

DEVELOPMENT OF AN ANDROID BASED DOCTOR'S TELE-INFORMATION SYSTEM

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENT

FOR THE DEGREE OF BACHELOR OF TECHNOLOGY

IN BIOMEDICAL ENGINEERING

(2014)



Submitted by-

HIMANSU ROUT

110BM0637

Under the guidance of

Prof. Sirsendu Sekhar Ray

(Assistant Professor)

Department of Biotechnology and Medical Engineering
National Institute of Technology
Rourkela



National Institute of Technology, Rourkela

CERTIFICATE

This is to certify that the thesis entitled “**DEVELOPMENT OF AN ANDROID BASED DOCTOR’S TELE-INFORMATION SYSTEM**” submitted by HIMANSU ROUT in partial fulfilment of the requirements for the degree of Bachelor of Technology in BIOMEDICAL embodies the bonafide work done by him in the final semester of his degree under the supervision of the undersigned. The thesis or any part of it has not been submitted earlier to any other University / Institute for the award of any Degree or Diploma.

Dr. Sirsendu Sekhar Ray

Assistant Professor

Department of Biotechnology and medical engineering

National Institute of Technology, Rourkela

ACKNOWLEDGEMENT

Every effort is motivated by an ambition and all ambitions have an inspiration behind in the height of reaching a mile-stone in life. I owe my deep sense of gratitude that helps me in such a constructive work with grateful heart, I avail this privilege to express my deep sense of gratitude and indebtedness to my esteemed guide **Dr. Sirsendu Sekhar Ray** (Assistant Professor, Department of Biotechnology and Medical Engineering) for his guidance, supervision and critical suggestion throughout the course of this project.

I am sincerely thankful to **Prof. Krishna Pramanik**, Dept. of Biotechnology and Medical Engineering, NIT, Rourkela, for providing the necessary facilities for this work. I gratefully extend my sincere thanks to all faculties and to all teaching and non-teaching staff members of Department of Biotechnology & Medical Engineering, National Institute of Technology Rourkela, Orissa.

Sincere thanks are extended to the Department of Biotechnology and Medical Engineering, NIT Rourkela and my friends for their valuable advice and moral support. My thanks are due to my revered parents and loving sister for their affection, inspiration, patience and support.

Himansu Rout

ABSTRACT

The prime objective of “**DOCTOR’S TELE-INFORMATION SYSTEM**” is to create a fully-fledged Android application which could be used to find a list of doctors based on the department selected by the user. The user not only finds all the doctors in the city, but also he can make a choice of calling the doctor on his available mobile number. The user can also send a mail to the doctor through his/her available email –id on Gmail rendered to the user on the phone; the user has the facility to send a message through sms to the doctor’s cell phone and can also know the detailed address of the doctor he chooses.

PLATFORM:-

Android Application

Development using ADOBE FLASH BUILDER

Submitted By:-

Himansu Rout (110BM0637)

CONTENTS

Acknowledgement.....	i
Abstract.....	ii
Contents.....	iii
List of Figures	v
List of Table.....	vi
1 Introduction.....	1
1.1 What is Android	4
1.2 Features of Android	5
1.2.1 Application Framework	5
1.2.2 Dalvik Virtual Machine (DVM)	5
1.2.3 SQLite	5
1.2.4 Linux Kernal	6
1.3 Versions of Android.....	6
1.4 Motivation	7
1.5 Objective	7
1.6 Outline of thesis	8
1.7 Literature Review.....	9
2 General Description.....	12
2.1 Introduction	12
2.2 Product Perspective	12
2.3 User Characteristics	13
2.3.1 User	13
2.4 General Constraints	14
2.4.1 Server Side	14
2.4.2 Client Side	14
3 Requirement Analysis and Specification.....	15
3.1 Functional Requirements of the System	15

4	Design.....	17
4.1	Function Oriented Software Design	17
4.1.1	Context diagram	21
4.1.2	Level 1 diagram	21
4.2	Object Oriented Software Design	23
4.2.1	ER Diagram	23
5	Coding Phase.....	24
6	Implementation.....	33
6.1	Android API	33
7	Graphical User Interface.....	35
8	Conclusions and Future Work.....	37
8.1	Conclusion	37
8.2	Future Work	37
9.	References.....	38

List of Figures

1.1	Growth chart for Android OS	4
1.2	Pie Chart Of the Android API level distribution	6
4.1	Flow Chart.....	19
4.2	Application Implementation Diagram	20
4.3	Level Diagram.	22
6.1	Application Programming Interface Diagram.....	34
7.1	Splash Screen.....	35
7.2	User Screen.....	35
7.3	Doctor’s List.....	36
7.4	Doctor’s contact details.....	36

List of Tables

1.1	Level of Android Versions & distribution	6
-----	--	---

Chapter 1

INTRODUCTION:

Since human existence, contacting someone has been one of the most imperative approaches to communicate and share our information in various mode of life. Communication serves as a foundation for planning and establishing, promoting motivation, altering individual's attitudes and in socialization. It can be verbal or written such as emails, formal letters or memos. Communication is important because it allows people to share thoughts, happiness, and improve interactions. In the absence of communication, there won't be any common interests and advancements in society which is important because it helps people to stay organized and understand each other. Without communication teams could not put projects together, business could not operate, and people could not understand what is going on. It is the basic method through which humans interact.

Among the various modes of communication available nowadays which includes calling, texting, email, voicemail, writing letter etc. there are also other modes of communication, which include language, lettering, communication in person or communication at a distance by means of radio, telephone, television, telegraph, etc. The two types are: Formal & Informal. In case of formal the person has to formally plan a consultation or send an email and transfer some statement to them. In case of informal the person has to meet up someone casually and transfer some memo to them

Mobile communication in different forms has been in use in varied fields to contact any person at the time of want. This history focuses on communication devices that connect wirelessly to the general public switched phone network. The transmission features a long and varied history going back to Reginald Fessenden's invention and shore-to-ship demonstration of radio telephone. The primary mobile telephones were barely transportable compared to today's compact hand-held devices. In conjunction with the method of developing more transportable technology, forceful changes have taken place within the networking of wireless communication and therefore the prevalence of its use. As a result, varied activities square measure dead during a reliable & quicker means.

Mobile communications as we all know them didn't get being till the geniuses at Bell Labs worked out a way to produce overlapping "cells" that might enable a mobile user to stay connected whereas moving from purpose to purpose. Martin Cooper of Motorola is attributable with making the primary true "cell phone" in 1983: the Motorola DynaTac 8000X. Six years later the Motorola MicroTac 9800X came on the market; this phone was sufficiently little (barely) to suit during a. Nokia came on the scene in 1992 with the 1011. This was the primary mass created GSM mobile phone that might receive SMS (texts). The phone had a memory capable of storing ninety nine numbers jacket pocket. Motorola International 3200 in 1992 was the primary "digital" mobile, reckoning on GSM technology rather than the quality analog network.

