A PRELIMINARY PROJECT REPORT

ON

Personalised Product Recommendation System using Hybrid Approach Based on Content Based and Collaborative Filtering

Submitted to the Savitribai Phule Pune University in the

partial fulfillment of the requirements for the award of degree

of

BACHELOR OF COMPUTER ENGINEERING

BY

SHUBHAM KOTHARI	**B120234309**
AMIT PATIL	**B120234346**
PREM MOTGI	**B120234333**
SHWETA SALWADE	**B120234379**
AJINKYA KOVE	**B120234310**

DEPARTMENT OF COMPUTER ENGINEERING

Accredited by NBA

STES'

SINHGAD COLLEGE OF ENGINEERING

S.N. 44/1, Vadgaon Bk, Off Sinhgad Road

Pune - 411041

YEAR 2016-17

Sinhgad Institutes

SINHGAD COLLEGE OF ENGINEERING

S.N. 44/1, Vadgaon Bk, Off Sinhgad Road

Pune - 411041

CERTIFICATE

This is to certify that the preliminary project report entitled

Personalised Product Recommendation System using Hybrid Approach Based on Content Based and Collaborative Filtering

Submitted by

SHUBHAM KOTHARI	**B120234309**
AMIT PATIL	**B120234346**
PREM MOTGI	**B120234333**
SHWETA SALWADE	**B120234379**
AJINKYA KOVE	**B120234310**

is a bonafide work carried out and is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, Pune for the award of the Degree of Bachelor of Computer Engineering

This project work has not been earlier submitted to any other Institute or University for the award
of any degree or diploma.

Place : Pune.

Date :

Prof. C. A. LAULKAR
Internal Guide
Department of Computer Engineering

Prof. M. P. Wankhade
Head,
Department of Computer Engineering

Dr. S. D. Lokhande
Principal
SCOE, Pune

Acknowledgements

With deep sense of gratitude we would like to thanks all the people who have lit our path with their kind guidance. We are very grateful to these intellectuals who did their best to help during our project work.

We are profoundly grateful to **Prof. C. A. Laulkar** for her expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion.

We would like to express deepest appreciation towards **Dr. S. D. Lokhande** , Principal, Sinhgad College of Engineering, **Prof. M. P. Wankhade** , Head of Department of Computer Engineering whose invaluable guidance supported us in completing this work.

At last we must express our sincere heartfelt gratitude to all the staff members of Computer Engineering Department who helped me directly or indirectly during this course of work.

Abstract

In an age of information overload, the importance of personalized recommendation system for online products and services is rapidly growing. Such systems allows buyers to find what they want without wasting their time and also enables sellers to provide buyers what they are expecting to purchase, thereby furnishing benefits to both parties.

Mundanely, people take errant walks in large scale shopping complexes and during that period they mostly face problems such as missing benefactor deals on customer's side as well as reduction in sells on vendor's side. Apparent solutions to these problems emphasize on providing recommendation based on large set of data; while proposed system is able to execute on small set of data and generating optimal and precise recommendation to the customer with in contrast to Netflix and YouTube etc.

Proposed system emphasize on amalgamated version of content based approach as well as collaborative filtering approach for the recommendation engine. Vital problems such as sparsity, cold star, scalability and provides accurate solution executing on small set of data.

List of Figures

2.1 Time Line Chart 15

3.1 Architecture Diagram 24
3.2 Use Case Diagram 25
3.3 Class Diagram 26
3.4 ER Diagram 27
3.5 State Transition Diagram 28
3.6 Deployment Diagram 29

4.1 Patient App: NFC 59
4.2 Patient App: Disease Prediction 59

6.1 Admin App: Main App 66
6.2 Admin App: Training Dataset 66
6.3 Doctor App 67
6.4 Patient App 67
6.5 Patient App: Disease Prediction (a) 68
6.6 Patient App: Disease Prediction (b) 68

List of Tables

4.1 doctor 32
4.2 medicenelog 32
4.3 noofdiseases 33
4.4 patient 33
4.5 patient(...contd) 33
4.6 patientlog 34

5.1 Unit Testing 63
5.2 Unit Testing(...contd) 64

Contents

Certificate **I**

Acknowledgements **II**

Abstract **III**

List of Figures **IV**

List of Tables **V**

1 INTRODUCTION **2**

1.1 Background And Basics 3

1.2 Literature Survey 4

1.3 Project Undertaken 5

1.3.1 Problem Definition 5

1.3.2 Scope Statement 5

1.4 Organization of Report 5

2 PROJECT PLANNING AND MANAGEMENT **7**

2.1 Detail System Requirement Specification (SRS) 8

2.1.1 Proem 8

2.1.2 Overall Description 8

2.1.3 External Interface Requirements 10

2.1.4 System Features 11

2.1.5 Other Nonfunctional Requirements 13

2.2 Project Process Modeling 14

2.3 Project Scheduling 15
2.3.1 Time Line Chart 15

3 ANALYSIS & DESIGN 16

3.1 IDEA matrix 17
3.2 Mathematical Model (Algorithm and Methodology) 20
3.3 Architecture Diagram 24
3.4 UML Diagrams 25
3.4.1 Use-Case Diagrams 25
3.4.2 Class Diagrams 26
3.4.3 ER Diagrams 27
3.4.4 State Transition Diagrams 28
3.4.5 Deployment Diagrams 29

4 IMPLEMENTATION & CODING 30

4.1 Introduction 31
4.2 Database Schema 31
4.2.1 Doctor Table 31
4.2.2 Medicenelog Table 32
4.2.3 No of Diseases Table 32
4.2.4 Patient Table 32
4.2.5 Patient Log Table 33
4.2.6 Symptoms Table 34
4.3 Operational Details & Classes 34
4.3.1 Operations 34
4.3.2 Major Classes 35
4.3.3 Code Listing 36
4.4 Screenshots of Major Functionalities 59

5 TESTING 60

5.1 Unit Testing 61

5.2 Integration Testing 61
5.3 Acceptance Testing 62

6 RESULTS & DISCUSSION **65**
6.1 Main GUI Snapshots 66
6.2 Discussion 69

7 CONCLUSION **71**

8 References **73**

CHAPTER 1

INTRODUCTION

1.1 Background And Basics

This is the 21st century. Era of computer development and data manipulation. This is where information in digital form has become an undivided part of human knowledge processing. From social networking to putting steps on moon computer are become inevitable to make progress and in taking decisions. The true generation information-ally dependent on world wide web and big data technologies.

Online shopping has become new trend of human lifestyle. So while shopping it is always better have some suggestion, advise and most importantly masterly experience in buying. In the real world we take our friends and families to the shopping with us to take better decisions to buy and to enhance our option and eliminate tunnel visioning while roaming around shopping complex. So the four housemen in digital marketing and online shopping like Amazon, Netflix, Youtube, Google has come to know that providing better options of product to buy for customer has become backbone of competition in digital world.

For this purposes this vendors started to use technology called as recommender engines. The key reason why many people seem to care about recommender systems is money. For companies such as Amazon, Netflix, and Spotify, recommender systems drive significant engagement and revenue. But this is the more cynical view of things. The reason these companies (and others) see increased revenue is because they deliver actual value to their customers – recommender systems provide a scalable way of personalising content for users in scenarios with many items.

But all of these recommendation systems are used to buy online products while we will be designing the engine to make recommendations on products which they can buy from store surrounding them in shopping complexes or mall. The reason behind this is that by visiting the shop based on our product recommendation customers can actually get a chance to see and touch and feel the products as well as their originality.

Uniqueness of this system is that now a days algorithms used to develop recommendation engines follow single approaches while we will be implementing hybrid approach to tackle disadvantages of both the approaches. This will give upper hand to the system while working moderately small amount of data as well as out of the box structure of our system.

1.2 Literature Survey

Recommendation system is an emerging field in now a days trending computer science. Lot of research and experiments are conducting right now to make recommendation engines more accurate or at least as precise as possible.

There is system called Unified Collaborative and Content-Based Web Service Recommendation which. Provides a Unique Algorithms for Web Service Recommendation. It Combines two major approaches: Collaborative and Content based and finds an unique algorithm overcoming the drawbacks of both algorithm. A project called as Unifying User based and Item based Collaborative Filtering Approaches by Similarity Fusion which reformulates the memory based collaborative filtering problem in a generative probabilistic framework, treating individual user-item ratings as predictors of missing ratings.

Systems like Matrix Factorization Techniques For Recommender Systems To design the recommender system which rely on different sets of input data which are placed in a matrix, representing the individual interest. It show an approach to implement recommender systems using matrix factorization.

Amazon uses a recommendation engine which To suggest various algorithm which basically aggregates items from similar customers, it suggest the items which are not yet bought or rated by users. Netflix uses its own one of kind award winning recommendation engine which combines 105 algorithms and generate recommendations. Approaches like Item-Based Collaborative Filtering.

Recommendation Algorithm are used to produce high quality recommendations and perform many recommendation per second for millions of users and achieve high coverage in the face of data sparsity. There is a survey conducted on Collaborative Filtering Based Recommender Systems for Mobile Internet Applications which focuses on taking full advantages of this ubiquitous mobile internet data from rapid development and applications for recommender systems.

There are some supplementary systems needed to work with recommendation engine like collective intelligence and decision support system along with expert systems which are equipped with Mechanism that supports adaptiveness and collective intelligence while Avoid-

ing limitations of conventional recommendation systems.

1.3 Project Undertaken

1.3.1 Problem Definition

To Design a Personalised Product Recommendation System using Hybrid Approach Based on Content Based and Collaborative filtering.

1.3.2 Scope Statement

Most of the people take errant walks in large scale shopping complexes mostly face the problems, such as missing the beneficial deals on the customer side as well reduction in sales on vendor side. Apparent solution to these problems emphasize on providing recommendation based on large set of data.

The proposed system will generate recommendations by using hybrid algorithm on small set of data with optimal results. To ensure pr´ecised and efficient recommendation so as to improve economic success of vendors.

1.4 Organization of Report

Project report is a major activity that was given higher precedence in the entire project activity. Project report encompasses in all the chapters. The college certificate is attached at the starting of the report.

Chapter 1 - Introduction focuses on the background of the project and scope of the project.

Chapter 2 - Software Requirement Specification explains the overall application description, system features, external interface requirements and other nonfunctional requirements of the system. Also covers the estimation of the project and scheduling system followed. Time line chart is also included in this section.

Chapter 3 - Software Design Specification concentrates on UML diagrams like Use Case diagram, Component diagram, Sequence diagram, State-Machine diagram and Activity diagram which focus on major users, their scope, responsibilities and information flow.

Chapter 4 - It includes which testing strategy is used to test the system, various test cases for different error conditions. It covers unit testing, integration testing and system testing.

Chapter 5 gives the references.

CHAPTER 2

PROJECT PLANNING AND MANAGEMENT

2.1 Detail System Requirement Specification (SRS)

2.1.1 Proem

1. Purpose:

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

2. Intended Audience and Reading Suggestions:

   The project is the college level project and is implementing under the guidance of college professors. This project is useful for optimized recommendation and generation of error free results.

3. Product Scope:

   Most of the people take errant walks in large scale shopping complexes mostly face the problems, such as missing the beneficial deals on the customer side as well reduction in sales on vendor side. Apparent solution to these problems emphasize on providing recommendation based on large set of data. The proposed system will generate recommendations by using hybrid algorithm on small set of data with optimal results. To ensure precised and efficient recommendation so as to improve economic success of vendors.

2.1.2 Overall Description

1. Product Perspective:

   At present, there is no such system which provides offline recommendations for small data sets. The recommendation system which exists, mainly work on online large data sets. The proposed recommendation system will work offline on small data sets. It comprises of hybrid approach based on Collaborative and Content-Based Filtering algorithm. The proposed system will produce concise recommendation using minimal datasets. The system in return will help to increase customer satisfaction and financial growth of vendors.

2. Product Functions

   The project features of:

   i. Precise Recommendation.

   ii. Well-Organized user history.

   iii. Information about Vendors (ex. Opening, Closing and, Lunch Timings)

   iv. Best Deals and Offers

   v. List of Gravitating Products

   vi. Remote Ordering

3. User Classes and Characteristics

   Various Classes of Users Operating the Product:

   • Vendors: Initializes and reconditions the offers as per the ongoing trend. Updates the user's record whenever a product is been bought.

   • Customer: Accesses to recommendation provided by the system.

   • Administrator: Tracks and manages all the user records. Responsible for maintenance of server.

