[]$ i.e the disease id
8. Increment counter for each D_i
 - a. For each entry in D_i
 - i. $\text{counter}[\text{disease}[i]] = \text{counter}[\text{disease}[i]] + 1$
9. Get the total disease weights for all the diseases in $\text{disease}[]$ in $d_total[]$
10. for $i = 0$ to $\text{counter}[].\text{length}$
 - a. If $\text{counter} != 0$
 - i. $\text{probability}[i] = (\text{counter}[i] * d_total[i]) * 100$
 - ii. $\text{probability_id}[i] = i$

- 11.** Display all the probable diseases in decreasing order of their probabilities
- 12.** Find out the disease with the highest probability.
- 13.** Store the disease id with the current month in the recent history table.
- 14.** For the diseases predicted check if any exists in the recent history table
 - a.** If yes
 - i.** Count the number of iterations of each disease predicted in current month.
 - ii.** Get the disease id with the max no of occurrences, fetch the symptoms of the recent trend's disease which are not entered by user and display the disease name and ask the user whether suffering from the remaining symptoms of the recent trend disease.
- 15.** Error: Enter Correct Symptoms.

6.3 UML Diagrams

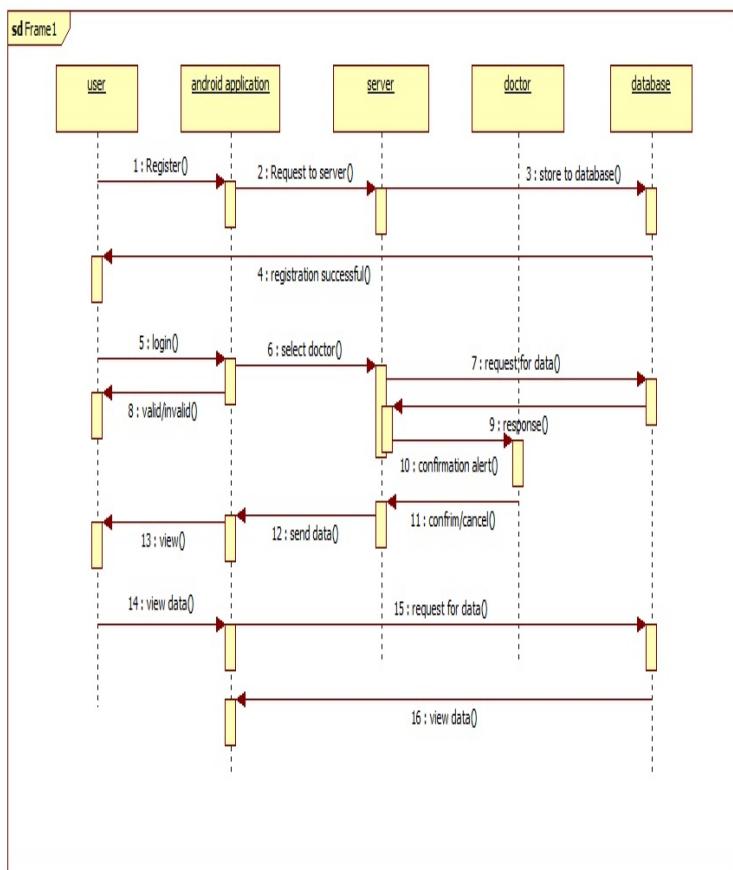
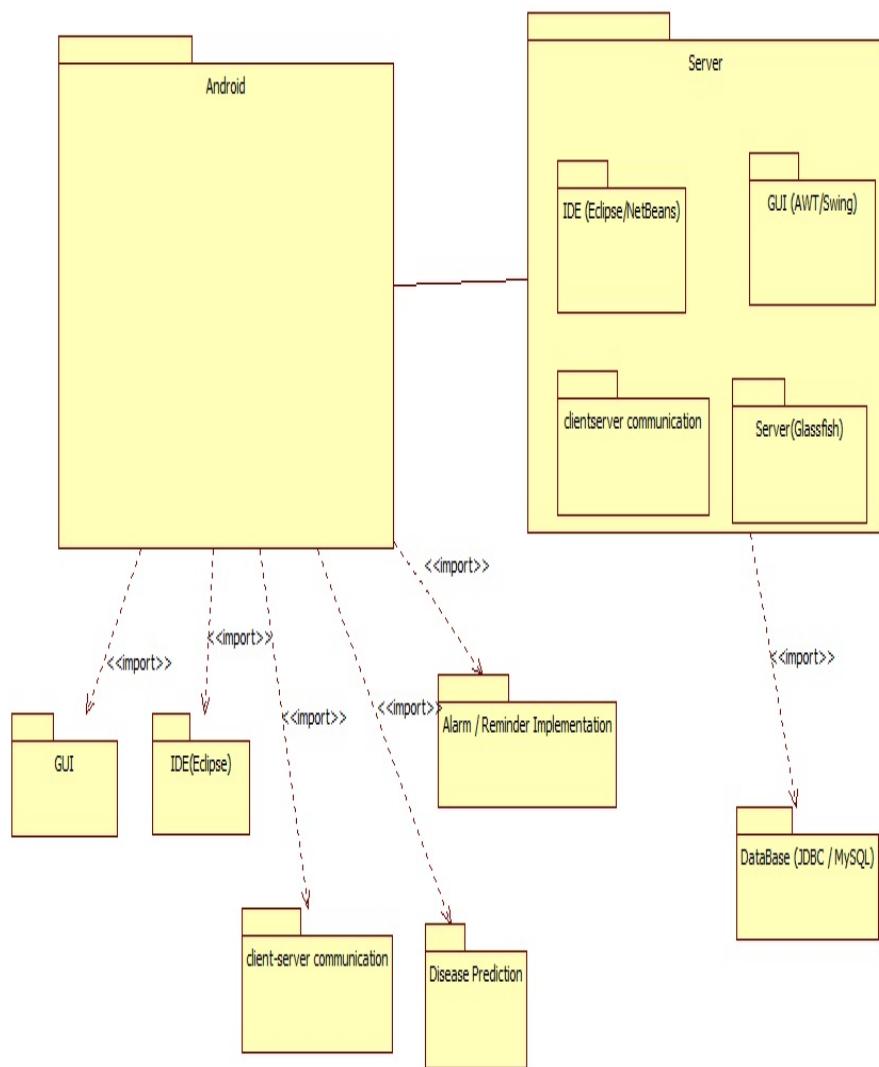
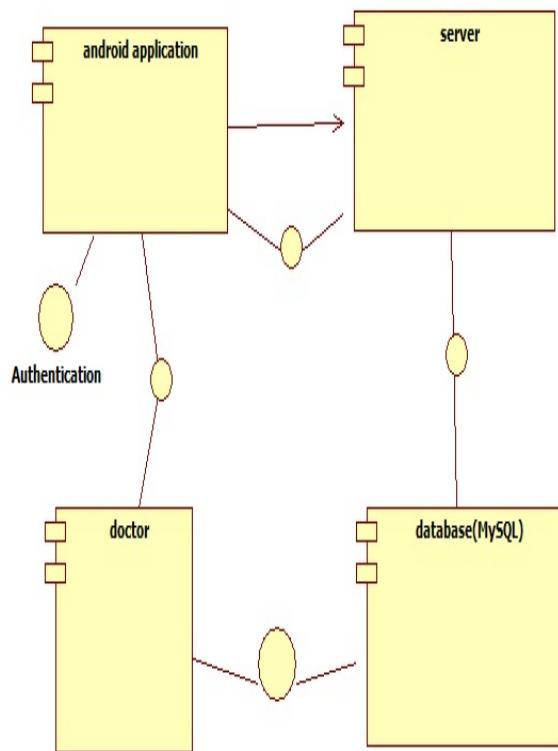


