ANDROID

Applications for Automation purposes

Under the guidance of

Prof. Paresh G. Kale

Submitted by

Swayan Jeet Mishra

(711EE3071)

Department of Electrical Engineering

NATIONAL INSTITUTE OF TECHNOLOGY ROURKELA

Android Applications for Automation purposes

A Thesis submitted to the

National Institute of Technology Rourkela

in partial fulfillment of the requirements

of the dual degree of

Bachelor of Technology

in

Electrical Engineering

and

Master of Technology

in

Control and Automation Engineering

by

Swayan Jeet Mishra

Roll No. 711EE3071

under the Supervision of

Prof. Paresh Govind Kale

May 2016

Department of Electrical Engineering

National Institute Of Technology Rourkela

Department of Electrical Engineering

National Institute Of Technology Rourkela

May 23rd, 2016

Supervisor's Certificate

This is to certify that the thesis entitled, **"Android Applications for Automation purposes"** submitted by **Swayan Jeet Mishra** in partial fulfilment of the requirements for the award of dual degree of **Bachelor of Technology** in **Electrical Engineering** and **Master of Technology** in **Control and Automation Engineering** during 2011-2016 at the National Institute of Technology Rourkela is an authentic work carried out by him under my supervision and guidance. Neither this thesis nor any part of it has been submitted for any degree or diploma to any institute or university in India or abroad.

Paresh G. Kale

Dedicated to

-My Dad and Mom

Hi Mom & Dad *waves*

- To my sister, hoping that someday she will try to copy this

-Three of my companions Pasha, Raj & Prayas through many a long nights of Engineering

-All of my control mates for sticking with me until the very end

-All those Technology Freaks and Geeks who are making the world an Awesome place to live in.

Swayan Jeet Mishra

Acknowledgement

My most recent five years venture at **National Institute of Technology, Rourkela** has added significant and valuable experiences to my life. I have utmost regard and adoration for this institute, and I shall always remember the individuals who have made this environment so exceptional and extraordinary. Now it is time to proceed onward to my future. Before I go any further, I like to express my sincere gratitude to those who have assisted me along the way.

First and foremost, praise and heartfelt thanks go to the Almighty for the blessing that has been showered upon me in all my endeavors.

I would like to express profound gratitude to my project supervisor, **Professor Paresh Kale,** for providing a project of my interest. His untiring exertion, commitment, and way of supervision is highly brain stimulating and extracts the best from a student. He has regularly given me the opportunity to envision, execute and analyze, but has coincidently guided me sharply to keep on track throughout the project work. It has helped me a lot for self-development for which I am obliged the most.

It gives me immense pleasure to acknowledge my batch mate **Bharat Apat,** of Department of Industrial Design for helping me in the graphical user interface design of the project.

May 23rd, 2016 Swayan Jeet Mishra

NIT Rourkela

Abstract

In recent years, the number of network-enabled smartphones everywhere has been increasing fast. With the rapid expansion of the Internet, people have been trying to reduce manual intervention as much as possible. A variety of sensors are embedded in today's smartphones which make interfacing with the outside world, easy. The majority of the smartphone users have Android as the operating system. So in the world of smartphone Android has the largest platform as compared to other operating systems. So in this project work Android is used to automate some of the simple day to day manual activities.

The thesis represents the design and development of simple android applications which are used to automate simple tasks. All the applications are compatible with Android 2.1 onwards. The designs of the proposed applications are on top of a Web interface which uses RESTful API as the communication protocol between client applications and web service. The applications take much advantage of sensors and techniques pre-installed in Android smartphones. The proposed applications follow optimizations according to the best practices recommended by Google, to increase user experience and reduce power consumption. The first one is PVSys, an Android application which gives the details of equipment required for solar pump installation at user's backyard or a full solar panel installation at user's house. The second one is BizCard, which automates the task of storing user's business cards in digital form and retrieving the contacts when required. The third application, Auto Attendance Manager, automates the task of taking attendance for the teachers and lecturers without the aid of any external device. The Auto Attendance Manager can be integrated easily with the present application of the Institute.

Automation of simple things with Android becomes easy. Moreover, a layman can work on Android interface since it is easy to implement and design the layouts. In today's world when every device is trying to communicate its information to the internet, Android interface with Bluetooth or Wi-Fi can be used for the same.

Keywords: Android, PVSys, BizCard, Auto Attendance Manager (AAM), Wi-Fi, Solar energy, Stroke width transform, Optical Character Recognition, Java, mobile applications, performance, design pattern.

Contents

List of Figures

List of Tables

Chapter 1. Introduction

Smartphones have become a part and parcel of everyone's life. With the launching of a vast number of Chinese smartphone vendors, these have even become cheaper, and almost everyone has been able to afford a smartphone. In the smartphone world, Android has managed to take a significant part of the global smartphone market, becoming a clear overseer in year-to-year growth [1]. With a vast and incredibly fast growing number of users and vendors, incorporated with different kinds of sensors, smartphones can be regarded as an attractive platform for automating simple manual tasks of day to day life.

Android is a mobile operating system currently developed by Google [2], based on the Linux kernel [2] and designed primarily for touchscreen-based mobile devices such as smartphones and tablets [2]. Android's user interface is mainly based on direct manipulation, using touch gestures that loosely correspond to real-world actions, such as swiping, tapping and pinching, to manipulate on-screen objects [2], along with a virtual keyboard for text input [3]. In addition to touchscreen-based devices, Google has further developed Android TV for televisions [4, 5], Android Auto for cars [6, 7], and Android Wear for wrist watches [8, 9], each with its specialized user interface [2]. A large number of netbooks [10], game consoles [11], digital cameras [12] and other electronics devices also use variants of Android as the operating system.

1.1 Introduction to Android

The World is contracting with the growth of cellphone technology. As the number of users is increasing day by day, facilities are also increasing. Starting with simple regular mobile phones which were used just for receiving and making phone calls, mobiles have changed our lives and have become a part of it. Now they are not used just for making calls, but smartphones have frequent uses, such as a Camera, Music Player, Tablet, T.V., and Web browser [13]. Moreover, with the new technologies, new software and operating systems are required.

Operating Systems have developed in last 15 years. Starting from black and white phones to modern smartphones or mini computers, mobile OS has come far away. Especially for smartphones, Mobile OS had significantly evolved from Palm OS [14] in 1996 to Windows Pocket PC in 2000 [15] then to Blackberry operating system [16] and Android [17].

One of the most widely used mobile operating system these days is Android. Android is a software bunch consisting not only the operating system but also middleware and key components. Andy Rubin, Rich Miner, Nick Sears and Chris White founded Android Inc. in Palo Alto of California, U.S., in 2003 [3]. Later Android Inc. was acquired by Google in 2005 [2]. After the first release, there have been updates in the version of Android.

1.1.1 History of android

Android is an operating system for portable devices such as smartphones and tablet computers [3].Android inherits the core functionality from Linux kernel. Google purchased the initial developer of the software, Android Inc., in 2005.The Open Handset Alliance [18] led by Google, resulted in the development of Android. The Open Handset Alliance [18], a crew of 84 hardware, telecommunication and software companies devoted to advancing open standards for a mobile device, first unveiled Android on November 5, 2007 [19].

The Open Handset Alliance shares a common motive of fostering innovation on mobile devices and giving consumers a far better user experience than much of what was available on earlier mobile platforms [19, 18]. By providing developers a different level of openness that enables them to work more collaboratively, Android accelerated the pace at which modern and compelling mobile services are made available to consumers. Google has symbolized Android by the green robot to the right [3, 2].

Android has evolved rapidly since its launch. Google has named all projects of Android after desserts [3, 2, 13]. The Table 1.1 below, lists all versions of Android along with their useful features. The table illustrates the rapid pace of development of Android project. Android is developed "on Internet time", that is much faster than the old style of development (for example Windows releases which are several years apart).

Table 1.1 An Overview of different versions of Android showing details of version name, date of release and main features [3]

1.5 Cupcake	• Release Date: April 27,2009 • Essential Features : 3rd party widgets, Keyboards, Video recording.
1.6 Donut	• Release Date: September 15, 2009 • Essential Features: Voice Search, Text to speech
2.0-2.1 Eclair	• Release Date: October 26,2009 • Essential Features: Better sync, exchange support
2.2-2.2.3 Froyo	• Release Date: May 20,2010 • Essential Features: Dalvik Virtual Machine, SD Card installation, Wireless Hotspot
2.3-2.3.7 Gingerbread	• Release Date: December 6,2010 • Essential Features: On-screen keyboard, Improved copy-paste feature, Near Field Communication
3.0-3.2.6 Honeycomb	• Release Date: February 22,2011 • Essential Features: Cohesive look, Fragments API
4.0-4.0.4 Ice Cream Sandwich	• Release Date: October 18,2011 • Essential Features: Navigation on-screen, Folder creation, Swipe gesture
4.1-4.3.1 Jelly Bean	• Release Date: July 9,2012 • Essential Features: Google Now, Expanded notifications, Project Butter
4.4-4.4.4 KitKat	• Release Date: October 31,2013 • Essential Features: Google Hanouts, Hidden status bar, Google Drive
5.0-5.1.1 Lollipop	• Release Date: November 12,2014 • Essential Features: Material Design, Android Runtime, OpenGL ES 3.1
6.0-6.0.1 Marshmallow	• Release Date: October 5,2015 • Essential Features: Device Protection, Sound Profiles

1.1.2 Flavors of Android

It is natural that there are several different flavors of Android. The Android platform is made available under developer-friendly open-source licenses, which gives device manufacturers and mobile operator's significant freedom and flexibility to design products. That flexibility also means there are several different "flavors" of Android. The biggest device manufacturers often put their theme on top of Android, which means the User Interface on a smartphone from one manufacturer may be different from the UI from another. Table 1.2 shows few examples of various flavors of Android with their manufacturer names.

Table 1.2 An Overview of various flavors of Android with their manufacturing companies [20]

Interface	Phone/Manufacturer
Vanilla	Nexus One, Nexus S, Galaxy Nexus
Touch Wiz	Samsung
Sense	ITC
Moto Blur	Motorola
Ux	Sony Ericsson

1.1.3 Android Revolution

It has been nine years since the people first bought Android smartphone. Within this period, Android has come a spectacular way from an early stage project with prototype devices to being the most popular OS in the Indian smartphone market [21]. Moreover, according to some surveys, it dethroned Nokia's Symbian from its ten-year top position [22]. Without any doubt, Android is now an international leader regarding a year-to-year growth and still has great potential for a future increase in sales. Table 1.3 shows the data collected in 2015 compared with 2014 [23]. Despite significant growth in the whole market (70% year-to-year), Android managed to increase its shares in 2015 by 15.5% selling approximately ten times more units [23].

Table 1.3 Worldwide smartphone sales by operating system (2015) [23]

Operating System Name	Units Sold in 2015	2015 Market Share (%)	Units Sold in 2014	2014 Market Share (%)
Android	298,797	84.7	254,354	83.3
iOS	46,062	13.1	38,187	12.5
Windows	5,872	1.7	9,033	3.0
Blackberry	977	0.3	2,420	0.8
Others	11,336	0.3	1,310.2	0.4

Figure 1.1 and Figure 1.2 shows market shares of smartphones by vendors as well as by operating systems. The table shows Android has almost captured the entire market. Despite the latest technologies used by Apple, Android has captured the Indian market since December 2013 due to its open source, an enormous amount of applications in App Store and cheap smartphones [24].

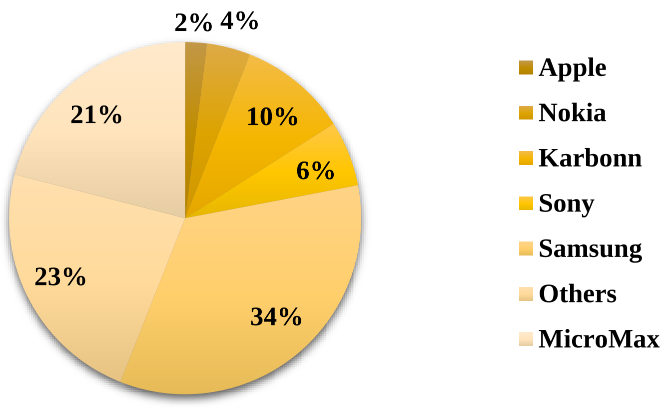


Figure 1.1 Pie Chart showing the Indian market shares by brand names [25]

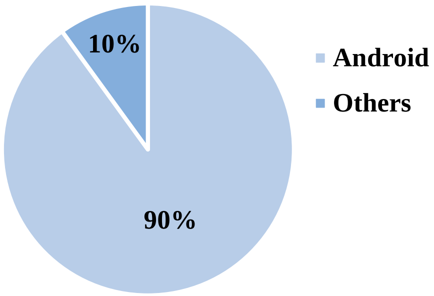


Figure 1.2 Pie Chart showing the Indian market shares of smartphones by operating systems [24]

Naturally, Android itself is not an individual operating system. It comes in many versions which update its architecture every half year or even more frequently. So far all new updates have backward compatibility. The most up-to-date chart (Figure 1.3) from Android Developer and website shows a distribution of Android versions. Data has been collected based on devices accessing the Google Play Store in a seven-day period ending on May 2, 2016. Therefore, these statistics exclude devices operating on various Android versions that do not access the Google Play Store, such as Amazon's Fire tablets [2].

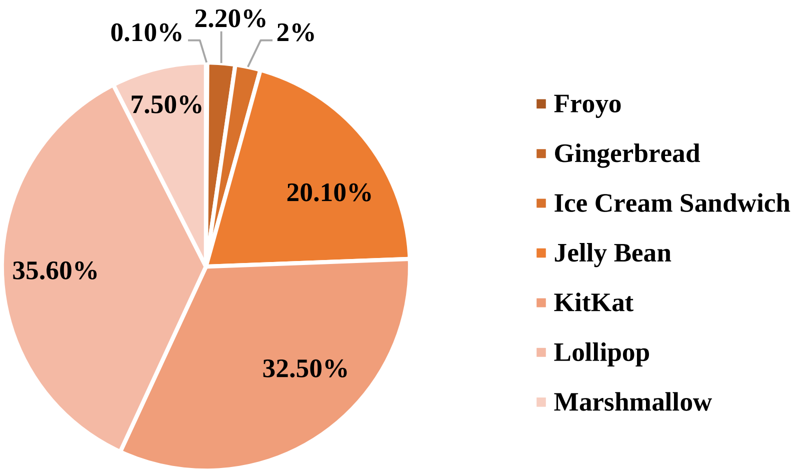


Figure 1.3 Pie Chart showing the distribution of different versions of Android by user percentage [26]

The recently released Android 5.0 (Honeycomb) and Android 2.3 (Gingerbread) with the following 2.3.3 quick update are not visible on the chart (2.7% together). Nevertheless, 93% of Android devices are now running 4.x versions or newer. Naturally, these statistics apply to devices which have connected to the Google Play Store Android Market. Hence, it is not speaking the actual distribution of Android versions on the global market. However, it definitely can be regarded as the long-term trend, showing the speed of the upgrade process. The above observation is vital as many developers have decided to release applications that support only versions from 4.1 onwards because the Android 4.x platform offers new, easy to use, pre-implemented solutions that prove to be useful.

The Appendix - I shows how fast the process of replacing the old version of Android with the new ones is. The vertical axis has an accuracy of barely two weeks, but still, this little amount of time is quite enough to observe a significant change. There are at least two good reasons why it is happening so fast. The first one is that smartphone vendors want to release new phones regularly (and customers also buy them regularly), so frequent updates create such a chance. The second is that Android is an open source project, so if the device's hardware fulfills the new version's minimal requirements, it is just a matter of time to get an upgrade [1].

When designing a new application in Android, one has to take into consideration not only a variety of platform versions but also user device's specification. An important feature is probably the screen resolution. Oppositely to Apple's iPhone, Android platform is not limited to a single device and Google does not manufacture its phones. As previously mentioned, Android is a software stack, and thus, it can be installed on practically any device that satisfies a minimal set of requirements, and it was meant to support a variety of resolutions, screen orientations, and densities. The next charts (Figure 1.4 and Figure 1.5) indicates that the vast majority of devices have either large or normal resolution screens, but it is good practice to support other options (especially as larger density screens are likely to appear shortly). To simplify the way that we design our user interfaces for different screen configurations, Android divides the range of actual display sizes and densities into several buckets as expressed by the table below [26].

