

**THE REPUBLIC OF TURKEY
BAHCESEHIR UNIVERSITY**

**DEVELOPING COMMUNICATION PROCEDURE
FOR MOBILE APPLICATION**

Master's Thesis

JAWAD RASHEED

ISTANBUL, 2015

**THE REPUBLIC OF TURKEY
BAHCESEHIR UNIVERSITY**

**GRADUATE SCHOOL OF NATURAL & APPLIED SCIENCES
ELECTRICAL & ELECTRONICS ENGINEERING**

**DEVELOPING COMMUNICATION
PROCEDURE FOR MOBILE APPLICATION**

Master's Thesis

JAWAD RASHEED

Supervisor: ASSISTANT PROF. DR. SELÇUK BAKTIR

Co-Supervisor: ASSOCIATE PROF. DR. GÖKMEN ALTAY

İSTANBUL, 2015

**THE REPUBLIC OF TURKEY
BAHCESEHIR UNIVERSITY**

**GRADUATE SCHOOL OF NATURAL & APPLIED SCIENCES
ELECTRICAL & ELECTRONICS ENGINEERING**

Name of the thesis: Developing Communication Procedure for Mobile Application

Name/Last of the Student: Jawad Rasheed

Date of the Defense of Thesis: April 14, 2016

The thesis has been approved by the Graduate School of Natural & Applied Sciences.

Assoc. Prof. Dr. Nafiz ARICA
Graduate School Director
Signature

I certify that this thesis meets all the requirements as a thesis for the degree of Master of Arts.

Assist. Prof. Dr. Ayça YALÇIN ÖZKUMUR
Program Coordinator
Signature

This is to certify that we have read this thesis and we find it fully adequate in scope, quality and content, as a thesis for the degree of Master of Arts.

Examining Committee Members

Signature

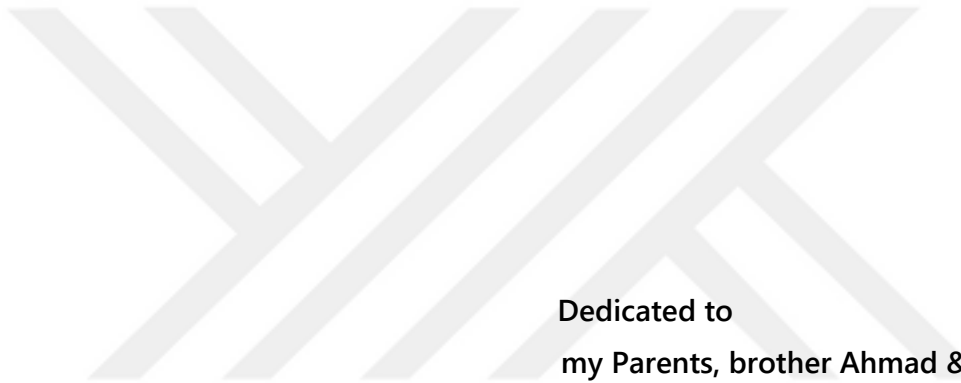
Thesis Supervisor
Assist. Prof. Dr. Selçuk BAKTIR

Thesis Co-supervisor
Assoc. Prof. Dr. Gökmen ALTAY

Member
Assist. Prof. Dr. Cemal OKAN ŞAKAR

Member
Assoc. Prof. Dr. Fethullah KARABIBAR

Member
Assist. Prof. Dr. Muhammad FAHIM



**Dedicated to
my Parents, brother Ahmad & Wife Jia**

ABSTRACT

DEVELOPING COMMUNICATION PROCEDURE FOR MOBILE APPLICATION

Jawad Rasheed

Master's Degree

Tez Danışman: Yrd. Doç. Dr. Selçuk Baktır

Tez Eş-danışman: Doç. Dr. Gökmen Altay

Nisan 2016, 79 Sayfalar

Bu çalışma, mobil uygulama için iletişim prosedürü önerir ve onun uygulama. Mobil kullanım, son on yılda önemli ölçüde büyüdü. Mobil uygulamalar (Genel olarak "uygulamalar"), hızlı bir kabul edilir Bilgi Sistemleri sektöründe büyüyen eğilimler.

Bilgisayarlar ve Mobil cihazlar ortaya çıkınca, Hayatta İnsana her alanda büyük destek genişleterek katılım sürecini otomatikleştirmek için yardımcı oldu. Bilim adamları ve geliştiriciler farklı telefonlar ve kağıt işi, bir firma, okullar ve üniversiteler, seyirci seminerler, toplantılar, uyeleri, için sadece öğrencilerin Çalışanların katılımını alan ihtiyacı ortadan kaldırmak için katılım sistemi dijital ortama donanımları geliştirdik.

Bu araştırma daha büyük sınıf oda, salon veya konferans salonu gibi, geniş alan üzerinde android cihaz Wi-Fi hotspot, kullanma katılımcı seyirci işaretleyebilirsiniz, Android hareket eden kullanma için iletişim yordamı geliştirmek için bir arzu ile başlar ve kullanıcı iş yükünü en aza indirmek için daha fazla kullanıcı dostu şekilde düzeni sunar.

Bu tezde geliştirilen mobil uygulama ve kodlarının tüm hakları Biyodata Teknoloji San. Tic.Ltd. Şti.'ye aittir.

Yoklama performansını optimize eder, bu araştırma önemli yönüdür. Bu Wi-Fi Hotspot fikir tanıtımı ve başarıyla uygulayarak elde edilir. Onun deney ve araştırma gösterir... bu app kapsama alanı çok daha iyidan diğer apps piyasada en fazla 24.5 metre yarıçap içinde çalışır.

Anahtar Kelimeler: iletişim Prosedürü, Android Uygulaması, Kablosuz bağlantı noktası.

ABSTRACT

DEVELOPING COMMUNICATION PROCEDURE FOR MOBILE APPLICATION

Jawad Rasheed

Master's Degree

Thesis Supervisor: Assist. Prof. Dr. Selçuk Baktır

Thesis Co-supervisor: Assoc. Prof. Dr. Gökmen Altay

April 2016, 79 Pages

This study proposes communication procedure for mobile application, and its implementation. Mobile usage has grown substantially in last decade. Mobile Applications (commonly referred as “apps”), are considered to be one of the fastest growing trends in Information Systems industry.

The emergence of computers and mobile devices helped human by extending vast support in every field of life, therefore attendance machines required to automate the attendance process. Scientists and developers have developed different devices and equipment, to digitize the attendance system to eliminate the need of paper work, that takes attendance of employees in a firm, students in schools and universities, audience in seminars, members in meetings, and crowd in ground.

This research starts with a desire to develop Communication Procedure for Android Mobile Application, which can mark attendance of attendees using android device Wi-Fi hotspot, over wide area, such as bigger class room, halls or auditorium and presents the layout in more user friendly manner in order to minimize the user work load.