4. Operating Environment

   Hardware

   - RAM: 8 Gb
   - Storage 1Tb+
   - Processor: i5+
   - Router and Antenna

   Operating System:

   - Server Side: Windows 7+
   - Vendor Side: Windows 7+/ Android 4.4+

- Client Side: Android 4.4+
  - Browser: Mozilla 40.0 + / Internet Explorer 8+
  - SQL Server

Design and Implementation Constraints:

- User's data is highly classified and should be accessed only by authorized Users.

- The results should be produced within the time limit of 30 seconds.

- Language Constraint: English

5. User Documentation – User Manual
   – On-Line Help
   – Tutorials

2.1.3 External Interface Requirements

1. User Interfaces

Features like accessing the recommended products needs user interaction. Vendor's interaction is necessary when the user data is to be updated. Customer interaction is needed to access the web page. User interface will be required to maintain database i.e. records of customer, vendor, products and deals.

2. Hardware Interfaces

The product can work on any internet connected mobile device. These devices should have some limited requirements to make the application run effectively. We expect the processor speed and internet speed are high because application always needs internet.

3. Software Interfaces

The input MAC address will be given by ARP protocol. The basic information for every customer will be provided by QR code. The other software interface needed is back-end database as data needed to be stored according to the customer's MAC address.

As back-end database include records, records of customer, vendor, products and deals. These data should be produced as soon as a request for the records is made.

4. Communications Interfaces

Web Browser required are Internet Explorer version 8.0+ or Mozilla Firefox version 40.0+. ARP protocol will be required for acquiring the MAC address. HTTP protocol as standard communication protocol.

2.1.4 System Features

- Recommendation Engine
  - Description and Priority
    - It is a kind of Decision Support System.
    - It will be implemented on the basis of Content-Based and Collaborative Filtering Algorithm.
    - This module has highest priority.
  - Stimulus/Response Sequences
    - System will produce recommendations on the basis of user's history or most trending products.
    - User will access the recommended product.
  - Functional Requirements
    - Handle Recommendation Requests.
    - Generate Precise Recommendation on limited but multi-criteria. approach.
- Security and Authentication
  - Description and Priority
    - To avoid unauthorized access this module is important.
    - The module is based on client-server architecture.
    - Authentication will be provided using MAC addresses.
  - Stimulus/Response Sequences
    - User will get connected to the network.
    - MAC address will be acquired by using ARP protocol for authentication.

- Functional Requirements
    - Signup module will be required for adding new vendors to the system.
    - Login module will be required by vendors for updating their offers.
    - If a vendor is not able to remember the password then, forgot password will be required for changing the password.

- QR Code Generator and Decryptor
    - Description and Priority
    - It will work as unique identification for each and every customer.
    - All the basic information about the customer will be encoded using QR code.
    - Vendors will update any user's data using QR code only.
    - Stimulus/Response Sequences
    - QR code will be generated for every user whenever for the first time the user will input the basic information.
    - QR code will be scanned by vendors using QR code scanners to update the record of any user.
    - Functional Requirements
    - Generate QR code for every user.
    - A module will be required for scanning QR code.
    - A decryption module is required to decrypt the scanned QR code and access the information.

- House Keeping and User Data
    - Description and Priority
    - It is responsible for storing and managing the information about the sessions.
    - Proper data is to be fed to recommendation module for accurate results.
    - It monitors the entire system.

  - Stimulus/Response Sequences
    * Stores all the records.
    * Feeds Proper data to recommendation module.
  - Functional Requirements
    * A module is required for updating the data.
    * A module is required for checking the data ambiguity.
    * A module is required for deleting the records older than 2 years.

- Result Generation and Optimization
  - Description and Priority
    * Responsible for Collective Intelligence to Filter Best Choice.
    * Converting results to proper readable format.
    * Also maintains the flow.
  - Stimulus/Response Sequences
    * Takes output of recommendation engine module as input.
    * Optimizes the result and then displays it.
  - Functional Requirements
    * Optimize the result and convert it to the most readable format.
    * Results to be displayed as various recommendations on user's screen.

2.1.5 Other Nonfunctional Requirements

(a) Safety Requirements

Records of customers are very important. Hence, loss of these information is not tolerable at any cost. To prevent this, backup will be performed on daily basis. This feature will help to prevent the possible loss of information in any situation. Backup will performed automatically on daily basis.

(b) Security Requirements

A proper authentication mechanism should be there to prevent the illegal use of the product.

As it may happen that unauthenticated people may try to access it illegally.

(c) Software Quality Attributes

Other supportable software quality attributes are:-

1. Reusablity:-

Some modules in the product will be reused from other softwares.

2. Reliablity:-

Software will be reliable enough as it will produce precised and accurate results.

3. Portablity:-

Software will be portable from one system to another

4. Robustness:-

Software will be robust to prevent the unauthenticated use.

2.2 Project Process Modeling

The process model adopted for this project is Waterfall Model. In this model, classic life cycle paradigm begin at system level and goes through Analysis, Design, Implementation, Testing and Deployment and Maintenance Task Sets. Waterfall Model is a systematic sequential approach for software development. Hence it is most widely used paradigm for software development.

2.3 Project Scheduling

2.3.1 Time Line Chart

The time line chart helps us to get the idea about how the progress takes place while development of the software. We have identified the following set of required activities for the project:

- Topic Analysis
- Literature Survey of the selected topic
- Mathematical Modeling using Set Theory
- Feasibility Analysis using NP Theory
- Design Modeling
- Implementation and Coding
- Software Testing

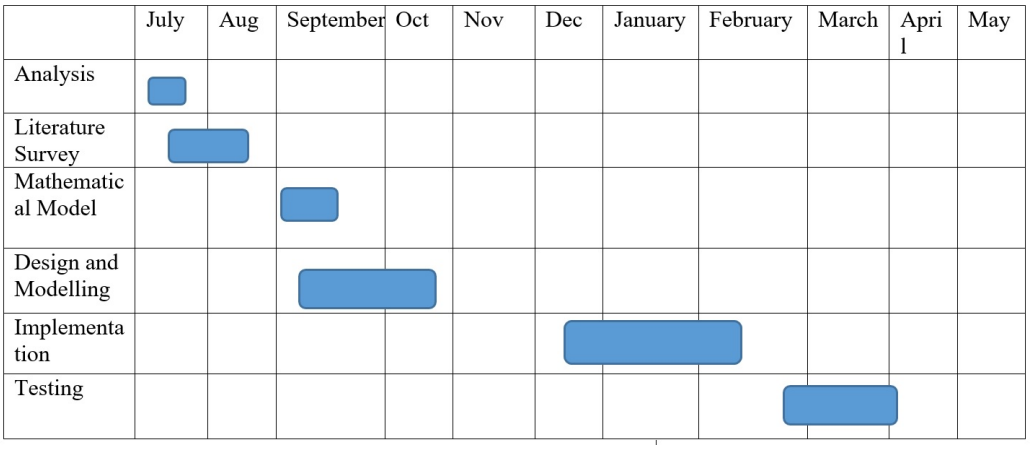

Figure 2.1: Time Line Chart

CHAPTER 3

ANALYSIS & DESIGN

3.1 IDEA matrix

- Increase
  - Performance
    - More precise Recommendation.
  - Portability
    - Portability for multiple datatypes.
- Improve
  - Reliability
    - Decision support mechanism based on different choices created by recommendation engine.
  - Security
    - Tightly coupled integration of modules.
  - Data overflow management.
    - Collaboration of module based and memory based filtering.
- Ignored
  - Storage
  - Computing power
    - Metadata generated by not logged in users or devices.
- Invent
  - Middleware
    - Hybird appproach based on both collaboration and content based filtering.
- Deliver
  - Recommendation based on user data and user history.

    - Appropriate recommendation based on multi-criteria collective intelligence recommendation system.

- Decrease
  - Cost.
    - Reduce in large data manupulation and filtering.
  - Hardware
    - Reduction in inappropriate suggesstions.
  - Man-power

- Evaluate
  - Pattern recognization
    - Uninterrupted evalution of recommendation generated by system to monitor wheather system is working its potential.
  - Data control and manupulation
    - Evaluate user history to create better recommendation.
  - Throughput

- Eliminate
  - Decision support sysytem.
    - To eliminate or not precise suggesstions by recommendation engine.
  - Track Monitor
    - To study system updated to eliminate data vulnarability.

- Accelerate
  - QR-code related operations.
    - No. of accurate recommendations suggested by system.

  - Metadata supervision.
    * House keeping and history management.
  - QR-code deletion and decreption
- Associate
  - Content-based filtering.
    * Association of results of collaboration and content-based filtering.
  - Collaborative filtering.
    * Associate various user data to suggest better recommendation.
  - Data management
- Avoid
  - Misinterpretation and misintertion of unknown datatypes.

3.2 Mathematical Model (Algorithm and Methodology)

Let S be the System

Where, S = I,O,F,Success,Failure

Where I is Input Set of Data

I = I1,I2,I3,I4

Where,

I1 ⇒ User MAC Address

I2 ⇒ User Name

I3 ⇒ Mobile Number

I4 ⇒ User Data (Bought Products)

O is Set of Outputs

O = O1,O2,O3,. . . . . .,On

Where,

O1 to On is Recommendation Results

F is Set of Functions

F = F1,F2,F3,F4,F5

F1 ⇒ Recommendation Engine

F2 ⇒ QR Code Generator and Decyrptor

F3 ⇒ Security and Authentication

F4 ⇒ Result Optimization & Generation

F5 ⇒ History & User Data

Success= Appropriate Recommendations

Failure=System Failure , Irrelevant Recommendation,Technical Issues

<u>Mathematical Model for Authentication Module</u>

Let S be the System

Where, S = I,O,F,Success,Failure

Where I is Input Set of Data

I = I1

Where, I1 $\Rightarrow$ Ip Address

O is Set of Outputs

O = O1

Where O1 is MAC Address

F is Set of Functions

F = F1,F2

F1 $\Rightarrow$ Register

F2 $\Rightarrow$ Login

Success= Session started and ended properly, Accurate Address Detection

Failure=User Authentication Failure, Unable to detect address

Mathematical Model for QR Encryptor & Decryptor

Let S be the System

Where, S = I,O,F,Success,Failure

Where I is Input Set of Data

I = $\{I1, I2\}$

$Where, I1 \Rightarrow$ User Information

I2 $\Rightarrow$ MAC Id

O is Set of Outputs

O = O1

Where O1 is QR Code

F is Set of Functions

F = F1,F2

F1 $\Rightarrow$ Encrypt

F2 $\Rightarrow$ Decrypt

Success= QR code generated, Proper Information Retrieved

Failure=Wrong Information Retrival, QR Code generation failure

Mathematical Model for HouseKeeping & User Data

Let S be the System

Where, S = I, F,Success,Failure

Where I is Input Set of Data

I = I1,I2,I3,I4

Where, I1 ⇒ User Information

I2 ⇒ Results

I3 ⇒ History

I4 ⇒ Defragmentation Location

F is Set of Functions

F = F1,F2

F1 ⇒ Defragmentation

F2 ⇒ History Manager

F3 ⇒ Result Manager

F4 ⇒ Session Manager

Success= System working according to ACID properties

Failure=Abnormal termination of system working

Mathematical Model for Recommendation Engine

Let S be the System

Where, S = I,O,F,Success,Failure

Where I is Input Set of Data

I = I1,I2,I3

Where, I1 ⇒ User Information

I2 ⇒ Previous Purchases

I3 ⇒ Visited Objects

O is Set of Outputs

O = O1

Where O1 is Recommendation Result

F is Set of Functions

F = F1,F2,F3

F1 $\Rightarrow$ Demodule

F2 $\Rightarrow$ Pattern Matching

F3 $\Rightarrow$ Pool of Results

Success= Appropriate Recommendations

Failure=System Failure , Irrelevant Recommendation, Technical Issues

3.3 Architecture Diagram

The Architecture Diagram of any system represents internal modules of the systems. Given architecture diagram represents all modules of the system.