Figure 6.2: *Interaction Diagram (Sequence Diagram)*

Figure 6.3: *Package Diagram*

Figure 6.4: *Component Diagram*

Chapter 7

Technical Specifications

7.1 Technology details to be used in the project

Frontend

Android application (built using Eclipse ADT plug-in)

Backend

Database: MySQL

7.2 References to Technology

- Abraham Silberschatz and Henry Korth, Database System Concepts, Alpha, 2001
- Herbert Schildt, The Complete Reference Java seventh edition, Tata McGraw-Hill, 2007
- Zigurd Mednieks, Android App Development, O'Reiley, 2005
- www.lynda.com - Android App Development with Java Essential Training
- www.javatutorialhub.com
- www.developer.android.com

Chapter 8

Snapshots of Project

8.1 Mobile Application

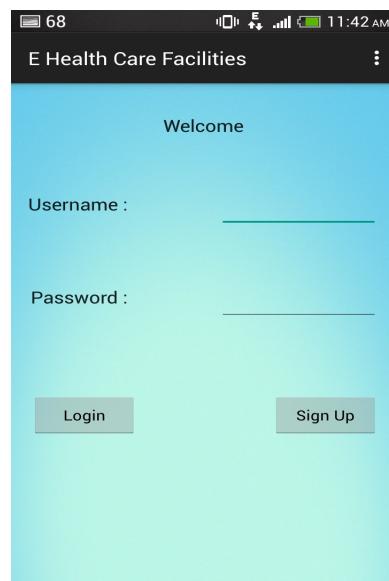


Figure 8.1: *Login/Sign up Page*

The screenshot shows a mobile application interface titled "Sign Up". At the top, it says "Please Fill In The Following". Below are several input fields:

- Name : Akshata Oswal
- Address : Wanowarie
- Blood Grp : B+
- Gender : Select
- Age : 20
- Birth Date : 04/02/1994
- Contact : 7799005412
- E-Mail : taoswal@gmail.com

Figure 8.2: *User/Patient Registration Page 1*

The screenshot shows a mobile application interface titled "Sign Up". It includes the following fields from the previous page and adds new ones:

- Age : 20
- Birth Date : 04/02/1994
- Contact : 7799005412
- E-Mail : akshataoswal@gma
- Username : akshata
- Password : (redacted)
- Confirm Password : (redacted)

At the bottom is a "Sign Up" button.

Figure 8.3: *User/Patient Registration Page 2*

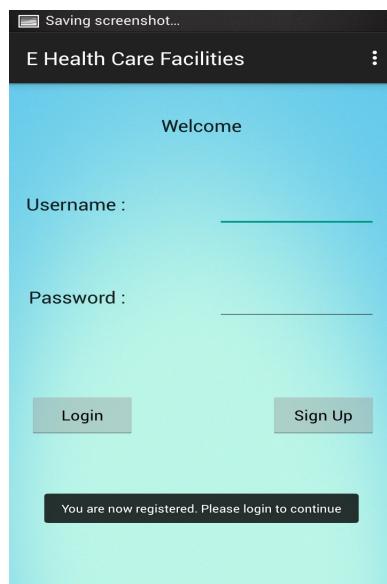


Figure 8.4: *Registration Confirmed*

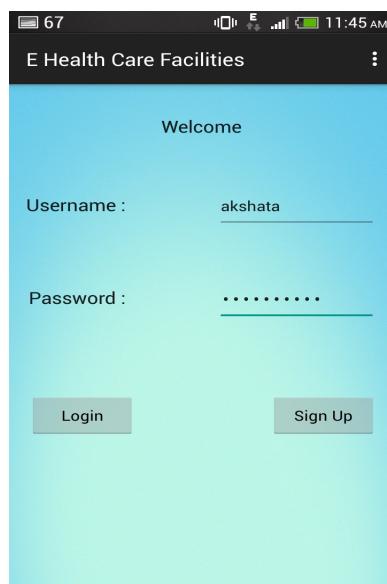
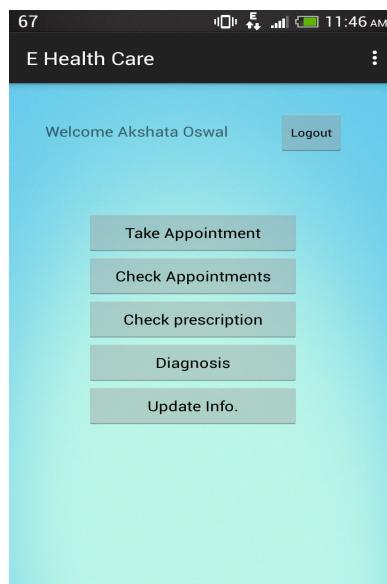


Figure 8.5: *Signing In*

Figure 8.6: *Startup Page*Figure 8.7: *Select the Speciality/Doctor/Date/Time of your choice*

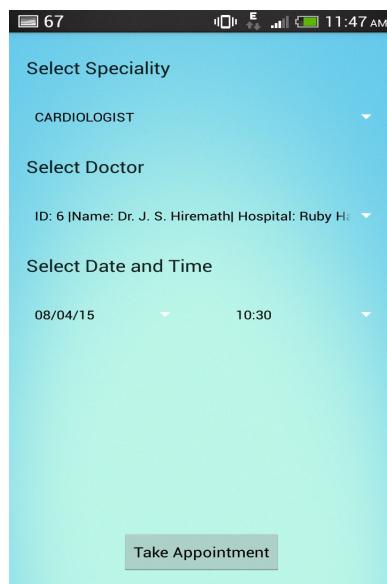


Figure 8.8: Verify the details before fixing an appointment

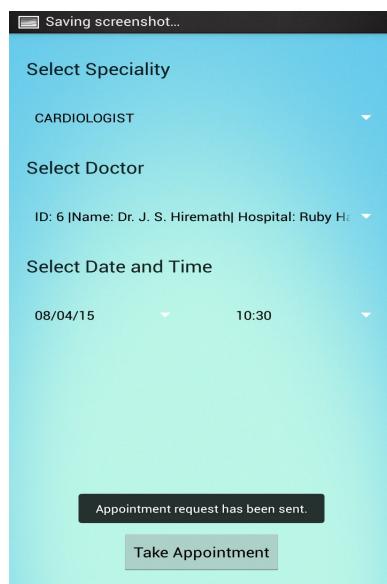


Figure 8.9: Request for the appointment sent

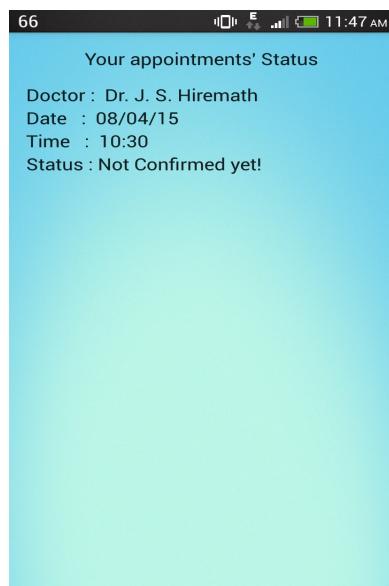


Figure 8.10: *Current Appointment Status*



Figure 8.11: *Appointment Status after Confirmation from doctors side*

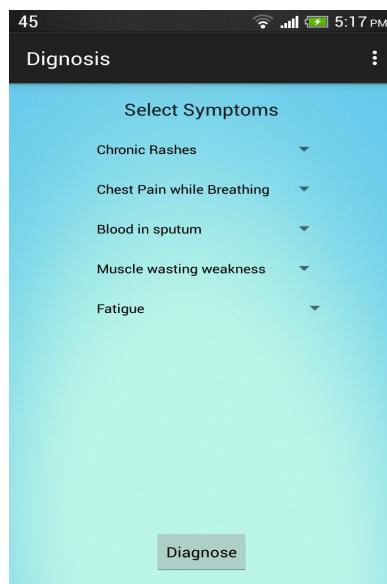


Figure 8.12: *Select Symptoms for Disease Diagnosis*



Figure 8.13: *Probable Diseases Diagnosed*



Figure 8.14: *Recent Trends*

A screenshot of a smartphone displaying a prescription check interface. The screen shows a black header with signal strength, battery level, and time (12:09 PM). Below the header, the title 'Check Prescription' is displayed. The main content area has a light blue background and contains the following text:

Your dosages

Id	Medicine	Dosage	Morning
1	Crocin	3	No
2	Travasil	2	Yes
3			
4			
5			

Figure 8.15: *Medicines to be taken; when Prescribed*

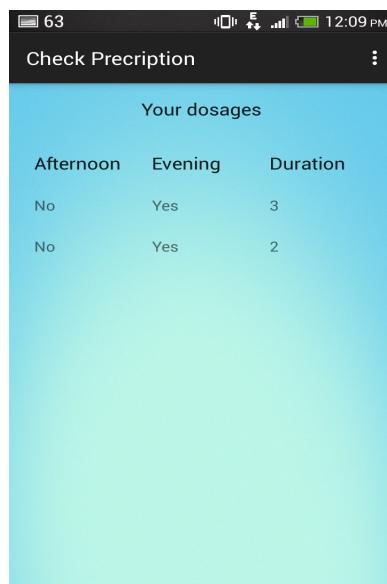


Figure 8.16: Time when the Medicines have to be taken

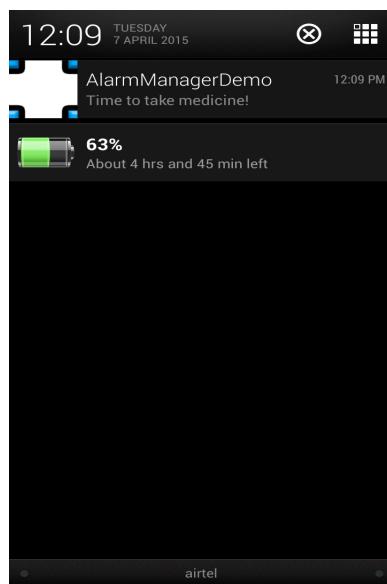
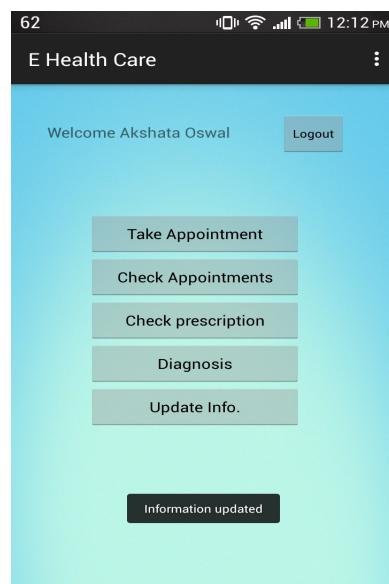