The App and its code developed for this research belongs to Biyodata Teknoloji San. Tic. Ltd. Şti, and all its rights belong to Biyodata Teknoloji San. Tic. Ltd. Şti,

The important aspect of this research is that it optimizes the performance of taking attendance. This is achieved by introducing the Hotspot Wi-Fi idea and implementing it successfully. The research and its experiment shows that this app works over the coverage area of maximum 24.5 meters' radius, which is quite better than other apps in the market.

Keywords: Communication Procedure, Android Application, Wi-Fi Hotspot.

CONTENTS

TABLES.....	ix
FIGURES.....	x
ABBREVIATIONS.....	xiii
1. INTRODUCTION.....	1
1.1 COMMUNICATION PROCEDURE FOR SCHOOL ATTENDANCE SCENARIO.....	2
1.2 COMMUNICATION PROCEDURE FOR MEETING SCENARIO.....	3
1.3 COMMUNICATION PROCEDURE FOR EMPLOYER EMPLOYEE SCENARIO.....	4
2. GENERAL INFORMATION.....	7
2.1 PRIOR WORK.....	7
2.1.1 Attendance with Mobile Phones.....	7
2.1.2 Other Automatic Ways of Taking Attendance.....	10
2.2 DEFICIENCIES IN PRIOR WORKS.....	10
3. DATA AND METHOD.....	13
3.1 ANDROID ENVIRONMENT.....	13
3.2 BRIEF OVERVIEW OF ANDROID MOBILE ATTEND. APP.....	14
3.3 TECHNIQUES AND METHODS.....	18
3.3.1 Operations Permissions.....	18
3.3.2 Switching Between Activities.....	21
3.3.3 Collecting Information from Website.....	22
3.3.4 Setting Hidden Wireless Hotspot Network.....	23
3.3.5 Establishing Data Communication Channel.....	24
3.3.6 Repeating Request for Attendance.....	26
3.3.7 Send Attachment in an Email.....	27
3.4 APPLICATION GUIDE FOR FEATURES CLARIFICATION.....	27
3.4.1 Login.....	28
3.4.2 Main Activity.....	32

3.4.3 Manager Activity.....	33
3.4.4 Attendee Activity.....	33
3.4.5 School Attendance Scenario.....	35
3.4.5.1 For teachers.....	35
3.4.5.1.1 Add course details.....	36
3.4.5.1.2 View courses list.....	39
3.4.5.1.3 Delete course.....	40
3.4.5.1.4 Course selected.....	41
3.4.5.1.5 View students list.....	42
3.4.5.1.6 Add student.....	43
3.4.5.1.7 Delete student.....	44
3.4.5.1.8 Automatic attendance.....	45
3.4.5.1.9 Manual attendance.....	47
3.4.5.1.10 View attendance.....	48
3.4.5.1.11 Export attendance sheet.....	50
3.4.5.2 For students.....	51
3.4.5.2.1 Add course details.....	52
3.4.5.2.2 View course list (Attendee side).....	52
3.4.5.2.3 Delete course.....	53
3.4.5.2.4 Request to mark attendance.....	53
3.4.6 Casual Meetings Scenario.....	55
3.4.6.1 For meeting manager or conductor.....	55
3.4.6.1.1 Add casual meeting.....	55
3.4.6.1.2 View meetings list.....	57
3.4.6.1.3 Delete casual meeting.....	57
3.4.6.1.4 Meeting selected.....	58
3.4.6.1.5 Automatic attendance.....	58
3.4.6.1.6 Manual attendance.....	59
3.4.6.1.7 View attendance.....	60
3.4.6.1.8 Export attendance sheet.....	60
3.4.6.2 For meeting attendee.....	60
3.4.6.2.1 Add casual meeting details.....	61

3.4.6.2.2 View meetings list (Attendee side).....	62
3.4.6.2.3 Delete casual meeting.....	62
3.4.6.2.4 Request to mark attendance.....	63
3.4.7 Business Environment Scenario.....	64
3.4.7.1 For employer.....	64
3.4.7.1.1 Add employee by importing list.....	65
3.4.7.1.2 Add employee manually.....	65
3.4.7.1.3 View employees list.....	66
3.4.7.1.4 Employees attendance.....	67
3.4.7.1.5 Automatic attendance.....	67
3.4.7.1.6 Manual attendance.....	68
3.4.7.1.7 View attendance.....	69
3.4.7.1.8 Export attendance.....	71
3.4.7.2 For employee.....	71
3.4.7.2.1 Add employer.....	72
3.4.7.2.2 View employers list (Attendee side).....	72
3.4.7.2.3 Delete casual meeting.....	73
3.4.7.2.4 Request to mark attendance.....	73
4. FINDINGS.....	74
5. CONCLUSION.....	76
REFERENCES.....	77

TABLES

Table 1.1: List of similar Android applications.....	9
Table 3.1: Targeted users of Attendance app.....	15
Table 4.1: Comparisons of Results with other attendance apps.....	75



FIGURES

Figure 1.1: Communication procedure for school attendance scenario.....	2
Figure 1.2: Communication procedure for meeting attendance scenario.....	2
Figure 1.3: Communication procedure for employer-employee scenario.....	2
Figure 3.1: Flow chart of Attendance app.....	15
Figure 3.2: Attendance app activities files structure.....	19
Figure 3.3: Login.....	28
Figure 3.4: Sample filled Login fields.....	29
Figure 3.5: Error invalid email entered in Login.....	29
Figure 3.6: Error no internet connection at Login.....	30
Figure 3.7: Sending pass code.....	31
Figure 3.8: Enter received Pass Code.....	31
Figure 3.9: Incorrect Pass Code entered.....	32
Figure 3.10: Main Activity of Attendance Program.....	33
Figure 3.11: Manager Activity.....	34
Figure 3.12: Attendee Activity.....	34
Figure 3.13: School Attendance (Manager Side)	35
Figure 3.14: Add course details (Manager Side)	36
Figure 3.15: Set starting date of the course.....	37
Figure 3.16: Import student list.....	37
Figure 3.17: Select the desired file from dropbox.....	38
Figure 3.18: Sample student list format to import.....	38
Figure 3.19: Student list imported successfully, confirmation message.....	39
Figure 3.20: View courses list.....	40
Figure 3.21: Delete course.....	40
Figure 3.22: Update view courses list after course deletion.....	41
Figure 3.23: Course course.....	42
Figure 3.24: View students list of selected course.....	42
Figure 3.25: Add student example.....	43
Figure 3.26: Update view students list after adding student.....	43