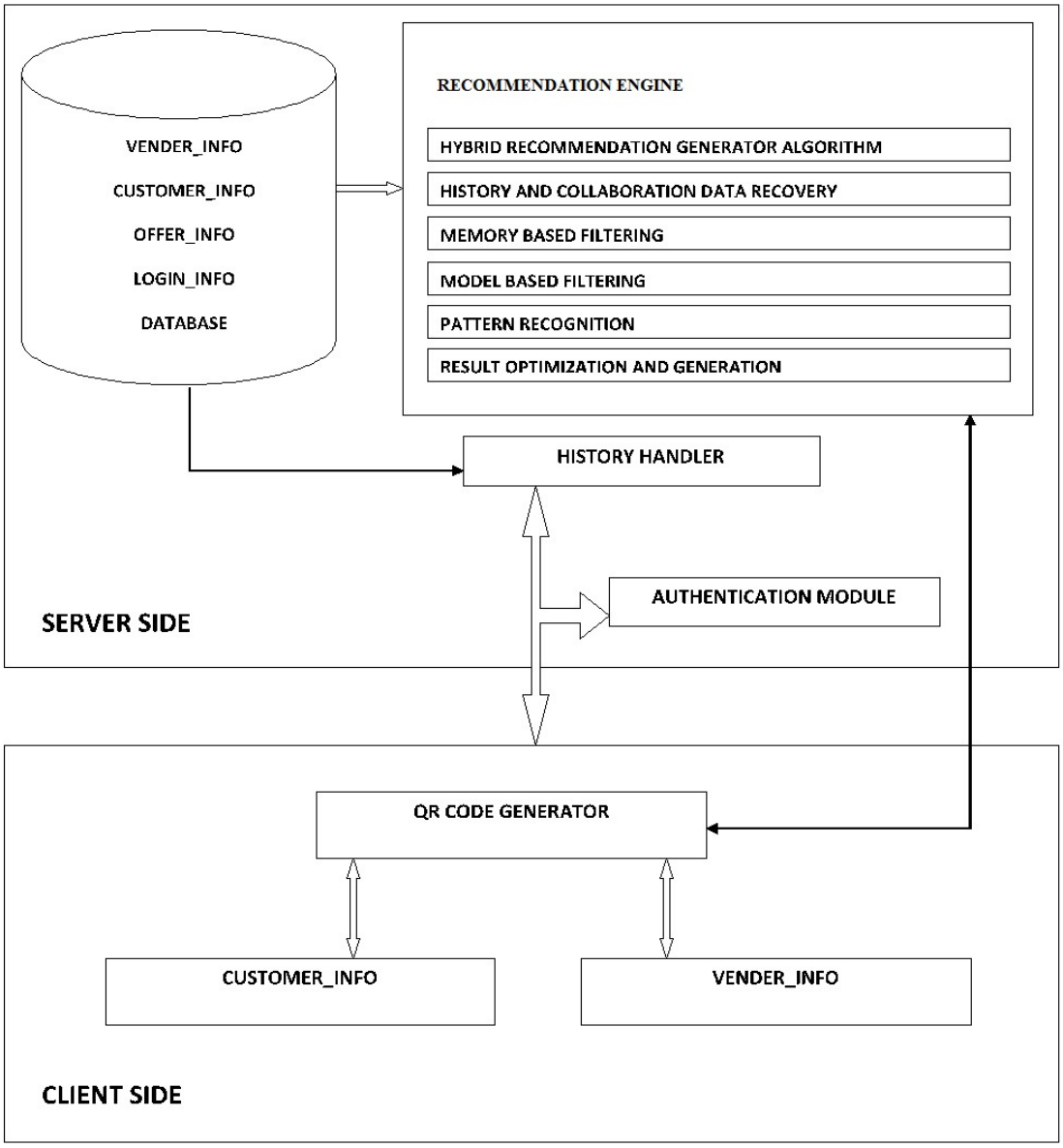

Figure 3.1: Architecture Diagram

The architecture comprises of the Server-Side and the Client-Side. Server-Side consists of the Database of the Customer and the Vendors, Recommendation engine, History handler and the Authentication Module. The Recommendation Engine performs or runs the activities as Hybrid Recommendation algorithm which is combining the Collaboration Filtering and the

Content-Based Filtering. Also includes Data recovery, Memory based Filtering, Model-based Filtering algorithms, pattern recognition and result generation and optimization. Client side include the QR-Code which is passed to the server for the authentication. On the client-side the on formation of the Customer and the Vendors are displayed.

3.4 UML Diagrams

3.4.1 Use-Case Diagrams

Use-case Diagrams of any system represent various users of any system. Roles played by all Users into given any particular system Are Also Represented by the Use-Case Diagram.

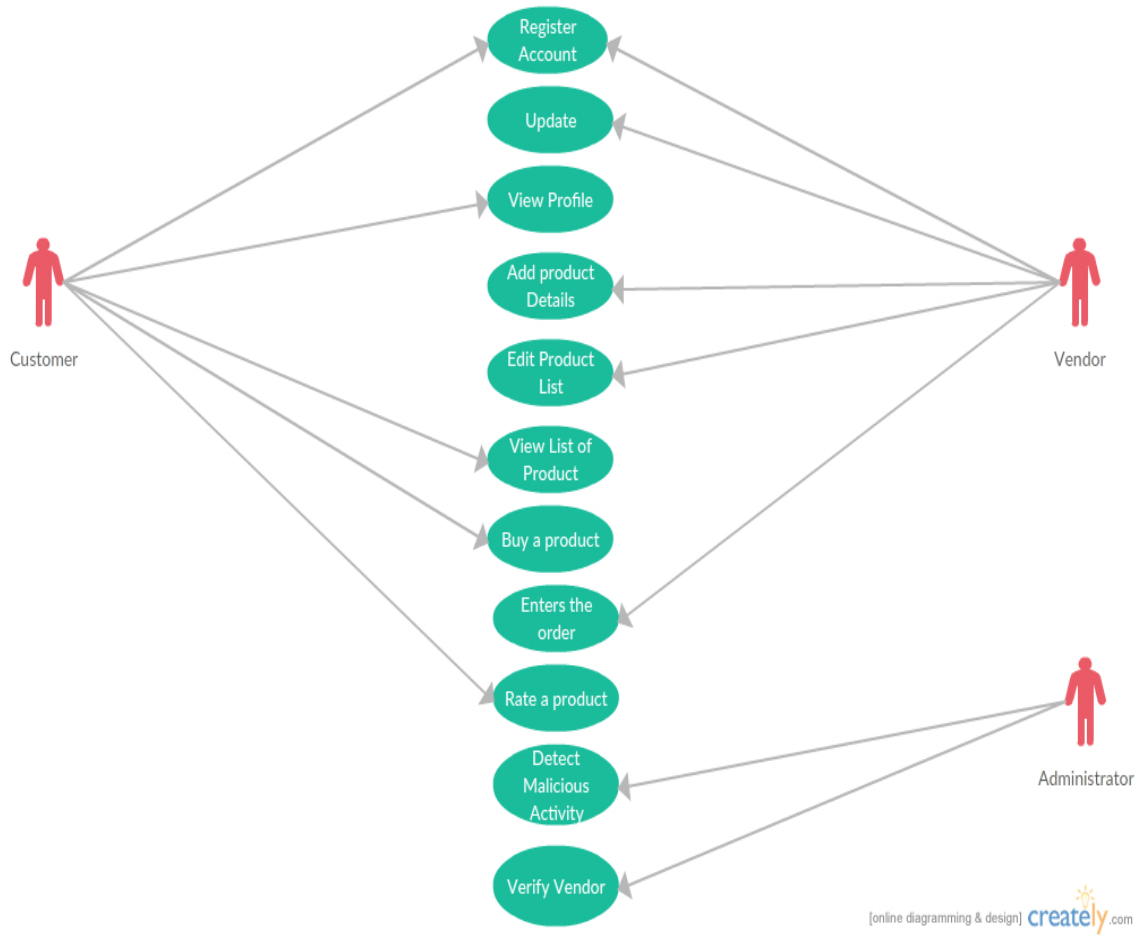

Figure 3.2: Use Case Diagram

The above use-case diagram describes the role of the three actors in the proposed system namely, Customer, Vendor and the Administrator. Customer as well as the Vendor must register their account at initial and if necessary can update it. The vendor is responsible for adding the

product and editing the offers. The Customer buys a product rates a product Administrator is responsible for managing data, detecting the malicious activities and verifying the vendor.

3.4.2 Class Diagrams

Class Diagram represents the classes and relationships between them. The given diagram represents all the classes of the System.

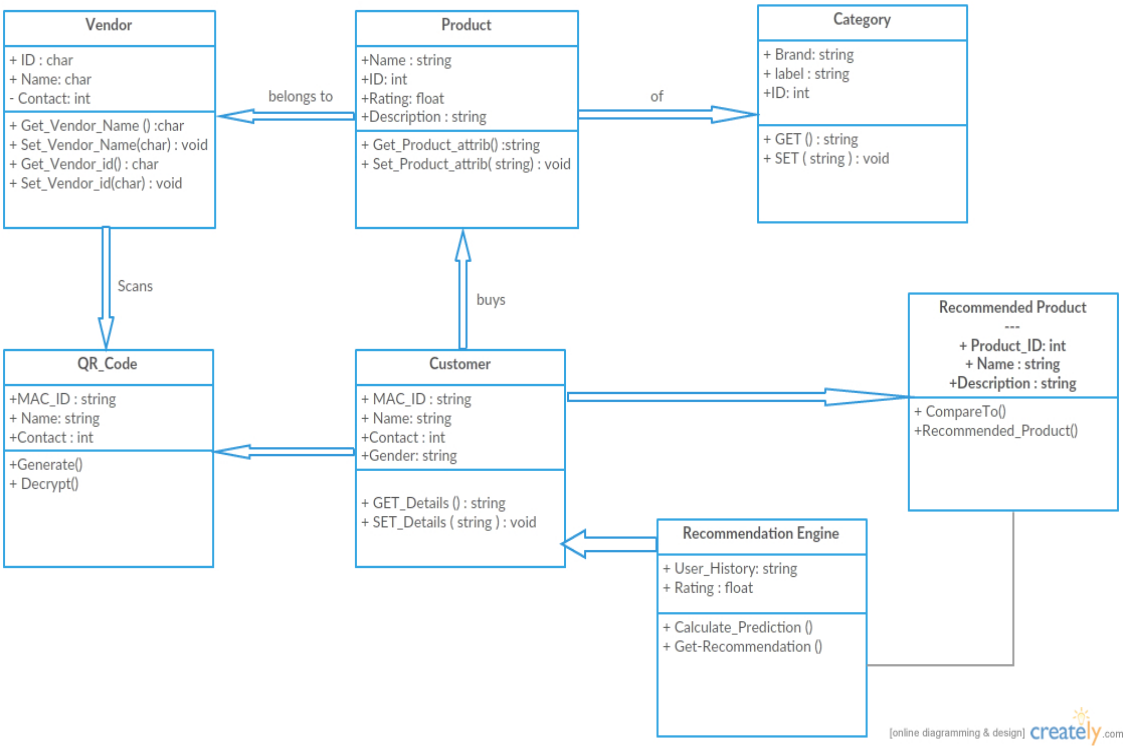

Figure 3.3: Class Diagram

The product includes the name of the product, id, rating and description about the product, these attributes are declared as public. There are two functions used getattrib() with return type string and setattrib() with return type void. The product belongs to the vendor which has the attributes as the id, name and the contact. The Product is of the category which is displayed with the brand, label and the id. The vendor scans the QR-Code of the customer which includes MAC-id, Name, and Contact of the customer. The Customer details are displayed as, Name, Mac-address, Contact, Gender etc. The recommendation engine processes the result and displays the Recommended Product to the customer.