Table 1.4 The distribution of different screen densities and resolution by user percentage [26]

Screen Size / Resolution	**ldpi**	**Mdpi**	**tvdpi**	**hdpi**	**xhdpi**	**Xxhdpi**
Small	2.1%					
Normal		4.5%	0.1%	41.1%	24.9%	15.1%
Large	0.2%	4.6%	2.2%	0.5%	0.5%	
Xlarge		3.2%		0.3%	0.7%	
Total	2.3%	12.3%	2.3%	41.9%	26.1%	15.1%

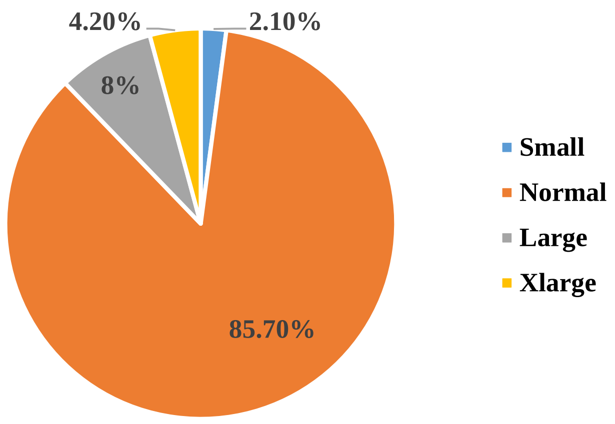


Figure 1.4 Pie Chart showing the user percentage distribution by sizes of screens [26]

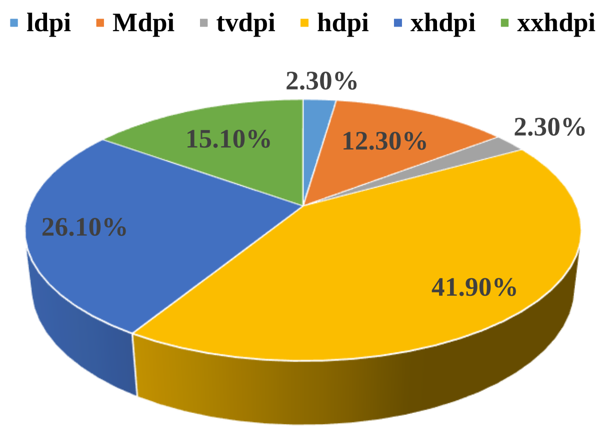


Figure 1.5 Pie Chart showing the user percentage distribution by resolution of screens [26]

1.2 Application Framework – fundamentals

Even though Android is a new software stack, it comes with proper documentation mentioned on the official developers page [26]. There are some publications and course books for people who seek basic information and introduction to the system. Additionally, a vast community of developers is actively exchanging questions and answers on web services like Google Groups [27], Android Blog [28] or Stack Overflow [29]. It does not make sense to quote large fragments of Android developers guide. Instead, this section covers the key elements of designing an Android application which is necessary to understand the following discussion about the proposed Android apps.

1.2.1 Manifest File of Android Application

Every Android application must have a single manifest file in a root directory (named AndroidManifest.xml) [30]. This file contains essential information about the application, required by the system to run it. All new components have to be registered here along with permissions needed to perform actions requested by the application. The file is also used to inform the system if a developer wishes to expose the application's data to other apps [1]. It specifies the application's reaction to various system events such as incoming calls or photo 12 capturing. Within the Manifest file developers can specify if they want one of their activities (described below) to become a launch activity [30]. For the purpose of this thesis, applications which contain a launch activity are called launch applications. Such applications are added to the Android's home screen so they can be launched independently from other apps.

1.2.2 Main application components of an Android Application

All Android applications contain the following elements:

- Activity [31]: a single piece of layout/user interface (UI). In an application, it runs independently as compared to other items, although users have a smooth experience of different activities appearing on the screen in a sequence. Activities usually contain some graphical elements (customized or taken from a pre-defined set) and constitute a specific action that users can perform like writing an email or selecting a contact from a list.

- Service [32]: a separate part of the activity which does not have a graphical layout (running in a background). Service itself does not run in a separate thread because it runs in the context of the activity or the broadcast receiver activity which has begun it. It usually should use a working thread to perform some complex computation or lengthy I/O operations.
- Content Provider [33]: a method that allows applications to share data between each other or to save some information simply persistently. It is an interface which provides standard methods to access data like query, insert, update, delete. The system imposes no specific type of data structure; it can be a single file or an SQLite database. The content provider is uniquely identified by its authority and can contain many types of data objects (many tables in a database). The most common method of accessing an object is to query Content Provider with a specific content URI, which has a standard form of content.
- Broadcast Receiver [34]: a part of the application which responds to actions broadcasted by the system itself (dimming the screen, capturing a new picture) or by other areas of the application (the same one or not). Broadcast Receiver does not have its graphical representation, but it can initiate certain actions to keep users informed about its work. Creating a notification in a status bar or updating a desktop widget initiates the process.

As mentioned previously Android, is an event-driven system. The events are called Intents [35]. Intent API is an important feature that handles interactions between application components across the whole system. Developers do not have to write any additional lines of code to integrate their implementation with other ones as long as they understand their set of required Intents. Moreover, it does not make any difference whether the application sends an Intent to a separate application or another part of the same application – the procedure is the same. A good example is the Map application which supports Intents that contain a request to display specific geo-location. Developers can simply broadcast the geo-location Intent whenever they want to show a map with a specific location. Additionally, should users have installed a different app which masks the functionality of the Map application (by filtering Intents), this new application would be used automatically instead? The Intent mechanism 13 allows developers to reuse different system components easily across the platform. They can focus on really innovative and natural functionality and simply attach new components like maps or navigation to their applications which make them even more compelling and useful.

1.2.3 Process Handling of Android Applications

One of the key points when designing an operating system for mobile devices is memory management. Android is trying to do as much task as possible in the background thread without bothering anybody, but also, there is much memory for alterations and modifications. Therefore, developers should at least be acquainted with the most basic aspects of process handling not to create applications that would be superior or even ’hostile’ to the system.

Very often Android has to deal with situations of memory shortage. The system has to reclaim resources to allow new processes to run. In such cases Android ranks all active processes according to the following order:

- Foreground process [36]: the process which carries out a foreground activity or a service that is bound to a foreground activity. It means the part of the system that the user is currently interacting with, so at any given time there are only a small number such processes. These processes are killed only in ideal critical situations as not terminating them will most likely cause lack of responsiveness (or even showing error messages to the user).
- Visible process [36]: a process that does not have any active components, but still is visible to the user. A good example is a process which executes a launching activity that launches another not-full-screen activity. Visible processes are considered necessary and get killed if the system cannot find enough memory for foreground processes.
- Service process [36]: a process which hosts a service and at the moment it is neither a foreground nor a visible process. Those processes, however, may be carrying out some essential tasks for the user like playing music in a background or downloading some useful data from the Internet.
- Background process [36]: a process which handles an activity that is currently

not visible. These processes are likely to get terminated at any given time, so the operating system keeps them in an LRU (least recently used) list to ensure that the process with the activity that was most recently seen by the user is last to be killed [36].

- Empty process [36]: a process that doesn't hold any active application components. The only reason why the system keeps these processes alive is for caching purposes [36].

Developers should constantly be acquainted with the fact that their processes could get killed and re-established by the system in the background thread even if they did not suspect such a situation to happen in the first place. To allow a smooth navigation between different screens, Android introduces some helping methods to save and restore the state of activities [31]. It is good practice not only to take advantage of these activity-state methods but also to use it wisely. For instance, no lengthy tasks should ever be performed on them. A good example is an activity where users can edit a new text message being interrupted by an incoming call. The details of the previous activity are captured by the new call activity, decreasing the priority of the old one. Therefore, the process which executes the text message-editing activity is more likely to get killed. Users would probably expect to be able to keep editing the text of the message after finishing the call. However, continuation would be impossible if the application had not persistently saved the text activity details beforehand.

1.3 Motivation and Objective

The main purpose of the proposed thesis work is to show how an Android running smartphone can be used to solve a variety of problems by using the Camera and Wi-Fi technologies. The solution to the problems without the utilization of any external devices or sensors which propose a stand-alone and independent solution to the problems. Moreover, the proposed solutions are backward compatible since they do not require to change the current infrastructure. There were certain architectural challenges [1]:

- The client based interfaces of the application had to use a RESTful API to communicate with the server [1].
- The applications had to add some new features and remove the deprecated ones in comparison to other applications of the same type.
- The application had to be portable, maximizing the number of target devices.
- The application should make a use of the pre-implemented solutions designed [1] by Google developers and added to the Android platform. One objective is to explore these solutions and potentially incorporate them into the new applications [1].
- The application had to be responsive and at the same time secure. Naturally, technology constraints imposed some compromises; nevertheless, diminishing the role of any of these three factors or disregarding them was entirely unacceptable. The goals are to pre-implemented solutions and use them wisely.

Chapter two provides background information about the Android platform itself. It focuses on fundamental components of Android operating system from application developers and their community. Furthermore, it shows the most up-to-date Indian smartphone market data with some statistics about the Android platform versions.

The third chapter presents an overview of PVSys, an application designed to provide the information about equipment for solar power installation and study the development of Android applications. It also describes the use of the modules of the proposed application. The end of the chapter describes the core algorithms and user interface solutions implemented for the proposed application. In simple words, the chapter imparts answers to the following questions: what users can do with this application, how they can use it and what are the key architectural elements that help in completing the task.

The fourth chapter presents a brief description of BizCard, an application designed to automate the task of information retrieval from business cards. It also describes some of the applications which have the same usage and their methods of retrieval. The methodology section contains the architecture as well as the algorithms for text detection and segregation purpose. This chapter presents an answer to the following questions: how the users can use the application and what are the key elements that take part in automating the task. The final section highlights the performance issues of the application, followed by a discussion about the application's future and conclusion.

The fifth chapter presents an insight into Auto Attendance Manager, an application designed to automate the task of attendance collection for professors and information retrieval for students. It also describes some of the applications which have the same usage and the technology which they use. The main task of this chapter is to throw light on the Wi-Fi based communication feature of the smartphone and how he feature can be used to automate simple tasks. The methodology section contains the architecture as well as the algorithms for Wi-Fi based attendance method. The last part throws light on the difficulties faced by the application, followed by the discussion and conclusion.

The final chapter describes some information about the performance issues and the challenges face for using Android for the proposed purposes, followed by final findings and discussion about the project's future.

Chapter 2. PVSys Overview

The idea to develop PVSys came from the fundamental need of people to know the right quantity of optimized resources to harness solar energy for various uses. The basic requirement was to provide a platform for calculating the components required for installation of solar pump and solar home lighting systems. The proposed application was to be simple enough to be operated by Indian farmers and people who belong to middle-class families. Moreover, the same platform can be used by any user with the smartphone running the Android system (version 2.1 and further) to receive alerts about solar subsidy systems and solar installation information.

The current form of PVSys application is just the first, which houses only the solar pump and the home lighting system modules. The current widget was simply a smaller version and properly defined a module of the bigger, dedicated application. It also discusses the majority of planned functionality of the widget. Thanks to the design guidelines described by the Android system as well as PVSys' application's architecture, the process of developing and integrating more modules with the bigger launch application (visible on the menu screen) should not be complicated and definitely, does not require any substantial changes in the existing code. The application is available for download at https://play.google.com/store/apps/details?id=com.nitrec.pvarray&hl=en.

2.1 Use Cases of PVSys Application

The main concept of PVSys application was to design a system that would be both functional and easy to use. The use cases of PVSys consists of two categories based on the usage of modules. The first one is a farmer who wants to use PVSys to install a solar pump in his farm. They just have to put the quantity of water to be pumped and the height to which the pump raises the water. The farmers can also share their data with other fellow farmers. The module section describes the details of the first module. The second perspective is the users who want to use home lighting module to calculate the requirements of installing devices to harness solar energy for the purpose of home lighting. The users have to enter the details of the devices/equipment they want to run from solar power, and PVSys does the rest. It also supports data sharing. Users can generate PDFs if required.

2.1.1 Module 1 – Solar Pump Module

The Solar pump module is useful for farmers' and other people who want to use solar energy for water pumping. The main aim of this module is present the user the equipment details as simply as possible. The interface was probably designed in the simplest possible manner so that even farmers' can easily use it. The module takes two inputs. The first one is the depth from which the pump raises the water in meters. The second one is the amount of water to be raised with its unit in Liters per hour per day. The module provides the number of Photovoltaic panels and the wattage of the motor required. The application calculates the number of Photovoltaic panels by taking the wattages of some commonly available solar panels. The users can view the details of the calculations by clicking on the PDF button.

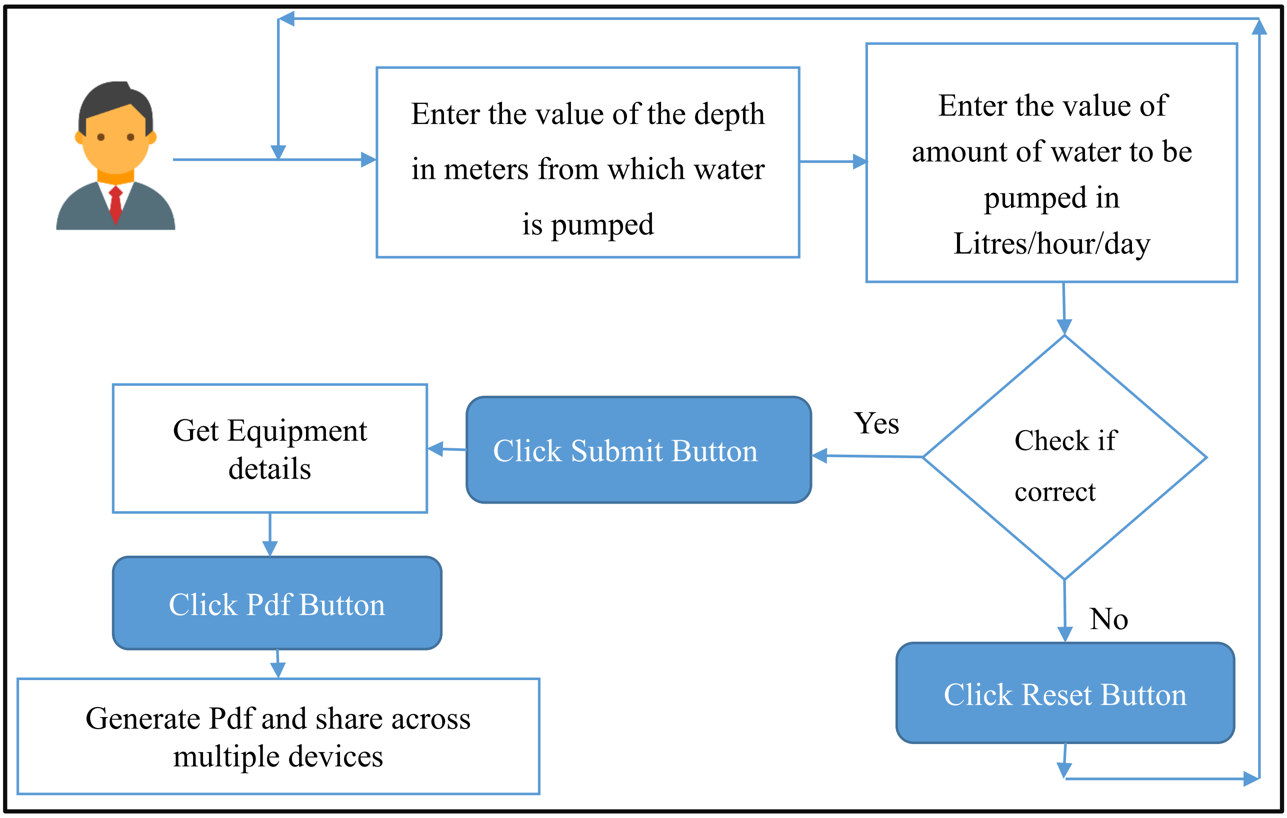


Figure 2.1 User Flow Diagram for Solar Pump Module (Module - 1)

2.1.2 Module 2 – Home Lighting Module

The Home Lighting Module is for the users who want to harness the power of solar energy in running the day to day appliances like television, ceiling fan, and other electrical and electronic devices. The main aim of this module is to provide the information of required devices as much simply as possible. The module takes its input in a tabular format as shown in Fig. The user selects the name of the appliance from the name list, the wattage of the appliance, the number of hours per day the appliance is put to use, and the number of appliances. The module calculates the battery bank voltage and Ampere hour of the battery, the total number of Photovoltaic panels required the number of panels connected in series, the number of panels connected in parallel. The module takes commonly available battery wattages and Ampere hour for calculation of wattage of photovoltaic panels. The user can view the details of the calculations by clicking on the PDF button. The sharing button only shares the download link of the application. In the next version of the application, users can share the data among themselves.

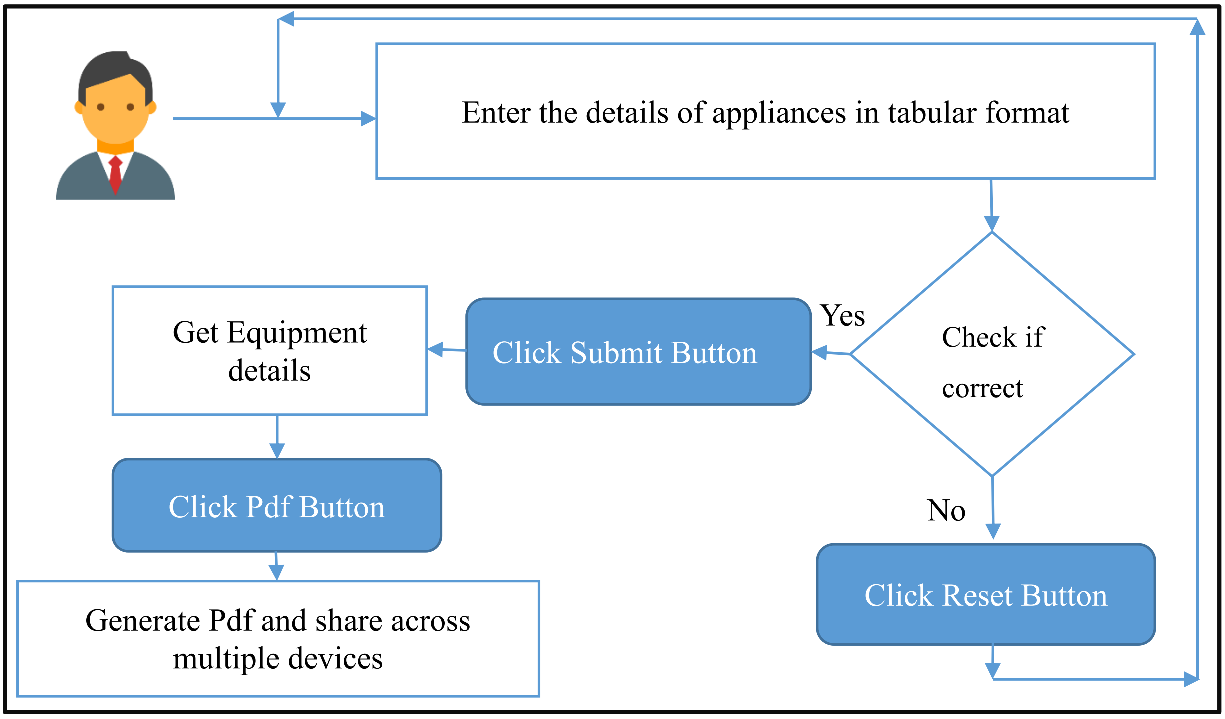


Figure 2.2 User Flow Diagram for Home Lighting Module (Module - 2)

2.2 Architecture of PVSys Application

This perspective presents the Architecture of the PVSys app. The architecture consists of two parts. The first one is Core part which consists of the Activities which describe the functionality of the Application. The second one consists of resources part which describes the layout and the various resources used by the application.