Figure 3.27: Delete student.....	44
Figure 3.28: Updated view student list after deleting student.....	44
Figure 3.29: Automatic Attendance.....	45
Figure 3.30: Take attendance.....	46
Figure 3.31: Attendee’s attendance request receive.....	46
Figure 3.32: Manual attendance.....	48
Figure 3.33: View attendance record.....	49
Figure 3.34: View attendance record after marking attendance manually.....	49
Figure 3.35: Choose email service for exporting attendance file.....	50
Figure 3.36: Send attendance sheet to email.....	51
Figure 3.37: Attendee School Attendance.....	51
Figure 3.38: Add course details (Attendee Side).....	52
Figure 3.39: View courses list added by a student.....	53
Figure 3.40: Attendee connecting with manager’s device hotspot.....	54
Figure 3.41: Attendance confirmation message.....	54
Figure 3.42: Casual meetings.....	56
Figure 3.43: Add casual meeting details (Manager Side)	56
Figure 3.44: View casual meetings list (Manager Side)	57
Figure 3.45: Delete casual meeting (Manager Side)	58
Figure 3.46: Meeting selected.....	59
Figure 3.47: View casual meetings list (Manager Side)	59
Figure 3.48: View attendance of casual meeting attendees.....	60
Figure 3.49: Casual meeting attendee.....	61
Figure 3.50: Add casual meeting (Attendee Side)	62
Figure 3.51: View casual meeting list (Attendee Side).....	63
Figure 3.52: Employer Activity.....	64
Figure 3.53: Add employee and employee info.....	65
Figure 3.54: Add employee manually.....	66
Figure 3.55: View employees list.....	66
Figure 3.56: Employees attendance.....	67
Figure 3.57: Employee attendance manually.....	68
Figure 3.58: View employees attendance.....	69

Figure 3.59: View employees' attendance set date.....	70
Figure 3.60: View employees attendance of select date.....	70
Figure 3.61: Employee activity.....	71
Figure 3.62: Add employer.....	72
Figure 3.63: View employers list.....	73



ABBREVIATIONS

AVD	:	Android Virtual Drive
API	:	Application Programming Interface
Apps	:	Application
CSV	:	Comma Separated Values
DB	:	Database
GPS	:	Global Positioning System
IDE	:	Integrated Development Environment
iOS	:	IPhone Operating System
IP	:	Internet Protocol
JDK	:	Java Development Kit
OS	:	Operating System
RFID	:	Radio Frequency Identification
SDK	:	Software Development Kit
SSID	:	Service Set Identifier
WI-FI	:	Wireless Fidelity
UI	:	User Interface
URI	:	Universal Resource Identifier

1. INTRODUCTION

Mobile usage has grown substantially in last decade. Besides it's calling feature, it is used across the world for variety of applications, such as reading/sending emails, playing online games, streaming videos and many more.

Mobile Applications (commonly referred as “apps”), are considered to be one of the fastest growing trends in Information Systems industry (Eddy, 2011). As mobile platforms continue to improve in performance, without introducing unnecessary design complexity in the user's life, it provides such great features that ease the work load. This is why interacting with business systems via mobile applications is becoming more popular than any other because of its easy interface.

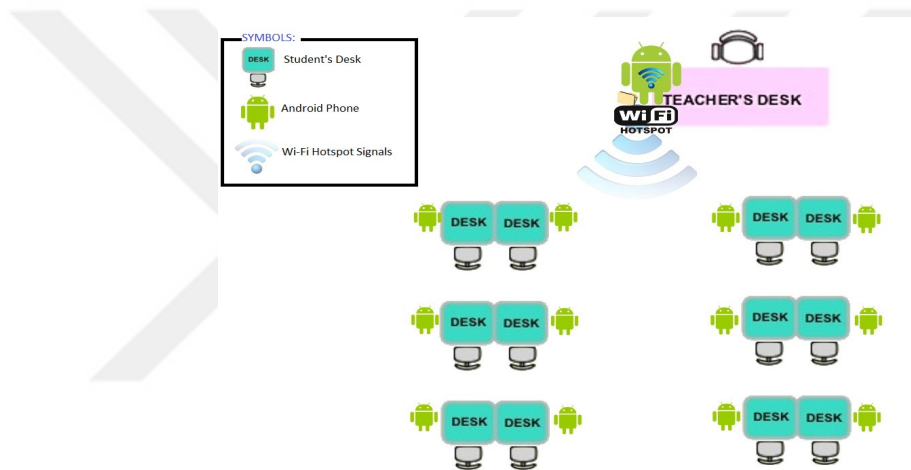
Google play store provides vast variety of software and applications keeping in view the customer needs and expectations. Such as attendance applications for android users to mark attendance of the audience or students. Most of these are based on Bluetooth connectivity, which limits the connectivity range as well as its functionalities, thus affecting the practical usage of this application over wide range of area. Development teams of these applications, faced difficulties like technology constraints (Abrahamson, 2007), limited physical resources, rapid specification changes, evolving and inherent constraints (Hayes, 2003), thus limiting them with their application implementation to Bluetooth connectivity.

This research starts with a desire to develop Communication Procedure for Android Mobile Application, which can mark attendance of attendees using android device Wi-Fi hotspot, over wide area, such as bigger class room, halls or auditorium and presents the layout in more user friendly manner in order to minimize the user work load. Based on the next chapters, preliminary findings show that there is no single android attendance application that is based on Wi-Fi hotspot.

1.1 COMMUNICATION PROCEDURE FOR SCHOOL ATTENDANCE SCENARIO

There is teacher and student scenario, in which teacher wants to take attendance of students via his mobile phone. Figure 1.1 shows the scenario, which is further illustrated stepwise.

Figure 1.1: Communication procedure for school attendance Scenario



Suppose there are twelve students in the class, each have android mobile phone. Teacher not only wants to take attendance of each and every student but also wants to keep the presence time of each student. The proposed communication procedure for this scenario is described as below.

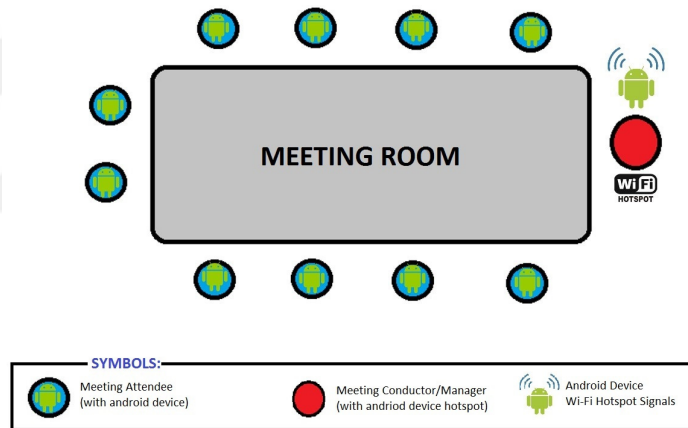
- a. Teacher have saved students name who are enrolled in the course
- b. When class starts, teacher will turn on his android mobile device Wi-Fi hotspot.
- c. Each student's device will connect with teacher's device hotspot
- d. After connecting, each student's device will send student's ID to mark attendance
- e. The teacher's device will receive messages sent by the student's device
- f. Teacher's device will send the confirmation message to each student's device
- g. When the student's device receives the confirmation message, it will disconnect with teacher's device hotspot, so that other students can connect

- h. This process will repeat after every 20 minutes to keep record of presence time of each student in the lecture.