In 1996, Nokia additionally introduced the Nokia mortal 9000. This was the world's 1st "smart-phone". Over the past decade, mobile phones have steady become smaller and smarter, with prognosticative text perform, auto-correct, GPS, and more. Siemens and Samsung shortly joined the mobile trade, and in 2007 the primary iPhone hit the market - marketing a million devices in only seventy four days. Antennas went internal, and profiles became slimmer and cleaner.

Today mobile phones vary from flip phones sufficiently little to suit during a front jean pocket to sleek, slim-profiled bit screens that square measure as good because the average desktop. With the appearance of smart-phones came "apps" - mini-applications which will do everything from finding a close-by eating house to scanning a coupon to count calories. Camera phones are commercially created since 2000, and most phones these days will take video furthermore. Today, most smart-phones either run Apple's programming or the mechanical man counterpart. Cell networks square measure still standard; however several phones will get rid of Wi-Fi and 3G or 4-G networks. Apps will convert calls into VoIP to reduce mobile vocation rates, and additional folks have mobile phones than own landlines. The mobile is here, and it's here to remain. In a reliable & faster way.

IMPORTANCE OF DOCTOR'S TELE-INFORMATION SYSTEM:

Mobile based application is a topic of modern technology. In the old system, the individual takes the telephone directory and search for the doctors in different departments of a hospital. The information regarding contacting the doctor is then executed through landline. In the current system through mobile devices, the individual can directly contact the doctor over voice call, email or sms which is stored in the mobile database. At the time of email the user will be lead to Gmail and for sms to compose messaging directly.

This project benevolence an application which contain the directory of doctors who can be contacted at the time of emergency through mobile devices, which is developed using Android application. The application can be installed in every android user's mobile phones and can be used to have a mobile connectivity with the doctors in every health issues in offline mode. This mobile application procures the contact details of the doctors assigned to the respective departments and the doctors enrolled in that corresponding department present in that mobile database. This application is helpful to all those people staying in rural areas or remote areas. It is also helpful for old people, students and practically to every individual at the time of health related issues.

1.1 What is Android?

Android is a Linux-based operating system developed for smart phones or tablet computers. It is a stack of software that includes operating system, middleware and libraries and APIs written in C [10]. It was developed by Google and Open Handset Alliance in July, 2005. Android is an open source and Google releases the source code under Apache license. This open source and free license allow the manufacturers and the enthusiastic developers to freely develop and modify their applications in Java-like language that utilizes Google-developed Java libraries [2].

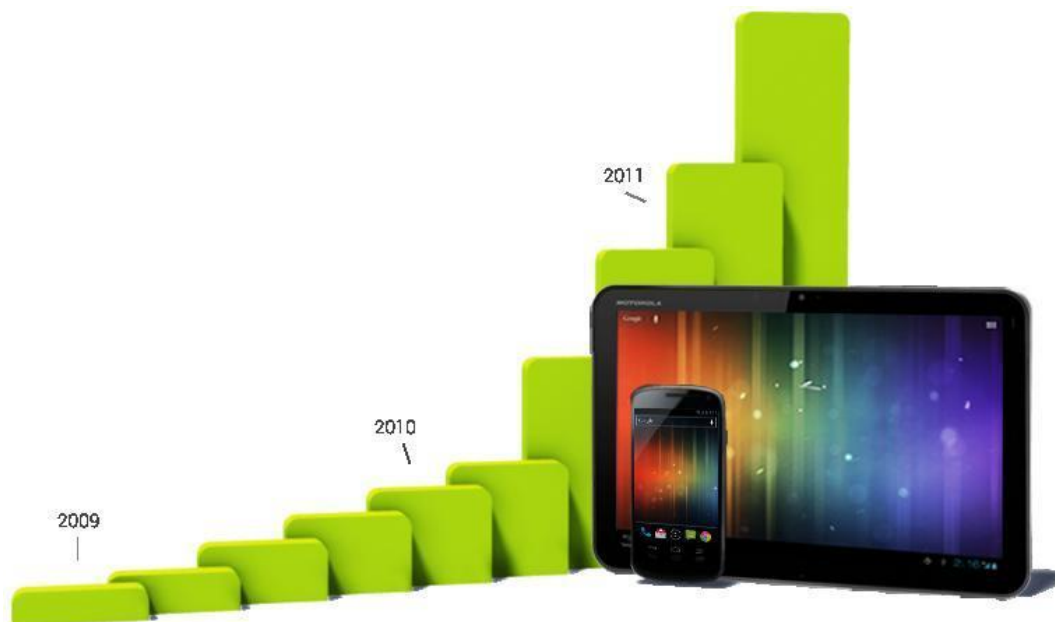


Figure 1.1: Growth Chart for the android OS [3]

The Android applications are developed using the Android software development kit (SDK). The SDK includes a comprehensive set of development tools which includes a debugger, software libraries, a handset based emulator which is based on QEMU (Quick Emulator) and tutorials [7]. The integrated development environment (IDE) which is officially supported for Android apps development is Eclipse which uses the Android Development Tools (ADT) plugin.

Advantages: - Because the user is using an open source platform with lots of industry support and with powerful market momentum, building our own software and installing the software on your own phone is quite reliable.

Disadvantages: When user is using an open source platform then lots of Security disquiets arises.

The following are the several features which help in the development of Android applications:

1.2 Features of Android :

1.2.1 Application Framework

Android application framework is supported by number of open source libraries like Open SSL, SQLite, and Libc. The application framework is also supported by the Android core libraries. The framework is based on UNIX file system permissions which ensures security as the applications can have only those abilities that mobile phone owner give at the time of installation. The application framework enables the reuse and replacement of components.

1.2.2 Dalvik Virtual Machine (DVM)

Dalvik is a process virtual machine used in Google's Android operating system. It is a low memory based virtual machine which is especially designed for Android to run on the embedded systems and work efficiently in low power. The programs are commonly written in Java and are compiled into byte code [2]. This byte code is then converted form JVM .class files to .dex file (Dalvik executable) before installation on a device.

1.2.3 SQLite

Android OS contains the SQLite database management classes which are used by an application to maintain its own private database. SQLite is a relational database management system contained in C programming library [10]. It is mostly preferred as embedded database for local or client storage in application software. It has many bindings to the programming languages.

1.2.4 Linux Kernel

Android uses Linux version 2.6 for the core system services like memory management, process management, security and network stack. The Linux kernel also acts as an abstraction layer between the hardware and the software stack [4].

1.3 Versions of Android

This section provides data about the relative number of devices running a given version of the Android platform.