The Core part consists of 6 activities, namely: Home Screen Activity, Module I Activity, Module II Activity, Learn More Activity, Contact Us Activity and FAQs Activity. When a user clicks the Launcher icon, Home Activity is launched which in turn provides access to other activities. Whenever a user clicks on the respective icon on the home screen, the corresponding activity is launched. All the activities are developed on the official Android libraries. The Activities contain the functional part of the application. Apart from these core activities, there is another activity called pdf generator activity which generates pdf files from the string buffer sent to it. The pdf generator activity uses the itext android pdf library for generating Pdf files.

The resources part consists of the layout files, the menu files, the drawable files and other miscellaneous resources used by the application. The Layout file describes the layout of each of the activity. The menu file describes the system menu associated with the activity. The drawable files contain the images of the launcher icons and other menu icons used in the application. Other resources file include string values used in the application.

Figure 2.3 shows the basic decomposition of the PVSys Application. Thanks to the Android system architecture treats all of the Java modules as separate containers responsible for providing different functionality. The Home activity can only start other activities. The core activity files are not visible to the user. The user can see only the layout files. All modules are written in Java technology with many XML files used for storing static resources.

The Android platform enables developers to generalize all types of resources like layouts, strings, the configuration of drawable or system components into XML files. The style mentioned above is not always obligatory, but it helps in maintaining the source code and making it more vivid. Moreover, developers who decide to follow the guideline mentioned above (which is one of the most 20 important best practices in developing Android applications) do not need to pay any heed about handling the different device's configurations (like density and the size of the screen).

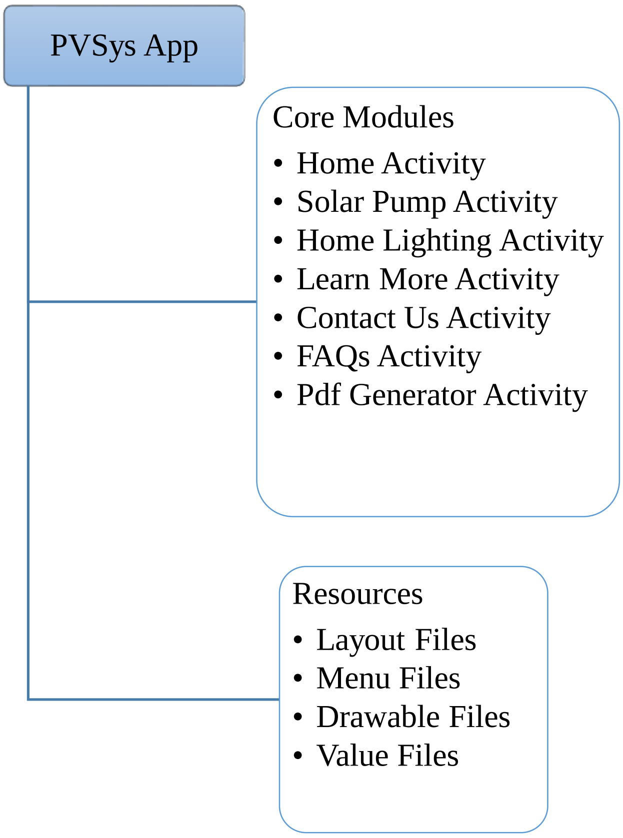


Figure 2.3 PVSys' Application architecture showing the division of various components

Android automatically detects a predefined set of folders like drawables-hdpi, drawables-mdpi, drawables-ldpi and developers should only populate them with the required versions of their layout files. The system chooses the best available file for them according to the device's configuration. The auto choosing of files based on a system's configuration is a very powerful mechanism which is available right out-of-the-box and developers are strongly encouraged to make an extensive use of it. PVSys uses layout for

XML files as in the next version of the application feature for local languages can be introduced. To produce a second language version of PVSys, just copy the XML files inside the values folder and then translate all of the values inside it into a new language.

2.3 Design Principles and PVSys' Interface

The design has become one of the vital points in developing Android applications. With the rapidly growing number of available apps, users expect them to be at the same time intuitive and elegant. It ultimately means that developers should constantly struggle to find a balance between making their interface clear but rich and making it simple. Users often perceive simple in a negative connotation as something not useful, not functional. On the other hand, the rich interface is not always intuitive, and not all of the users are willing to learn how to make use of new applications if it takes too much time. Android comes with a variety of ready-to-use UI components, and it is highly recommended to use those 'blocks' to achieve consistency with the pre-installed Android applications. However, the system does not specify any unified layouts for third party applications to allow developers to change them according to their will.

From the time of the first release of Android system, Google's developers have been collecting ideas about different design patterns which they could recommend to the broad community. These are both original ideas of people who work for Google and suggestions about UI posted on various forums or already implemented in third party applications released on the Android Market. During the previous Google, I/O conference which took place in San Francisco in May 2010 the developers have presented in a session called Android UI design patterns. The design patterns contain two separate groups. The first one is related to game development. Hence, it is not interesting from this paper. The second one provides guidelines for developing standard user interfaces, and the PVSys application takes advantage of many of them.

The current form of PVSys application uses the Material Action Bar theme provided by Google. Consequently, implements one of the five design patterns presented by Google (Action Bar). Nevertheless, apart from properly implementing the core tasks, the design was the second most vital part in developing the PVSys application. The design section contains into eight sub-sections. These sub-sections represent the core functionality of the application.

2.3.1 Home Screen of PVSys Application

Figure 2.4 (b) provides an overview of the home screen of the application with all its components. The main elements of Home Screen are:

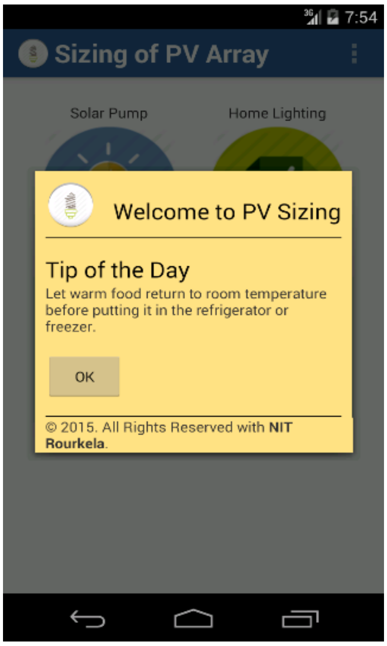
- Action Bar: The action bar contains the app name and the logo of the application. The proposed application loads the standard action bar provided by the Android and then the application runtime changes its attributes during the execution. The action bar also shows the inheritance button on moving from launching activity to its child activity. The action bar also houses the launching activity's menu.
- Custom Gird View: The custom grid view class inherits itself from the standard grid view class, but the only difference is that the custom class is used to create a grid view of image buttons by dividing the screen into two or more halves. The Home screen uses icons from Design Icons website [37]. The corresponding activity links every button's onClick attribute by passing an intent to start the related activity.

2.3.2 Tips Screen of PVSys Application

The tips screen is an interactive modal window which displays tips at the start-up of the application. The initialization of the launching activity starts the tips screen, but after its XML layout has been flattened to create objects. The design of the tips screen is in a separate XML file. The tip screen is shown only once at the start-up of the app. A Structured Query Language Database in the smartphone client stores the tips so that editing the tips will be easy.

Figure 2.4 (a) shows an overview of the tips screen of the PVSys application. The okay button cancels the tips' screen and lets the user view the home screen. The screen also gets hidden by user's gesture to click on the outside area of the screen.

(a)



(b)

Figure 2.4 (a) The snapshot of the home screen of the PVSys application (b) The snapshot of the tips' screen of the PVSys application

2.3.3 Solar Pump (Module – 1) Interface of PVSys Application

The diagram in Figure 2.5 describes the overview of the user interface of the Solar Pump module. Screen (a) describes the data to be entered by the user. It consists of two text fields with two text labels above them. The Submit button contains an onClick attribute to a sub-activity of solar pump module. The sub-activity does the calculations through an asynchronous thread and after completing the calculations submits the data to the module's principal activity.

The main module for receiving the data updates its main screen to show the output as shown in screen (b). After having generated the data, the user clicks on the PDF button to generate a pdf of the detailed calculations of the module. The application generates the pdf in an asynchronous class, which inherits the libraries from the itext Android Pdf library. The asynchronous class gets its input from the string buffer of the main activity of the Solar pump module. The pdf is stored in a specified folder and on successful storage the application shows a context menu to show the Pdf. The context menu opens the pdf using the native Android applications available in the client's smartphone to launch and read pdf documents.

The asynchronous classes are required since it enables a non-blocking GUI main thread. If the application performs calculations and pdf generation on the main thread, then the GUI becomes unresponsive for the time until the code executes. The share button just shares the download link of the application. In the next version of the app users can share their equipment data with others.

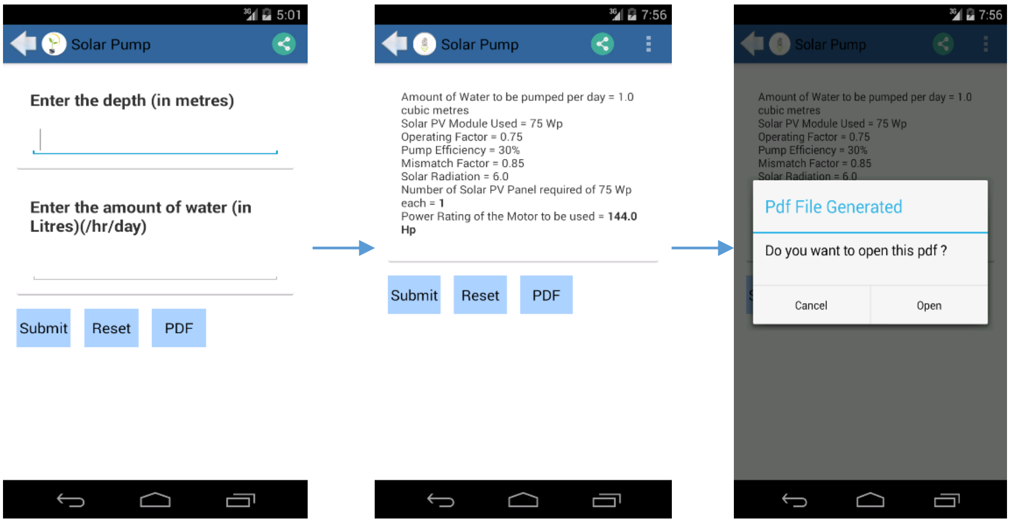


Figure 2.5 The snapshot of the module-I screen of the PVSys application where (a) shows the value entering interface (b) shows the result interface (c) shows the context menu after the application generates the pdf.

2.3.4 Home lighting Module (Module – II) Interface of PVSys Application

The diagram in Figure 2.6 describes the snapshot of the user interface of the Solar Pump module. Screen (a) describes the data to be entered by the user. It consists of a grid view with the text labels and text entry widgets in a tabular format. The Submit button contains an onClick attribute to a sub-activity of the home lighting module. The sub-activity does the calculations through an asynchronous thread and after completing the calculations submits the data to the module's principal activity.

The main module for receiving the data updates its main screen to show the output as shown in screen (b). After having generated the data, the user clicks on the PDF button to generate a pdf of the detailed calculations of the module. The application generates the pdf in an asynchronous class, which inherits the libraries from the itext Android Pdf library. The asynchronous class gets its input from the string buffer of the main activity of the home lighting module. The pdf is stored in a specified folder and on successful storage the application shows a context menu to show the Pdf. The context menu opens the pdf using the native Android applications available in the client's smartphone to launch and read pdf documents. The asynchronous classes are required since it enables a non-blocking GUI main thread. If the application performs calculations and pdf generation on the main thread, then the GUI becomes unresponsive for the time until the code executes.

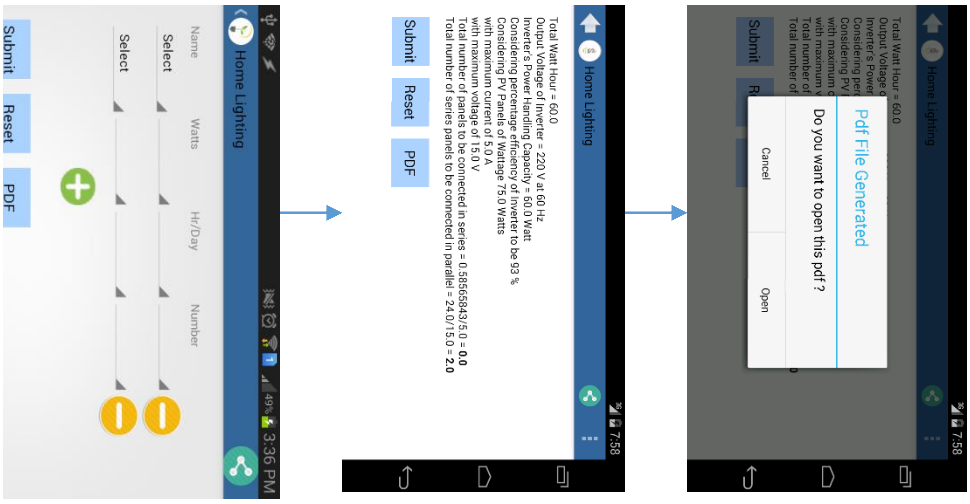


Figure 2.6 The snapshot of the module-II screen of the PVSys application where (a) shows the value entering interface (b) shows the result interface (c) shows context menu after the application generates pdf.

2.3.5 Learn More Screen of PVSys Application

Figure 2.7 shows a snapshot of the Learn More Activity. The layout of the activity consists of an expandable list adapter class, and an SQLite database stores the links and data. Clicking a heading allows the list to expand as shown in Fig (b) and on clicking the links, it launches an intent to open the link in the default browser of the user.

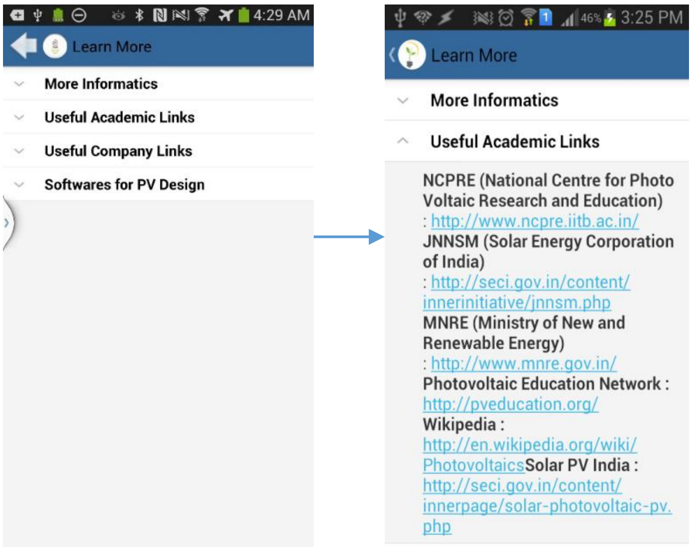


Figure 2.7 The snapshot of the learn more screen of the PVSys application where (a) shows the expandable list adapter (b) shows the expanded list adapter after the user clicks it.

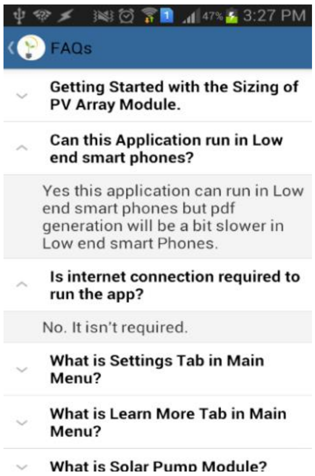
The Learn more tab provides more information about solar photovoltaic, the companies working on solar energy, research institutes, and manufacturing companies. It consists of four subheadings:

- More Informatics: This submenu contains the websites which provide more information on PV Sizing.
- Useful Academic links: This submenu contains websites which provide useful academic information on solar energy and Photovoltaic Modules.
- Useful Company links: This submenu contains websites which provide valuable enterprises in the field of solar energy and photovoltaic.
- Soft wares for PV Design: This submenu contains the connection of the soft wares which are useful for PV Design.