1.2 COMMUNICATION PROCEDURE FOR MEETING SCENARIO

There is meeting scenario, in which meeting conductor or manager wants to take attendance of meeting attendees via his mobile phone. Figure 1.2 shows the scenario, which is further illustrated stepwise.

Figure 1.2: Communication procedure for meeting attendance scenario



Suppose there are some attendees who are attending the meeting and a meeting manager is conducting that meeting, each have android mobile phone. Meeting manager wants to keep record of each attendee's attendance who is attending the meeting via his android mobile device and total time presence time of each member attendee separately. The proposed communication procedure for this scenario is described as below.

- a. The meeting manager will turn-on Wi-Fi hotspot on his device
- b. Each meeting attendee will connect with his/her device with manager's device hotspot
- c. After connecting, each attendee's device will send the attendee's name to manger's device over connected Wi-Fi hotspot

- d. Manager’s device will receive the messages sent by each attendee’s device
- e. After receiving message, manager’s device will record the attendance of that attendee and sends confirmation message back to attendee’s device
- f. When attendee’s device will receive the confirmation message, it will disconnect with the manager’s device, so that other attendees can connect and request for their attendance
- g. This process repeats after every 20 minutes, in order to record the presence time of each attendee.

1.3 COMMUNICATION PROCEDURE FOR EMPLOYER-EMPLOYEE SCENARIO

There is an employer-employees’ scenario, in which employer wants to take attendance of employees via his mobile phone. Figure 1.3 shows the scenario, which is further illustrated stepwise.

Figure 1.3: Communication procedure for office attendance scenario



Suppose that, employees are entering in the office, each have android mobile phone. Employer or the office receptionist has an android device too and wants to keep record of each employee's check-in and check-out time of the office, in order to calculate the hours an employee works each day. The proposed communication procedure for this scenario is described as below.

- a. Employer or receptionist will turn on his/her android device hotspot
- b. Each employee who enters the office, connects with the employer's device hotspot
- c. After connecting, employee's device will send the employee ID or name to employer's device over Wi-Fi hotspot
- d. Employer's device will receive the message sent by the employee and record the check-in time of the employee
- e. Employer's device will send a confirmation message to employee's device
- f. When the employee's device receives the confirmation message, it disconnects with the employer's device
- g. On the other side, if employee is leaving the office, it will connect with the employer's device hotspot
- h. After connecting, employee's device will send the employee's ID or name to the employer's device to record the attendance.
- i. Employer's device will receive the message sent by employee and record the check-out time of that employee
- j. Employer's device will send confirmation message to employee's device
- k. When employee's device receives the confirmation, it will disconnect with employer's device hotspot

This thesis illustrates stepwise description of different programming techniques used in Android application development process that are implemented which incorporates main features like Wi-Fi hotspot, importing the attendees sheet and export the attendance sheets which other applications does not have.

The fundamental features of this android attendance application are:

- a. Connecting devices over Wi-Fi Hotspot (for latest android versions also)
- b. Automatic attendance marking with time stamps
- c. Manual attendance mark
- d. Calculating the attendee presence time
- e. Importing the attendees list
- f. Add/Delete course/student/instructor/employer/employee manually
- g. Exporting attendees' attendance sheet

To implement the above mentioned features, various connection types and data sources are needed which can be achieved with following programming techniques:

- a. Accessing APIs
- b. Socket programming
- c. Thread programming
- d. Accessing external memory
- e. Programmable components

This thesis constitutes of five chapters. Chapter 2 includes the Literature Review & General information, which points out work done previously in similar applications, thus clarifying the importance of this research and its implementation. Chapter 3 illustrated the android programming environment setup, application features and its usage, methods used to implement these features. Chapter 4 states the difference with other applications and its performance. Thesis ends with conclusion, observation and limitations of this research in Chapter 5.

From now on the words application and applications will be abbreviated to app and apps, respectively

2. GENERAL INFORMATION

This section provides the information necessary for a reader to understand why this research is carried out. It covers the ways and methods of taking attendance with respect to time and situation, the work done in this regards and their comparison with each other and point out the deficiencies, keeping in view the human need in today's time, thus clarifying the importance of this research and its implementation.

2.1 PRIOR WORK

The emergence of computers and mobile devices helped human by extending vast support in every field of life, therefore attendance machines required to automate the attendance process. Thus the attendance process also changes over time with the lifestyle and the evolution of computer. As life advances, human came up with better attendance procedure and process, switching from manual error prone process of maintaining manual attendance to automating attendance. Scientists and developers have developed different devices and equipment, to digitize the attendance system to eliminate the need of paper work, that takes attendance of employees in a firm, students in schools and universities, audience in seminars, members in meetings, and crowd in ground.

2.1.1 Attendance with Mobile Phones

After the invention of mobile phone, tablets and other mobile devices, many daily bases work has been shifted to it, which decreases the time consuming paper work. Developers have developed billions of apps for mobile devices, mainly for android operating systems and iOS (mobile operation system for apple mobile devices) to ease different kind of calculations, communication and other things. As topic of this thesis is related to Android, therefore the main focus will be on it.

Android is a mobile operating system (OS) and relatively new platform based on Linux kernel and designed primarily for touchscreen mobile devices. Initially it was developed by Android, Inc., which then was bought by Google Inc. (Elgin, Ben, 2005) and its first release was unveiled in 2007 (Meier, 20010).

The real game began when Google open the Google Play Store (app store) for Android users. Its users can download android applications through Google Play, which replaced the old Android Market (Bishop, 2012).

In year 2013, a survey of mobile app developers found that seventy-one percent of developers create applications for android (Developer Economics Q3 analyst report, July 2013). In the same year Google play store has had over one million Android applications published, and over 50 billion applications downloaded (Phonearena.com, 2013). It was further revealed in Google I/O conference that there were over one billion active monthly Android users (Techspot, 2014).

Google app store allows developers to create apps and provide these on the Google Play store. Among these billions of apps, developers have provided attendance apps for a common android user.

On the idea of marking the attendance on phone, very little work on android has been done. As per this thesis research, similar apps on Google Play store regarding attendance are available but not much, few of those are listed in Table 1.1. These apps were developed for older versions of Android but also works fine on newer Android.

Developers of these apps have targeted various Application Program Interfaces (API) using different kinds of Software Development Kits (SDK). Most of these apps mentioned in Table 1.1 targeted the school, college and university attendance. Very few apps are made for general purpose of attendance. Each app has its own features as per requirement of the customer, which are listed below.