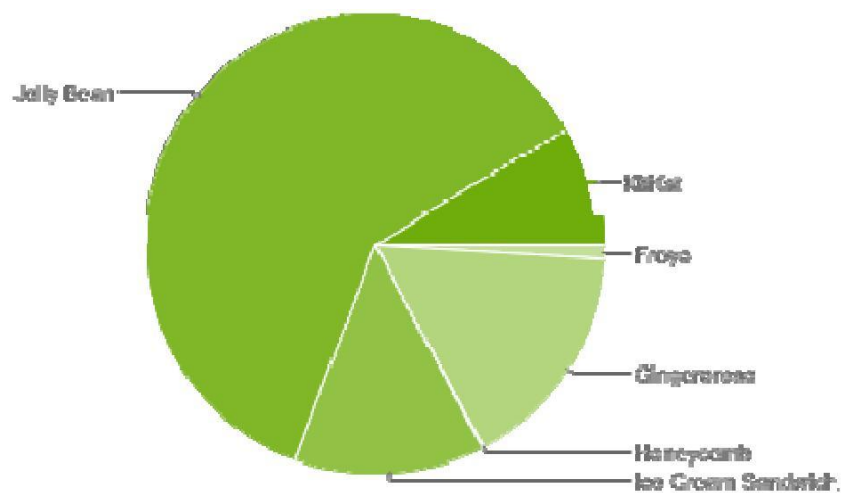


Figure 1.2: (Pie chart of the Android API level distribution) [3]

Version	Codename	API	Distribution
2.2	Froyo	8	3.7%
2.3.3-2.3.7	Gingerbread	9	0.1%
3.2	Honeycomb	13	0.1%
4.0.3-4.0.4	Ice-cream Sandwich	15	27.5%
4.1x 4.2x 4.3 4.4	Jellybean	16	26.1%
4.4	Kitkat	19	8.5%

Table 1.1: Level of Android Versions & their distribution

1.4 Motivation

The connectivity to doctors through mobile devices is one of the most important systems needed by every individual to contact the doctors at the time of need. The previous conventions followed for contacting doctors was very tedious task and require a lot of time. It was not automated and so handling and maintaining was a tough job [8]. The previous connectivity system uses large databases in the form of hardcopies; directory etc. needed to get the information about a doctor and thereafter contact him. But it was quite time taking & was not reliable. So there was a need to automate the connectivity system and to reduce the manual effort needed in storing the records and maintaining it. The doctor's directory system through mobile devices is fully reliable, fast and is easy to use [4, 5].

Nowadays, Android phone has emerged as the world's most popular mobile platform. Android is the world's most popular mobile platform. It's the largest installed base of any mobile platform and growing fast. Millions of users are using android phones and android application is becoming more and more popular [8].

1.5 Objective:

Design a Doctors Directory system to keep contact details of doctors in mobile devices for easy and faster connectivity at the time of health issues.

1. Develop and implement a Doctors Directory system.
2. To create an Android mobile application to provide a User Interface to interact with the system.

1.6 Outline of thesis:

The thesis consists of seven chapters following this chapter:

Chapter 2: General Description

We discuss about the General constraints, Product Prospective and User characteristics etc.

Chapter 3: Requirement analysis and specification

We discuss about the functional requirements of the system, developed the SRS model of the system.

Chapter 4: Design

This chapter discusses different Design models. The Function Oriented design approach focuses on the functions of the system; the functionalities are represented with the help of DFD's. The object oriented design approach focuses on the ER model of the system.

Chapter 5: Coding Phase

The coding phase deals with transforming the user requirements as described in SRS document, into a form i.e. implementable using programming language. The programming language that I used to develop is JAVA and XML.

Chapter 6: Implementation

This chapter outlines the three tier architecture of the application developed. The Client side is the android application interface that is accessed by the user. MySQL server acts as the back end.

Chapter 7: Graphical User Interface. This chapter describes the different user interfaces of the applications and the forms shown to the user.

Chapter 8: Conclusion & Future Work

Chapter 9: References

1.7 Literature review:

The mobile services and applications worth chain has well-versed several changes throughout the past few years (1990-2012) because of the evolution of mobile devices and their capabilities. Chohick in 2011, a platform is “a product which will be extended by a user for the advantage of different users”, and this statement will be verified by that the iPhone or golem as a platform compete an elementary role within the shift of developers attention to the present new chance, yet as shopper education and awareness of the advantages of applications to make sure a viable system. As additional and additional community are transferred to mobile technologies, well, in truth, the quantity of individuals owning a Smartphone or a pill and different new devices has skyrocketed over the previous couple of years, each economic facet is moon-faced with a replacement perspective in approaching customers. Within the context of package progress, many voluminous mobile house owners and social media, insurance firms and their various ecosystems cannot afford to be unbroken out of the digital loop. From ancient and long approaches towards a replacement era of, bit and solve”, let’s define the most advantages that insurance industries got to gain from adopting mobile applications and the way your insurance app ought to look like: superior consumer base by branching resolute multiple mobile platforms (Android, iPhone, tablets, etc.) and canopy a bigger objective demographic advocate product data.

With the advancement of contemporary technology and especially mobile technology it's no marvel that cell phones and their quality are on the increase. It’s maybe thanks to their easy use and their ability to facilitate different applications. It's thanks to this that a lot of consultants in psychological science have begun to review the results of cell phones on our society.

A study by Adriana Bianchi and Dr. James G. Phillips, Ph.D., (2005), victimization predictors from addiction literature, the study sought-after to predict problematic transportable usage. It keyed in on factors they believed to cause transportable disorders. 195 subjects of each genders and a median age of thirty six (ranging from 18-85 years old) were interviewed. The results cause the creation of the transportable downside Use Scale. This scale was compared to the Addiction Potential Scale. Fundamentals that were accustomed compare the scales and build the transportable downside Use Scale were the number of your time one uses their mobile phone and therefore the form of usage. The behavioral predictors that influenced the number of mobile phone usage were: being extroversive, possessing low vanity, and being young.

Improvements in mobile technology have permitted a widespread range of applications to be used by individuals on the travel. Some of the issues that arises includes small screen size, inadequate connectivity, high power depletion rates and restricted input modalities for devices.. The success or failure of the application depends on the user as these devices are designed to use anywhere. This characteristic of usability is repeatedly disregarded in prevailing usability facsimiles, as these replicas are intended for applications which are infrequently used in a mobile framework.

This definition identifies 3 factors that should be considered when evaluating usability.

User: Person who interacts with the product;

Goal: Intended outcome;

Context of use: Users, tasks, equipment (hardware, software and materials), and the physical and social environments in which a product is used.

Each of the above factors may have an impact on the overall design of the product and in particular will affect how the user will interact with the system using traditional models of usability.

Mobile Context: When using mobile applications the user is not snarled to a single position. He/she may also be interacting with neighboring people, stuffs and environmentally friendly rudiments which may divert his/her attention.

Connectivity: It is frequently sluggish and undependable on mobile devices. As a result the enactment of mobile applications utilizing these facilities is affected.

Small Screen Size: To deliver lightness mobile devices comprise of very limited screen size and so the amount of information that can be displayed is limited.

Dissimilar Display Resolution: The resolution of mobile devices is abridged from that of desktop computers resulting in inferior quality images.

Restricted Processing Ability and Control: Mobile devices often contain less processing capability and control, in order to offer transportability. As a result the type of applications that is limited for mobile devices.

Data Entry Methods: The input methods are different from computers and entail a certain level of expertise. Thus the possibility of inaccurate input is increased and the rate of information entry is decreased.

From the review it is apparent that many existing models for usability do not consider mobility and its consequences, such as additional cognitive load. This complicates the job of the usability practitioner, who must consequently define their task model to explicitly include mobility. One might argue that the lack of reference to a particular context could be strength of a usability model provided that the usability practitioner has the initiative and knows how to modify the model for a particular context.