3.4.3 ER Diagrams

E-R Diagrams represent the entities of system. Also the relationships between them are represented using various notations of E-R Diagram. The given diagram represents various Entities present in the system. The Customer buys many products. Customer has the attributes Contact,

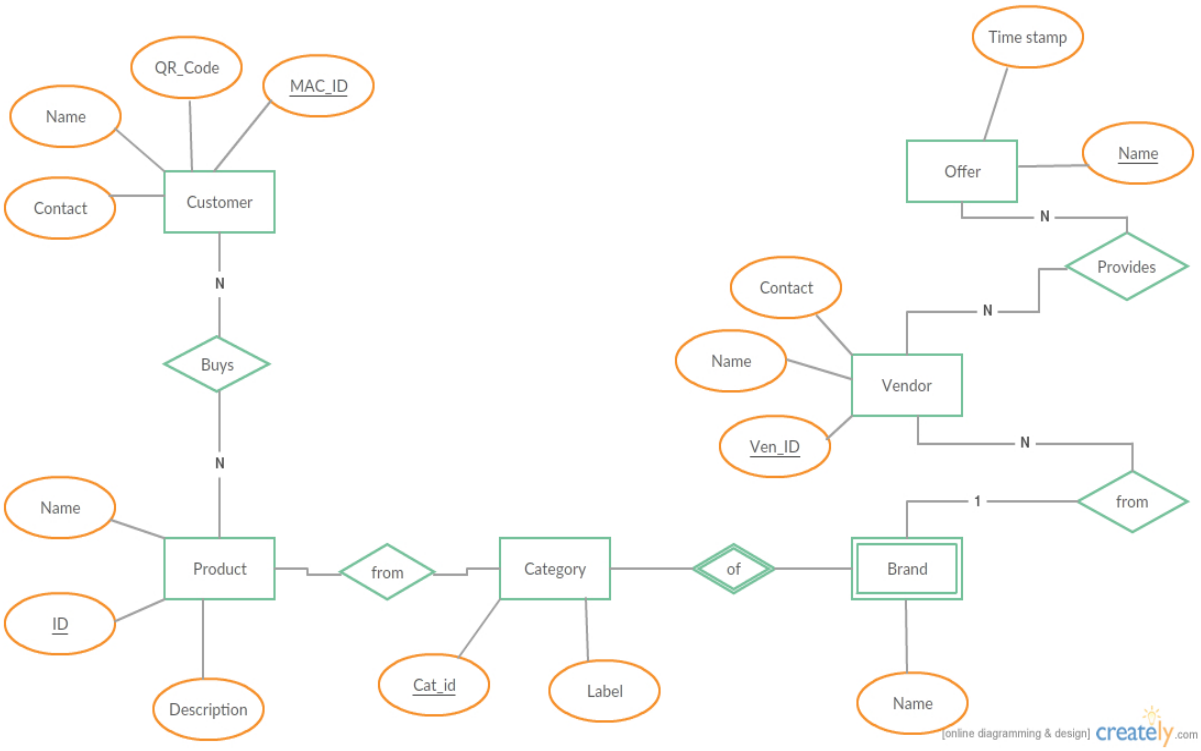

Figure 3.4: ER Diagram

Name, QR-Code, Mac-id is primary key. Product has the id as its primary key. The Product is of the particular Category. Category is identified with the primary key cat-id. The category is classified with the brands. Brand is the weak entity set with only one attribute as name. Many Vendors may have product from same brand. Vendor has primary key as ven-id. Many Vendors provide different offers classified with their names.

3.4.4 State Transition Diagrams

State Transition Diagram represents all states in the system. The system among all the states is represented using state diagram. The given below diagram shows all states of the system.

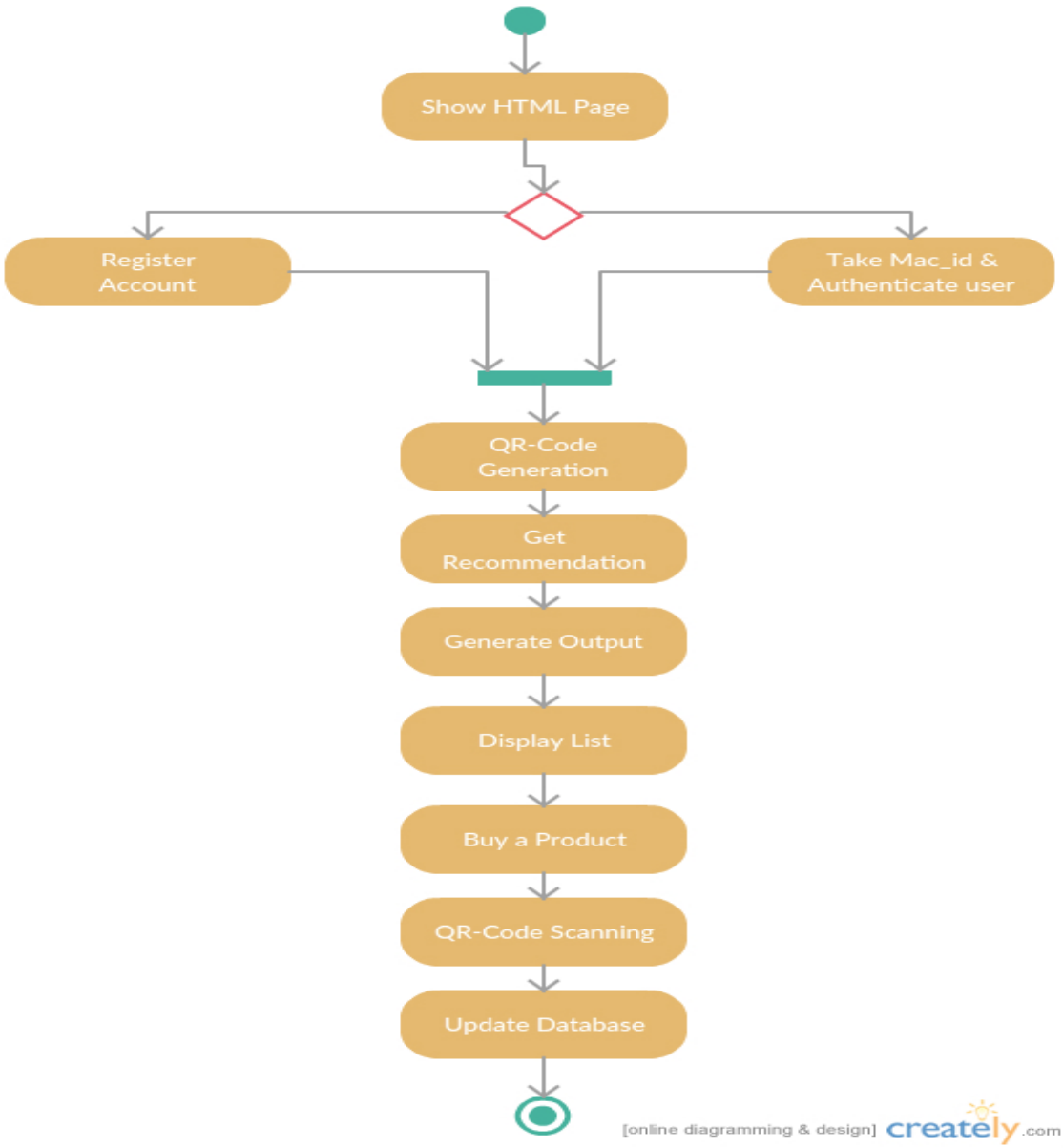

Figure 3.5: State Transition Diagram

The HTML page is displayed at initial. If the user is already registered the Mac address is taken and the user is authenticated. If not the user is registered then, the user must create the account. After, the QR-Code is generated for the particular user. Then, the user gets the recommendations based on the processed data and the output is generated or displayed to the

user. The customer buys a product. Entries are made in the Customer's account by scanning the QR-Code and Updating the database.

3.4.5 Deployment Diagrams

Deployment Diagrams represent all the components to be deployed at the time of deployment of the system. The given diagram represents the components of the system to be deployed De-

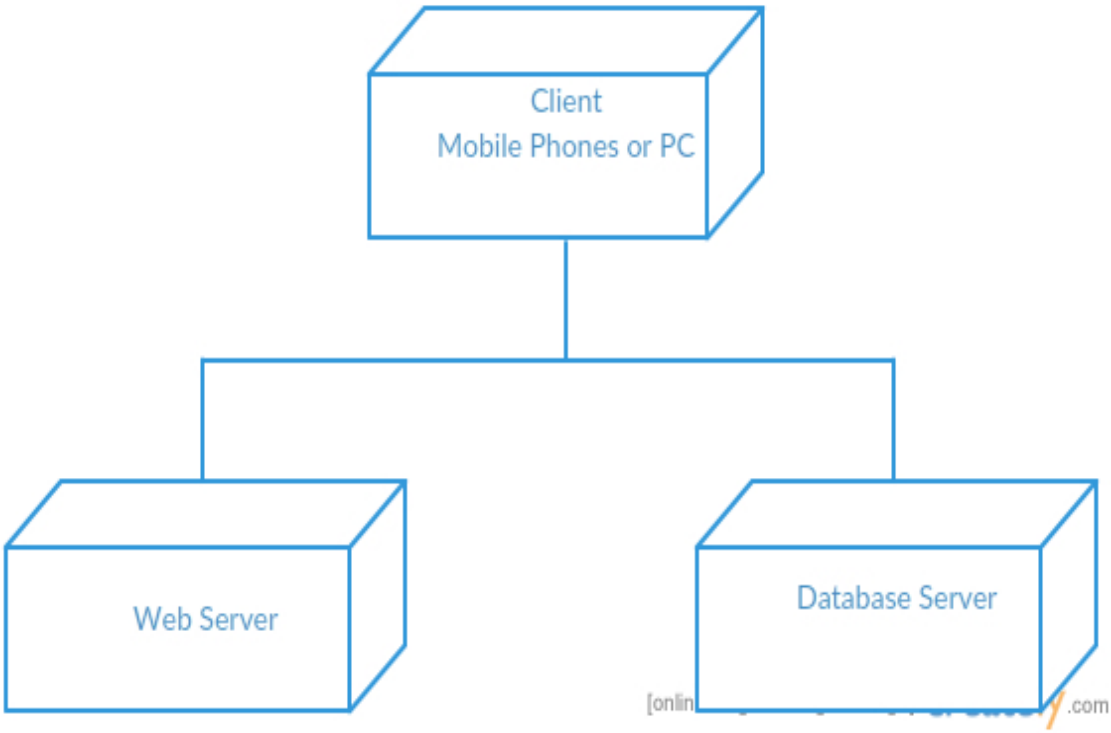

Figure 3.6: Deployment Diagram

ployment Diagram are used to describe the components required when the system is deployed to the end-user. The system requires the Web-Server, Database-Server and the Customers or Clients Mobile phones and PC. Database Server manages the Customer as well as the Vendors data. The Data processing is done by the database server. The Web-Server is required to manage the web pages and the user interface on the Customer and the Vendor side. The Client mobile phones or the PC are the used as the interface to the Web-server are the Database server.

CHAPTER 4

IMPLEMENTATION & CODING

4.1 Introduction

This chapter covers the listing of the code for major functionalities. Also covers the classes along with the responsibilities. For implementation purpose we need Java J2SE and JDK J2SE (Java 2 Standard Edition) Java is the language for development of the project. JDK is the development kit used to compile java programs. We have implemented Android application module in the eclipse platform and the server module is implemented in NetBeans platform. The Glassfish server is designed using various web services. Admin GUI and prediction algorithm is designed using NetBeans IDE. MySQL is used for designing database for ANN algorithm. Hardware NFC coding is done in both the android applications. Modules are connected using JDBC.

4.2 Database Schema

List of tables:

1. doctor
2. medecinelog
3. noofdiseases
4. patient
5. patientlog
6. remedies
7. symptoms
8. upload

4.2.1 Doctor Table

Did - Did is a doctor's unique identification number given to every doctor in a hospital.

Password - Password column contains password given to a particular doctor.

Fname - fname is first name of doctor.

Lname - lname is last name of doctor.

Doctor table is used for storing doctor's information in various columns.

did	password	fname	lname	address	mobile	gender	emailid

Table 4.1: doctor

4.2.2 Medicenelog Table

Did - Did is a doctor's unique identification number given to every doctor in a hospital.

MediName - This field contains name of the prescribed medicine given by the doctor with given Did.

MediType - This field contains type of medicine.

MediTimeMorn - This field shows the morning time of medicine intake.

MediTimeAft - This field shows the afternoon time of medicine intake.

MediTimeEven - This field shows the evening time of medicine intake.

Medicenelog table is used to store prescription and medicine timings given by the doctor and other detailing of medicine.

did	MediName	MediType	MediTimeMorn	MediTimeAft	MediTimeEven

Table 4.2: medicenelog

4.2.3 No of Diseases Table

No of disease table contains 10 diseases. This table is used by ANN algorithm to give output of the prediction. Patient will enter symptoms and ANN algorithm will choose one of the diseases from this table.

4.2.4 Patient Table

Pid - Pid is a patient's unique identification number given to every patient in a hospital.

LoginId - LoginId is patient's personal login name.

diseaseName
Hypertension
Malaria
Diabetes
Leprosy
Measles
Tuberculosis
Pneumonia
Hepatitis
Diarrhoea
Hypothyroidism

Table 4.3: noofdiseases

Symptoms - Symptoms is list of symptoms experienced by patients.