- a. Manual attendance
- b. Create/Delete course

- c. Add/Delete student
- d. Import student list
- e. Export attendance sheet
- f. Statistical representation of attendance
- g. View attendance reports
- h. Automatic attendance (only three apps listed in Table 1.1 has this feature)

Despite other features, automatic attendance feature is only in three android apps, namely “Bluetooth Attendance” which take the attendance over the Bluetooth and is just for faculty members, “Attendance Keeper” that just target firms & organization attendance by marking employees’ location with the help of Global Positioning System (GPS) and internet by accessing users’ account on workplace server, and third one “Hello Office” also targets employees’ attendance if an employee is connected with a specific router with pre-mentioned mobile device.

Table 1.1: List of similar Android applications

Sr. No.	Application Name	Developed by	Price (TRY)
1	Attendance Tracker	PeterMan	4.22 – 12.44
2	Bluetooth Attendance	Mihir Dhakan	30 days trial verison
3	Attendance Keeper	Azure Droid	4.66 per item
4	Hello Office	N/A	2.56 – 4.00 per item
5	When I work Time & Attendance	When I Work, Inc.	In app purchases
6	Smart Attendance	N/A	Free
7	Attendance	Andre Restivo	Free
8	Easy Attendance Register	thebasicapp	Free
9	Attendance Manager	Dot Inc.	Free
10	Attendance Tracker (Students)	Shaunak Gupte	Free
11	Attendance Register (Students)	Rudra Nirvan	Free
12	Employee Attendance Tracker	Alice Inc.	Free

2.1.2 Other Automatic Ways of Taking Attendance

Mobile phones are not the only solution to take or mark attendance. Scientists have developed alternative ways to mark the attendance by introducing various types of devices.

Like dictation machines, fax machines, and carbon paper, attendance tracking solutions have evolved to meet the demands of the contemporary workplace. More and more businesses are phasing out outdated paper-punch card systems with biometric fingerprint attendance or retina eye scan attendance technology, that uses a special scanner to scan the finger print or eye retinal structure of a human. As each human being has different finger print and eye retinal structure which makes the machine to easily identify every individual.

Some came up with the idea of Radio Frequency Identification (RFID) system that uses radio waves to identify the object, human. RFID takes the barcoding concept and digitizes it for the modern world providing the ability to uniquely identify.

As these solutions are less related with our thesis topic, so for information about Radio Frequency Identification system reader can read “RFID Applications: An Introductory and Exploratory Study” by “Ahsan, Shah & Kingston, Jan 2010), for Biometric Scanning reader can go through a brief survey by of Biometric in IRIS Technology by “C. D. Patel, S. Trivedi, S. Patel”.

2.2 DEFICIENCIES IN PRIOR WORKS

When it comes to choose the attendance system, there are two options: first is on premises systems in which installing software within the business is required, and second solution is based on cloud system in which installing a software is not essential. These examples of mobile applications and other systems can be roughly grouped into above defined two groups. A broader survey of various categories of mobile applications can be found in “Language and Infrastructure Design for Mobile

Computation” (June 2000). Although solutions discussed above are accurate and fast, but still each of these has its own pros and cons, which are listed below;

- a. Expensive; for RFIDs and Biometric Scanning systems, companies/schools/firm have to invest a lot in purchasing and installing the equipment
- b. Time Consuming; except RFID system which can scan more than 1000 tags in one minute (Mirva Saarijarvi, Dec 2011), most of the solutions listed above are time consuming, means system consume a lot of time to mark the attendance, such as Biometric scanning system and android mobile apps with manual attendance feature only. Mobile apps with Bluetooth connectivity system also takes time to mark the attendance, while bonding and pairing with other devices.
- c. Infrastructure Required; Mobile apps with automatic attendance over WiFi connectivity requires mid-point routers, a server and GPS interface, which means more money.
- d. Shorter Coverage Area; As Bluetooth is a wireless technology standard for exchanging data over short distances so Android attendance apps which record the attendance via Bluetooth connectivity has shorter coverage range around nice meters (B. K. Mandal, D. Bhattacharyaa, & T. Kim, 2014).
- e. Less Connection Channel: In mobile Bluetooth connectivity there is a limit of seven slave nodes in a Bluetooth network as three bits addressing is used. So no mobile can mark attendance of more than seven other mobile persons at the same time.

It is worth noting that a large proportion of these mobile apps involve much human interaction, generally the apps with manually attendance feature only. The biggest drawback to time and attendance systems is that they are more expensive than the old manual method where employees wrote down their hours each day or teacher took attendance of each student by calling out students' names.

An internet search does not provide any example of Android Mobile Attendance application that exploit pre-selected features not implemented in any other attendance app. Therefore, it can be concluded that above mentioned problems formed the bases of this research & to implement these feature (specially over Wi-Fi Hotspot). The

successful implementation is shown in next chapter that also describes methods used for these implementations and its working.



3. DATA AND METHOD

This section covers the solution to implemented the application features mentioned in previous chapter, its usage and methods. It outlines the steps required to setup an android environment, explains the fundamental concepts of Android application development and specify few common techniques frequently used for Android Mobile Attendance app.

While designing this application, most number of android mobile users are targeted, by setting the minimum SDK to API 16 (Android 4.1, Jelly Bean), so that app runs on 94.8 percent (according to Android Studio) of the devices. Lower API levels target more devices, but have fewer features available, therefore designing of this app is in such a way that it covers more devices without affecting our main feature of attendance over Hotspot connection.

3.1 ANDROID ENVIRONMENT

Android environment setup is the first step before starting the implementation of android mobile attendance proposed solution (that is attendance over Hotspot connection) in this thesis. For this thesis, Android Studio is used as Android environment. To setup android environment following steps are acquired from Android official website.

- a. Install Android Java Development Kit (JDK) from Sun/Oracle website, as all the android coding is based on java code.

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

- b. Install Android Official Integrated Development Environment (IDE) from Android Developers website (from the below link given below), as it is one of the best android application developing tool kit with user-friendly interface and world-class code editing, debugging and instant build/deploy system facility.

<http://developer.android.com/sdk/index.html>

This provides the tools and resources needed to test Android apps. Upon Installation, users should verify the installation of all the necessary components such as different APIs via Software Development Kit (SDK).

To know more about android studio, reader can visit android developer website.

<http://developer.android.com/tools/studio/index.html>

- c. Use live testing environment. In Android Studio there is option to setup Android Emulator for testing purposes from Android Virtual Drive Manager to create a new Android Virtual Drive (AVD). But it will be more appropriate to test it using a personal Android device as a debugging tool, as establishing a hotspot connection between two devices is the main part of this application.