Chapter 2

General Description

2.1 Introduction

The doctor's directory application gives an ease in the working, managing and day-to-day requirements of health related issues in an easy way through an intuitive interface that makes it easy for every individual to utilize it fully and is virtually maintenance free.

The subsection 2.2 explains the product perspective, 2.3 explains the functions of the product. Section 2.4 explains the type of user and characteristics. Section 2.5 explains the constraints and requirements.

2.2 Product Perspective

My doctor's directory application simplifies the connectivity to doctors through mobile devices. The system is deployed on every individual's mobile and the individual can utilize the app for their respective health needs by contacting the respective doctor through cell phone, sms or email which are stored in the mobile database. The software focus on the easy way of storing and contacting the doctors in the server with the use of mobile phones.

2.3 User Characteristics

The system has single user that is the owner of the mobile phone. Here is a summary of the permissions enjoyed by the user

2.3.1 User

- The user logs in through his cellphone and gets the details of the various medical departments and the doctors corresponding to that department.
- The user can click on the required doctor's name to access the details of that doctor present in the mobile internal database.
- The user can then call or sms or email the doctor as per his/her comfort to contact the doctor in a fast & reliable way.
- The user can also view the doctor's photo (if available) in the server through his mobile phone.

2.4 General Constraints

The general constraints of this system are of two types

1. Hardware Constraints
2. Software constraints

Hardware constraints include any devices which can run Android Operating System. And the following are the software constraints of the system.

2.4.1 Server Side

Database server: PHPMYAdmin or higher

PHP: PHP 4.4.0 or higher (5.2 recommended)

2.4.2 Client Side

Any network enabled device which is able to connect to the server and running the Android Operating System.

Chapter 3

Requirement Analysis and Specification

The Application developed follows the following steps of SDLC. Under which the SRS model developed for the system is as listed below

The SRS model Contains

- Functional Requirements.
- Non-Functional requirements

3.1 Functional Requirements of the System

The functional requirements parts discuss the functional behavior that should be possessed by the system. Each requirement maps to a high level function (fi) that transforms the given set of input data (ii) into output data [1, 7].

Different functional requirements possessed by the system are:-

R.1. Login

Description: The user will login into the application from his/her mobile phone. If the user don't have adobe flex installed in his/her mobile phone, user will be prompted to install the same from google play else error message will be displayed.

R.2 Welcome Screen

Description: After login, the user can view the welcome screen showing hospital doctor plus logo which is quite professional to the application.

R.2.1 Selection of Hospital Departments

Description: The user can view the list of departments available in the hospital. Input: Select a department and click on get doctor's list.

Output: List of doctors under that department.

R.2.2 Select among the list of available doctors

Description: User chooses the desired doctor.

Input: Click on the specific doctor's name

Output: Contact details of that doctor is shown

R.3 Connectivity

Description: After getting the details of the doctor he/she can contact the doctor through any of the details available.

Input: Select cell phone or email or sms

Output: Activity (dial, mail, sms) executed as per selection

Chapter 4:

Design

The design phase emphasizes on the transformation of customer requirements as defined in the SRS document, into a form that is suitable for coding. The design phase can be broadly classified in two levels.

- 1) Preliminary or high level design
- 2) Detailed design

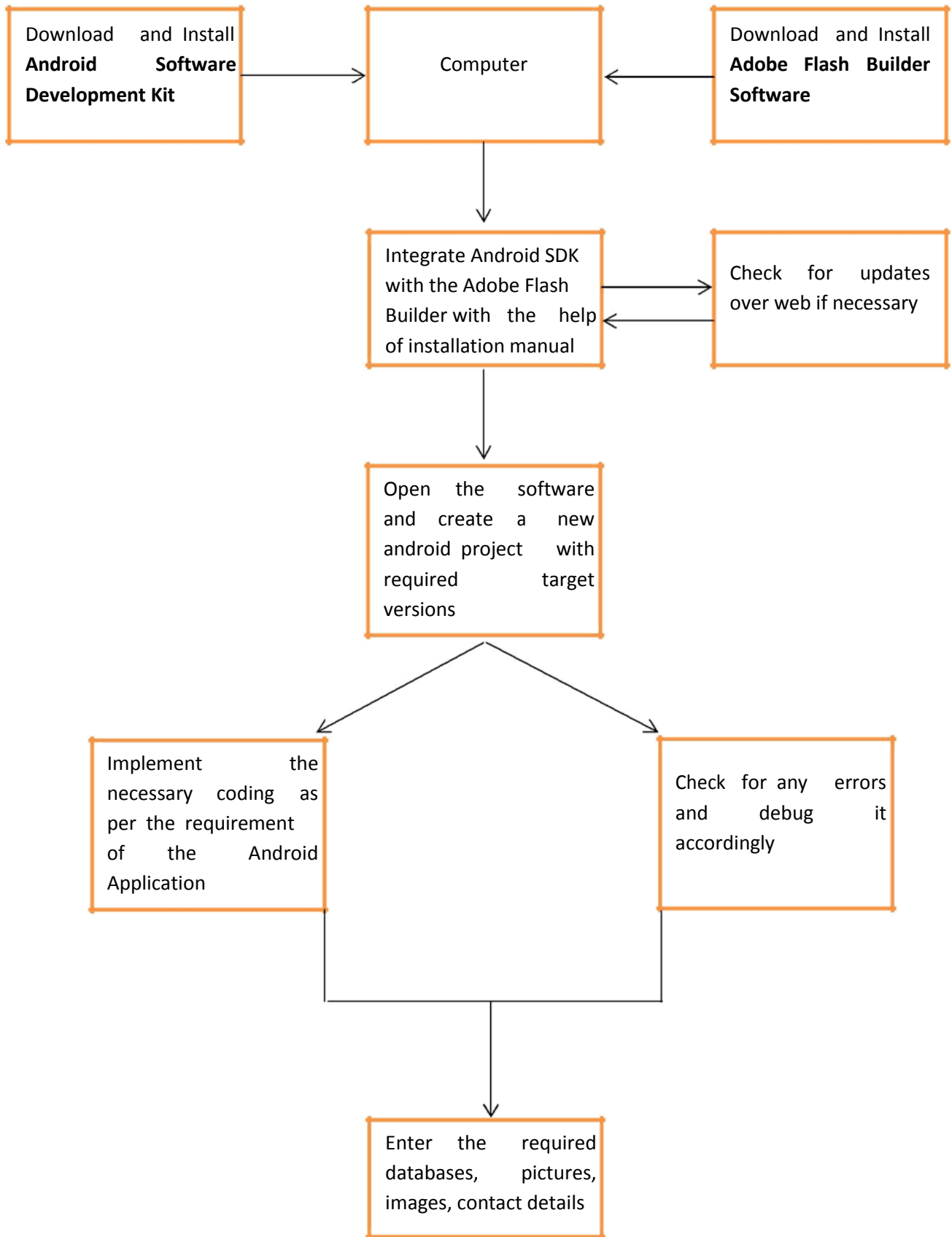
The preliminary design can be further divided into two sub categories

- Function Oriented Software Design
- Object Oriented Software Design

4.1 Function Oriented Software Design

This design model can be represented by drawing the DFDs (Data Flow Diagrams) for the given SRS document.