Figure 8.17: Alert/Reminder for the Medicines to be taken

Figure 8.18: *Update Information*Figure 8.19: *Information Updated*

8.2 Website

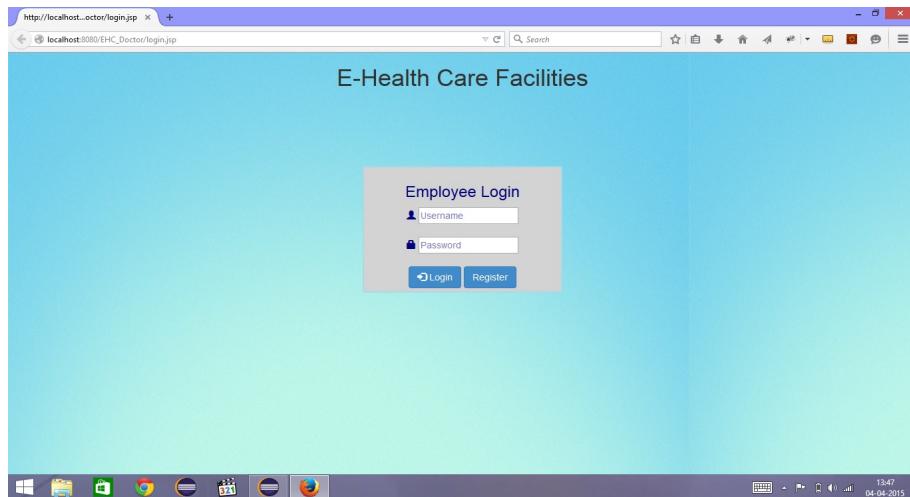


Figure 8.20: *Login/Sign up Page*

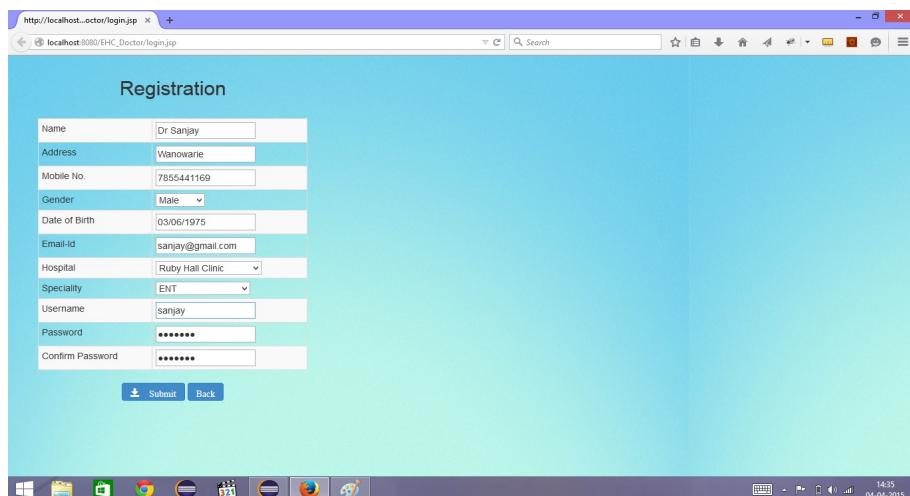
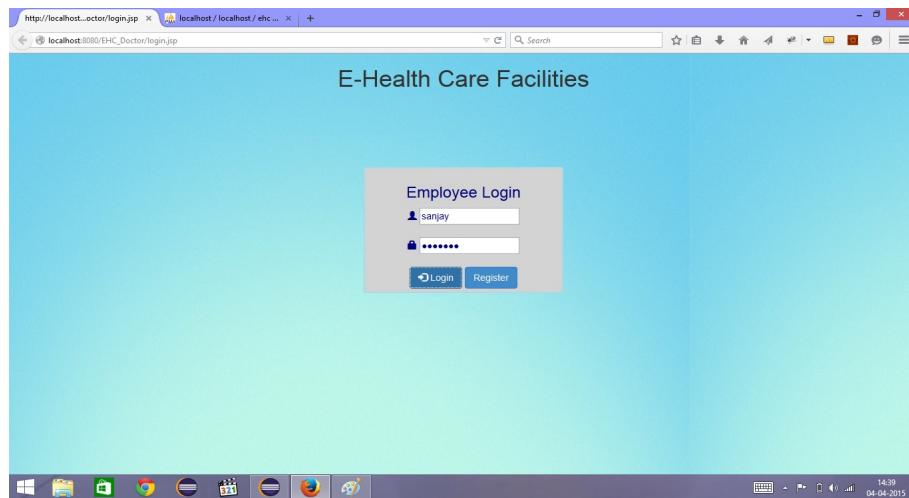
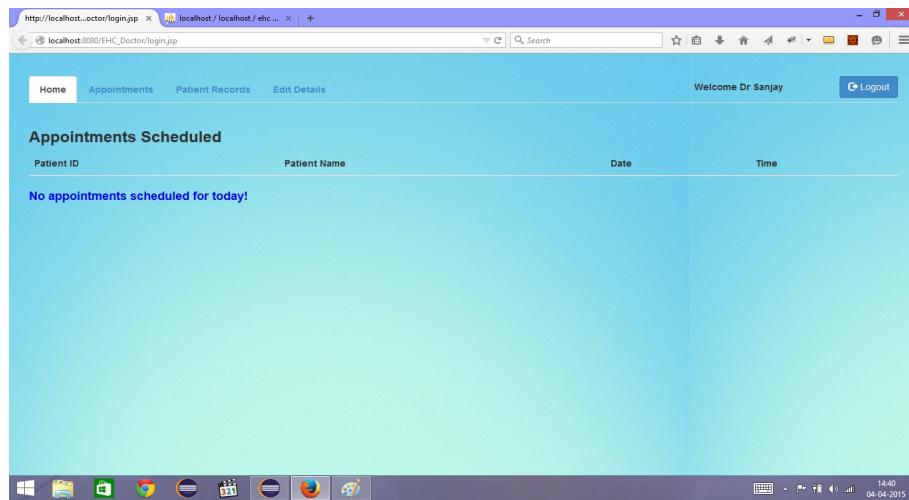
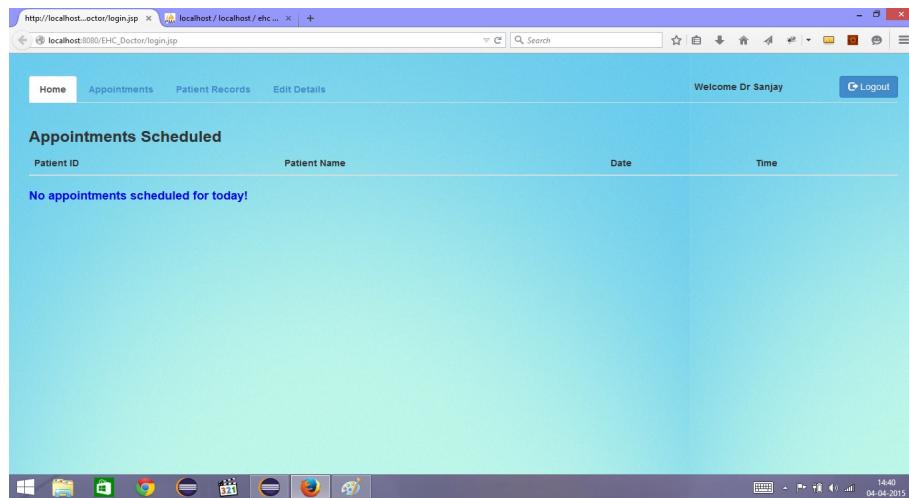
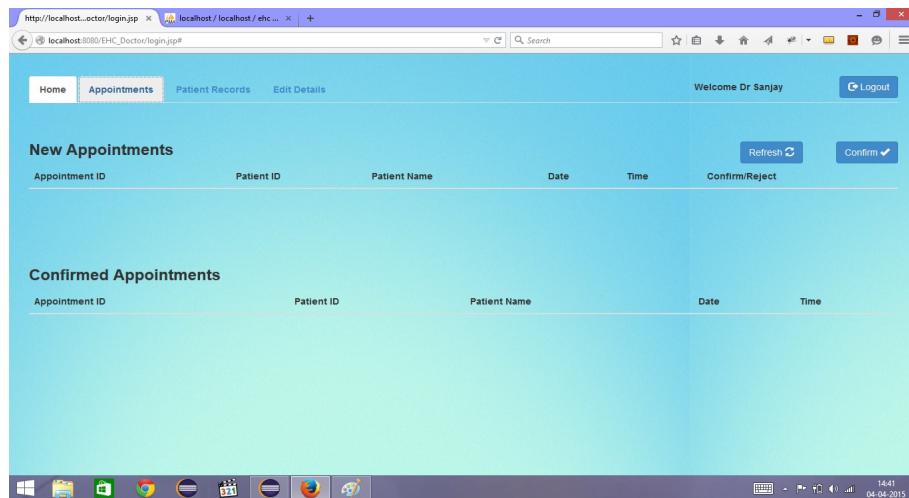
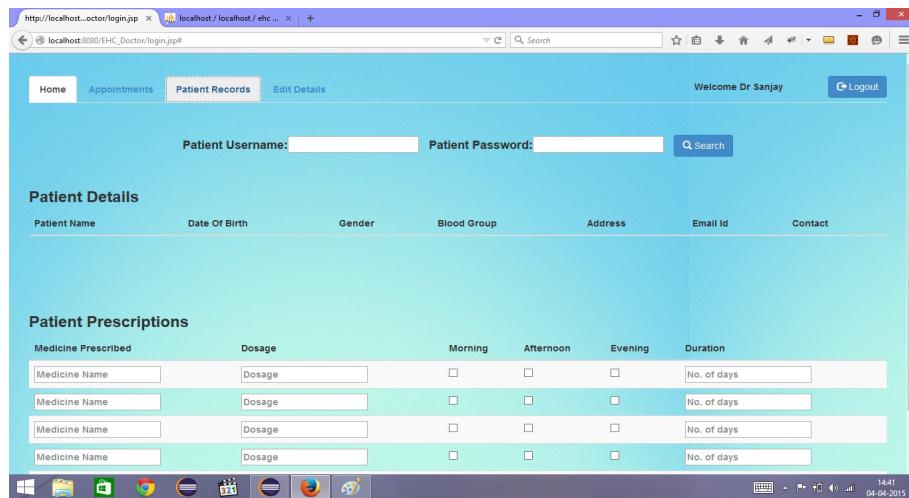
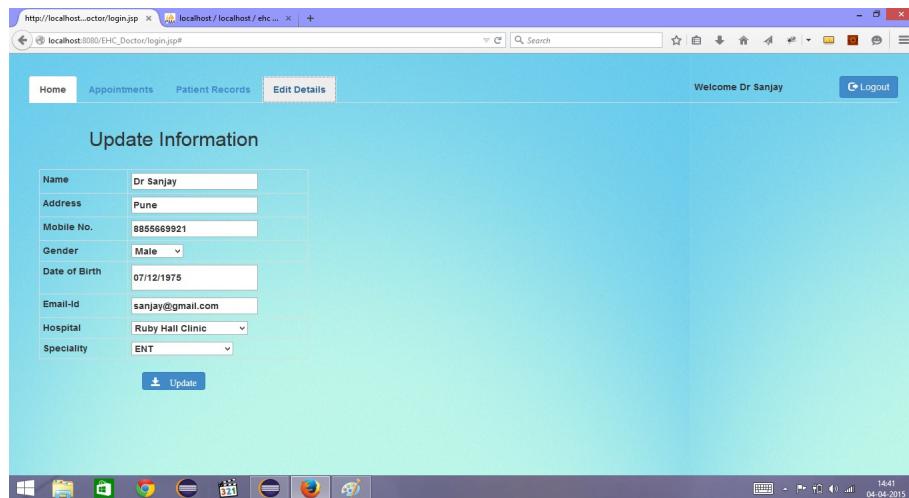
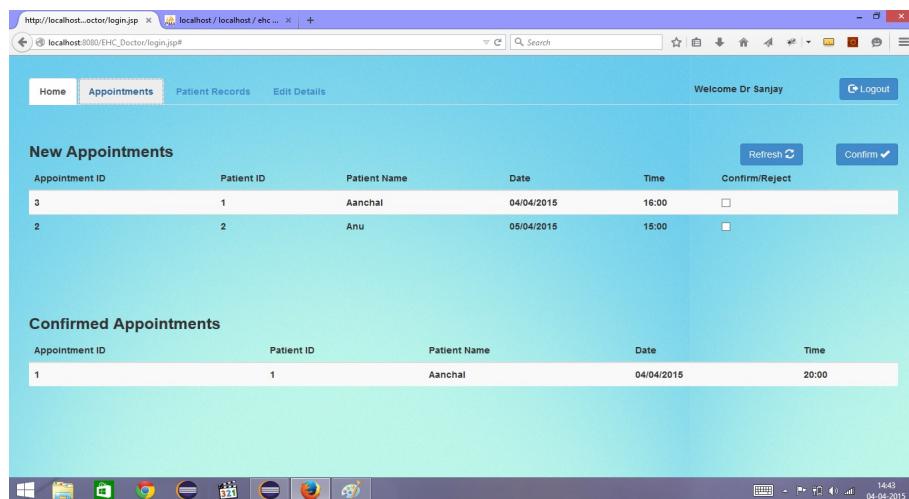
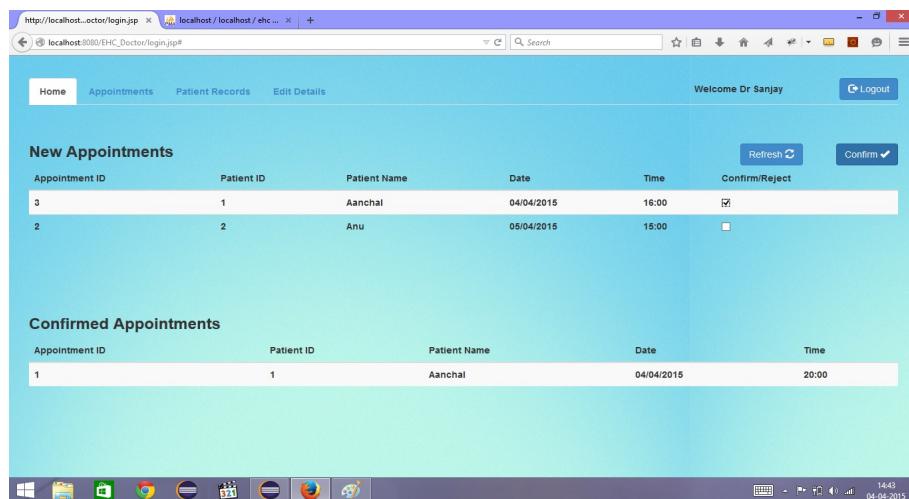