Patient table is used for storing patient's information in various columns on the basis of symptom field in the patient table, ANN will predict most probable disease. Patient table contains all the physical health details which can be useful for doctors in diagnosis.

pid	loginId	password	fName	symptoms	lName	address	mobile	gender	emailid

Table 4.4: patient

city	age	height	weight	bloodgroup

Table 4.5: patient(...contd)

4.2.5 Patient Log Table

Patient log table contains the details of patients visit to doctor. This table shows all the prescription details, dates of next visit, comments and report explanation.

lid	pid	logDate	prescription	reportDetails	precautions	nextVisit	comments	duration

Table 4.6: patientlog

4.2.6 Symptoms Table

This table contains 36 symptoms in total, which can be used for disease prediction. Once the patient selects symptoms from the list, the selected data is sent to ANN algorithm to predict the most probable disease.

4.3 Operational Details & Classes

4.3.1 Operations

Patient App:

Patient will be able to view prescription given by doctor. Alerts will be generated according to prescription timings for specific medicine. Application will identify NFC tags on medicine and patient will recognize the correct medicine by alert message generated on patient's application. Patient can upload reports to server using application.

Doctor App:

Doctor will be able to view patient or patient information. Doctor's application will have facility to give prescription and timing to any patient irrespective of the location. Doctor will be able to see reports sent by admin. Doctor will give NFCs for specific medicine. Doctor will be able to change/update the medicine or the prescription timings.

Admin:

Admin will add/manage patients' and doctors' entries. Apply Data mining using ANN for generating disease prediction using patient's history and symptoms. Admin will be able to view report uploaded by the patient. That report can be forwarded to the doctor.

Server:

Manage the database through request sent or received. Authentication process is done through server. The server used in this system will be home server.

Database:

Store the patients' information such as name, address, medical history, etc. Store the doctors' information such as name, contact, specialty, etc.

NFC:

NFC tags will be used to identify the medicines. Near field communication (NFC) tags are of two types, write once read multiple times and write read multiple times. NFC's working range is 4 inches. When a NFC enabled phone makes contact with NFC tag, it'll read the NFC tag. NFC tag will be assigned with unique id's according to particular medicine.

4.3.2 Major Classes

Patient Class :-

Login() : This function authenticates the patient. A password and user name field is given and accordingly patient is identified.

View prescription() : This function enables patient to view prescription given by doctor at anytime anywhere.

View Medicine TimeTable() : It shows timetable of medicines so that patient is able to view which medicine to take.

Alert when Medicine Time() : This function generate alarm at the time of medicine intake.

Upload report to doctor() : In this function, a facility of uploading documents like reports, scans to the doctor is given.

Identify medicine using NFC tag() : This function identifies the particular medicine which is to be taken with the help of NFC tags. Each NFC tag is given a specific ID which is used to identify the medicine.

Logout()

Doctor Class :-

View Patients List() : By this function doctor will get list of patients along with their details.

Give Prescription() : This function enables doctor to give prescription to patient on the app.

View History() : Doctor will be able to view the history of any patient.

Give Medicines with time() : Medicines will be given along with the timings.

Medicine with NFC code() : Each medicine is given a particular NFC tag which will identify the medicine.

Admin Class :-

Register Patient() : Patient will be registered along with all the details.

Manage Patient() : Patient can be deleted and its details can be updated.

Add Doctor() : Admin will register doctor along with all the details.

Manage Doctor() : Doctor can be deleted and details can be updated.

Apply DM on Symptoms() : Patient symptoms will be given as a input to the prescription algorithm.

Prediction() : ANN algorithm will predict the most probable disease according to the patient's symptoms.

Server Class :-

Add all Details() : It will add all the parameters received from admin GUI to the database.

Receive Request() : The applications will request data for the different operation to the server.

Send reply() : Server will send required data to the applications.

Authentication() : It will check authentication of user.

Database Class :-

Add() : It will add doctors and patients entry in the respective tables.

Remove() : It will remove doctors and patients entry in the respective tables.

Update() : It will update doctors and patients entry in the respective tables.