A data flow diagram is a graphical representation of the data flow through an information system which is used to model the process aspects of the system. DFD is the preliminary step used to create an overview of the system. DFD is used for structured design.



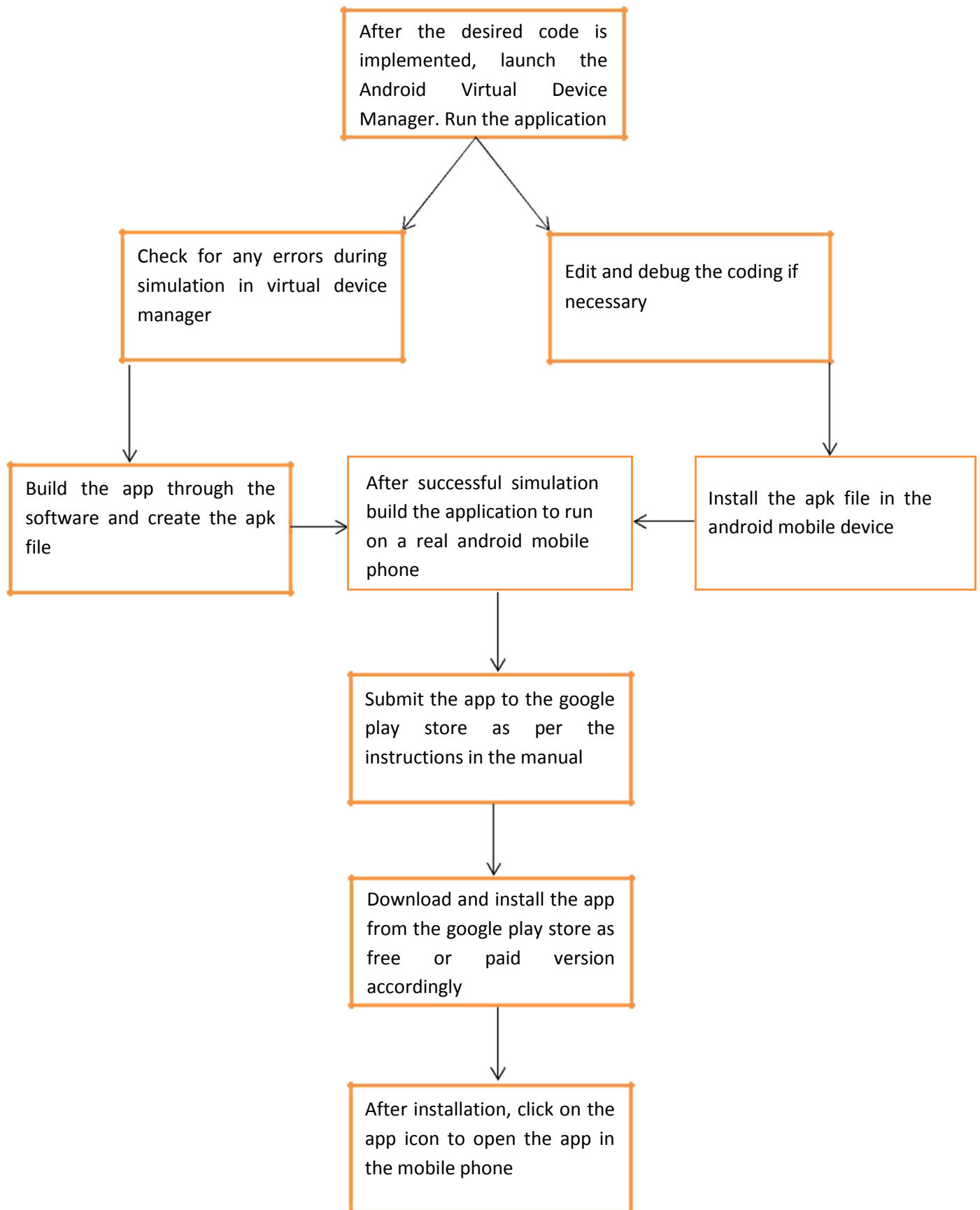


Figure 4.1 : Flowchart

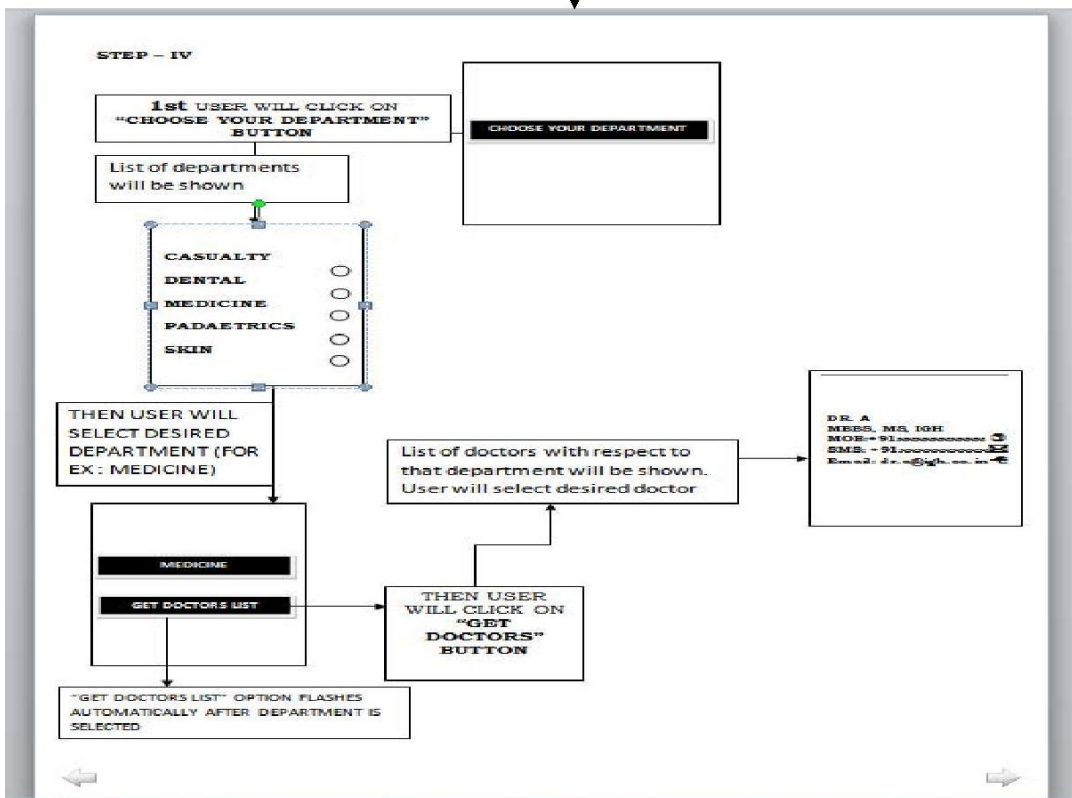
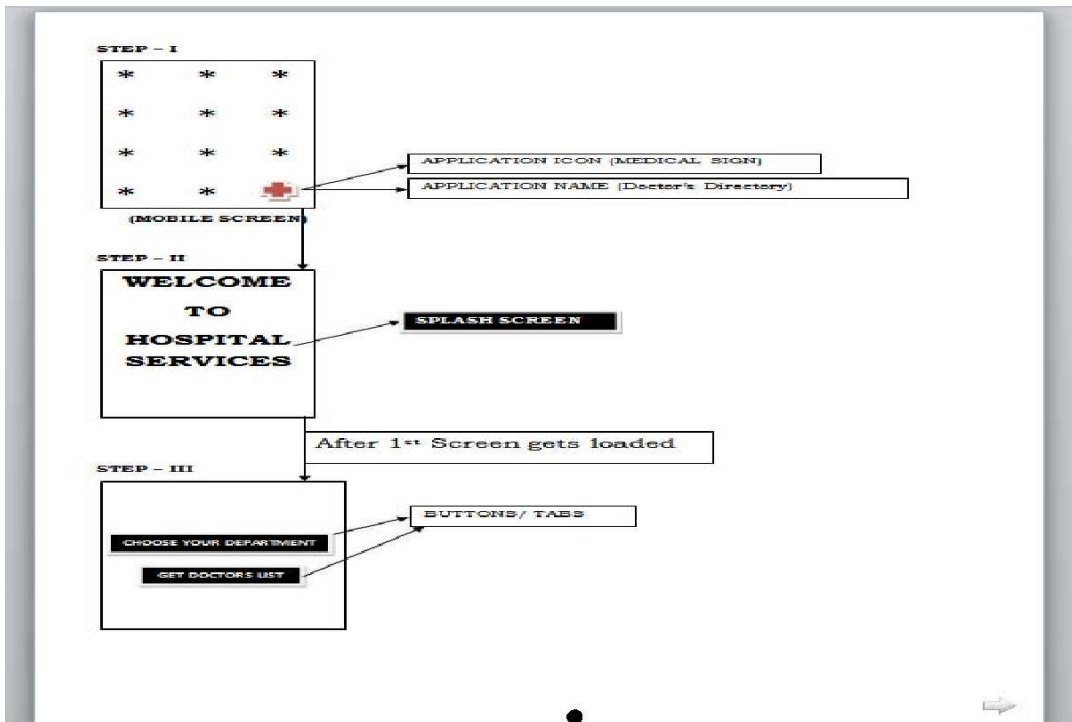