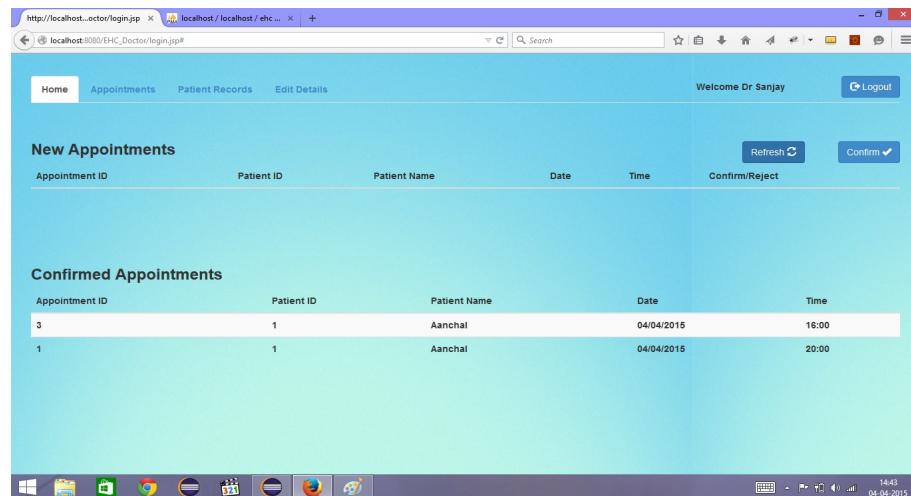
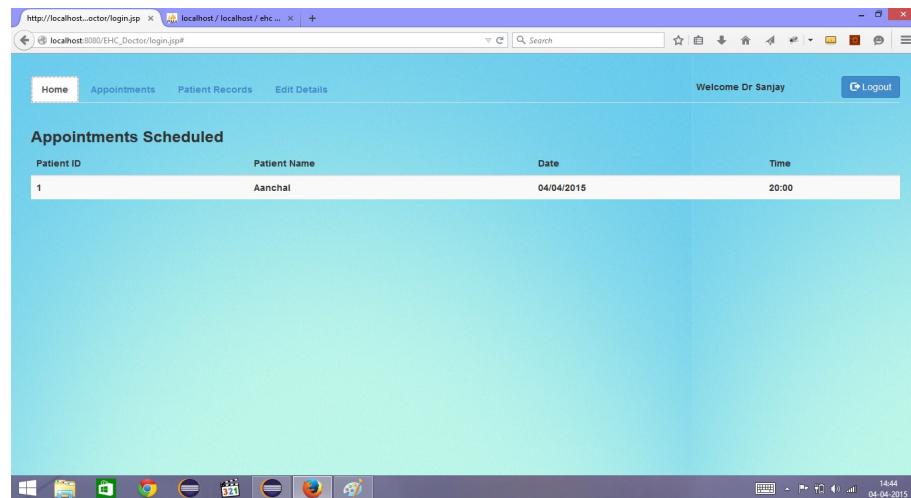
Figure 8.21: *Sign Up Page*

Figure 8.22: *Logging in*Figure 8.23: *Home Page*

Figure 8.24: *Appointment Schedule*Figure 8.25: *Appointments Page*

Figure 8.26: *Patient Records*Figure 8.27: *Update Information*

Figure 8.28: *Appointments List*Figure 8.29: *Confirm/Reject Appointments*

Figure 8.30: *Confirmed Appointments*Figure 8.31: *List of Appointments Scheduled*