2.3.6 Contact Us and FAQs Screen of PVSys Application

Figure 2.8 shows an overview of the Contact Us screen and FAQs screen. The Contact Us screen consists of a Text Label to give the details of the Project. The email label when clicked opens in Gmail application directly. The provide feedback button when clicked launches an intent that inaugurates a feedback form in the default browser of the user. The FAQs screen implements an expandable grid adapter. An SQLite database stores questions and answers. When the user clicks a question, it opens up to display the answer. The user gets answers to the most commonly asked issues in the FAQs section.

(a)



(b)

Figure 2.8 The snapshot of the (a) Contact Us screen of the PVSys application (b) the expanded list adapter of the FAQs screen

2.3.7 Data Sharing Options

The present version of the proposed application contains the options to share the download link of the application. The share button, when clicked launches an intent to share a link with the messaging and social networking platforms like Whatsapp, Facebook, and Twitter. The next version of the application implements a common model of the data object which can be serialized and shared between different users so that they can be able to share their data values.

2.4 Functionality of the Modules of PVSys Application

The functionality of the app describes the basic algorithm used for the development of both the modules of the application. This section also describes the library and pdf class used for Pdf generation. The source code of the algorithm is developed using the Java language in the Android Studio. Both the modules implement standard optimization procedures.

2.4.1 Algorithm – Solar Pump Module (Module – I)

The algorithm for Module-I is a developed version of the problem mentioned in the solar photovoltaics book [38]. The only difference is that after calculating the wattage of motor required, the application's algorithm calculates the number of solar panels needed to wattages of some commonly available solar panels, and the shows the minimum number of panels to the user. Figure 2.9 describes the algorithm for Solar Pump Module (Module - I).

2.4.2 Algorithm – Home Lighting Module (Module – II)

The algorithm for Module-II is a developed version of the problem mentioned in the solar photovoltaics book [38]. The only difference is that after calculating the wattage of power required, the application's algorithm calculates the number of solar panels needed to wattages of some commonly available solar panels, and shows the minimum number of panels to the user. Figure 2.10 describes the algorithm in steps for Home Lighting Module (Module - II).

2.4.3 PDF Library Integration

The proposed application uses iText Pdf library for pdf generation. The application uses a pdf class, which uses the APIs of the library to generate pdf from the string buffer provided to it as input. The application stores the pdf file in the default document directory of the smartphone. The file contains the detailed calculations used for the module. The snapshots of the sample pdf files of both the modules are shown in Figure 2.11 and Figure 2.12.

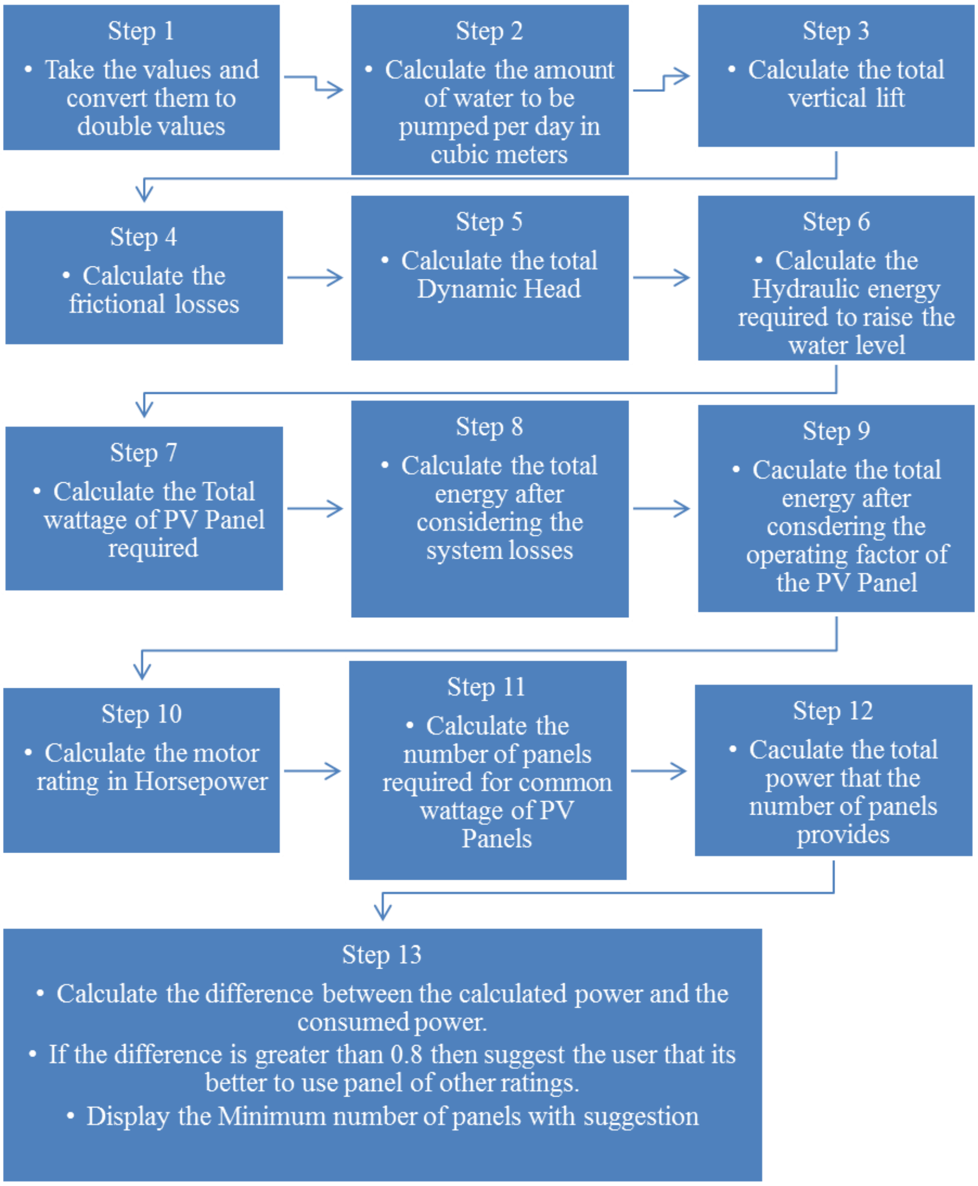


Figure 2.9 Algorithm of the Solar pump module (Module - I) of the PVSys application

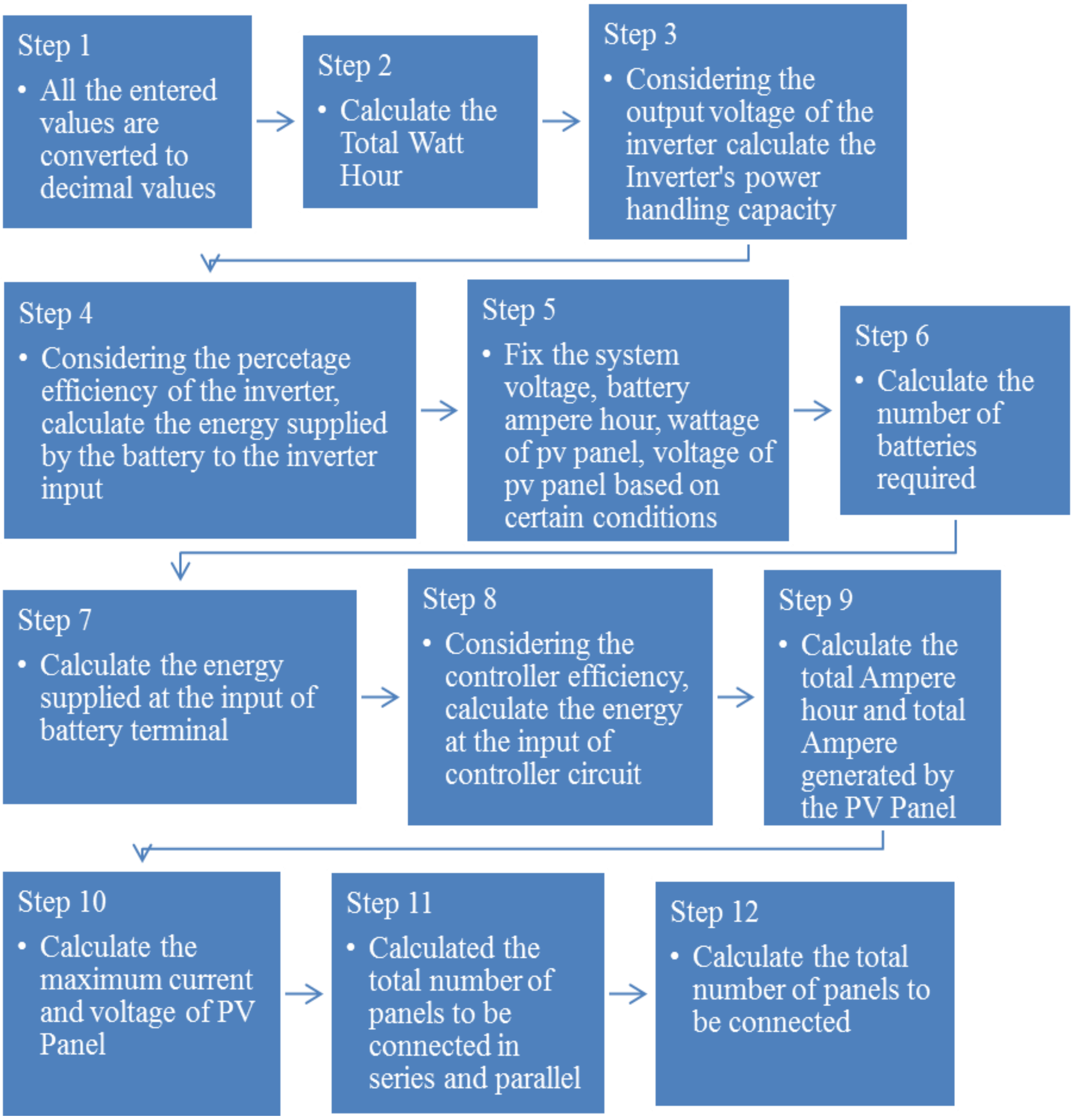


Figure 2.10 Algorithm of the Home Lighting Module (Module - II) of the PVSys application

Amount of Water to be pumped per day = 0.1 cubic metres
Total Vertical Lift = 12.0 metres
Water density = 1000 kg/m^3
Acceleration due to gravity = 9.8 m/s^2
Solar PV Module Used = 75 Wp
Operating Factor = 0.75
Pump Efficiency = 30%
Mismatch Factor = 0.85
Frictional Losses = 5% of Vertical Lift = 0.6000000000000001
Total Dynamic Head = 12.6
Hydraulic energy required to raise water level = Mass * g * tdh = 3.43
Solar Radiation = 6.0
Total Wattage of PV Panel = Total Hydraulic energy/ no. of hours pf peak sunshine/day = 0.5716666666666667
Considering System Losses = 2.241830065359477
Considering the operating factor for PV Panel = 2.9891067538126364
Number of Solar PV Panel required of 75 Wp each = 1
Power Rating of the Motor to be used = 0.004 Hp

Figure 2.11 Snapshot of the pdf file generated by Solar Pump Module

Total Watt Hour = 112.0
Output Voltage of Inverter = 220 V at 60 Hz
Inverter's Power Handling Capacity = 112.0 Watt
Considering percentage efficiency of Inverter to be 93 %
Energy supplied by the Battery to the inverter input = 112.0/0.93 = 120.43011 Wh
Choosing PV System Voltage to be 6.0 V
Battery Depth of Discharge = 0.7
Battery Ampere Hour = 48.0 Ah
Number of Batteries Required = 18.666666/(48.0*0.7)= 1
Energy Supplied at the Input of Battery Terminal = 120.43011/0.85 = 141.68248 Wh
Considering Controller efficiency to be 0.9 %Energy at the input of Controller Circuit = 141.68248/0.9 = 157.42499 Wh
Total Ampere Hour Generated by PV Panel = 157.42499/6.0 = 26.237497 Ah
Total Amperes generated by PV Panel = 26.237497/6.0 = 4.372916 A
Considering PV Panels of Wattage 20.0 Watts
with maximum current of 2.5 A
with maximum voltage of 8.0 V
Total number of panels to be connected in series = 6.0/8.0 = 1
Total number of series panels to be connected in parallel = 4.372916/2.5 = 2

Figure 2.12 Snapshot of the pdf file produced by Home Lighting Module

Chapter 3. BizCard Application

BizCard is an Android-based business card recognition application to automate the information [39] recording and retrieval process. The application recognizes Indian-based business cards from a wide variety of fonts [39] and formats. The algorithm for the application consists of three stages: text detection and pre-processing, Optical Character Recognition (OCR), and text segregation phase. The application pre-processes the image captured by a smartphone in the first stage, and then textual regions are extracted. The second stage converts the detected text into the strings [39]. In the final phase, the string obtained from the second step is segregated into different contact entities and verified with the help of databases. The application is available for download at https://play.google.com/store/apps/details?id=com.software.daffodills.Bizcard.

3.1 Need of business card automation and available software review

Business cards are cards bearing business information about a company or individual. Sharing cards during formal introductions is believed to be a convenient and a memory aid. However, it becomes a problem in itself when a pile of received cards grows larger. Storing the cards physically, categorizing them, and retrieving the information when needed is challenging in itself. Digitizing or automating the storage, classification, and retrieving is certainly a way out. A smartphone with a decent camera can play a pivotal role in making the digitization handy [39].

Many Android-based commercial card-scanning applications are already available on the market. However, every one of them has some or other drawback, e.g. lack of data sharing across social media [40], interactive address mapping [41], and most importantly longer processing time (from one second to five seconds) [42]. Some other problems to mention are the inability of the user to comment on contacts saved, failure to store the image of the business card for future reference [40, 41, 42]. ABBYY Card Reader’s free version stores only the name and phone number [43]. It cannot handle text with a dark background and white letters [44]. It further lacks communication niceties like Facebook, Twitter, and LinkedIn. Business Card Reader has a purposeful interface. However, it fails to detect the text correctly from most of the cards [45]. CamCard's OCR engine tripped up on some fancy characters (a stylized "&" became an "8"). If phone and fax numbers are in the same line, it put both into the same field [46]. Presto BizCard's OCR accuracy is fluky (missed part of address import) [47]. ScanBizCards has WebSync service, but it struggles with most cards and exhibits major OCR errors [48]. WorldCard Mobile found to miss certain contact entities in the scans [49]. Yolu Card Reader uses manual human corrections to OCR technology but the recognition time can last up to nine hours especially when the server is busy [50].

A card scanner should be able to detect the text, the symbols, and the numbers from the card omitting any irrelevant information such as logo or graphics. Generally to achieve the above-said functions, the software uses one or more image processing technique. Images captured with a low or medium level smartphone camera exhibits poor quality due to inadequate imaging conditions (e.g. absence of natural light or flash, sensor quality, variability both in geometry and appearance). Captured image thus undergoes a pre-processing stage which removes the background noise, corrects the geometrical distortions, and extracts the textual area from the card. Popular pre-processing techniques are de-Skew [51, 52], despeckle, binarization [53], line and word detection [54], script recognition and segmentation [55]. Most applications use binarization and text detection using clustering approach but fail on images with a complex background and poor lighting [56]. The linear Niblack algorithm was proposed to extract connected components but fails on the character of different sizes [56]. Local Color Quantization (LCQ) is another approach in which connected components are extracted based on color, but it has high processing time since, performed for all 256 colors [57].

Commonly used text detection techniques are text detection using gradient information, text detection using color information, and learn based text detection. Edge and gradient based algorithms use sharp edges considers as text. The algorithm is scale dependent, that is, only text with certain font size is detected [58]. Text detection using color information is suitable for an image with monochrome backgrounds e.g. newspaper advertisements, web pages [59]. Learn based text detection uses neural networks and machine learning algorithm to classify the features of an image, thus requires high processing power to implement [60]. An attempt is made to put forward a free-to-use card-scanner which wold eliminate all such drawbacks, making the process of automation truly simple. The proposed algorithm uses three stages: Text detection using Stroke Width Transform (SWT) [61, 62], OCR using Tesseract engine, text segregation using simple reasoning and database-based verification.

3.2 Methodology of Information Retrieval for BizCard

The method of the proposed application, which has a client-server architecture, uses pre-processing and textual area detection, and text segregation. Text detection and pre-processing include the separation of the textual area from the image with proper contrast and brightness. Text segregation involves classifying the obtained text into contact entities. Figure 3.1 presents the major steps carried for the application schematically.