Figure 4.2: Application Implementation Diagram

4.1.1 Context diagram

A context diagram is a structured graphical tool used for identifying the functional areas and the processes which are performed within and between the system and outside the system. Context diagram supports a data-oriented approach for designing system. It helps in investigating the output and the process requirement of the system. It helps in defining the boundaries of the proposed system. The symbols used in the context diagrams are for external entities, data storage and data flows and process.

Here in the following context diagram, I represent my doctors directory app which has one external entity i.e. user of the app and the data flowing in and out of the system is the doctor's details.

4.1.2 Level Diagram

The context-level DFD is then exploded to produce a Level DFD which models the details of the system. The Level DFD shows how the system is divided into sub-systems (processes), and how each processes deals with one or more of the data flows to or from an external entity, and how the processes together provide all of the functionality of the system. The level DFD also identifies the internal data stores which must be there for the system to do its job, and shows the data flow between the various parts of the system.

In the below Level 1 DFD 4.2, the doctors directory app has been decomposed further into 5 processes which are namely Login, Welcome Screen, View department's list, View doctor's list, Contact details of doctor. Each process is accessed by the user.

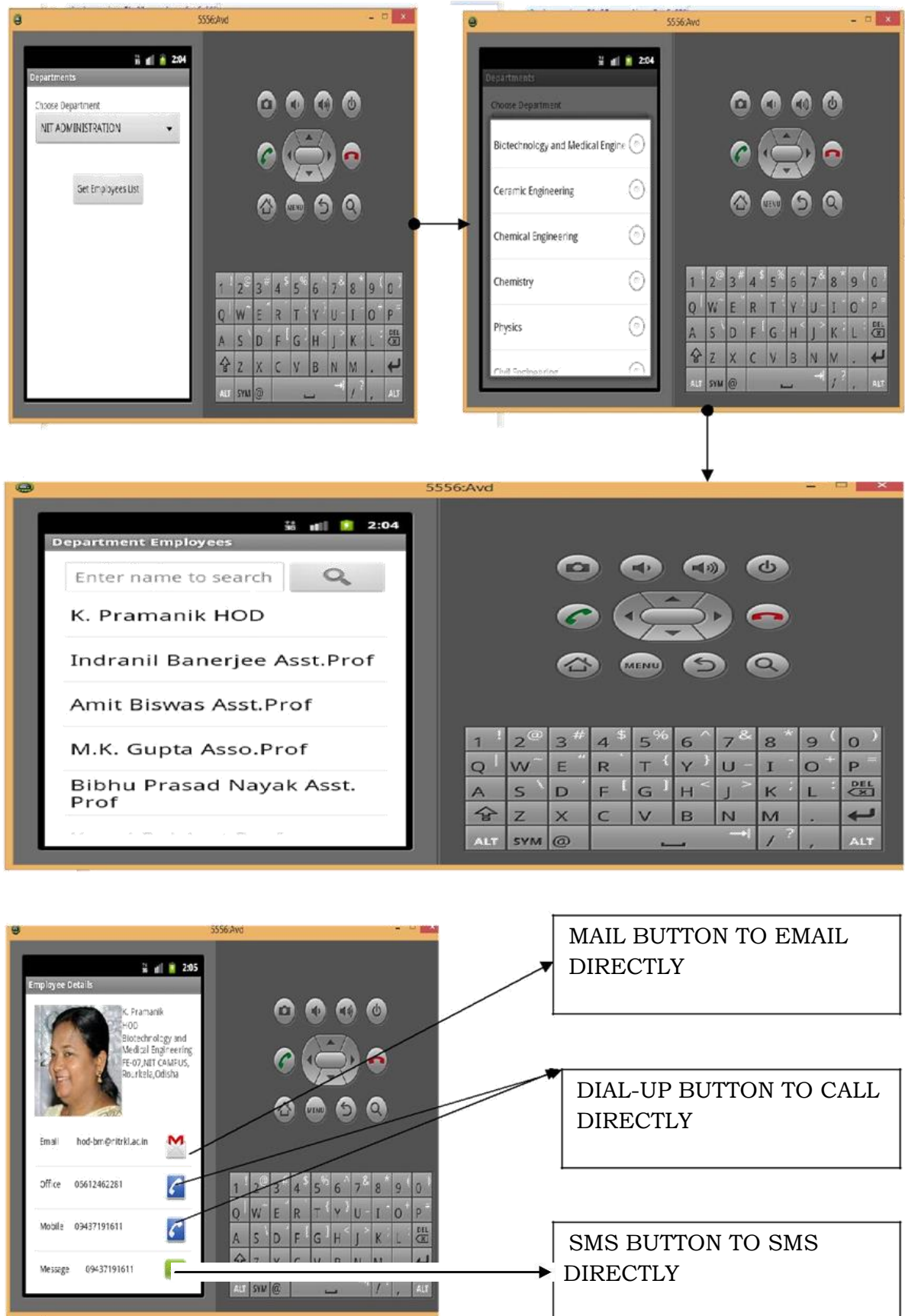


Figure 4.3: Level Diagram

And there is data storage namely Server database, Department details, Corresponding doctor's list, Doctor's details which are used in the system.