3.2 BREIF OVERVIEW OF ANDROID MOBILE ATTENDANCE APP

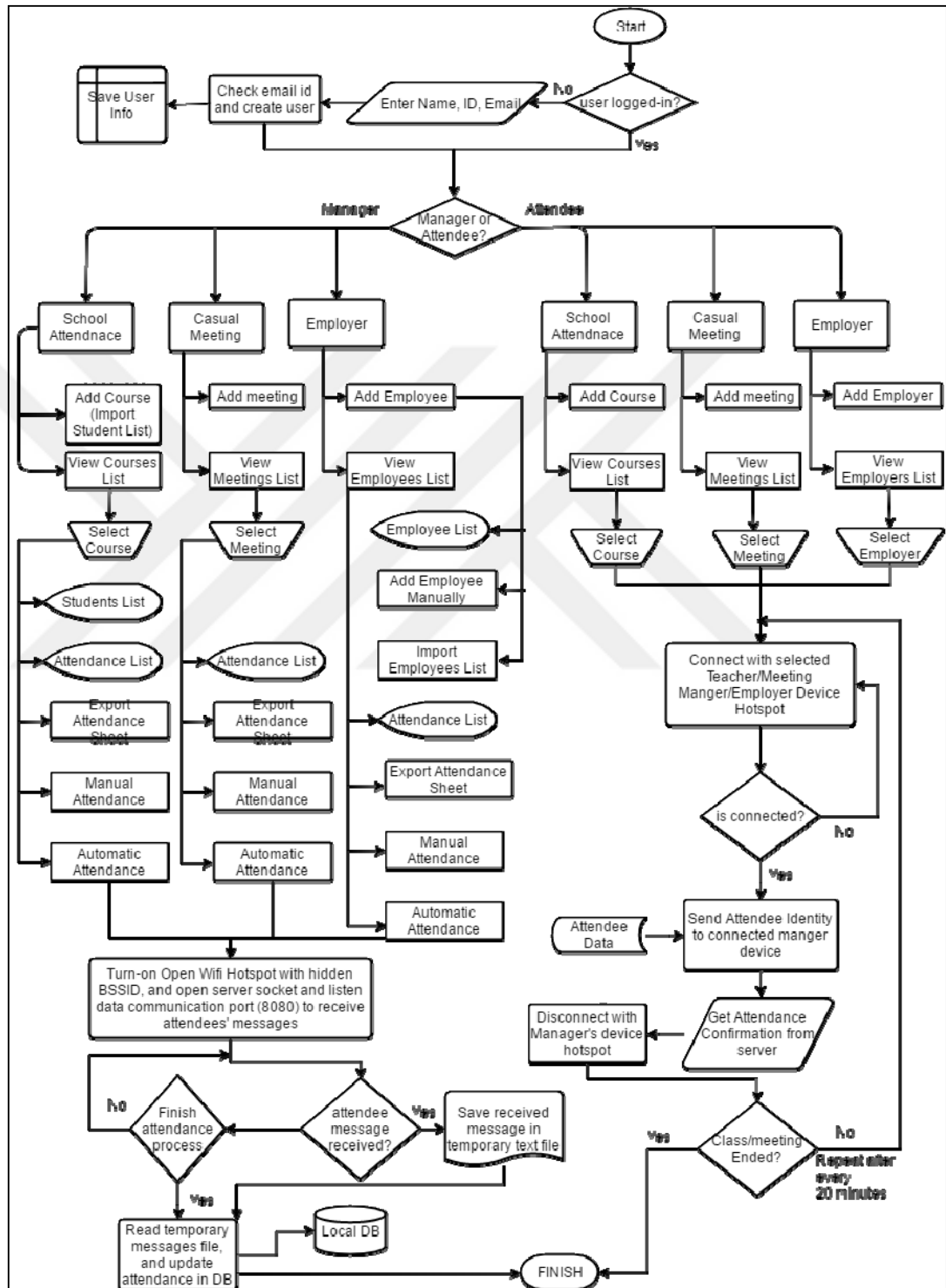
Android Attendance app is fully automatic attendance that takes attendance by establishing connection with hotspot Wi-Fi device of the manager and sending attendance request over data communication channel via socket programming. The main idea and work flow of this app is represented in Figure 3.1.

The research is to develop a general attendance app which can be used in firm, small organization or company, meetings or conference, school or university class rooms. This app divided these in three categories, “School Attendance” for class attendance, “Casual Meetings” for meetings and conferences and “Employer – Employee” for small firm, company or organization. Each of these are further categorized in two groups (manager, and attendee). Users of each category and group is shown in Table 3.1

Table 3.1: Targeted users of Attendance app

Category\Group	Manager	Attendee
School Attendance	Teacher, Instructor, Professor	Students
Casual Meeting	Meeting Conductor	Meeting members, Audience
Employer - Employee	Employer, Boss	Employee

Figure 3.1: Flow chart of Attendance app



In other words, each group has three categorizes, “School Attendance”, “Casual Meetings” & “Employer-Employee”. Before proceeding further, brief functionalities of each group is listed below for readers to get idea about flow of application.

Manager Side of this app is that who will be responsible to look after the attendance and mark attendance of the attendee. **Attendee Side** of this app is for those who wants to request their concerned authority to mark attendance.

School Attendance: This part of app is for schools and university professor and students. Following points briefly explains how the attendance system works for teacher and students.

- a. **Manager Side:** This activity is for teacher, instructor or professor of school, college or university. In this activity an instructor can add course, and import the students list saved in Comma Separated Values (CSV) file from his/her drop-box account. Once the course details along with students are saved, the instructor can view the courses lists he added. By clicking on the any course in the list, the instructor can view the students list, add or delete student, view attendance, mark attendance manually and take automatic attendance.
- b. **Attendee Side:** School Attendance for Attendee side is basically for students of the school, college or university. Here a student can add the course and its instructor details. Student can create as many courses as he/she wants and can view the added courses list. If student wants to send request of marking attendance to instructor, he should wait till the instructor opens the attendance process. Once the attendance process opens, student can send request of attendance just by clicking on the course name. After that a background service, after every 20 minutes, will send request to teacher to mark his/her attendance. This background service will last till the lecture ends.

Casual Meetings: This segment of attendance app is for casual meetings and conferences. schools and university professor and students. Below points briefly states attendance system that work for meeting manager and meeting attendees or audience.

- a. **Manager Side:** This side is for meeting manager or meeting conductor who will be responsible of taking the attendance. In the Casual Meeting of Manager side,

the meeting manager can add a meeting just by enter the meeting name and data. A manager can open any added meeting details by clicking on meeting name in the meetings list. Meeting manager can further start and finish automatic attendance with one button. It has also option to take manual attendance, view attendance record and export the attendance of any meeting to his/her email address.

- b. Attendee Side:** Members of the Casual Meeting can request the meeting manager for his/her attendance by accessing the Casual Meeting of Attendee Side of the app. Attendee can request for attendance by adding the meeting name and hotspot name of that meeting. Once the attendance process opens, attendee can connect with the manager device over hotspot Wi-Fi and send his/her name to mark attendance just by clicking on the meeting name in the lists of meetings. After that a background service start automatically and requests meeting manager to mark the attendance. The background runs after every 20 minutes' interval, and last till the meeting ends.