4.3.3 Code Listing

Major part of the coding contains reading and writing the NFC tags and ANN algorithm for disease Prediction.

```java
/*
//==============================================================

This file contains the code to read the NFC tags
on Patient's app.

=============ReadNfcActivity.java=============

*/

package com.example.mediboxpatient;

import java.util.Arrays;

import android.nfc.NdefMessage;
import android.nfc.NdefRecord;
import android.nfc.NfcAdapter;
import android.nfc.Tag;
import android.nfc.tech.NfcF;
import android.os.Bundle;
import android.os.Parcelable;
import android.app.Activity;
import android.app.PendingIntent;
import android.content.Intent;
import android.content.IntentFilter;
import android.util.Log;
import android.widget.TextView;

public class ReadNfcActivity extends Activity {
```

```java
private TextView mTextView;
private NfcAdapter mNfcAdapter;
private PendingIntent mPendingIntent;
private IntentFilter[] mIntentFilters;
private String[][] mNFCTechLists;
String FINALDATA = "";

@Override
protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_read_nfc);
        mTextView = (TextView) findViewById(R.id.tvisisi);
        mNfcAdapter = NfcAdapter.getDefaultAdapter(this);
        if (mNfcAdapter != null) {
                mTextView.setText("Read an NFC tag");
        } else {
                mTextView.setText("This phone is not NFC enabled.");
        }
        // create an intent with tag data and deliver to this activity
        mPendingIntent = PendingIntent.getActivity(this, 0,new Intent(this,
            getClass()).addFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP),0);
        // set an intent filter for all MIME data
        IntentFilter ndefIntent = new IntentFilter(
                NfcAdapter.ACTION_NDEF_DISCOVERED);
        try {
                ndefIntent.addDataType("*/*");
                mIntentFilters = new IntentFilter[] { ndefIntent };
        } catch (Exception e) {
                Log.e("TagDispatch", e.toString());
```

```
        }
        mNFCTechLists = new String[][] { new String[] { NfcF.class.getName()
}

@Override
public void onNewIntent(Intent intent) {
        String action = intent.getAction();
        Tag tag = intent.getParcelableExtra(NfcAdapter.EXTRA_TAG);
        String s = action + "\n\n" + tag.toString();
        // parse through all NDEF messages and their records and pick text t
        Parcelable[] data = intent
                .getParcelableArrayExtra(NfcAdapter.EXTRA_NDEF_MESSAGES);
        if (data != null) {
                try {
                        for (int i = 0; i < data.length; i++) {
                                NdefRecord[] recs = ((NdefMessage) data[i]).g
                                for (int j = 0; j < recs.length; j++) {
                                        if (recs[j].getTnf() == NdefRecord.TN
                                                && Arrays.equals(recs[j].getT
                                                        NdefRecord.RTD_TEXT))
                                                byte[] payload = recs[j].getP
                                                String textEncoding =
                                                    ((payload[0] & 0200) == 0
                                                        : "UTF-16";
                                                int langCodeLen = payload[0]
                                                s += ("\n\nNdefMessage["+ i+
                                                    NdefRecord["+ j+ "]:\n\""
                                                    String(payload, langCodeL
                                                    payload.length - langCode
```

```
                                                                textEncoding) + "\"")
                                                        FINALDATA = new String(payloa
                                                            langCodeLen + 1,
                                                                payload.length - lang
                                                                textEncoding);
                                                }
                                        }
                                }
                        } catch (Exception e) {
                                Log.e("TagDispatch", e.toString());
                        }
                }
                System.out.println("FINAL DATA IS ************************ "
                                + FINALDATA);
                mTextView.setText(s);
        }

        @Override
        public void onResume() {
                super.onResume();

                if (mNfcAdapter != null)
                        mNfcAdapter.enableForegroundDispatch(this, mPendingIntent,
                                mIntentFilters, mNFCTechLists);
        }

        @Override
        public void onPause() {
                super.onPause();
```

```
                if (mNfcAdapter != null)
                        mNfcAdapter.disableForegroundDispatch(this);
        }
}

//==========================================================

/*

This file contains the code to write NFC tags
on doctor's app.

=============NfcActivity.java=============

*/

package com.example.doctorapp;

import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.io.ObjectInput;
import java.io.ObjectInputStream;
import java.io.ObjectOutput;
import java.io.ObjectOutputStream;
import java.io.UnsupportedEncodingException;
import java.util.Calendar;

import org.ksoap2.SoapEnvelope;
```

```
import org.ksoap2.serialization.PropertyInfo;
import org.ksoap2.serialization.SoapObject;
import org.ksoap2.serialization.SoapSerializationEnvelope;
import org.ksoap2.transport.HttpTransportSE;

import LibPack.mediDescription;
import android.nfc.FormatException;
import android.nfc.NdefMessage;
import android.nfc.NdefRecord;
import android.nfc.NfcAdapter;
import android.nfc.Tag;
import android.nfc.tech.Ndef;
import android.os.AsyncTask;
import android.os.Bundle;
import android.app.Activity;
import android.app.Dialog;
import android.app.PendingIntent;
import android.app.TimePickerDialog;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.TimePicker;
```

```java
import android.widget.Toast;

public class NfcActivity extends Activity {

        NfcAdapter adapter;
        PendingIntent pendingIntent;
        IntentFilter writeTagFilters[];
        boolean writeMode;
        Tag mytag;
        int did, pid;
        Context ctx;
        Button btWrite, btRead;
        EditText medName;
        TextView txtmorn, txtAft, txtEven;
        Spinner spinType;
        String MedicineType[], DAYTIME = "MORNING", FinalData = "", RESPONSE = "NO";
        static final int TIME_DIALOG_ID = 1111;
        private TextView output;
        public Button btnClick;
        mediDescription medi;
        private int hour;
        private int minute;

        @Override
        protected void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.activity_nfc);
                ctx = this;
                btWrite = (Button) findViewById(R.id.btwriteData);
```

```
medName = (EditText) findViewById(R.id.edtMedicineName);
btRead = (Button) findViewById(R.id.btreadData);
spinType = (Spinner) findViewById(R.id.spintype);
txtmorn = (TextView) findViewById(R.id.text1);
txtAft = (TextView) findViewById(R.id.text2);
txtEven = (TextView) findViewById(R.id.text3);
Bundle bdata = getIntent().getExtras();
did = bdata.getInt("did");
pid = bdata.getInt("pid");
System.out.println("Val of PID:****************" + pid);
System.out.println("Oncreate...");
MedicineType = new String[4];
MedicineType[0] = "Tablet";
MedicineType[1] = "Syrup";
MedicineType[2] = "Injection";
MedicineType[3] = "Drop";
spinType.setAdapter(new ArrayAdapter<String>(ctx,
                android.R.layout.simple_list_item_1, MedicineType));
output = (TextView) findViewById(R.id.output);
/********* display current time on screen Start ********/
final Calendar c = Calendar.getInstance();
// Current Hour
hour = c.get(Calendar.HOUR_OF_DAY);
// Current Minute
minute = c.get(Calendar.MINUTE);
// set current time into output textview
/********* display current time on screen End ********/
// Add Button Click Listener
addButtonClickListener();
```

```
btRead.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {
                Intent i = new Intent(getApplicationContext(),
                                ReadNfcActivity.class);
                startActivity(i);
        }
});
btWrite.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
                LongOperationAddMediLog lo1 = new LongOperationAddMed
                lo1.execute("");
                try {
                        FinalData = medName.getText().toString() + "#
                                + spinType.getSelectedItem().toString
                                + txtmorn.getText().toString() + "#"
                                + txtAft.getText().toString() + "#"
                                + txtEven.getText().toString();
                        System.out.println("FINAL
                            DATA IS****************************** "+
                        if (mytag == null) {
                                Toast.makeText(ctx, "Error Detected",
                                    Toast.LENGTH_LONG).show();
                        } else {
                                write(FinalData, mytag);
                                Toast.makeText(ctx, "Ok Writing",
                                    Toast.LENGTH_LONG).show();
                        }
```

```
                        } catch (Exception e) {
                                Toast.makeText(ctx, "Error Writing",
                                    Toast.LENGTH_LONG).show();
                                e.printStackTrace();
                        }
                }
        });
        adapter = NfcAdapter.getDefaultAdapter(this);
        pendingIntent = PendingIntent.getActivity(this, 0, new Intent(this,
                        getClass()).addFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP)
        IntentFilter tagDetected = new IntentFilter(
                        NfcAdapter.ACTION_TAG_DISCOVERED);
        tagDetected.addCategory(Intent.CATEGORY_DEFAULT);
        writeTagFilters = new IntentFilter[] { tagDetected };
}
public void FillInfo() {
        medi.pid = pid;
        medi.mediName = medName.getText().toString();
        medi.mediType = spinType.getSelectedItem().toString();
        if (txtmorn.getText().equals("null")) {
                medi.mediMornTime = "No";
        } else {
                medi.mediMornTime = txtmorn.getText().toString();
        }
        if (txtAft.getText().equals("null")) {
                medi.mediAftTime = "No";
        } else {
                medi.mediAftTime = txtAft.getText().toString();
        }
```

```
        if (txtEven.getText().equals("null")) {
                medi.MediEvenTime = "No";
        } else {
                medi.MediEvenTime = txtEven.getText().toString();
        }
}
public void addButtonClickListener() {
        btnClick = (Button) findViewById(R.id.btnClick);
        btnClick.setOnClickListener(new OnClickListener() {
                @Override
                public void onClick(View v) {
                        showDialog(TIME_DIALOG_ID);
                }
        });
}
@Override
protected Dialog onCreateDialog(int id) {
        switch (id) {
        case TIME_DIALOG_ID:
                // set time picker as current time
                return new TimePickerDialog(this, timePickerListener, hour, m
                                false);
        }
        return null;
}
private TimePickerDialog.OnTimeSetListener timePickerListener = new
    TimePickerDialog.OnTimeSetListener() {
        @Override
        public void onTimeSet(TimePicker view, int hourOfDay, int minutes) {
```

```
                // TODO Auto-generated method stub
                hour = hourOfDay;
                minute = minutes;
                updateTime(hour, minute);
        }
};
private static String utilTime(int value) {
        if (value < 10)
                return "0" + String.valueOf(value);
        else
                return String.valueOf(value);
}
// Used to convert 24hr format to 12hr format with AM/PM values
private void updateTime(int hours, int mins) {
        String timeSet = "";
        if (hours > 12) {
                hours -= 12;
                timeSet = "PM";
        } else if (hours == 0) {
                hours += 12;
                timeSet = "AM";
        } else if (hours == 12)
                timeSet = "PM";
        else
                timeSet = "AM";
        String minutes = "";
        if (mins < 10)
                minutes = "0" + mins;
        else
```

```
                minutes = String.valueOf(mins);
            // Append in a StringBuilder
            String aTime = new StringBuilder().append(hours).append(':')
                            .append(minutes).append(" ").append(timeSet).toString
            int temptime = hours;
            String AMPM = timeSet;
            output.setText(aTime);
            if (temptime >= 7 && AMPM.equalsIgnoreCase("PM") || temptime == 12
                            && AMPM.equalsIgnoreCase("AM") || temptime < 5
                            && AMPM.equalsIgnoreCase("AM")) {
                DAYTIME = "-EVENING";
                txtEven.setText("" + aTime + DAYTIME);
            } else if (temptime >= 5 && AMPM.equalsIgnoreCase("AM")
                            || temptime <= 11 && AMPM.equalsIgnoreCase("AM")
                            || temptime == 12 && AMPM.equalsIgnoreCase("PM")) {
                DAYTIME = "-MORNING";
                txtmorn.setText("" + aTime + DAYTIME);
            } else if (temptime <= 12 && AMPM.equalsIgnoreCase("PM")
                            || temptime <= 7 && AMPM.equalsIgnoreCase("PM")) {
                DAYTIME = "-AFTERNOON";
                txtAft.setText("" + aTime + DAYTIME);
            }
        }
        private void write(String text, Tag tag) throws IOException,FormatException {
            NdefRecord[] records = { createRecord(text) };
            NdefMessage message = new NdefMessage(records);
            // Get an instance of Ndef for the tag.
            Ndef ndef = Ndef.get(tag);
            // Enable I/O
```

```
            ndef.connect();
            // Write the message
            ndef.writeNdefMessage(message);
            // Close the connection
            ndef.close();
        }
        private NdefRecord createRecord(String text)
                    throws UnsupportedEncodingException {
            String lang = "en";
            byte[] textBytes = text.getBytes();
            byte[] langBytes = lang.getBytes("US-ASCII");
            int langLength = langBytes.length;
            int textLength = textBytes.length;
            byte[] payload = new byte[1 + langLength + textLength];
            // set status byte (see NDEF spec for actual bits)
            payload[0] = (byte) langLength;
            // copy langbytes and textbytes into payload
            System.arraycopy(langBytes, 0, payload, 1, langLength);
            System.arraycopy(textBytes, 0, payload, 1 + langLength, textLength);
            NdefRecord recordNFC = new NdefRecord(NdefRecord.TNF_WELL_KNOWN,
                            NdefRecord.RTD_TEXT, new byte[0], payload);
            return recordNFC;
        }
        public class LongOperationAddMediLog extends AsyncTask<String, Void, String>
            @Override
            protected void onPreExecute() {
                    // TODO Auto-generated method stub
                    super.onPreExecute();
                    medi = new mediDescription();
```

```
        FillInfo();
    }

    @Override
    protected String doInBackground(String... params) {
        System.out.println("Inside background method");
        SoapObject request = new SoapObject(WelcomeActivity.NAMESPACE
                        "AddMedicenes");
        System.out.println("after namespace");
        SoapSerializationEnvelope envelope = new SoapSerializationEnv
                        SoapEnvelope.VER11);
        PropertyInfo pi = new PropertyInfo();
        pi.setName("eIn");
        pi.setValue(objectToString(medi));
        pi.setType(String.class);
        request.addProperty(pi);
        System.out.println("After property");
        envelope.setOutputSoapObject(request);
        System.out.println("after envelope");
        HttpTransportSE androidHttpTransport = new HttpTransportSE(
                        WelcomeActivity.URL);
        try {
            androidHttpTransport
                            .call(WelcomeActivity.SOAP_ACTION, en
            SoapObject resultsRequestSOAP = (SoapObject) envelope
            String resp = resultsRequestSOAP
                    .getPrimitivePropertyAsString("return");
            RESPONSE = resp;
            System.out.println("resposne-------------------------
```

```
                                        + resp);
                } catch (Exception e) {
                        e.printStackTrace();
                }
                return null;
        }

        @Override
        protected void onPostExecute(String result) {
                // TODO Auto-generated method stub
                super.onPostExecute(result);
                if (RESPONSE.equalsIgnoreCase("YES")) {
                        Toast.makeText(getApplicationContext(), "DONE..!!",
                                        Toast.LENGTH_SHORT).show();
                } else {
                        Toast.makeText(getApplicationContext(), "Adding Error
                                        Toast.LENGTH_SHORT).show();
                }
        }
}
Object stringToObject(String inp) {
        byte b[] = Base64.decode(inp);
        Object ret = null;
        try {
                ByteArrayInputStream bis = new ByteArrayInputStream(b);
                ObjectInput in = new ObjectInputStream(bis);
                ret = in.readObject();
                bis.close();
                in.close();
```

```
        } catch (Exception e) {
                System.out.println("NOT DE-SERIALIZABLE: " + e);
        }
        return ret;
}
String objectToString(Object obj) {
        byte[] b = null;
        try {
                ByteArrayOutputStream bos = new ByteArrayOutputStream();
                ObjectOutput out = new ObjectOutputStream(bos);
                out.writeObject(obj);
                b = bos.toByteArray();
        } catch (Exception e) {
                System.out.println("NOT SERIALIZABLE: " + e);
        }
        return Base64.encode(b);
}
@Override
protected void onNewIntent(Intent intent) {
        if (NfcAdapter.ACTION_TAG_DISCOVERED.equals(intent.getAction())) {
                mytag = intent.getParcelableExtra(NfcAdapter.EXTRA_TAG);
                Toast.makeText(this, "Ok Detection" + mytag.toString(),
                                Toast.LENGTH_LONG).show();
        }
}
@Override
public void onPause() {
        super.onPause();
        WriteModeOff();
```

```
        }
        @Override
        public void onResume() {
                super.onResume();
                WriteModeOn();
        }
        private void WriteModeOn() {
                writeMode = true;
                adapter.enableForegroundDispatch(this, pendingIntent, writeTagFilters
        }
        private void WriteModeOff() {
                writeMode = false;
                adapter.disableForegroundDispatch(this);
        }
}

//==========================================================

/*

This file contains the algorithm for disease
prediction using ANN.

=============NeuralNetwork.java=============

*/

package algoPack;
```

```java
import java.io.Serializable;

public class NeuralNetwork implements Serializable{
    public SingleLayer layers[];
    public int totalLayers;
    public double learningRate;
    public int neuronInEachLayers[];
    //called from OpinionAnalysis_ANN-->ann_opinion()
    public NeuralNetwork(double learningRate, int totalLayers, int neuronInEachLayers
       // System.out.println("Layers:  "+totalLayers);
        this.totalLayers = totalLayers;
        this.neuronInEachLayers = neuronInEachLayers;
        this.learningRate = learningRate;
        layers = new SingleLayer[totalLayers];
        for (int i = 0; i < totalLayers; i++) {
            if (i != totalLayers - 1) {
                layers[i] = new SingleLayer(neuronInEachLayers[i], neuronInEachLayers
            } else {
                layers[i] = new SingleLayer(neuronInEachLayers[i], 0);
            }
        }
    }
    public void setInputs(double inputs[]) {
        for (int i = 0; i < neuronInEachLayers[0]; i++) {
            layers[0].neurons[i].value = inputs[i];
        }
    }
    double limiter(double x) {
        return (1.0 / (1 + Math.exp(-x)));
```

```
}
public double[] runNetwork() {
    double outputs[];
    int i, j, k;
    outputs = new double[neuronInEachLayers[totalLayers - 1]]; // Temp ouput arra
    for (i = 1; i < totalLayers; i++) {
        for (j = 0; j < neuronInEachLayers[i]; j++) {
            layers[i].neurons[j].value = 0;
            for (k = 0; k < neuronInEachLayers[i - 1]; k++) {
                layers[i].neurons[j].value = layers[i].neurons[j].value +
                layers[i - 1].neurons[k].value * layers[i - 1].neurons[k].weights
                //Multiply and add all the inputs
            }
            layers[i].neurons[j].value
                = layers[i].neurons[j].value + layers[i].neurons[j].bias; //Add b
            layers[i].neurons[j].value = limiter(layers[i].neurons[j].value);
                // Squash that value
        }
    }
    for (i = 0; i < neuronInEachLayers[totalLayers - 1]; i++) {
        outputs[i] = layers[totalLayers - 1].neurons[i].value;
    }
    return outputs; //Return the outputs
}
double sigmaWeightDelta(int layer_no, int neuron_no) {
//Calculate sum of weights * delta. Used in back prop.
//layer_no is layer number. Layer number and
//neuron number can be zero
    double result = 0;
```

```
        for (int i = 0; i < neuronInEachLayers[layer_no + 1]; i++) {
        //Go through all the neurons in the next layer
            result = result + layers[layer_no].neurons[neuron_no].weights[i] *
            layers[layer_no + 1].neurons[i].delta; //Comput the summation
        }
        return result;
    }
    public void train(double inputs[], double outputs[]) {
    //The standard Backprop Learning algorithm
        int i, j, k;
        double target, actual, delta;
        setInputs(inputs); //Set the inputs
        runNetwork(); //Update all the values
        for (i = totalLayers - 1; i > 0; i--) { //Go from last layer to first layer
            for (j = 0; j < neuronInEachLayers[i]; j++) {//Go thru every neuron
                if (i == totalLayers - 1) { //Output layer, Needs special atential
                    target = outputs[j]; //Target value
                    actual = layers[i].neurons[j].value; //Actual value
                    delta = (target - actual) * actual * (1 - actual); //Function to
                    layers[i].neurons[j].delta = delta; //Compute the delta
                    for (k = 0; k < neuronInEachLayers[i - 1]; k++) {
                        layers[i - 1].neurons[k].weights[j] +=
                        delta * learningRate * layers[i - 1].neurons[k].value;
                        //Calculate the new weights
                    }
                    layers[i].neurons[j].bias = layers[i].neurons[j].bias +
                    delta * learningRate * 1; //n_value is always 1 for bias
                } else { // for all intermediate layers
                    //Target value
```

```
                    actual = layers[i].neurons[j].value; //Actual value
                    delta = actual * (1 - actual) *
                        sigmaWeightDelta(i, j);//Function to compute error
                    for (k = 0; k < neuronInEachLayers[i - 1]; k++) {
                        layers[i - 1].neurons[k].weights[j] += delta * learningRate *
                            layers[i - 1].neurons[k].value; //Calculate the new weigh
                    }
                    layers[i].neurons[j].bias = layers[i].neurons[j].bias +
                        delta * learningRate * 1; //n_value is always 1 for bias
                }
            }
        }
    }
}
```