4.2 Object Oriented Software Design

In this design approach the system is viewed as a collection of entities (objects). Each object manages its own properties (attributes) and state.

4.2.1 ER Diagram

The entity-relationship diagram is a data modeling technique that graphically represents information systems entities and the relationships between those entities. An ER diagram is a conceptual and representational model of data which is used to represent the system framework infrastructure.

The ER diagram contains following elements:

- Entities
- Relationships
- Attributes

In designing the ER diagram, I identify and define all the entities, determine the interactions between the entities and determine the cardinality of the relationship. The ER diagram for the doctor's directory app represents all the entities namely User, Department, Doctor's Details and the various activities for connectivity used in the system. The User entity contains the attributes login, welcome screen. The department entity contains the attributes department button, list of departments, department selection tab and get doctor's list button.

Chapter 5

Coding Phase:

The coding phase deals with transforming the user requirements as described in SRS document, into a form i.e. implementable using programming language. The programming language that I used to develop is JAVA and XML. For our design to be easily implantable in a convenient programming language, the following items were design during the coding phase.

Different models were required to implement the entire design solution. The modules constituted of units such as welcome module, selection module, view module & activity module. The innovation relationship among different modules. This includes which modules call other modules.

The interface among different module identifies the exact data items exchanged among the modules.

SOURCE CODE OF DOCTORS TELEINFORMATION SYSTEM

➤ *For welcome screen:*

XML CODE:

```
<xml version="1.0" encoding="utf-8">
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:orientation="vertical">
```

```

<ImageView
    android:id="@+id/imgLogo"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:src="@drawable/nitlogo" />

</LinearLayout>
<menu xmlns:android="http://schemas.android.com/apk/res/android" >

    <item
        android:id="@+id/action_settings"
        android:orderInCategory="100"
        android:title="@string/action_settings"/>

</menu>

```

JAVA CODE:

```

package com.nit.nitdirectory;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;

public class SplashActivity extends Activity {

    // Splash screen timer
    private static int SPLASH_TIME_OUT = 2000;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash);
    }
}

```

```

        new Handler().postDelayed(new Runnable() {

            @Override
            public void run() {
                // This method will be executed once the timer is over
                // Start your app main activity
                Intent i = new Intent(SplashActivity.this,
                DepartmentsActivity.class); startActivity(i);
                // close this activity
                finish();
            }
        });
        SPLASH_TIME_OUT);
    };

```

➤ *For Home Screen:*

XML CODE:

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent" android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".DepartmentsActivity" >

```

```

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/caption_dept_activity" />

```

```

<!--<ListView

```

```
        android:id="@+id/deptList"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/textView1"
        android:layout_marginTop="22dp" >
</ListView> -->
```

```
<Spinner
    android:id="@+id/spnr_dept"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/textView1"
    />
```

```
<Button
    android:id="@+id/btnSubmit"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/btnSubmit"
    android:layout_marginTop="120dp"
    android:layout_centerInParent="true"
    android:text="Get Employees List" />
```

```
</RelativeLayout>
```

JAVA CODE:

```
package com.nit.nitdirectory;
import android.app.Activity;
import android.content.Intent;
import android.content.res.Resources;
import android.net.Uri;
import android.os.Bundle;
```

```
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ImageView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

public class EmployeeDetails extends Activity {

    ImageView img;
    TextView emp_nm;
    TextView emp_title;
    TextView emp_dept;
    TextView emp_city;

    ListView listView1;

    String fnm;
    String lnm;
    String title;
    String city;
    String dept;
    String ofc;
    String mob;
    String eml;
    String pic;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_employee_details);
    }
}
```

```

        img = (ImageView) findViewById(R.id.emp_img); emp_nm
        = (TextView) findViewById(R.id.emp_dtls_nm); emp_title
        = (TextView) findViewById(R.id.emp_dtls_title); emp_dept
        = (TextView) findViewById(R.id.emp_dtls_dept); emp_city
        = (TextView) findViewById(R.id.emp_dtls_city);

        Intent iin= getIntent();
        Bundle b = iin.getExtras();

        if(b!=null)
        {

                fnm = b.getString("firstname");
                lnm = b.getString("lastname");
                title = b.getString("title");
                city = b.getString("city");
                dept = b.getString("department");
                ofc = b.getString("office");
                mob = b.getString("mobile");
                eml = b.getString("email");
                pic = b.getString("pic");

                System.out.println("EmployeeDetails:: "+fnm+ " " +lnm+ " " +title+ " " +city+ " "
                +dept+ " " +ofc+ " " +mob+ " " +eml+ " " +pic);

        }

        Resources res = getResources();
        int resourceId = res.getIdentifier(pic, "drawable",
        getPackageName()); img.setImageResource( resourceId );

        //String str = fnm+ " " +lnm+"\n"+title+"\n"+dept+"\n"+city;
        emp_nm.setText(fnm+ " " +lnm);

```

```
emp_title.setText(title);
emp_dept.setText(dept);
emp_city.setText(city);
```

```
Emp_Det_Row Emp_Det_Row_data[] = new Emp_Det_Row[]
{
    new Emp_Det_Row(R.drawable.mail, "Email \t "+eml), new
    Emp_Det_Row(R.drawable.phone, "Office \t "+ofc), new
    Emp_Det_Row(R.drawable.phone, "Mobile \t "+mob), new
    Emp_Det_Row(R.drawable.sms, "Message \t "+mob),

};
```

```
Emp_Detls adapter = new Emp_Detls(this,R.layout.employee_details_list,
Emp_Det_Row_data);
listView1 = (ListView)findViewById(R.id.emp_contact);
listView1.setAdapter(adapter);

listView1.setOnItemClickListener(new OnItemClickListener() {
```

```
@Override
public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,long arg3) {
// TODO Auto-generated method stub
switch(arg2){

case 0:
if(!eml.equals("Not Available")) {
Intent emailIntent = new Intent(android.content.Intent.ACTION_SEND);
    emailIntent.setType("plain/text");
    emailIntent.putExtra(android.content.Intent.EXTRA_EMAIL,
Uri.parse("Email:"+eml));
startActivity(Intent.createChooser(emailIntent, "Please select Email Client"));
```

```

        }
    else{
        Toast.makeText(getApplicationContext(), "Email not available.",
        Toast.LENGTH_LONG).show();
    }
    break
    case 1:
        if(!ofc.equals("Not Available")) {
            Intent intent = new Intent(Intent.ACTION_CALL);
            intent.setData(Uri.parse("tel:"+ofc));
            startActivity(intent);

            else{
                Toast.makeText(getApplicationContext(), "Office phone number not available.",
                Toast.LENGTH_LONG).show();
            }
        }
        break;

    case 2:
        if(!mob.equals("Not Available")) {
            Intent intent = new Intent(Intent.ACTION_CALL);
            intent.setData(Uri.parse("tel:"+mob));
            startActivity(intent);