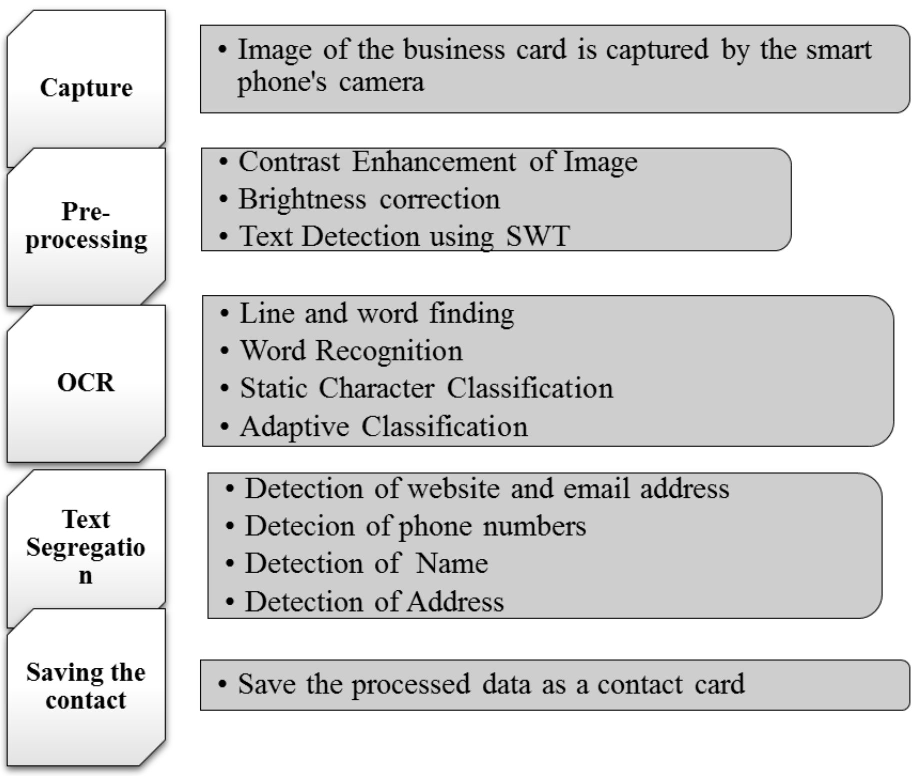


Figure 3.1 Flowchart of the algorithm for pre-processing of the image taken with a smartphone

3.2.1 Architecture of BizCard Application

The architecture of the proposed application, as shown in Figure 3.2, consists of two parts: a mobile interface and a server interface. Both the interfaces communicate with each other in the Http Protocol with the help of REST APIs. The pre-processing part of the proposed application is done using Python program implementing a modified version of SWT on the server After the OCR. The obtained text is then segregated and converted to JSON format and sent through the HTTP protocol to the client.

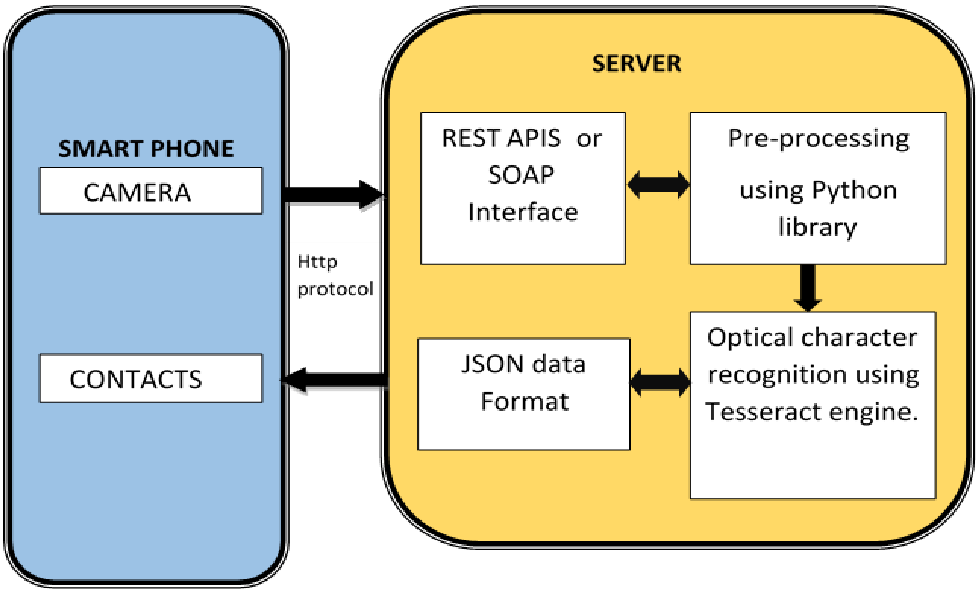


Figure 3.2 A schematic diagram showing client-server interactions of BizCard Application

3.2.2 User Flow Diagram for BizCard Application

The user flow diagram describes the stages of the proposed application from the start to end. Proposed application's user flow diagram also describes the individual screens where interactions take place and the process of moving from one screen to another. A screen offers some possibility and takes the user from one step of the application to the next step. In Figure 3.3 the user flow diagram of the proposed application is described. The user launches the activity, takes a picture of the card and submits the image to the server. The server performs the required operations on the picture and returns the segregated string in JSON format to the client application. The client application then prompts the user to save the data as a contact card or edit the mistakes.

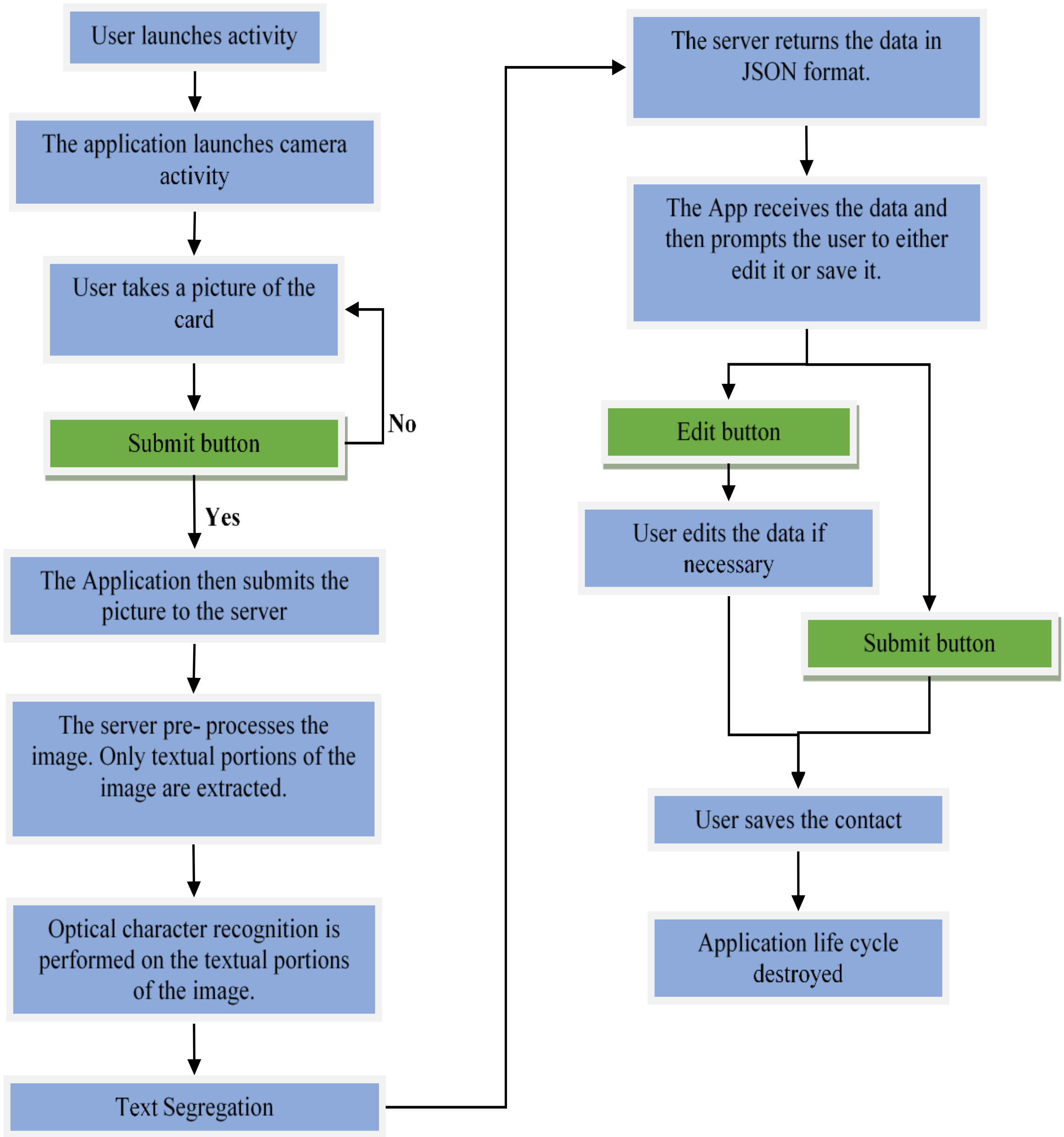


Figure 3.3 User Flow Diagram indicating the start and end of the process life cycle of the BizCard application

3.2.3 Algorithm for Text Detection and Pre-processing

The images taken with a smartphone have a broad range of contrast, brightness, and geometrical distortions. Hence, the image is required to be pre-processed for the extraction of the textual portions of the picture. The steps below describe the pre-processing and text area extraction algorithm.

Step 1 Edge detection using canny edge detection algorithm [62]

Step 2 Calculation of the stroke width of every pixel

Step 3 Storing the stroke width of every pixel in a map and grouping them into letter candidates:

The grouping procedure is done first by grouping the pixels with similar stroke width and then applying the rules mentioned below to distinguish the letter candidates. To allow smooth variation of stroke widths in an alphabet, the ratio of stroke widths should be less than 3 [61].

The rules to distinguish the letter candidates are:

- The variance of the stroke-width within a component must not be too large.
- The aspect ratio of a component must have little variation, to reject large and small components.
- Ignore the components whose size is too large or too small. The application does the above procedure by limiting the length, width, and pixel count of the component.

Step 4 Grouping closely positioned letter candidates into regions of text:

The rules for two letter candidates to be consolidated into a pair are:

- Two letter candidates should have nearly same stroke width. Hence, the width of the median stroke widths should be less than some threshold limit [62].
- The ratio between the heights of the letter characters and the widths of the letters must not exceed 2.5 when capital letters are next to lower case letters [61].
- The distance between letter characters should not be greater than three times the width of the wider one.

- Letters of the same word have nearly same color. Therefore, we compare the average color of the candidates for pairing.
- Discard the regions with less than three letters

3.2.4 Algorithm for Text Segregation

Text Segregation procedure involves the process of classifying the obtained text into phone numbers, landline numbers, address, name, designation, website and email id. To segregate the text for different contact fields. The program uses simple reasoning and verification method (verification of the data from the address and name from the database). Regular expressions are used to match the occurrence of string patterns. However, only matching of string patterns is not sufficient to extract the name and address. Hence, the database is required for the two step verification of name and address.

The text extracted from the images contains all the data i.e. name, address, designation, phone number and various other contact points lumped together. So the data needs to be segregated into different categories. The proposed application uses the following algorithm for text segregation.

Step 1 Detection of web address:

The web address must begin with a "www" having a collection of characters in between and must be ending with a dot and a collection of characters. The pattern for extraction of web address is www.<Any number of characters, dots or any special characters>.<single word>.

Step 2 Detection of email address:

The pattern of the email address can have as many characters at the start but should have a "@" in between and should end with a "." and extension domain name. The pattern for extraction of the email address is <Any number of characters>@<website name / any number of characters>.<single word>.

Step 3 Detection of mobile numbers:

The mobile number contains only numbers consisting of a maximum of 10 digits for India. The patterns for detection of mobile number are:

- +91-<10 digits>
- <10 digits>
- +91-<3 digits>-<3 digits>-<4 digits>
- +91-<5 digits>-<5 digits>
- +91-<4 digits>-<3 digits>-<3 digits>
- +91 <3 digits> <3 digits> <4 digits>
- +91 <5 digits> <5 digits>
- +91 <4 digits> <3 digits> <3 digits>

Step 4 Detection of landline numbers:

It consists of two parts i.e. a STD code and a phone number. The STD code consists of a maximum of 5 digits a continuously. From the STD code the country and place name is detected, hence, a part of the address is verified by the landline number. The patterns for detecting STD code are:

- 0-<3digits>
- +91-<3 digits>
- 0<3 digits>
- +91<3 digits>
- +91<single or multiple space><3 digits>
- 0<single or multiple space><3 digits>

The patterns for detection of landline numbers are:

- <A maximum of 7 digits>
- <3 digits>-<4 digits>
- <4 digits>-<3 digits>
- <4 digits><single/multiple space><3 digits>
- <3 digits><single/multiple space><3 digits>

Step 5 Detection of name:

The name contains two parts: the first name, and surname. The proposed application extracts surname by matching against a database of surnames, and the proposed application categorizes the rest as the first name. The first name can have a maximum of two to three words before or after the family name.

Step 6 Detection of Address:

The zip code is extracted from a continuous word of 6 digits and matched against a database of zip codes to give the particular address string. The state name, district name are matched against a database of states and districts and the other parts of the address are isolated.

Step 7 Detection of job role:

The last part left is the job role and other noisy strings. The job role the strings are matched against a database of common job functions.

3.3 Discussion on the results of BizCard on different business cards

The different steps of the pre-processing phase are outlined in Figure 3.4. The steps are A. Canny Edge Detection, B. Stroke Width Transform, C. Text Line Aggregation, D. Filtering unimportant pixels, E. Finding letter candidates. The images show that maximum noises in the pre-processing stage arise from the geometrical distortions, logos, and other graphics elements. Figure 3.4 A shows the canny edged image. Figure 3.4 B displays the Stroke width of all the characters presents based on a scale of 0 to 10. Figure 3.4 C and D show the images after the aggregation of text line and filtering of unnecessary pixels. Figure 3.4 E shows 'Asst. Professor' is not detected in the final stage as the color of fonts gets blended with the background. The presence of text on logos also leads to the identification of the same in the letter strings.

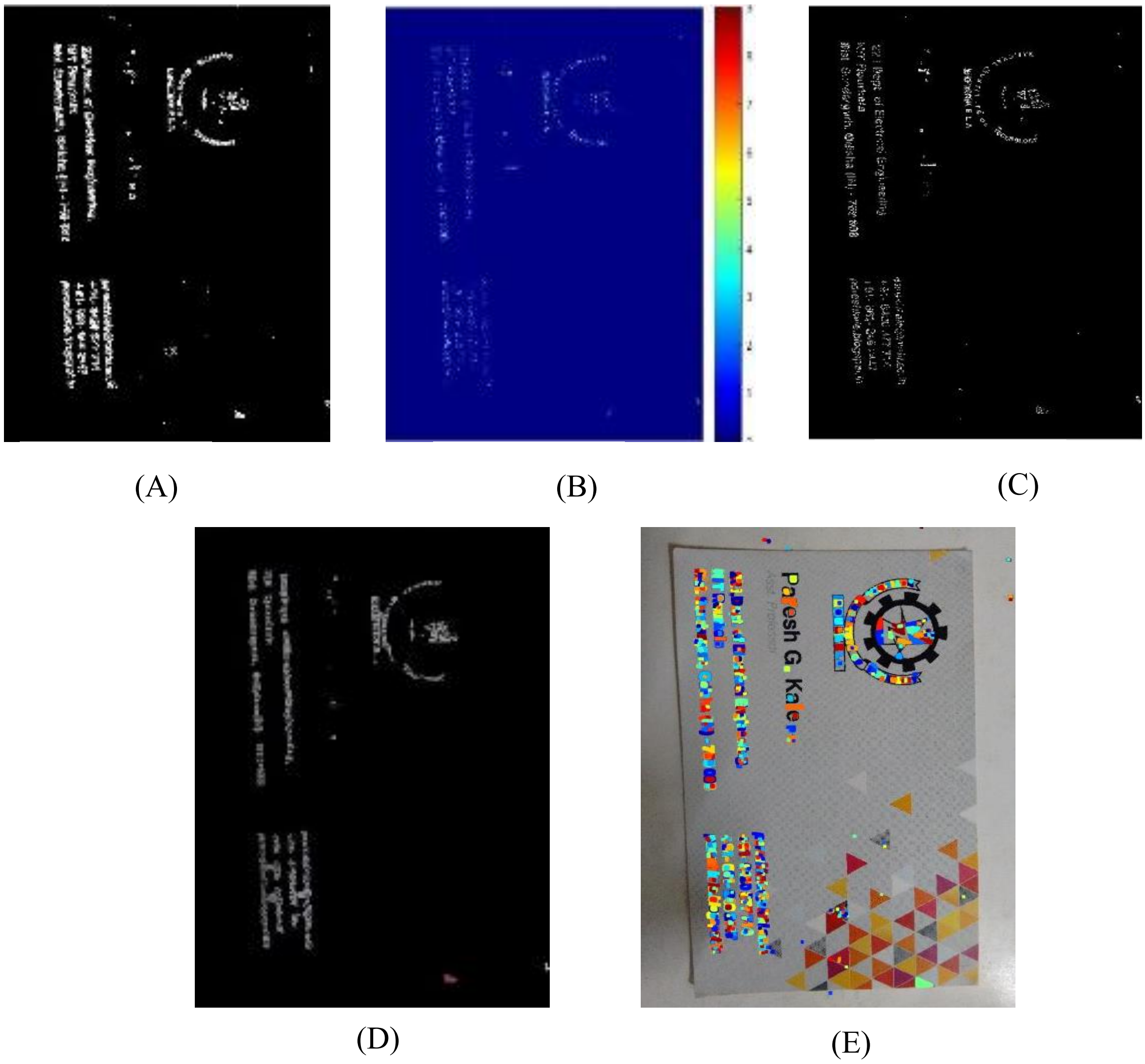
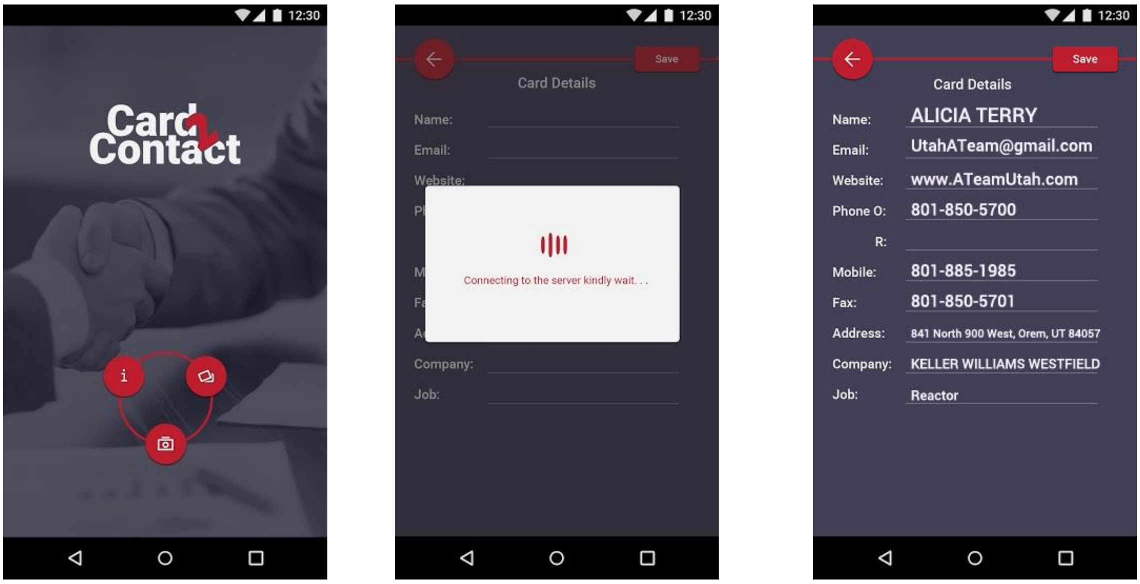


Figure 3.4 Various steps of the text detection and pre-processing phase A. Canny edge detection B. Stroke Width Transform C. Text Line Aggregation D. Filtering unimportant pixels E. Finding letter candidates.