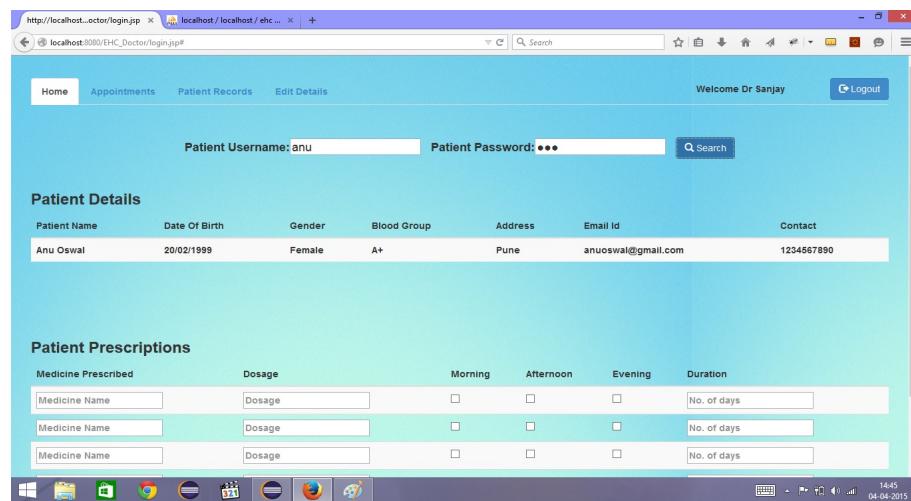


Figure 8.32: Patient Details

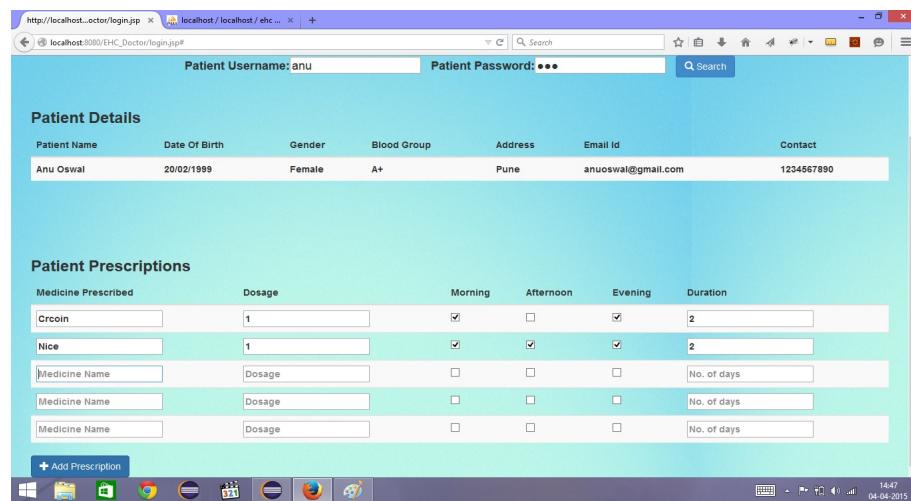


Figure 8.33: Edit Patient Details/Prescribe Medicines

Chapter 9

Conclusion

Our proposed application will be feasible for use in the work environment as well as for family and friends. Also it will be user friendly and can be used with ease by the novice users as well as professional users. The proposed application will be used for the process of taking appointment, cancelling it, adding reminders for medical dosage. It will also predict the disease according to the inputted symptoms by user. This application will be innovative in its own aspect as it will minimize the dependency of user on the devices. This app will combine a number of functionalities into one, so the user need not download a number of applications for having access to information regarding ticketing process or booking a ticket.

The plus points of the applications are as follows:

1. It eliminates paper!
2. It eliminates the need to wait in queue for taking appointment.
3. It is secured and keeps personal data private from other users.
4. Improved customer experience Its interactive and in real time! Mobile app.

Thus what we are seeing is a significant force that could potentially change the entire healthcare industry.

Chapter 10

References

10.1 Papers

- Nirmala Devi.M, Appavu alias Balamurugan.S, Swathi U.V, An amalgam KNN to predict Diabetes Mellitus, IEEE International Conference on Emerging Trends in Computing, Communication and Nanotechnology (ICECCN), 2013.
- Asma A. AlJarullah, Decision Tree Discovery for the Diagnosis of Type II Diabetes, International Conference on Innovations in Information Technology, 2011.
- V. Manikantan & S. Latha, Predicting the Analysis of Heart Disease Symptoms Using Medicinal Data Mining Methods, International Journal on Advanced Computer Theory and Engineering (IJACTE) Volume-2, Issue-2, 2013.
- Rajkumar Gaur Grewal, Babita Pandey, Two level Diagnosis of Breast Cancer using data mining, International Journal of Computer Applications (0975 8887) Volume 89 No 18, March 2014.
- Fatima Eddaoudi, Fakhita Reragui, Abdelhak Mahmoudi and Najib Lamouri, Masses Detection Using SVM Classifier Based on Textures Analysis, Applied Mathematical Sciences, Vol. 5, 2011.
- Rahul Isola, Rebeck Carvalho, Amiya Kumar Tripathy, Knowledge Discovery in Medical Systems Using Differential Diagnosis, LAMSTAR, and k-NN, IEEE Transactions on Information Technology In BiomedicineE, VOL. 16, NO. 6, November 2012.
- Rebeck Carvalho,Rahul Isola,Amiya KumarTripathy MediQuery-An Automated Decision Support System,IEEE,ISSN :1063-7125.

- George L. Tsirogiannis, Dimitrios Frossyniotis, Konstantina S. Nikita, and Andreas Stafylopatis, A Meta-classifier Approach for Medical Diagnosis,G.A. Vouros and T. Panayiotopoulos (Eds.): SETN 2004, LNAI 3025, pp. 154163, 2004.
- M.Akhil jabbar, B.L Deekshatulu, Priti Chandra, Classification of Heart Disease Using K- Nearest Neighbor and Genetic Algorithm, International Conference on Computational Intelligence: Modeling Techniques and Applications(CIMTA), 2013

10.2 Websites

- <http://en.wikipedia.org>
- <http://www.google.co.in>

Appendix A

Mathematical Models

A.1 Mathematical Model

A.1.1 Set Theory

$$Z = \{U, S, D\}$$

Where,

$$U = \text{Set of user} = \{P, H\}$$

$$P = \{\text{patient1, patient2 ...}\}$$

$$H = \{\text{hospital1, hospital2 ...}\}$$

$$U = P \cup H$$

$$D = \text{Set of database} = \{D_t\}$$

$$D_t = U \cup S$$

$$S = \text{Set of services} = \{\text{Access Records, Take Appointment, Reminders, Predict Diseases}\}$$

A.2 Functional Decomposition of the system

DIFFERENT FUNCTIONS:

Let

$f(g)$ = function of sign up.

$f(h)$ = function of appointment scheduling.

$f(i)$ = function of appointment conformation.

$f(j)$ = function of accessing medical records.

$f(k)$ = function of reminders.

$f(l)$ = function of disease prediction.

$f(m)$ = function of uploading records.

FUNCTIONAL DEPENDENCY:

$f(g) : D_t \rightarrow U$

A: $f(i) \rightarrow f(h)$

B: $f(j) \rightarrow f(m)$

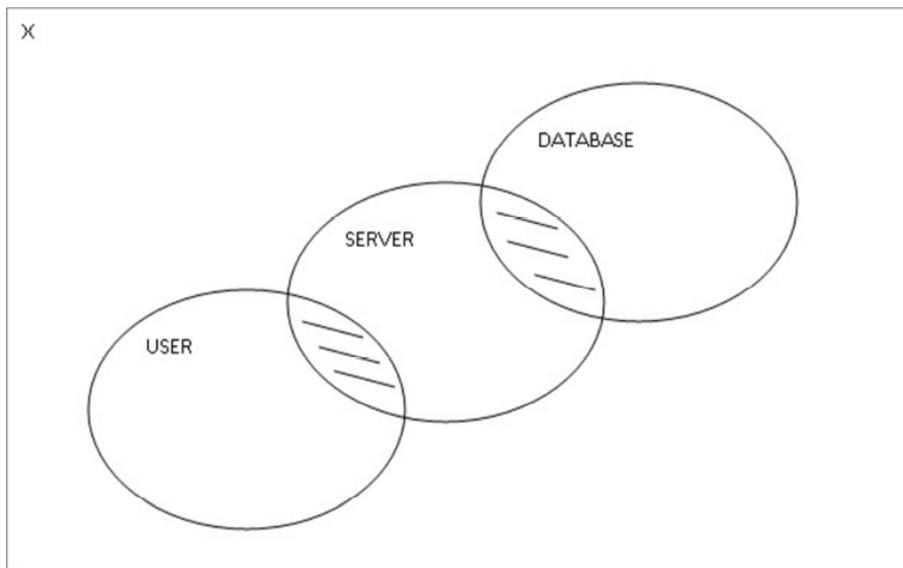
$$C: f(k) \rightarrow f(i) \cup f(m)$$

$$D: f(l) \rightarrow D_t$$

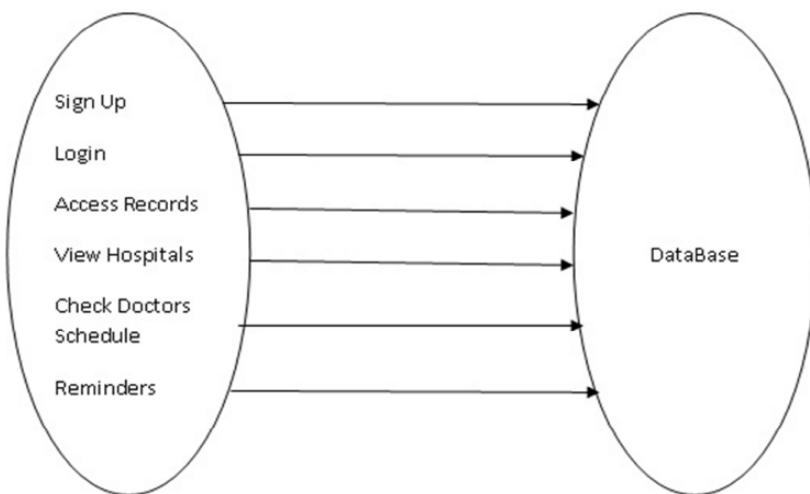
$$E: f(l) \rightarrow f(k)$$

$$F: f(m) \rightarrow f(l)$$

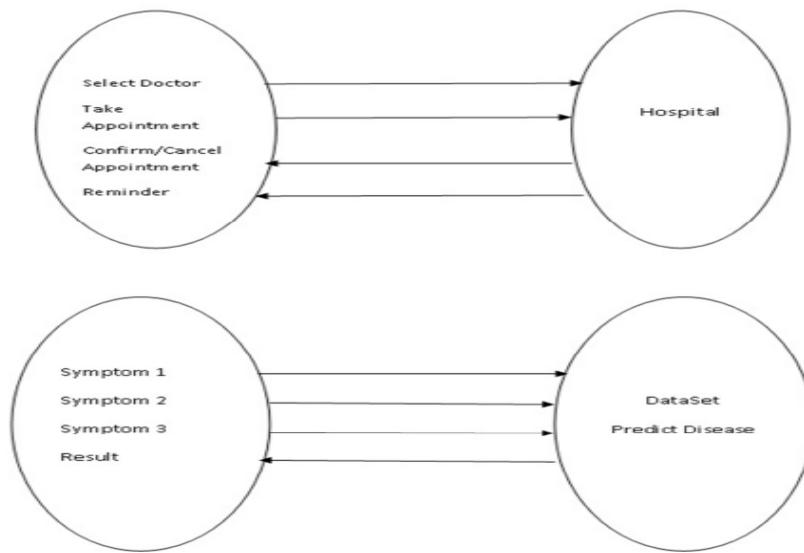
A.2.1 Venn Diagram



A.2.2 Functional Mapping-1



A.2.3 Functional Mapping-2



Appendix B

Testing/Reliability of Design

B.1 Test Plan for Design

Purpose of this plan is to explain step by step testing methodology for a developed android Urban Railway Ticketing application and is to be used this as a guide for the testing activity.

This test plan is designed to test android urban railway ticketing application. The test cases are intended to be ran manually. The test plan centers around ensuring correct functionality at the code and application levels.

The first level of testing is unit testing to ensure that the code that has been written functions correctly and continues to function correctly as the code is changed.

The next level of testing is manual testing to ensure that the functionality of the application are correctly implemented as per the user requirement (SRS and wireframe Documents.).

Testing Stages covered under manual testing.

1. Pre - Test.
2. User Interface.
3. Navigation.
4. Functionality testing.
5. Performance

Objective of this test plan is to ensure that the urban railway ticketing app functionality and component of the application are tested to the high standard and quality requirements are fulfilled.

1. Test plan Defining tasks and responsibilities
2. Identify required resources and related information.
3. Identify testing risks
4. Provide testing Schedule.

Testing Requirements

1. Each urban railway ticketing application program or method should be tested using Unit test at the time of development.
2. Graphical user interface should be tested using manual testing.
3. Functionality of software will be tested using functional testing.

Testing Strategy

The steps in testing consist:

1. Creation of all the test scenarios and test cases
2. Preparation of a test case document that has a brief description of the test case , steps to conduct tests and expected result.
3. Defect Report generation.

B.2 Details of Design Testing

The main testing types that will be performed.

1. **Pre Test:** This Pre - Testing provides general tests about launching the application.
2. **User Interface:** This User Interface testing provides tests about delivery, running the application, display, and using of the application.
3. **Navigation:** This Navigation testing provides tests about all menus, widgets and functions are accurate and correctly working.
4. **Functionality testing:** This User Interface testing provides tests about implementation of all the features in the application and work as per expectation. Sources for the information are SRS and wireframe Documents

5. **Performance Testing:** This Performance testing provides tests about response time (speed) and utilization of Application to execute its required functions in comparison with performance standard and The Application should perform at its best.

Features to Test In this application we will be testing all the features listed below:

1. **Authentication & Validation:** The username and password should be validated and authenticated.
2. **Taking Appointment:** Checking the proper dates and timing of the appointment request.
3. **Confirmation of Appointments:** The appointments have to be either confirmed/cancelled by the doctors side.
4. **Disease Prediction:** According to the symptoms entered most accurate diseases should be predicted.
5. **Retrieval of Data:** The data should be retrieved from the database whenever it is needed by the user.
6. **Checking Prescriptions:** Prescriptions can be uploaded by the doctor only and that will be visible to the patient.

B.3 Test Cases

Test Case Id	Test Case Name	Test Steps	Expected Result	Actual Result	Test Status (P/F)
1	Validate Registration	Enter mobile no less than/greater than 10 digits	An error message "Mobile no should be 10 digits long"	An error message "Mobile no should be 10 digits long"	P
2	Validate Registration	Enter email address without '@' or.'	An error message "Enter valid email address"	An error message "Enter valid email address"	P
3	Validate Registration	Enter password less than 6 characters	An error message Enter password with length greater than 7 and less than 12	An error message Enter password with length greater than 7 and less than 12	P
4	Validate Registration	Enter username less than 4 characters	An error message Enter valid username	An error message Enter valid username	P
5	Validate Registration	Fill incomplete sign up details	An error message Fill all details	An error message Fill all details	P
6	GUI	Check font, button size and position of buttons and text eld	They should be of same size and properly positioned	Size is same with proper positioning	P
7	Usability	Verify usability and interactivity	It should be user friendly and self-learning.	It is user friendly and self-learning	P
8	Performance	Time for disease diagnosis	It should be within limit with ecient performance	Desirable speed	P

9	Functionality	Functionality of recent trends in diseases and its accuracy	Prediction should be accurate	Its accurate	P
10	Appointment	Take appointment for a date / time before current date / time	An error message Enter valid date/time	An error message Enter valid date/time	P
11	Prescription	Reminders for prescription (Morning, Evening, Night)	Reminders at respective time	Reminders at respective time	P
12	GUI	Check for the background color and visibility	Background color should be constant and text should be clearly visible	Consistency is maintained and clear visibility	P

Appendix C

Project Management and Project Status

C.1 Project Management Approach

We have defined one tracking sheet for BE COMP where we are updating plan and actual dates as per the phase of project. HOD and Project coordinator track this sheet on monthly basis. Project status review takes place with project guide, project coordinator and HOD-COMP. Project guide reviews the documentation and project milestone in detail. Project coordinator tracks the timeliness and correctness of concepts. HOD ensures that all project team members are contributing to the project, team work is happening and project is within the framework as defined by the university guidelines.

Project team members meet once a week with the sponsoring company to provide the status and plan the activities for the current and next milestone.