4.4 Screenshots of Major Functionalities

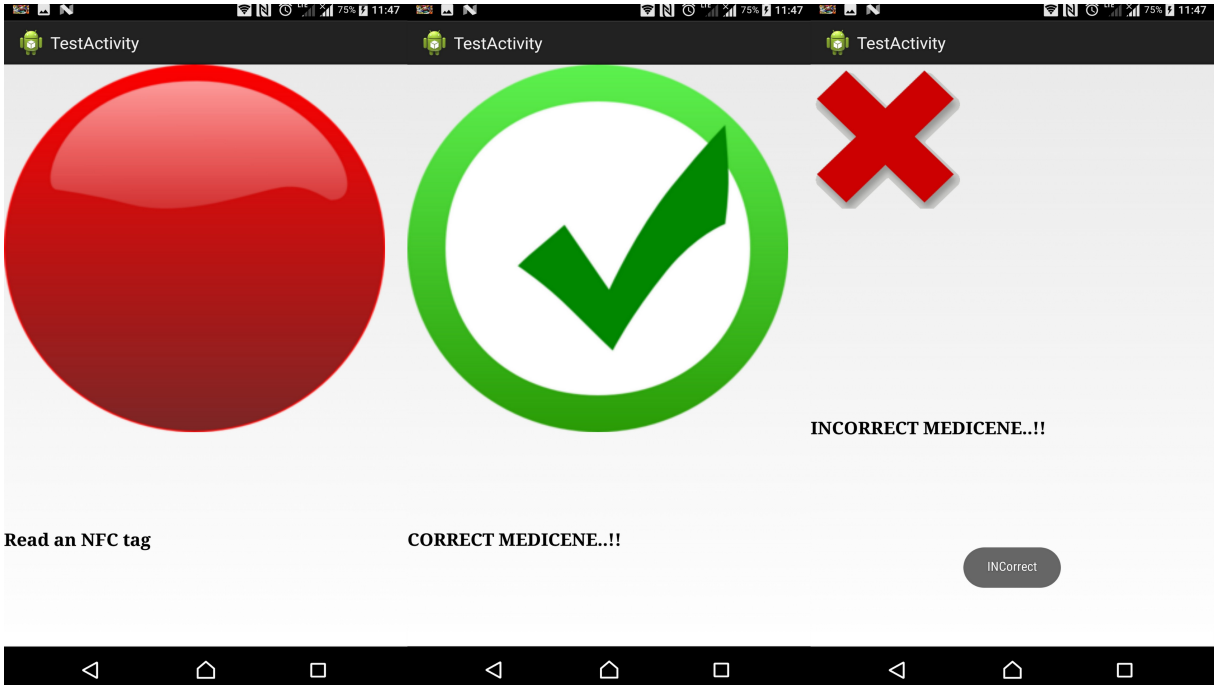

Figure 4.1: Patient App: NFC

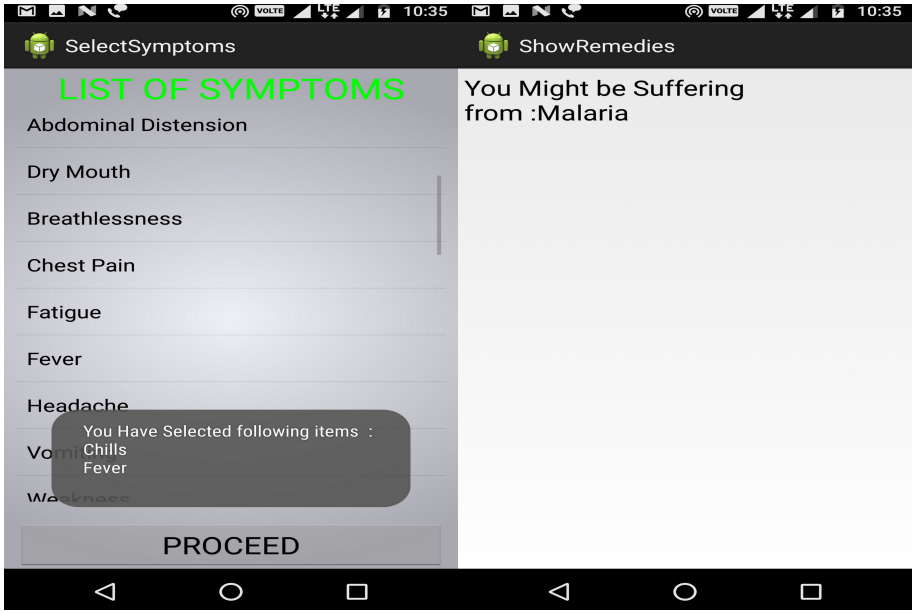

Figure 4.2: Patient App: Disease Prediction

CHAPTER 5

TESTING

5.1 Unit Testing

This type of testing is performed by the developers before the setup is handed over to the testing team to formally execute the test cases. Unit testing is performed by the respective developers on the individual units of source code assigned areas. The developers use test data that is separate from the test data of the quality assurance team. The goal of unit testing is to isolate each part of the program and show that individual parts are correct in terms of requirements and functionality.

Limitations of Unit Testing: Testing cannot catch each and every bug in an application. It is impossible to evaluate every execution path in every software application. The same is the case with unit testing. There is a limit to the number of scenarios and test data that the developer can use to verify the source code. So after he has exhausted all options there is no choice but to stop unit testing and merge the code segment with other units.

5.2 Integration Testing

The testing of combined parts of an application to determine if they function correctly together is Integration testing. There are two methods of doing Integration Testing Bottom-up Integration testing and Top Down Integration testing.

Bottom-up Integration Testing: This testing begins with unit testing, followed by tests of progressively higher-level combinations of units called modules or builds.

Top-down Integration Testing: This testing, the highest-level modules are tested first and progressively lower-level modules are tested after that.

In a comprehensive software development environment, bottom-up testing is usually done first, followed by top-down testing. The process concludes with multiple tests of the complete application, preferably in scenarios designed to mimic those it will encounter in customers' computers, systems and network.

5.3 Acceptance Testing

This is arguably the most importance type of testing as it is conducted by the Quality Assurance Team who will gauge whether the application meets the intended specifications and satisfies the client's requirements. The QA team will have a set of pre written scenarios and Test Cases that will be used to test the application.

More ideas will be shared about the application and more tests can be performed on it to gauge its accuracy and the reasons why the project was initiated. Acceptance tests are not only intended to point out simple spelling mistakes, cosmetic errors or Interface gaps, but also to point out any bugs in the application that will result in system crashers or major errors in the application.

By performing acceptance tests on an application the testing team will deduce how the application will perform in production. There are also legal and contractual requirements for acceptance of the system.

Test Module	Test Case Name	Case ID	Test Procedure	Input	Expected Result	Remark
Admin GUI	Register Doctor	Ta1	Accept doc details and add the entry to the database	Doctor's details	Entry should be added to the Database	Passed
	Register Patient	Ta2	Accept patient details and history to add in the database	Patient's details	Entry should be added in the database	Passed
	Update Doctor	Ta3	Accept new data and update the entry in database	New data	Entry in database should be updated	Passed
	Update Patient	Ta4	Accept new data and update the entry in database	New data	Entry in database should be updated	Passed
	View Attachments	Ta5	Download attachment from the database and open	Select file	Should be able to view attachment	Passed
	Import dataset & Train ANN	Ta6	Train ANN using training dataset to generate disease prediction	csv file	ANN should be trained to generate prediction	Passed
Doctor App	Login	Td1	If userId and Password correct then success	User name & password	Authentication should be correct	Passed
	Change Password	Td2	New Password will be sent to the mail id	New password	Password should work once created	Passed
	Patient list	Td3	Select a patient from the list to add log or view log	Select patient	Patient's details should be accessible and updatable	Passed
	Add Log	Td4	Add comments, precautions etc to the log table	Comments, precautions etc	Log should be added to the patient log table	Passed
	View Log	Td5	View the logs added to the log table	Query for displaying log	Logs should be displayed on the app	Passed
	Read NFC	Td6	Read the NFC tag using doctor's app	Data on NFC tag	Tag id should be displayed on app	Passed
	Give Prescription - write NFC	Td7	Add Medicine id to the NFC tag	Medicine name	NFC tag should be written successfully	Passed

Table 5.1: Unit Testing

Test Module	Test Case Name	Case ID	Test Procedure	Input	Expected Result	Remark
Patient App	Login	Tp1	If userId and Password correct then success	User name & password	Authentication should be correct	Passed
	View log	Tp2	View log given by the doctor	Click on view button	Log should be Displayed	Passed
	Upload attachment	T3	File will be selected from the explorer and uploaded to the database	jpg or png file	File uploaded successfully	Passed
	Show remedies - Disease prediction	Tp4	Symptoms will be selected from the list and sent to ANN for prediction	Symptoms given by patients	Disease prediction	Passed
	Change Password	Tp5	New Password will be sent to the mail id	New password	Password should work once created	Passed

Table 5.2: Unit Testing(...contd)