Figure 3.5 shows the user interface of the proposed application. The application consists of two screens one where the proposed application uploads the image or takes a picture of the card, second where the proposed application shows the scanned information to the user. Figure 3.6 displays the result of the draft application on three other business cards. Table 3.1 shows all the steps of the text segregation algorithm.



(a) (b) (c)

Figure 3.5(a) Android Application Interface for uploading Pictures (b) Android Application Interface showing spinner after uploading of Pictures (c) Interface for displaying the details of business card in android applications

Figure 3.6 Image of Specimen cards captured by smartphone camera as an input to first step i.e. image pre-processing

The results show detection of wrong characters due to similarity in appearances, e.g. ‘i’ is detected instead of ‘l’ in email ‘pareshkale.nitrkl.ac.in’ is detected as ‘pareshkaire.nitrki.ac.m’, web address ‘pareshkale.blogspot.in’ as ‘pareshkaire.biogspot.in’, ‘l’ is detected as’t’ in the email address and name in specimen card C. Wrong characters are detected if a certain part of textual areas are missed in text detection phase. The results also show a high level of noise in text detection. For example, in specimen card B ‘Open up’ is detected as ‘OA k Opes’ and it is bundled with address, in specimen card A address has an extra ‘or,' in specimen card C ‘Black Swan’ is detected as ‘Black%.' High noise content may be due to a broad range of brightness, geometrical distortions, detection of letters from the logo and other graphics of the card content High noise content cause false recognition of the characters leading to false classification the information. For example, the presence of noise may result in replacing ‘www’ by ‘w*w’, which makes it difficult to classify the content as the web address; the replacement of a digit of mobile number may lead to a mobile number not being recognized.

Table 3.1 The results of the BizCard application of three different business cards showing card number, step name and extracted text

Specimen Card	#	Step	Extracted Text
	1	Email Extraction	pareshkaire.nitrki.ac.m
	2	Website Extraction	pareshkaire.biogspot.in
	3	Phone Number extraction	Mobile : 9420 377 214 Landline : 661- 246 2447
	4	Zip Code extraction	Zip Code: 769 008
	5	Name extraction	Paresh G. Kale
	6	Address Extraction	or 224, Dept. of Electrical Engineering, NIT Rourkela Dist. Sundargarh, Odisha (IN) – 769 008
B	1	Email Extraction	
	2	Website Extraction	
	3	Phone Number extraction	9040308464
	4	Zip Code extraction	
	5	Name extraction	Abhiram Bank
	6	Address Extraction	New City Communication ; OA k OPes " Sec-19, Near Nilam Hotel
C	1	Email Extraction	kattot.das@orientbtackswan.com
	2	Website Extraction	www.universitiespress.com www.orientbtackswan.com
	3	Phone Number extraction	Landline : 2766 5446/5447/2849/2850/3514 Mobile : 93910 04842
	4	Zip Code extraction	3*6452 500 029
	5	Name extraction	KaLLot Das
	6	Address Extraction	Universities Pass Orient Black% an Assistant Manager Content Research 84 Development (SINN) Orient Blackswrni Prk ate I muted 3*6452 Himayatnagari Hyderabad 500 029

Segregation process of proposed application depends on pattern matching. Failure of identification of patterns and detection of wrong patterns may result in erroneous classification. Another pitfall of the algorithm is the dependence of surname detection on the database. The database used in the proposed application consists of common surnames of Indian origin. Hence, detection of any uncommon surname not found in the database fails. The detection of bold texts from the card may improve the name detection procedure. The address detection also depends on database verification. The algorithm for address matching fails if the database does not contain the address.

3.4 Conclusion and Future work on BizCard

The paper shows the implementation of the business card reader as an android application and displays an accuracy up to 80%. The overall application consists of three phases: text detection and pre-processing, OCR and text segregation. Text detection and pre-processing stage use an improved version of SWT algorithm. The SWT algorithm detects textual regions with high accuracy as compared to other text detection algorithms. Proper lighting conditions may reduce the noise content of the text detection phase. Performing SWT for a smaller value of threshold also reduces the noise content.

The OCR phase uses artificial neural network based Tesseract engine which improves the accuracy of the recognition further. The training of neural network on a broad array of data may improve the accuracy of Tesseract engine. Text segregation based on simple reasoning requires low-end servers. The output at the end of segregation stage has errors. Hence, the user is given the final choice to correct it before saving the contact on the phone. Dictionary search algorithm may remedy the words with noise content. The inclusion of natural language processing algorithms may improve the name detection.

Chapter 4. Auto Attendance Manager Application

Professors used to call the name of students for the purpose of attendance checking. The Professors verify the identity of student's participation by human recognition using facial and voice matching. The above approach is time-consuming because the number of students' is getting increased day by day. Moreover, they may have to recheck any of the students' presence at the end of the period manually. The proposed application offers an automated attendance checking method to take advantage of Wi-Fi 802.11x technology [63]. The instructors generate a Quick Response (QR) code [64] that contains a unique identity. The instructor shows this code on the projector and the students' scan it. The scan is sent to the server using the Wi-Fi routers connected in the class. The server verifies the student id and then stores the scan. In addition to being a stand-alone system, this method proposes a novel concept that is less time consuming as well as requires much less effort from the side of the professor.

4.1 Reason for Automating the Attendance System and available methods of automation

Checking students' attendance in schools, universities, kindergartens, and travel agencies is a time-consuming process because the instructor has to call each person by person when the number of students/users are big. So, instructors/leaders have to consume more time for students to check attendance. This paper presents a smartphone based attendance system using Wi-Fi signals. In this method, we do not have to recognize or tag such a RFID card or the reader. Attendance is automatically checked only if the students have the smartphone. The statistics about absences, lateness, and attendance, is calculated automatically.

In the past, the instructors used to call the name of members (including students, travelers, and children) for the purpose of attendance checking by the instructor (teachers, leaders, and employer). Usually, the professors verify the identity of students by human facial and voice recognition. The instructor macthes the similarity of facial and voice recognition against the present status of the student. The professor may recheck any of the student's presence during the lecture by manually checking the updated attendance list that shows the attendance during or after class. Recently, an automatic presence checking by using a Radio Frequency communication is widely used. However, if the RFID card is faulty or students/users do not get the RF card, this RF based attendance checking cannot work properly [65, 66]. Moreover, it is not enough to cover the entire areas of an individual lecture room for Near Field Communication.The proposed application avoids these issues. This application offers a new participation checking technique by the advantageous and right approach to exploit the Wi-Fi 802.11x innovation on smart mobile devices.This novel method provides the following advantages:

- This process prevents the student from leaving the classroom right after checking a present or from answering the call for checking instead of other students.
- During open air exercises, it can check whether all the kids are present, or somebody is away.
- When gatherings of vacationers move (for purposes, for example, tourism and business trips) some place utilizing a sanction transport, a tour guide can without much of a stretch check whether all the general population are ready to board the vehicle.

We can use the proposed application to take attendance since it will be much less time consuming and the instructor has just to generate a unique token and on connecting to the wifi the token will be verified, and thus, the application records the attendance status.To verify that a user is within a predefined area we can use GPS. If GPS does not work, we can use Indoor localization [67]. In this paper, as the students are inside a lecture hall, the Internet Protocol (IP) address [68] of the router is used to verify that the students are inside the class.

There are many proposals for Automatic Attendance Systems in the literature. Most of these suggestions focus on applications installed on the lecturer device, whether a smartphone or a laptop. This section briefly mentions few of these proposals. Reference [69] proposes software to be installed in the professor's smartphone. It enables to query students' smartphone via Bluetooth connection and, through transfer of students' smartphones' media access control (MAC) addresses [70] to the professor's smartphone, the proposed application confirms the presence of the student. However, if the strength of the class is large, then there can be interference due to a large number of Bluetooth devices sending out the identification signals.

In Reference [71] there is another example of a proposal that uses real-time face detection algorithms integrated into an existing learning management system (LMS). It automatically detects and registers students attending a lecture. The above proposal represents a supplemental tool for professors, combining adaptive algorithms used in machine learning used to track facial changes during a much longer period. On the other hand, in [72], the proposal uses biometric fingerprint verification technique. The above reference proposes a system which checks fingerprint; that automates the complete process of giving attendance. Since biometrics are concerned as the measure of unique human physical features, it has been used to verify the identity of students'. The disadvantage of the system is that it is hard to be able to monitor the presence of the authenticated student throughout a session. Thus, another proposal [73], discusses a system that uses facial recognition technology to track authenticated students. The above proposal implements a neural network based algorithm for face detection and an eigenface algorithm for recognition.

4.2 Proposed Methodology of the Auto Attendance Manager

The framework lies between internet learning and traditional learning as assistance for the participation record-keeping process, in a way that advances the address time so that the Professor uses it in giving valuable materials instead of squandering the time taking participation.

The system requires login process by the class Lecturer through its Android Application or Web Application to generate an encrypted QR code with accurate information. The QR code can be generated any time before the class. During the lecture, or at its beginning, the Professor displays a QR code to the students. The students scan the displayed QR code using the Android Application, provided to them through the smartphone. Along with the scanned information, the current IP address and location information of the mobile are also communicated to the Server Module to confirm attendance. The whole process should take less than a minute for any student as well as for the entire class to complete their attendance confirmation. Smartphones will communicate with the server via the local Wi-Fi coverage offered by the institution since it is needed to verify if the scan is from the lecture hall or any other place.

4.2.1 Architecture of the Auto Attendance Manager Application

The system is composed of three modules: the Server, the Android application Student module, and the Android application Professor Module. Figure 4.1 describes the architecture of the proposed application. The current attendance platform used by the institution may integrate the Server module, or it can be a standalone application depending on the choice of the developer. The following subsections will describe the tasks for each module.

Server Module

The server module facilitates the following tasks:

- Marks the student present or absent according to the information collected from Wi-Fi response.

- Generates QR Code for the Professor
- Checks QR Code ID and Internet Provider's (IP) address so as to prevent proxies
- Provides the report to the students as well as to the Professor.

When a student scans a QR code, he sends the following details via the Android Application to the server i.e. the Student ID, the lecture name, date, time, the attendance status and the IP address. This way, the Server module will register the appropriate presence status. To generate the QR code, the professor logs in the Server Module, to enter the information needed by the system to the QR code. The generated QR code includes an identity code which may be encrypted or empty according to the levels of protection. When the application matches the identification code against the table of QR codes it will give the following information:

- Course ID
- Date, beginning time and ending time of lecture
- Professor ID

The teachers, in turn, can save this QR code as an image and display it on the slides of the lecture. If the teacher policy is to allow late students in his class and would like to mark them as late attendees, then the QR code should also be copied on some of the other slides as the teacher wishes which will indicate the time when the student entered the lecture. When the students enter the class, they pull out their smartphones, open the Android Application, and scan the QR code.

The third function of the Server Module is to run an identity check against the registered students. The application accomplishes the above task by comparing the QR code ID against a registered QR code id. The IP address of the student is also verified to check that the attendance is from the particular part of the academic area. Students uploading QR code from the internet will be marked absent. A matching score will be added to the attendance sheet so the professor could perform a manual job either during the lecture or after the lecture. The application does the identity check once the attendance registration transaction is received. The purpose of the above task is to allow the professor to check the results of the identity check before the end of the lecture if he/she wishes to do so.

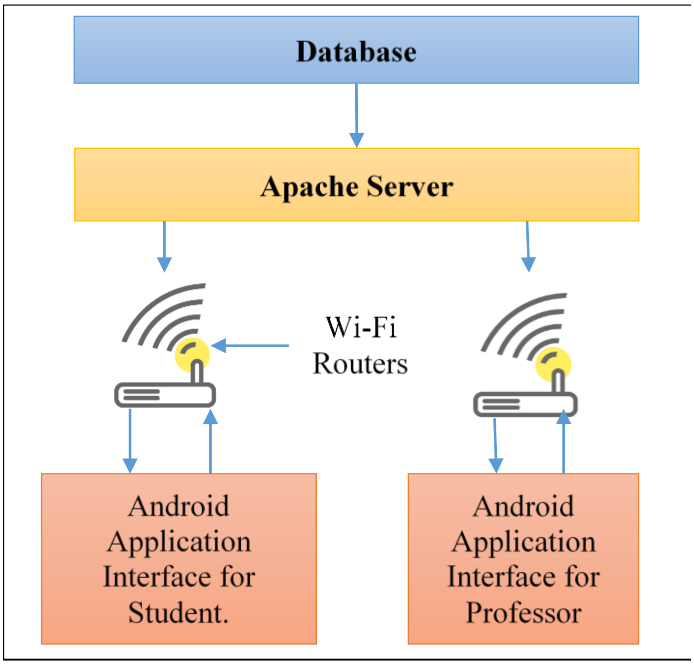


Figure 4.1 Overall System Architecture of Auto Attendance Manger showing its division into Apache Server, Android Application Interface for Student and Android Application Interface for Professor

Android Application Module for Students

The Students' Module is the part that students usually install on their smartphones. This module is a standalone application that communicates with the Server Module. It provides enrolled subject details of the student and also the number of absents and presents in the subject. As mentioned earlier, the communication will be through the institute local Wi-Fi network.

When the student sees the QR code on the screen, he/she opens the Android application. If it is the first time after installing the app, the system requests the student to enter a roll number and password. Once logged in, the application allows the student to click on the scan QR code button. Once the student scans the code, the system sends the response code to the server and resumes working in the background. With that, the process is considered completed. The server, in turn, will send back a success or error message which will be displayed to the user.

Android Application Module for Professors

The Professors' Module is the part that the professors usually install on their smartphones. This module is the standalone application that communicates with the Server Module. It provides subject details and also the particulars of the students' enrolled. On clicking on the particular topic, he can generate a QR code and save it for later use. The professor module also generates the detailed reports as necessary. The professor can either login via the web application or Android Application.

4.2.2 User Flow Diagram for the Auto Attendance Manager Application

The user flow diagram describes the stages of the proposed application from the start to end. Proposed application's user flow diagram also describes the individual screens where interactions take place and the process of moving from one screen to another. A screen offers some possibility and takes the user from one step of the application to the next step. In Figure 4.2 the user flow diagram of the proposed application is described for both the modules i.e. Professor Module and Student Module. The professor launches the Android application, generates a QR code and then puts it on the slide for the students' to scan. The professor can also view and generate the attendance report of the students. The students' scan the QR code and the application records their presence.