C.2 Status of Project Work done

Sr. No	Task	Status
1	Synopsis	Completed
2	SRS Document	Completed
3	Design Completion	Completed
4	Coding	Completed
5	Testing	Completed

Table C.1: Project Status

Appendix D

Papers Published

1. "*A Survey on Disease Diagnosis Algorithm*", Aanchal Oswal, Vachana Shetty, Mustafa Badshah, Rohit Pitre, Manali Vashi, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume - 3, Issue - 11, November 2014
2. "*Digital Medical Care with Iterative and Probabilistic Disease Prediction*", Aanchal Oswal, Vachana Shetty, Mustafa Badshah, Rohit Pitre, Manali Vashi, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume - 4, Issue - 5, May 2015

Appendix E

User Manual

E.1 Preface

E.1.1 About This Guide

This document provides information about the services and functions available with the E-Health Care Facility App (referred to simply as "the app" in the remainder of this document) and how to access them.

E.1.2 Audience

This guide is intended for people who want to use their Android phone to access remote interactive services for E-Health Care Facilities.

E.2 System Overview

This is an Android application of E-Health Care Facilities where a user (patient) can get a brief idea about what disease he/she is suffering from just by inputting some symptoms using our smartphone application. Users can even make a pre-appointment with the doctor of their choice. Using our phones facilities the application can enable reminders to take the specific medicine at the specific time according to our medical prescription which is also uploaded by our doctors. Basically, the user can fix appointments with the doctor of their preference, suitable to their convenience of time and date; view his reports; set alerts and reminders.

E.3 System Summary

This app operates on mobile devices with Android operating system. It is compatible with Android 1.5 API level 3 and higher versions. The

application requires connection to Internet in order to have access to the functionalities of the app. After installation on the device, app can be used immediately without any further configuration.

E.3.1 User Access Levels

Everyone can use application, but registration has to be done prior to the usage of application.

E.3.2 Contingencies

In case of power outage data are not saved in internal memory of the operating device. In case there is no Internet connection available then the application cannot be used.

E.4 Getting Started

Getting started section briefly describes E-Health Care Facility Application and installation of it on the device. The section briefly presents the log in and sign up process.

E.4.1 Installation

The app is not available (yet) on Google Play (a.k.a. the Android Market). Consequently, to install the app, you will first have to enable "third-party app installation" in the "developer" settings on your phone. You can then transfer the application package (version 1.0) and install it manually on your device. If you have the ".apk" package on a desktop PC, first transfer it to your device, using your USB connection, to a directory of your choice (for example "/download"). If you transferred the ".apk" package from a desktop PC or if your system has not automatically prompted you for installation, you then need to install a file manager on your device. I recommend OI File Manager. With this tool, open the "/download" directory (or the one you chose to store your copy of the ".apk" package) and "click" on the ".apk" file to run the installation. The app is currently optimized only for "standard" screen sizes (320x480 pixels) but should not have any problem with other sizes and densities, except for sub-optimal graphics.

E.4.2 Signing In

To launch the app, tap the application icon on the home screen of your Android device. The Sign In screen appears. If you do not have an account then create one using signup and then sign in. Enter the username and password. When you are done, tap Sign In. Note: Your screens might look somewhat different from the examples in this document, but the basic elements are the same.

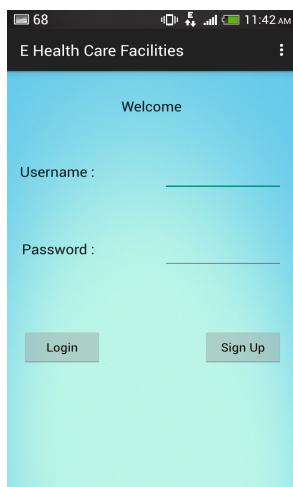


Figure E.1: *Sign in*

E.4.3 Signing Up

If you are a new user then the sign up process has to be completed which requires basic information. Fill the information and register.

E.5 System Menu

The system menu comprises of five main tabs : Take Appointments, Check Appointments, Check Prescription, Diagnosis and Update Information.

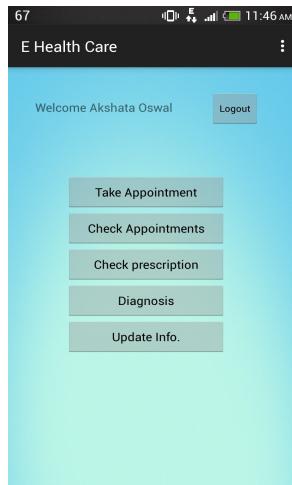


Figure E.2: *Menu Page*

E.5.1 Take Appointment

Here the User (Patient) can select the doctor of his/her choice by selecting the speciality, the doctor, time and date of convenience and then fix an appointment.

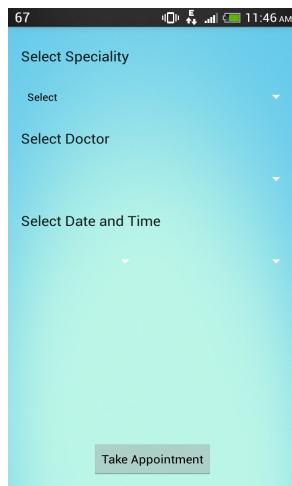


Figure E.3: *Take Appointment*

E.5.2 Check Appointment

On selecting the check appointment you can get the status of your appointment whether it has been confirmed from the doctors side or yet has to be confirmed.



Figure E.4: *Status of Appointment*

E.5.3 Check Prescription

If the doctor has prescribed any medicines for the patient then that can be viewed here along with the dosage of the prescribed medicines and the time it has to be taken(Morning/Evening/Night).

A screenshot of a mobile application titled "Check Prescription". The screen displays a table of "Your dosages" with the following data:

Id	Medicine	Dosage	Morning	Evening	Night
			Yes	No	Yes
1	Crocin	3	No	Yes	Yes
2	Travilis	2	Yes	No	Yes
3					
4					
5					

Figure E.5: *Check Prescription*

E.5.4 Diagnosis

On selecting the disease diagnosis option the patient can find out what disease he/she is suffering from by just selecting the symptoms. Here one has to simply select the symptoms he's facing and click the diagnose button to avail a probable idea of the disease which he/she might be suffering from. Once the user finds out what disease he/she is suffering from they can select the recent trends option and be more assured.



Figure E.6: *Select Symptoms*



Figure E.7: *Diagnosis*



Figure E.8: *Recent Trends*

E.5.5 Update Information

The basic information of the user can be edited any time from the system menu by clicking on the Update Info. button.

A screenshot of a mobile application interface titled "Update Information". The screen shows a form with the following fields:

- Name : Akshata Oswal
- Address : Wanowarie, Pune
- Blood Grp : B+
- Gender : Female
- Age : 20
- Birth Date : 04/02/1994
- Contact : 7799005412
- E-Mail : akshataoswal@gma

Each field is preceded by a label and followed by an input field or dropdown menu. The entire form is set against a light blue background.

Figure E.9: *Update Information*

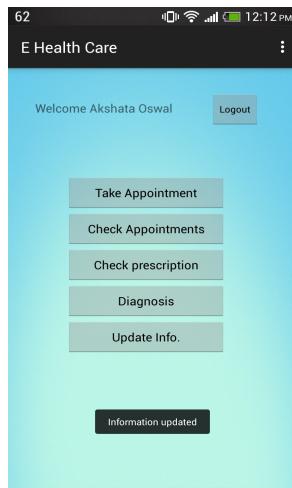


Figure E.10: *Information Updated*

E.6 Exit System

The system can be exited simply by logging out from the system menu after carrying out the activities of your choice. This can be done by clicking on the log out button present in the main page.

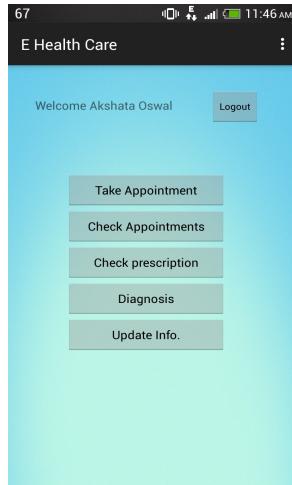


Figure E.11: *Menu Page*