CHAPTER 6

RESULTS & DISCUSSION

6.1 Main GUI Snapshots

Figure 6.1: Admin App: Main App

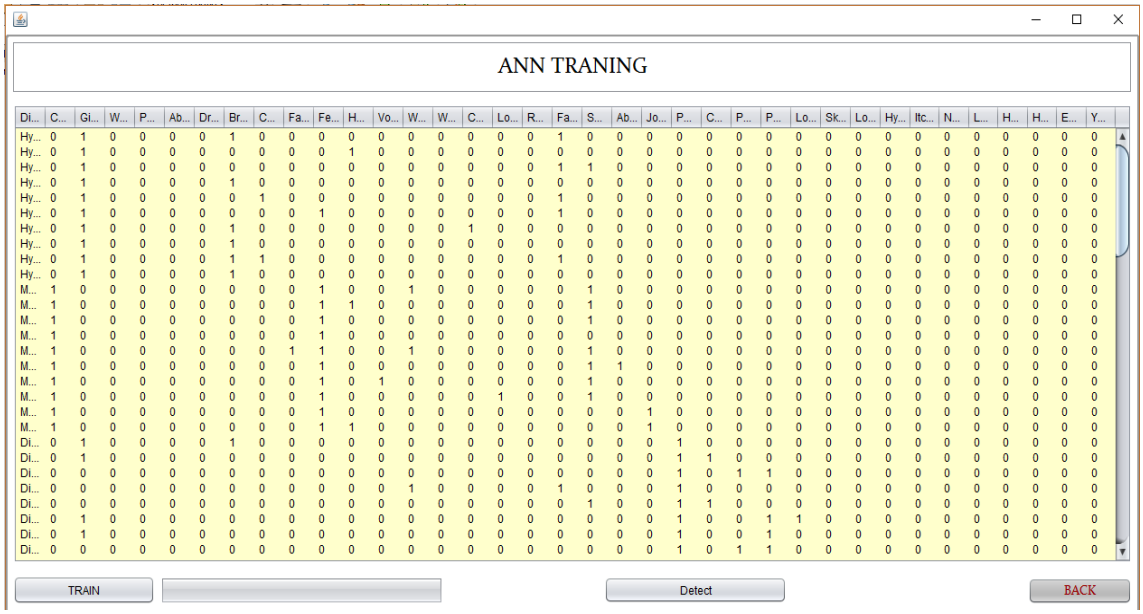

Figure 6.2: Admin App: Training Dataset

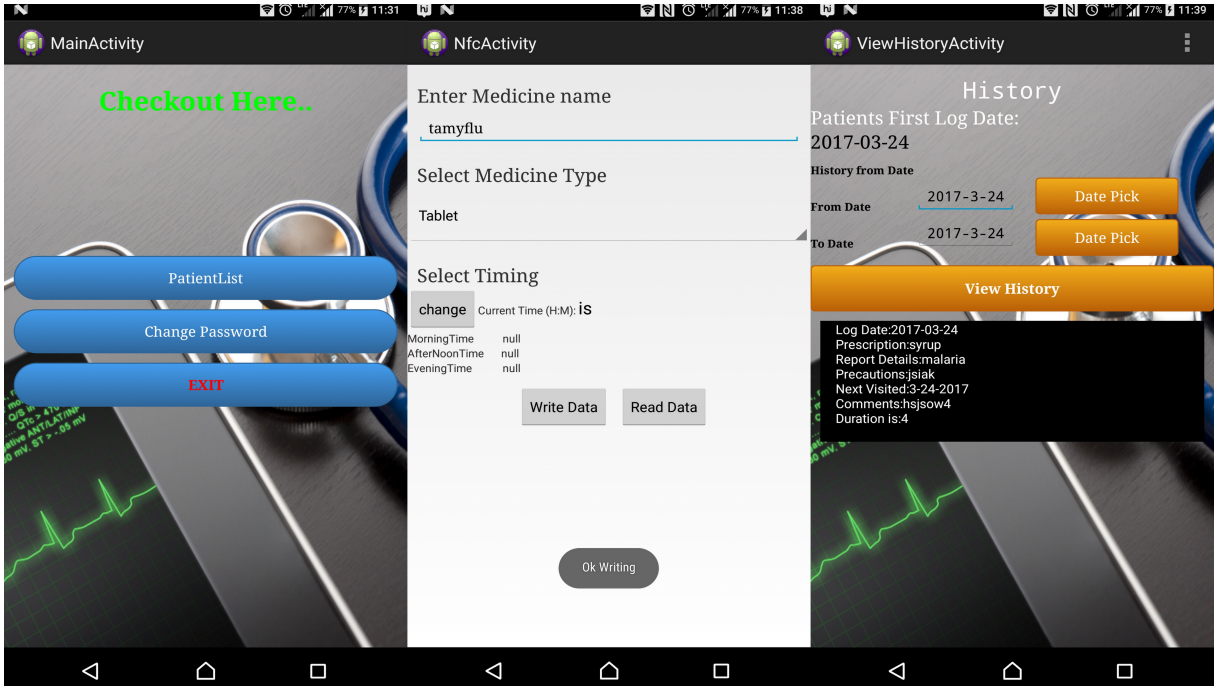

Figure 6.3: Doctor App

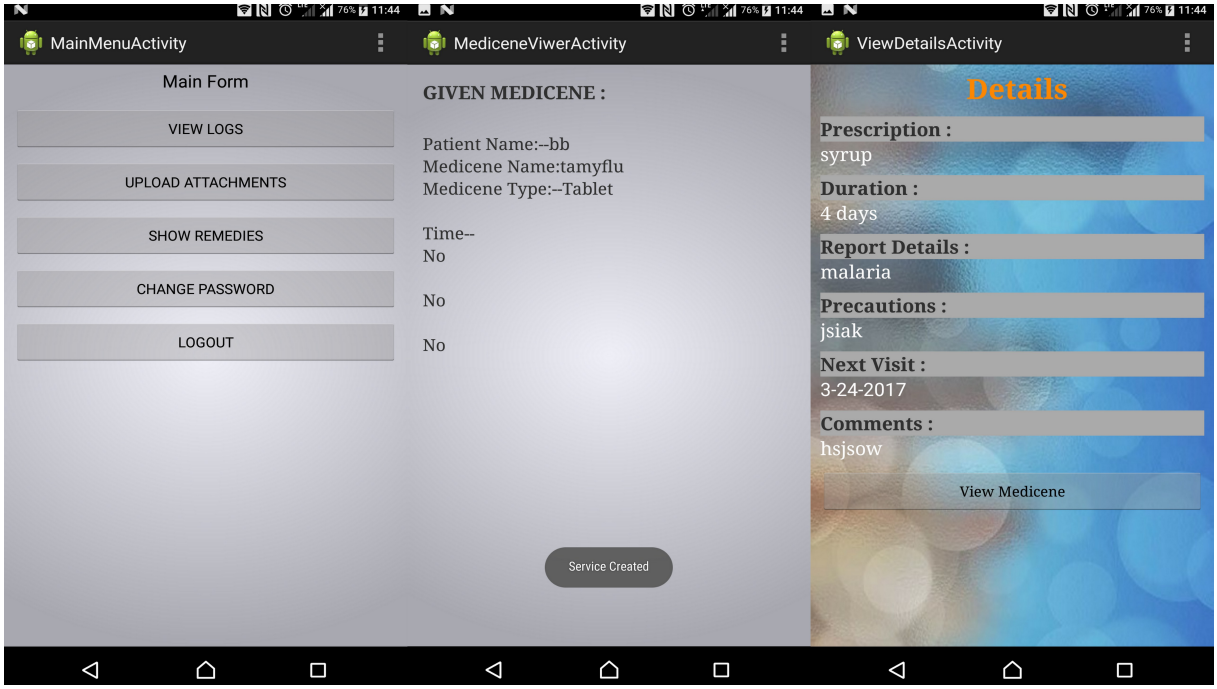

Figure 6.4: Patient App

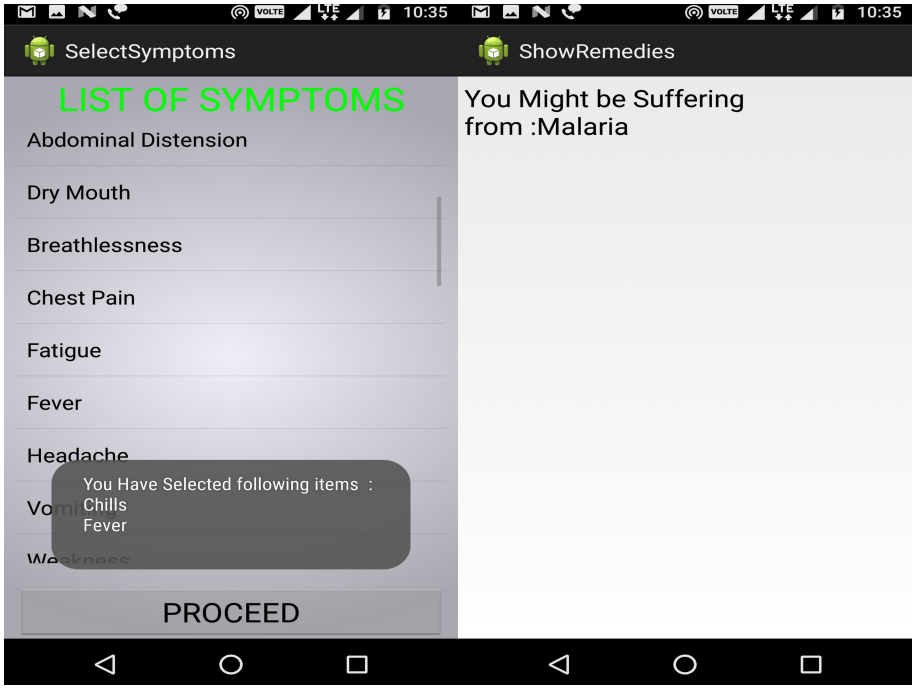

Figure 6.5: Patient App: Disease Prediction (a)

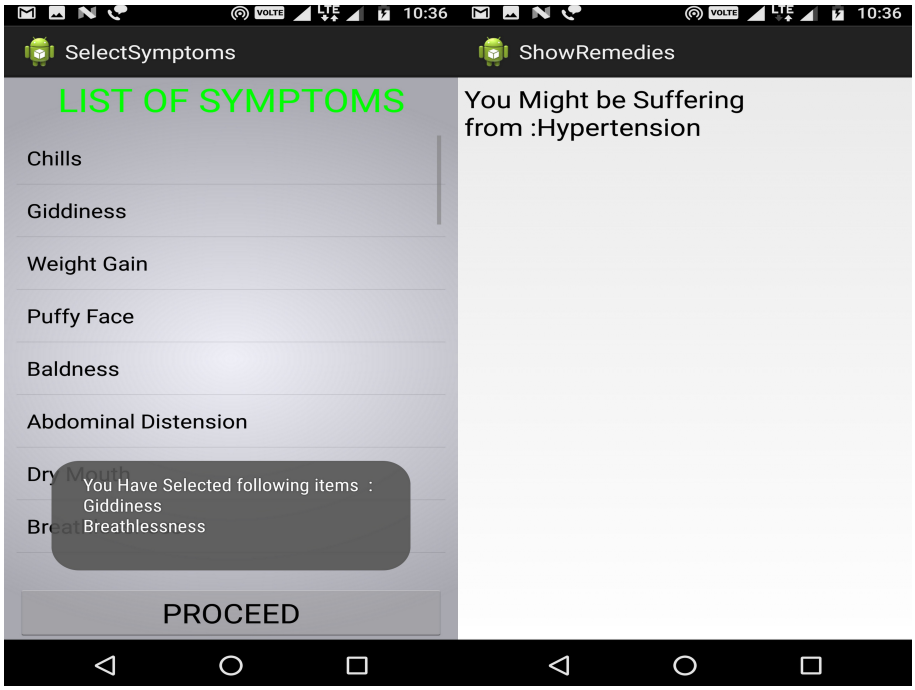

Figure 6.6: Patient App: Disease Prediction (b)

6.2 Discussion

The core objective of this system is to work as a assistance application for healthcare and medicine remainder, eliminating the possibility of taking wrong medicine as some people are not able to read medicine. In the current era, one of the greatest concerns in healthcare is global aging and prevalence of chronic diseases. Lots of resources get wasted because of lack of digitization. Therefore we have designed this project keeping in mind the need of moving routine medical checks and other healthcare services from hospital centric to home- centric environment . By implementing this project, we reduce Society's financial burden and burden on hospital resources also the paper usages which was being used in reports and multiple prescriptions. The five main modules namely, Patient App, Doctor App, Admin, Server, Database are the core functionality of the software. To get a bird's eye on what exactly these modules do, let's understand them in short.

The admin module is responsible to add/manage patients' and doctors' entries. Apply Data mining using ANN for generating disease prediction using patient's history and symptoms. Admin will be able to view report uploaded by the patient. That report can be forwarded to the doctor. This module is needed to make synchronisation between the doctors and patients. Next is server module which manages the database through request sent or received. Authentication process is done through server. The server used in this system will be home server.

Database module stores the patients' information such as name, address, medical history, etc. Store the doctors' information such as name, contact, specialty and other details. The another module is the Doctor's application. This module is implemented to make doctor able to view patient or patient information. Doctor's application will have facility to give prescription and timing to any patient irrespective of the location. Doctor will be able to see reports sent by admin. Doctor will give NFCs for specific medicine. Doctor will be able to change/update the medicine or the prescription timings.

In patient's application module, patient will be able to view prescription given by doctor. Alerts will be generated according to prescription timings for specific medicine. Application will identify NFC tags on medicine and patient will recognize the correct medicine by alert

message generated on patient's application. Patient can upload reports to server using application.

All these modules and their appreciable features make the system perfect tool for taking healthcare services to the next level.

CHAPTER 7

CONCLUSION

This application is for the people who are too busy to remember the medicine timings and also for the old people who can't remember the timings. This application mainly works as assistance application for healthcare and medicine remainder, eliminating the possibility of taking wrong medicine as old people are not able to read medicine. This project comes under EHR which will reduce paper usage in great extent which was being used in reports and multiple prescription. Main objective of this project is to increase digitization. By implementing this project, we reduce burden on hospital resources i.e lot of rush and unavailability of beds will not be a problem. Society's financial burden will also be reduced. Time and money will be saved which were previously being consumed due to transportation (multiple visits) and also waiting in queue for the doctor. This project reduces the possibility of losing hard-copy of the records by saving the history on database. The ANN algorithm in this project will predict the probable disease the patient is having according to the symptoms given by the patient.

CHAPTER 8

REFERENCES

[1] Geng Yang, Li Xie, Matti Mäntysalo, Xiaolin Zhou, Zhibo Pang, Li Da Xu, Sharon Kao-Walter, Qiang Chen, Lirong Zheng, "A Health-IoT Platform Based on the Integration of Intelligent Packaging, Unobtrusive Bio-Sensor and Intelligent Medicine Box ", 2013, IEEE

[2] Tania Cerquitelli, Elena Baralis, Lia Morra and Silivia Chiusano, "Data Mining for Better Healthcare: A Path Towards Automated Data Analysis? ", 2016, IEEE

[3] Amiya Kumar Tripathy, Rebeck Carvalho, Keshav Pawaskar, Suraj Yadav, Vijay Yadav, "Mobile Based Healthcare Management using Artificial Intelligence ", 2015, IEEE

[4] Prethi.M, Ranjith Balakrishan, "Cloud Enabled Patient-Centric EHR Management System ", 2014, IEEE

[5] Gillian Pearce, Lela Mirtskhulava, Koba Bakuria, Julian Wong, Salah Al-Majeed, Nana Gulua, "Artificila Neural Network and Mobile Applications in Medial Diagnosis ", 2015, IEEE

[6] Qiang YE, Tao LU, Yijun LI, Wenjun SUN, "Neural Network with Forgetting: An ANN Algorithm for Customers ", 2005, IEEE

[7] https://en.wikipedia.org.