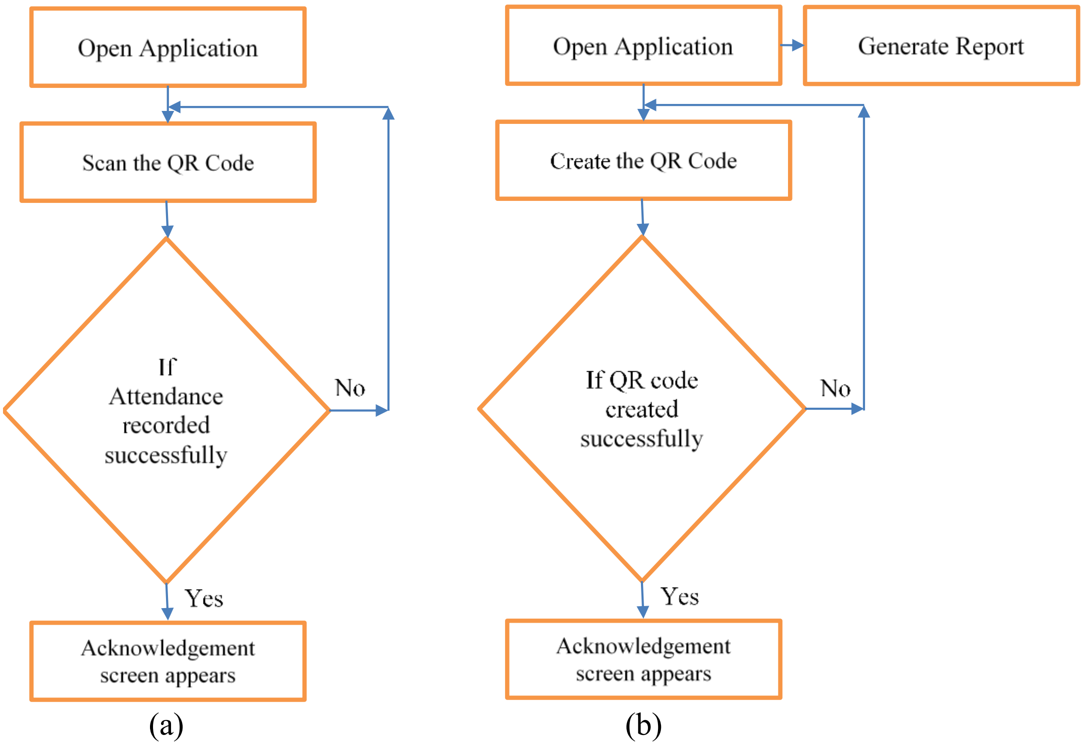


Figure 4.2 (a) User Flow Diagram for Student Module of Auto Attendance Manager (b) User Flow Diagram for Professor's Module of Auto Attendance Manager

4.2.3 Database Diagram for Auto Attendance Manager Application

The database diagram describes the relationship between the tables of the database created for the proposed application. The Figure 4.3 shows the database diagram of the proposed application. The Attendance Recording table has the ID as the primary key and the user ID, QR Code ID, and Subject ID as the foreign keys respectively which link to the Users table, QR Code Details table and the Subjects table respectively. The Groups table contains the listing of groups which consists of Professors and Students as two groups. The Users table stores the details of Users and is linked to the Group Details table and other tables through Group ID and User ID respectively. The QR Code Details table stores the details of QR Code being generated and the Users Table links to the same and Subjects Table through the respective foreign keys. The subject allocation table stores the students allocated to a particular course by the User ID from the Users table and the Subject ID from the Subjects table. The Timetable stores the timetable of a particular professor and is linked to the Users table and Subjects table.

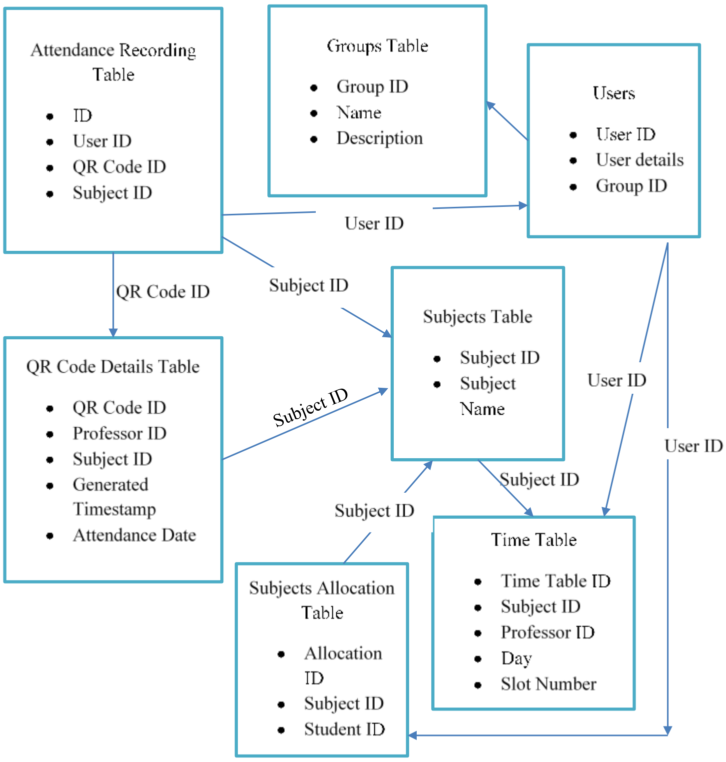


Figure 4.3 Database diagram indicating the relationship between different tables of the Auto Attendance Manager

4.2.4 Integration of Indoor Localization Library with Auto Attendance Manager

The integration of indoor localization with Auto Attendance Manager facilitates the second checkpoint to make sure that the students are inside the class. The proposed application locates an individual with the help of Wi-Fi strength fingerprinting. Auto Attendance Manager implements Wi-Fi positioning with the support of Redpin open source indoor positioning system [74]. Redpin is used since it provides room level accuracy and avoids the time consuming and training phase known from other indoor positioning systems.

The indoor positioning system consists of logger module and a server module. The logger module logs the fingerprints of Wi-Fi access points, associating them with a particular room. Figure 4.4 shows the schematic of the indoor localization process. During the logging process, the logger module saves the fingerprints in the server module. As the Redpin library associates the fingerprint with a particular room, hence during the locating process, the problem gets reduced to the problem of finding a single fingerprint that matches the best measurement. Hence to determine the current location of a mobile device, the library tries to find one known fingerprint that matches the current measurement best.

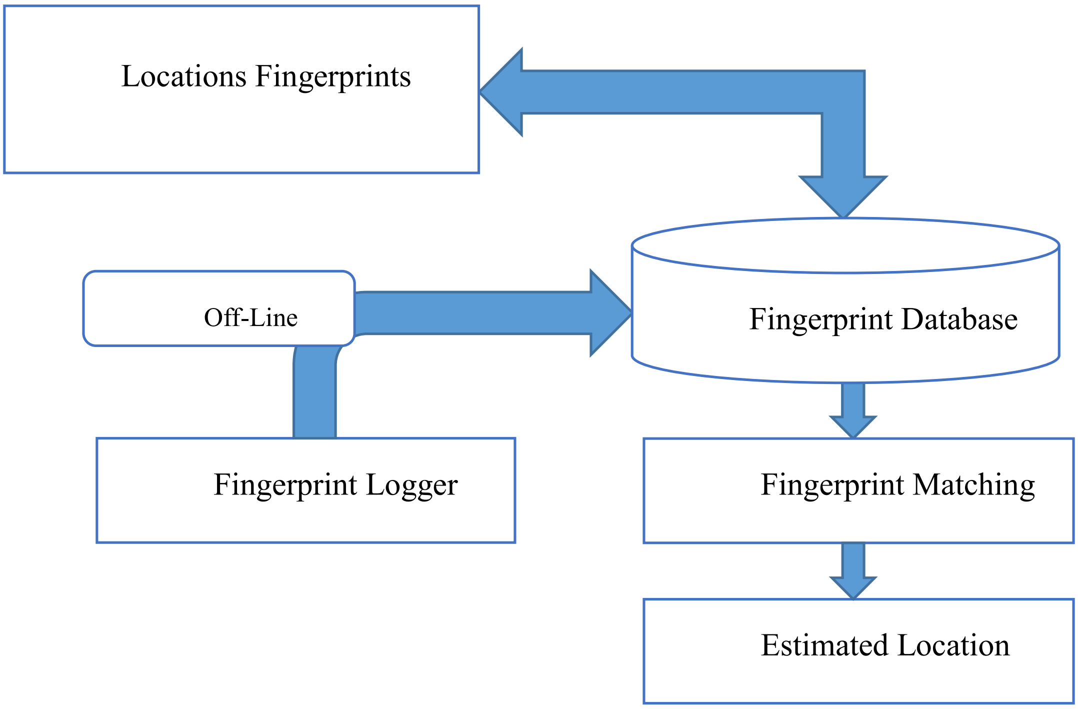
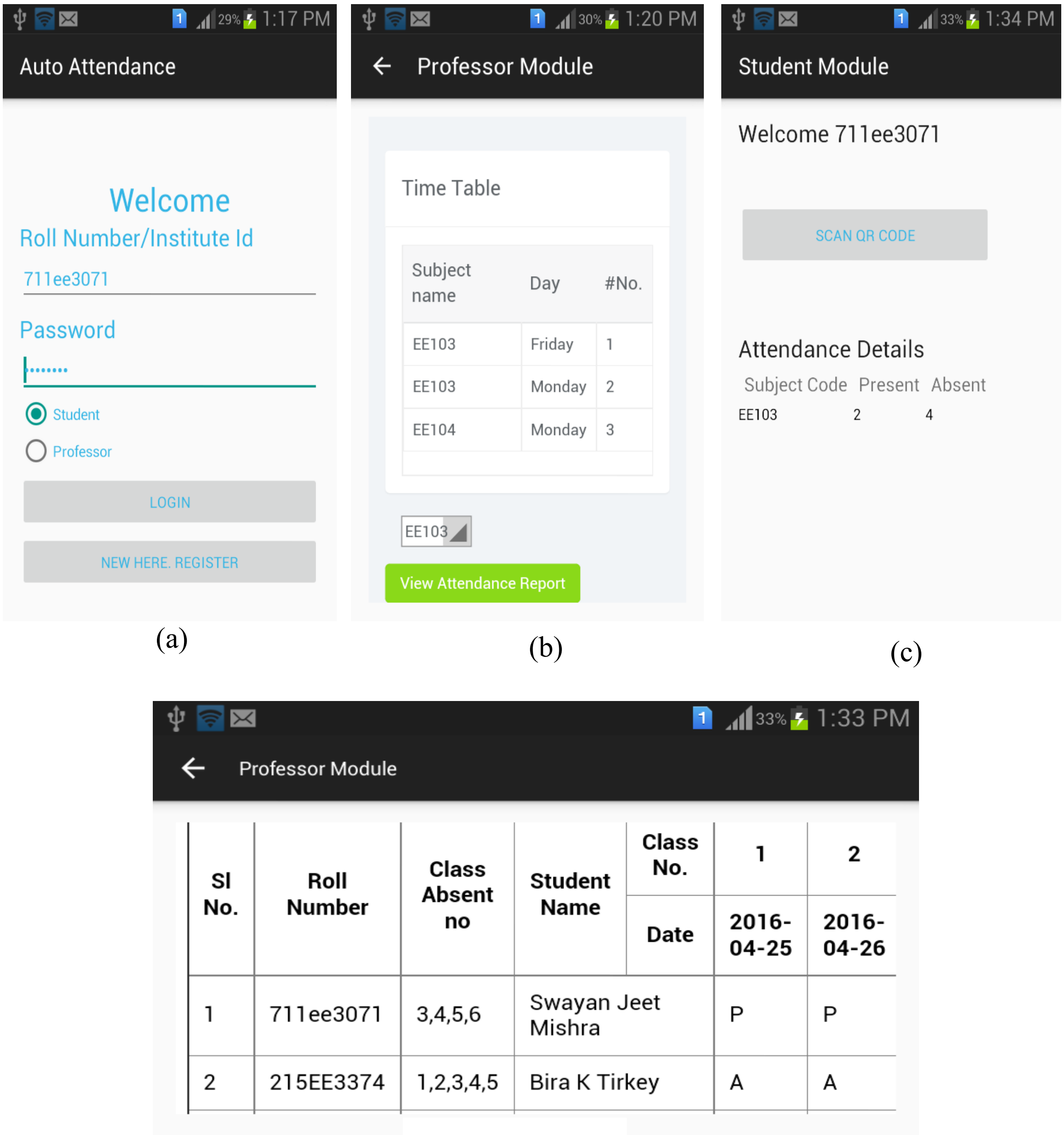


Figure 4.4 Schematic Diagram of Indoor Localization Process

4.3 Results and Discussion on the Auto Attendance Manager Application

Figure 4.5 shows the User Interface screens of the Auto Attendance Manager. The screen (A) presents the login screen of the application. The screen (B) shows the professor module interface. The timetable is clearly visible and below the time table the professors can click on the buttons provided to view the details of the subject's attendance. Double clicking the particular row of the timetable generates a QR code for the given day and subject. The next screen (C) shows the student section of the application. The following table the button shows the attendance details of the students. The scan QR code button scans the QR code and upon successful scanning shows an acknowledgment message to the student. The screen (D) shows the registration screen of the Auto Attendance Manager.



Sl No.	Roll Number	Class Absent no	Student Name	Class No. / Date	1 / 2016-04-25	2 / 2016-04-26
1	711ee3071	3,4,5,6	Swayan Jeet Mishra		P	P
2	215EE3374	1,2,3,4,5	Bira K Tirkey		A	A

Figure 4.5 Different screens of the Auto Attendance Manager where (a) shows the login screen (b) shows the Professor Module (c) shows the Student Module (d) shows the Report generated by the Professor Module

The Auto Attendance Manager application was thoroughly tested for a class. Firstly, some students had problems detecting QR codes as the students sitting in class have a wide variety of smartphones, which do not have their camera settings adjusted. The camera settings mostly involved setting the white balance to proper value and enabling the auto focus feature if available. Setting the individual settings to proper value is not possible, hence, in the next version a third party application will be used to scan QR codes since it may have professional settings for Camera. Secondly, environmental lighting and projector brightness also played a key role in the detection of QR code. A proper combination of the above mentioned two features is critical for QR code detection. Thirdly, the structure of the room also affected the detection of QR code. If the room was poorly structured such that fan and projector came in the way of QR code detection, then the visuality of camera is impaired. Fourthly, the quality of projector also played an important part.

4.4 Conclusion and Improvements on Auto Attendance Manager

The chapter shows the implementation of an automatic attendance management system. The overall application consists of two modules: Professor Module and Student Module. The Professor generates a QR code from his module and displays it to the students for a degree of time. The students scan the QR code using the Student module, and the application records their attendance. Students can also view their attendance details from the Student Module. For the purpose of QR code scanning, a third party application is used which provides a coherent camera setting QR code which facilitates easily scanning of QR code.

The QR code scanning depends on various factors, important of which are environmental lighting, projector quality, and room structure. Moreover, there is a difficulty in identifying which students are inside the room and who are outside. Indoor location verification using Wi-Fi fingerprinting may improve the accuracy of the application.

Chapter 5. Conclusion and Future Work

The conclusion and future work section describe the summary of the work done. The section is contains two parts, first is the conclusion, and the second is future work. The conclusion section illustrates the summary of methodologies of the applications developed. The future work sections describe the problems encountered and the possible solutions, the developers may implement.

5.1 Conclusion

Android provides an easy platform for the developers to take advantage of all the different kinds of sensors interfaced with the smartphone, but as the solutions become more and more complex, it is difficult to make full use of Android's functionality. There are many pre-implemented solutions (especially in 2.x and now 3.x versions) like Account Sync mechanism, LRU cache, Alarms, Location and Sensor Managers that developers need to understand extensively to use them wisely (without a negative effect on the system). Moreover, for developing an application that requires functionalities of the camera, the developer needs to maintain same camera settings across all smartphone devices which are difficult. Also, different smartphones have different varieties of Wi-Fi and Bluetooth sensors. So, maintaining a uniform scale of measurement for the strengths of measuring the Wi-Fi and Bluetooth signal is quite difficult. The implementation of client based applications using RESTful API and services for Android is easy to implement and all Internet of Things based applications are based on RESTful API and services for Android.

5.2 Future Work

The future work section outlines the features of the proposed applications that are new in the upcoming versions. The update in the applications will also remove the bugs in the current versions of the application. The next version of PVSys application implements a uniform data model throughout the application. The implementation of the data model allows the users to share their data across devices and also on social networking platforms. PVSys includes a third module named as Solar Lantern Module in the next version. The module is useful for people who want to build a solar lantern for their use. The next release of the PVSys application also implements costing details current version.

BizCard application has improved detection in the next release. The application directly detects name from the bold texts from the card. Furthermore, different artificial neural network and natural language processing tools and algorithms are used to improve the segregation process in the next version of the application. Auto Attendance Manager can have an additional checking method for students. The method involves indoor location check by using Wi-Fi fingerprinting technique. The Auto Attendance Manager Application implements improved and responsive interface in the next version.

.

Bibliography

[1] P. Bedyński, “Andood – an Android application,” University of Warsaw, 2011.

[2] Wikipedia, “Android (Operating System),” Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Android (operating system). [Accessed 20 March 2016].

[3] Wikipedia, “Android History,” Wikipedia, [Online]. Available: http://android.wikia.com/wiki/Android_History. [Accessed 5 December 2015].

[4] Wikipedia, “Android TV - Wikipedia,” Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Android_TV. [Accessed 30 January 2016].

[5] Google, “Android TV,” Android, [Online]. Available: https://www.android.com/intl/en_in/tv/. [Accessed 30 January 2016].

[6] Wikipedia, “Android Auto - Wikipedia,” Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Android_Auto. [Accessed 30 January 2016].

[7] Google, “Android Auto,” Android, [Online]. Available: https://www.android.com/auto/. [Accessed 30 January 2016].

[8] Wikipedia, “Android Wear - Wikipedia,” Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Android_Wear. [Accessed 30 January 2016].

[9] Google, “Android Wear,” Android, [Online]. Available: https://www.android.com/intl/en_in/wear/. [Accessed 30 January 2016].

[10] Engadget, “HP hedges its bets, unveils a 14-inch laptop running Android,” Engadget, [Online]. Available: http://www.engadget.com/2014/06/01/hp-slatebook/. [Accessed 30 January 2016].