The last category of this app is for business environment such as employers and employees of a firm, organization or company. Following points shortly describes the attendance system for employer and employees.

- a. Employer (Manager Side):** Employer is primarily for the Manager or person who is responsible to take attendance in a firm, organization or company. In this part of the app, an employer imports the list of the employees, from dropbox account, working for the company, add or delete employee manually. It has also options of Automatic attendance, Manual attendance and view hours worked by employees of desired date. The attendance of employees is with time stamp, in order to keep the check-in and check-out time of each employee. Furthermore, he/she can import the attendance and time record of employees
- b. Employee (Attendee Side):** In this activity of the app, an employee requests for attendance to an employer by adding employer name and hotspot name in the app. An attendance is requested at the Check-In time and Check-Out time just by clicking on the employer name in the list.

3.3 TECHNIQUES AND METHODS

An android Program works in the form of activities. Activities are classes within packages that interact with the user (“Activity”, 2012). These classes extend an “Activity” type and thus inherit methods and other related information needed for successful implementation of Android Mobile Attendance app. For each activity in attendance application a new class is defined. The activities file structure of this app is shown in Figure 3.2.

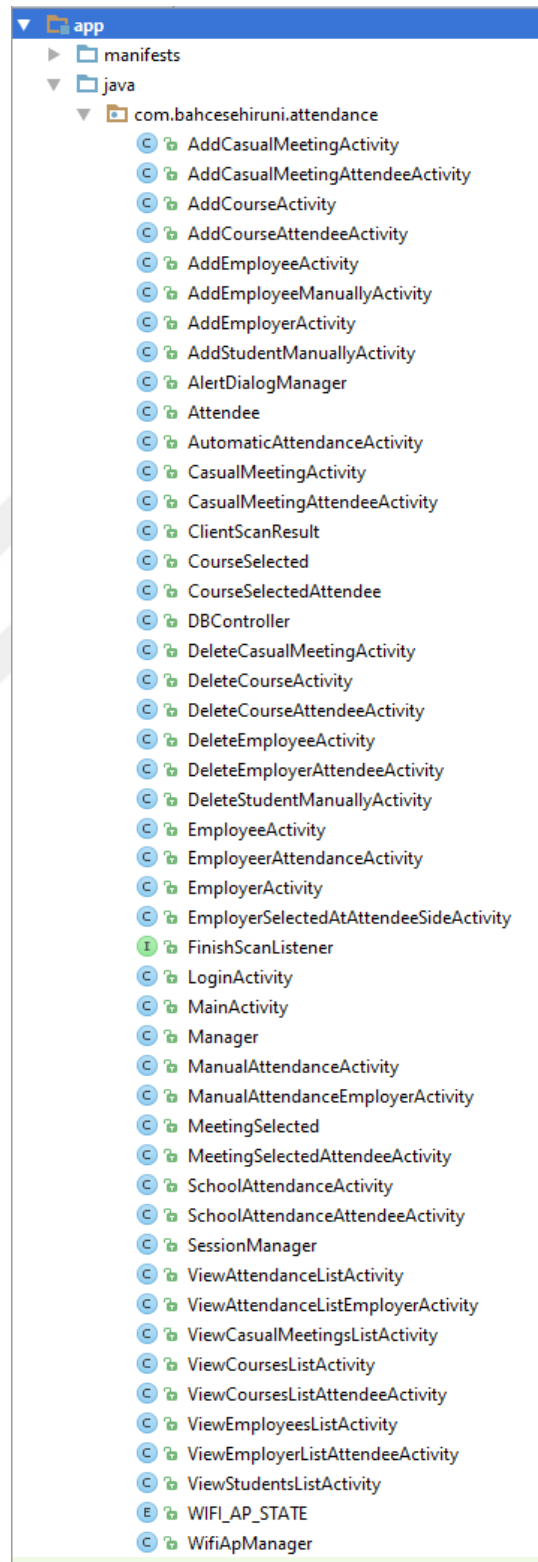
To implement in application for most of the android user is a great challenge. As Android 6.0 (API Level 23) is introduced with variety of system changes and API behavior changes, so code is designed in a way that it supports Android Jellybean as well as Android Marshmallow. Following are the techniques and methods used for the implementation.

3.3.1 Operations Permissions

Android is based on Linux in which each applications runs with distinct system identity, Linux user ID and group ID, which isolates applications from each other and from the system (“Systems Permission”, 2016). In Android no app, by default, has permission to perform any operations such as reading or writing another application files, keeping the device awake, changing network configurations or performing network access, that would adversely impact other apps, the operating system or the user. AndroidManifest.xml file is the most critical file in Android app structure that allows developer to define type of permissions required for application, available activities in the application and setting the Android API level requirement.

Felker (2011) states that the AndroidManifest.xml file “keeps track of everything your application needs, requests and has to use to run”. Therefore, for the successful implementation of this thesis research, following permission are given in AndroidManifest.xml file of the application in order to allow application features to

Figure 3.2: Attendance app activities files structure



work fully and properly. Following list shows the permissions given with respect to operations required in this app;

- a. Accessing Internet; to access the internet with in the app, such as at the time of email verification in login activity or access dropbox folder, internet permissions are given.

```
<uses-permission android:name="android.permission.INTERNET" />
```

- b. Read CSV file; to import and read the CSV file from the memory, read permissions should be set as below,

```
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
```

- c. Write file; to write the text files such as temporarily saving received messages of attendees at manager side in a text file, or preparing CSV attendance file of the attendees to export, writing permissions must be set as follows,

```
<uses-permission android:name="android.permission.WRITE_SETTINGS" />  
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

- d. Turn on hotspot; turning on hotspot with the conditions to configure it as open connectivity network, mean without password but with hidden network means general people cannot scan it, requires special permission for this configuration.

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />  
<uses-permission android:name="android.permission.CHANGE_NETWORK_STATE" />  
<uses-permission android:name="android.permission.CHANGE_CONFIGURATION" />
```

- e. Connecting with Hotspot Wi-Fi; for connecting with hotspot Wi-Fi of the manager following permissions are required.

```
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />  
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
```

- f. Check network state; whether network is connected or not, application should have following permission to check it.

```
<uses-permission android:name="android.permission.INTERNET" />
```

3.3.2 Switching Between Activities

Each major action or operation done in this app is defined as an activity by introducing these activities inside the <application> tags in the Manifest.xml file.

```
<activity  
    android:name=".SampleActivity"  
    android:label="@string/app_name" />
```

Every android application has one at least one launcher activity, which means when the application will start, it will always start with the activity that is defined as the “launcher” activity in the Manifest.xml file. For this attendance app *MainActivity* is the launcher activity.