        }else{
            Toast.makeText(getApplicationContext(), "Mobile phone number not available.",
            Toast.LENGTH_LONG).show();
        }

        break;

    case 3:

        if(!mob.equals("Not Available")) {
            Intent intent = new Intent(Intent.ACTION_SENDTO);

```

```
intent.setData(Uri.parse("smsto:" +
Uri.encode(mob))); startActivity(intent);
    }
else{
Toast.makeText(getApplicationContext(), "Mobile phone number not available.",
```

Chapter 6

Implementation

The Application is implemented according to three tier architecture which is commonly known as Client-Server architecture, where Client is the consumer of the services, also the requester of services whereas the server side is the provider of services. The third layer that is the middle layer that converts the users requests into server understandable form.

The system developed contains Android application in its front end i.e. the client side. The middle layer that connects the client side to the server side is the Jason (JavaScript Object Notation) Parser that is primarily used to transmit data between server and server application.

The server side comprises of the MySQL. The android API uses the PHP to connect to the database or server database.

6.1 Android API

API stands for Application Programming Interface. It contains the following features-

- Accept request by POST/GET method
- Execute PHP class to update/access data from database
- Give output in Jason format

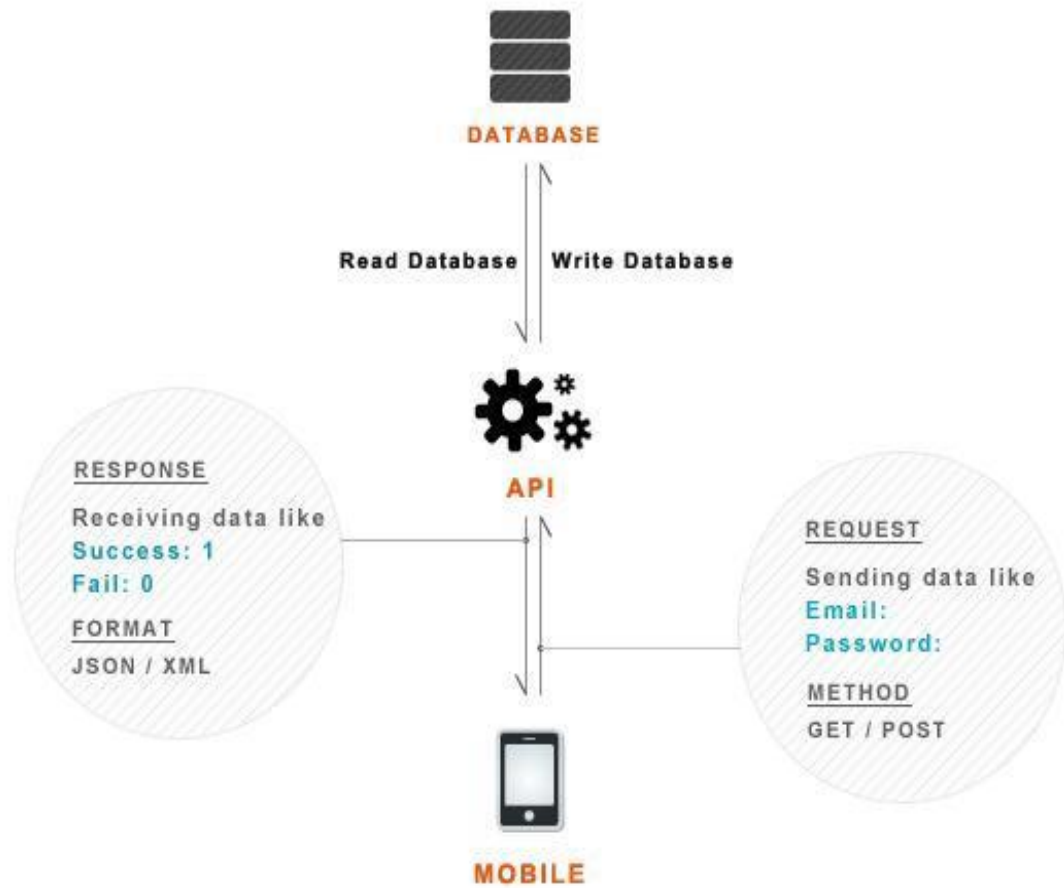


Figure 6.1: Application Programming Interface

Chapter 7

Graphical User Interface

The application GUI is developed using the android SDK and the various forms are as shown below.



Figure 7.1: Splash Screen



Figure 7.2: User Screen

Department Doctors

Enter name to search

Dr. Kastu Sahu MBBS

Dr. S N Panda MD


Dr. Kiran Mishra MD

Dr. Laxmikanta Biswal MBBS

Dr. Rakesh Kumar Jain MBBS

Figure 7.3: Doctors List under Selected Department

Doctor Details



Dr. Kastu Sahu
MBBS
Orthopaedics
34, Tarngini Market, Civil Township, Panposh Road, Rourkela, Odisha

Photo not available


Email: sahkastu@gmail.com

Office: 06612518952

Mobile: 9437004560

Message: 9437004560

Doctor Details



Dr. Laxmi Narayan Mishra
BDS
Dentist
Plot No-C/94, Sec19, Rourkela, Odisha

Photo not available

Email: mishraln@gmail.com

Office: 06612645929

Mobile: 9437085230

Message: 9437085230

Figure 7.4: Doctors Contact Details

Chapter 8

Conclusions and Future Work

8.1 Conclusion

The doctor's directory system through mobile devices is a very effective tool which can be used to a great extent. The system is portable and can be easily installed and used on any mobile phones supporting Android OS.

The use of this system can result in a reduction of number of hours spent in searching for doctors and contacting them at the time of need. It also provides an interface which is easy to understand by the users and greatly helps in adapting to the use of this system.

8.2 Future Work

The application can be further enhanced and several other functionalities can be added. The application can be made for other platforms like windows, apple, and blackberry. The present system is only applicable to android devices.

We can enhance the application by creating directory of doctors at country, state or district level. The feature to update the details of the doctors at a later stage can also be implemented. The system can also be enhanced by using voice recognition feature of the Android.

References

- [1] Android Developer Webpage. <http://www.developers.android.com>.

- [2] Roger S Pressman. Software Engineering: A Practitioner's Approach. 2010.

- [3] Visual Paradigm International. Visual Paradigm for UML 10.1 Community Edition.
<http://www.visual-paradigm.com>.

- [4] Gabriel Fielding. Android tutorial webpage. <http://www.androidhive.info>.

- [5] Jens Nagel & Rich Fulcher, "Android Design Structure in Android Design".

- [6] IBM Worklight V6.0.0 Getting started, "Setting up your Android Development Environment.

- [7] Scott Davis, "Google Maps API.V2".

- [8] <https://www.stackoverflow.com/android/experts/blogspot.html>

- [9] <https://www.vogella.com/android/tutorials.html>

- [10] <https://www.mykyong.com/android/tutorial.html>

- [11] <https://www.developers.google.com/console.html>