[11] Wikipedia, “OUYA,” Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Ouya. [Accessed 30 January 2016].

[12] Tech PP, “5 Best Android-Powered Cameras to Blend Processing Power with Quality Lenses,” Tech Personalized, [Online]. Available: http://techpp.com/2014/05/16/best-android-cameras/. [Accessed 30 January 2016].

[13] The Android Mania, “What is Android ?,” The Android Mania, [Online]. Available: http://www.theandroid-mania.com/what-is-android/. [Accessed 8 September 2015].

[14] Palm Source, “Palm SO | Palm Source,” Palm Source, [Online]. Available: https://www.palmsource.com/palmos/. [Accessed 30 January 2016].

[15] Pocket PC Central, “Windows Mobile Pocket PC Information Center,” Pocket PC Central,

[Online]. Available: http://pocketpccentral.net/device_menus/pocket_pcs.htm. [Accessed 30 January 2016].

[16] Wikipedia, “Blackberry OS | Wikipedia,” Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/BlackBerry_OS. [Accessed 30 January 2016].

[17] Android Training Institute in Pune, “Android Training,” Android Training Institute in Pune, Pune, 2008.

[18] Open Handset Alliance, “Open Handset Alliance,” Open Handset Alliance, [Online]. Available: http://www.openhandsetalliance.com/. [Accessed 31 January 2016].

[19] masterinandroid, “Home Page,” masterinandroid, [Online]. Available: http://masterinandroid.com/. [Accessed 20 March 2016].

[20] Begin with Android, “Beginner's Guide to Android,” Begin with Android, [Online]. Available: http://www.beginandroid.com/intro.shtml. [Accessed 5 February 2016].

[21] Slideshare, “Snapshot of Digital India,” Slideshare, [Online]. Available: http://www.slideshare.net/Ethinos/snapshot-of-digital-india-september-2014. [Accessed 25 Novmber 2015].

[22] Reuters, “Google topples Nokia from smartphones top spot,” Reuters, 31 January 2011. [Online]. Available: http://www.reuters.com/article/us-google-nokia-idUSTRE70U1VW20110131. [Accessed 5 February 2016].

[23] Gartner Analysis, “Gartner Says Emerging Markets Drove Worldwide Smartphone Sales to 15.5 Percent Growth in Third Quarter of 2015,” Gartner, 18 November 2015. [Online]. Available: http://www.gartner.com/newsroom/id/3169417. [Accessed 9 February 2016].

[24] Times of India, “Android's market share in India over 90%: IDC,” TOI, [Online]. Available: http://timesofindia.indiatimes.com/tech/tech-news/Androids-market-share-in-India-over-90-IDC/articleshow/22793647.cms. [Accessed 15 February 2016].

[25] IDC, “Smartphone OS Market Share, 2015 Q2,” IDC, [Online]. Available: http://www.idc.com/prodserv/smartphone-os-market-share.jsp. [Accessed 15 February 2016].

[26] Android Developer, “Dashboards | Android Developer,” [Online]. Available: https://developer.android.com/about/dashboards/index.html.

[27] Google, “Google Groups,” [Online]. Available: https://groups.google.com/forum/#!forum/android-developers. [Accessed 16 December 2015].

[28] Android Developers Blog, “Home | Android Developers Blog,” Google, [Online]. Available: http://android-developers.blogspot.in/. [Accessed 5 August 2015].

[29] Stack Overflow, “Stack Overflow,” [Online]. Available: http://stackoverflow.com/.

[30] Android Developers, “App Manifest,” Android, [Online]. Available: https://developer.android.com/guide/topics/manifest/manifest-intro.html. [Accessed 29 July 2015].

[31] Android , “Activities | Android Developers,” Android, [Online]. Available: https://developer.android.com/guide/components/activities.html. [Accessed 8 August 2015].

[32] Android, “Services | Android Developers,” Android, [Online]. Available: https://developer.android.com/guide/components/services.html. [Accessed 5 August 2015].

[33] Android, “Content Providers | Android Developers,” [Online]. Available: https://developer.android.com/guide/topics/providers/content-providers.html. [Accessed 8 August 2015].

[34] Android, “BroadcastReceiver | Android Developers,” Android, [Online]. Available: https://developer.android.com/reference/android/content/BroadcastReceiver.html. [Accessed 10 August 2015].

[35] Android, “Intents and Intent Filters,” Android, [Online]. Available: https://developer.android.com/guide/components/intents-filters.html. [Accessed 10 August 2015].

[36] Android Developers, “Processes and Applications,” [Online]. Available: https://developer.android.com/guide/topics/processes/process-lifecycle.html.

[37] Design Icon, “Icons,” [Online]. Available: http://www.design_icons_1.com.

[38] C. S. Solanki, “Solar Photovoltaics,” in Solar Photovoltaics: Fundamentals Technologies And Applications, Bombay, Prentice Hall, 2011.

[39] S. J. Mishra and K. P, “BizCard - An Android Application to Automate the task of Information Retrieval from Business Cards,” ESTIJ, vol. II, no. 6, 2016.

[40] M. Perenson J., “7 business card apps for smartphones: Scan 'em and store 'em,” 3 December 2013. [Online]. Available: http://www.computerworld.com/article/2486045/mobile-apps/mobile-apps-7-business-card-apps-for-smartphones-scan-em-and-store-em.html. [Accessed 28 6 2015].

[41] Andoroid Headlines, “Top 10 Business card scanner apps for android,” Android Headlines, 2 January 2015. [Online]. Available: http://www.androidheadlines.com/2015/01/featured-top-10-business-card-scanner-apps-android.html. [Accessed May 2015].

[42] J. Wallen, “Five apps to tame your business card chaos,” 8 October 2014. [Online]. Available: http://www.techrepublic.com/blog/five-apps/five-apps-to-tame-your-business-card-chaos/. [Accessed 5 8 2015].

[43] ABBY, “Business Card Reader,” [Online]. Available: http://www.abbyy.com/business-card-reader-for-windows/. [Accessed 22 May 2015].

[44] ABBY, “Full Contact Business Card Reader,” FullContact InC., [Online]. Available:

https://www.fullcontact.com/cardreader/. [Accessed 20 May 2015].

[45] Business Cards Info, “BCI Home,” Business Card Info, [Online]. Available: http://www.businesscardsinfo.net/. [Accessed 21 May 2015].

[46] CAMCARD, “CAMCARD | Professional Business Card Reader,” INTSIG INFORMATION CORPORATION, [Online]. Available: https://www.camcard.com/. [Accessed 25 May 2015].

[47] Visioneer, “New Software Presto! BizCard Reader,” Visioneer, Inc., [Online]. Available: http://support.visioneer.com/products/Software/CardReader/BizCard.asp. [Accessed August 2015].

[48] CircleBack, “ScanBizCards App,” CircleBack, Inc., [Online]. Available: https://www.circleback.com/scanbizcards. [Accessed 28 August 2015].

[49] PenPower Inc, “WorldCard Mobile iPhone App,” PenPower TECHNOLOGY LTD., [Online]. Available: http://www.penpowerinc.com/product.asp?sn=392. [Accessed 29 August 2015].

[50] App Annie, “Yolu Card Reader,” App Annie, [Online]. Available: https://www.appannie.com/apps/ios/app/yolu-card-reader-business. [Accessed 31 August 2015].

[51] S. Ray, “A Simple and Efficient Skew Detection Algorithm via Text Row Accumulation,” Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on, vol. 2, pp. 1145-1148, 1995.

[52] C. L. Yu, Y. Y. Tang and C. Y. Suen, “Document Skew Detection based on Fractal and Least Squares Method,” Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on, vol. 2, pp. 1149-1152, 1995.

[53] C. T. Have, “Content recognition of business cards,” University of Copenhagen, Copenhagen, 2007.

[54] H. Saiga, Y. Nakamura, Y. Kitamura and T. Morita, “An OCR System for Business Cards,” in Document Analysis and Recognition, 1993., Proceedings of the Second International Conference on, Yamato Koriyama, 1993.

[55] W. Bieniecki, S. Grabowski and W. Rozenberg, “Image Preprocessing for Improving OCR Accuracy,” 2007 International Conference on Perspective Technologies and Methods in MEMS Design, pp. 75-80, 2007.

[56] L. R. Desai, P. B. Kadam and S. Shinde, “Review on Text Detection Methodology from Images,” International Journal of Advanced Research in Computer and Communication Engineering, vol. III, no. 2, 2014.

[57] J. Gllavata, R. Ewerth and B. Freisleben, “A Robust Algorithm for Text Detection in Images,” Image and Signal Processing and Analysis, 2003. ISPA 2003. Proceedings of the

3rd International Symposium on, vol. 2, pp. 611-616, 2003.

[58] Q. Ye, W. Gao and W. Wang, “A Robust Text Detection Algorithm in Images and Video Frames,” in ICICS-PCM, Singapore, 2013.

[59] J. Gao and J. Yang, “An Adaptive Algorithm for Text Detection from Natural Scenes,” Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on (Volume:2), vol. 2, 2001.

[60] A. Coates, B. Carpenter, C. Case, S. Satheesh, S. Bipin, T. Wang, D. J. Wu and A. Y. Ng, “Text Detection and Character Recognition in Scene Images,” 2011 International Conference on Document Analysis and Recognition, pp. 440-445, 2011.

[61] S. Kumar and A. Perault, “Text Detection on Nokia N900 using Stroke Width Transform,” Cornell University, 2010.

[62] B. Epshtein, E. Ofek and Y. Wexler, “Detecting Text in Natural Scenes with stroke width transformation,” Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, 2010.

[63] Wikipedia, “Wi-Fi,” Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Wi-Fi. [Accessed 27 January 2016].

[64] Wikipedia, “QR Code,” Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/QR_code. [Accessed 16 January 2016].

[65] M. Srinidhi and R. Roy, “A web enabled secured system for attendance monitoring and real time location tracking using Biometric and Radio Frequency Identification (RFID) technology,” in Coimbatore, Kharagpur, 2015.

[66] T. S. Lim, S. C. Sim and M. M. Mansor, “RFID based attendance system,” Industrial Electronics & Applications, 2009. ISIEA 2009. IEEE Symposium on , vol. 2, pp. 778-782, 2009.

[67] Wikipedia, “Indoor Positioning System,” Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Indoor_positioning_system. [Accessed 16 February 2016].

[68] Wikipedia, “IP Address,” Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/IP_address. [Accessed 16 February 2016].

[69] Jamil, Tariq;, “Automatic attendance recording system using mobile telephone,” Telecommunications Forum (TELFOR), 2011 19th, Belgrade, 2011.

[70] Wikipedia, “MAC Address,” Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/MAC_address. [Accessed 17 February 2016].

[71] V. Shehu and A. Dika, “Using real time computer vision algorithms in automatic attendance management systems,” Information Technology Interfaces (ITI), 2010 32nd International Conference on, pp. 397-402, 2010.

[72] C. Saraswat and A. Kumar, “An Efficient Automatic Attendance System using Fingerprint Verification Technique,” International Journal on Computer Science and Engineering, vol. 2, no. 2, pp. 264-269, 2010.

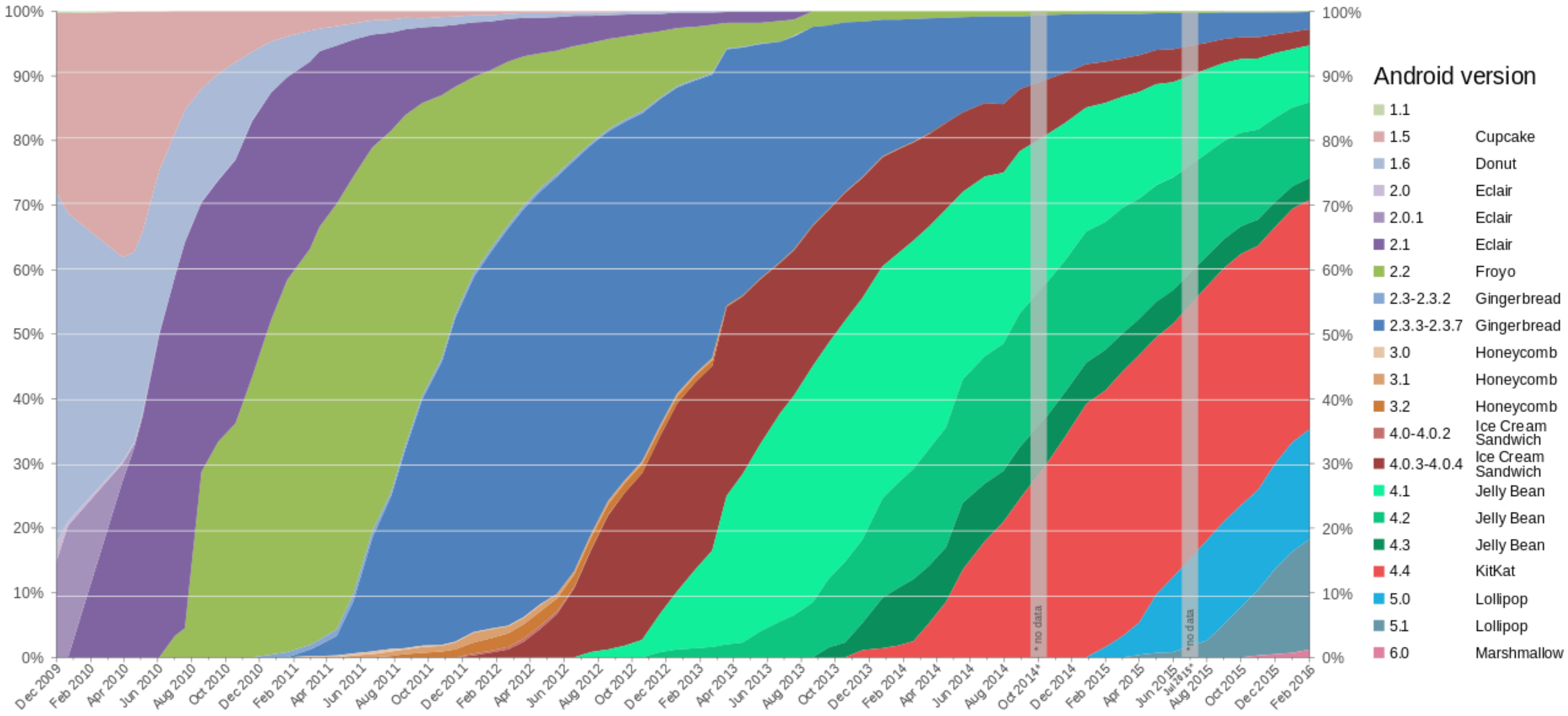
[73] Q. Xiao and X. D. Yang, “A facial presence monitoring system for information security,” Computational Intelligence in Biometrics: Theory, Algorithms, and Applications, 2009. CIB 2009. IEEE Workshop on, pp. 69-76, 2009.

[74] P. Bolliger, “Redpin - Adaptive, Zero-Configuration Indoor Localization through User Collaboration,” ETH, Zurich, 2008.

[75] Google, “Tesseract Engine,” [Online]. Available: https://code.google.com/p/tesseract-ocr/. [Accessed 20 June 2015].

Appendix

Appendix - I



A graph showing the upgrade speed of different versions of Android by percentage of users [3]

Dissemination

Journal Articles

1. "BizCard–An Android Application to Automate the Process of Information Retrieval from Business Cards", IRACST Vol.6, No.2, Mar-Apr 2016

Vitae

SWAYAN JEET MISHRA
ELECTRICAL ENGINEER

CALL ME
+91 9853034348

PERSONAL

Name	Swayan Jeet Mishra
Birthday	29th November, 1993
Relationship	Single
Nationality	Indian
Born	Rourkela, Odisha.
Langugaes	Odiya, Hindi, English.

CONTACT

Mobile	+91 9853034348
Home	+91661 2481555
Email	swayanjeetmshr657@gmail.com
Skype	swayanjeetmshr657
Address	PLOT-E/143, Phase - I, Chhend Colony, Rourkela, Odisha.

TIMELINE

2016
Electrical Engineering
NIT Rourkela.
CGPA 7.92

2011
12th
Deepika English Medium School,
Rourkela.
90.2 % | CBSE Board

2009
Matriculation
Chinmaya Vidyalaya (E.M),
Rourkela.
96.4 % | ICSE Board

ABOUT ME

I am sincere, hard-working, efficient, and reliable. I am dynamic and resourceful too. I am an innovative individual, who loves to bring novelty into work. I am ambitious and goal oriented. If I get the desired platform, I can prove myself an asset to the firm.

EDUCATION

Degree	Bachelor's Degree in Electrical Engineering

HOBBIES

- Reading Short Story Books
- Swimming
- Playing Cards

REFERENCES

Choudhury Niroj Nalini Parhi
Teacher, Deepika English Medium School,
Rourkela.
+91 9937115311
niroj13.parhi@gmail.com

Abhijeet Dash
Student, CV Raman College of Engg.
+91 7205219181
abhijeetdash001@gmail.com

PROJECTS

- Made the ground station receiver of the balloon satellite.
- Made a business card scanner based on Image Processing
- Made an automated attendance system based on QR code scanning and networking feature.

SUBJECTS OF INTEREST

- Transformers
- Power Electronics
- Control Systems

ONLINE PROFILE

Twitter
www.twitter.com/Swayanjeet

Facebook
www.facebook.com/swayanjeet

Google+
www.google.com/+SwayanJeetMshr

Linkedin
www.in.linkedin.com/swayanjeet

Index