Once application has two or more activities, switching from one activity to another in right way is critical part of android programming. For this, activities use Intents to communicate with each other. Intents are “messages” shared between Activities (Felker, 2011). Each Intent can store various information that a developer wants to pass it to another activity. To initiate an Intent, following lines of code is used within a class (Activity) to open new Activity by sending additional information as an extra values.

```
Intent nextActivity = new Intent(this, Target_Activity_Name.class,  
    Target_Activity_Name.class);  
// Passing the Extra Intent Values to Keep the track and other things  
nextActivity.putExtra("extra_value_name1", variable_Name1);  
nextActivity.putExtra("extra_value_name2", variable_Name2);  
startActivity(nextActivity);
```

In order to finish the previous activity, after starting new Target Activity, *finish()* command is add at the end of above stated code. At the Target Activity side, these passed variables are accepted for further manipulation, as shown below,

```
Intent i = getIntent();  
// getting attached intent data  
Int variableName1 = i.getIntExtra("extra_value_name1");  
String variableName2 = i.getStringExtra("extra_value_name2");
```

Passing additional information to next activity is to communicate various things, such as in this Android Mobile Attendance app sometimes extra information is passed in order to tell next activity that which “Subject”, “Meeting” or “Employer” is selected in order to start its attendance or other views.

The app is designed in a way that minimizes the code, taking full use of extra intent development option. With the help of these extra information, the code is minimized such as “Attendance” activity class is implemented only once in this app for all entities in group “Manager”.

3.3.3 Collecting Information from Website

Getting information or data from website was easy by including the Apache HTTP client library in the code, but Android 6.0 release removes support for the Apache HTTP Client (“Android 6.0 Changes”, 2016). Android has introduced new class “URLConnection” in API 23 that is more efficient because it reduces network use through transparent compression and response caching and minimizes power consumption with the help of following code.

```
URL url = new URL("http://www.sampleWebsite.com/");
URLConnection urlConnection = (URLConnection) url.openConnection();
try {
    InputStream in = new BufferedInputStream(urlConnection.getInputStream());
    readStream(in);
    finally{
        urlConnection.disconnect();
    }
}
```

From features mentioned in the previous sections, create user and verify the valid and working email address, and email verification system are the ones requiring the implementation of this technique. Once the user press Sign-Up at login page, it sends user’s email address to webpage server, to verify the email by sending a pass code. As a result, website returns some information. In order to get that information, it is best to implement app with the latest class of “URLConnection” so that it works on Android 6.0.

Android app has a main thread called the User Interface (UI) thread, which should contain functionality that can be quickly processed (Vogel, 2010b). Processing HTTP requests sometimes take long time, depending on the user's Internet connection and speed. If HTTP requests are placed in UI Thread, it can cause the application to crash due to unresponsiveness. As Vogel (2010b) suggests that HTTP request code must be placed in a separate Thread, which is accomplished by implementing *AsyncTask* class. A class is defined within the Login activity class in this app, which extends *AsyncTask* class, as following example syntax.

```
private class ExampleClassName extends AsyncTask<String, Void, String>
```

Types in above syntax can differ according to requirement. The first type is the string parameter being passed to a *doInBackground()* method, which is required to invoke within a class (Vogel, 2010b). Second type in this class is used for progress information, while third type is the type of result that *doInBackground()* method will return. (Vogel, 2010b). The result from *doInBackground()* method will be passed to *onPostExecute()* method as a parameter (Vogel, 2010b). An object of this class is initiated and called like sample code given below;

```
new ExampleInnerClassName().execute  
("http://sampleWebsiteURL/verifyemail.php?email=" + user_email);
```

The *HttpURLConnection* request is defined in *doInBackground()* method to get the returned data by website. While *doInBackground()* method is running, progress dialog appears on the screen. Since the main chunk of the work is done, the *ProgressDialog* is dismissed, and result (whether passcode sent or not) of *doInBackground()* method is passed to *onPostExecute()* method. The next action that a *Login Activity* takes according to the result of *doInBackground()* method is placed within *onPostExecute()* method.

3.3.4 Setting Hidden Wireless Hotspot Network

The primary feature of this research is taking the attendance over Manager's device Hotspot, without any intermediate router or access point. Manager's device hotspot

must be enabled at the start of taking attendance, so that attendees can connect with the manager to request for their attendance. Enabling the hotspot programmatically by setting the Manager's (User's) Name as the Hotspot device name so that all the known attendees can connect. But it should be without password, means configuring its security as "open", so that each attendee can connect with it easily.

If Hotspot will be password protected than manager has to tell it to each attendee, thus consuming time and extra work. Despite of making the hotspot Wi-Fi password protected, its security risk can be minimized by setting it open but configuring it as Hidden Wireless Network. Hidden Wireless Networks are those networks which do not broadcast their Network ID that is Service Set Identifier (SSID). This can be done as shown below;

```
WifiConfiguration netConfig = new WifiConfiguration();  
  
netConfig.SSID = ManagerDeviceHotspotName;  
netConfig.hiddenSSID = true;  
netConfig.status = WifiConfiguration.Status.ENABLED;  
netConfig.allowedAuthAlgorithms.set(WifiConfiguration.AuthAlgorithm.OPEN);
```

In this way unwanted guests are restricted to enter into the network and jam the network traffic. On the attendee side, it is programmatically configured that manager's device hotspot is configured as hidden network and it will search for network with the hidden SSID.

3.3.5 Establishing Data Communication Channel

The data communication channel is needed in one of the major feature, mention in previous sections, of this app, such as send the request to manager to mark the attendance. This feature is achieved by implementing socket programming techniques.

A socket is one end-point of a two-way communication link between two programs running on the network. Android has huge set of libraries among those are the socket classes, one for client side and one for server side, that are used to represent the connection between a client program and server program. The server in this app is the

manager side, who will turn on his hotspot device and server socket to get request from the clients who are attendees. For this app, java sockets are used to achieve the server-client (manager-attendee) communication. Manager side (that is server side) which is bonded to port (8080) on the device by opening Server Socket and is available to client using IP address and port combination. Attendee (which is client side) uses some random port for connection. Once connection is established from client side, it will then send the attendee's information to server and server will reply with confirmation message.

The “*AutomaticAttendanceActivity.java*” class contains the implementation of server, which create “*ServerSocket*” object in a separate thread. With this *ServerSocket* object an *accept()* function is used to wait for an incoming request and blocks until the connection is opened.

```
ServerSocket serverSocket = new ServerSocket(SocketServerPORT);  
Socket socket = serverSocket.accept();
```

And uses *dataInputStream* and *dataOutputStream* functions to receive and send the messages over the socket.

```
DataInputStream Input = new DataInputStream(socket.getInputStream());  
DataOutputStream Output = new DataOutputStream(socket.getOutputStream());
```

After establishing the connection, following code will block the server side to proceed further, unless client sends the message.

```
String messageFromClient = dataInputStream.readUTF();
```

The attendee side a class contains the implementation of client, which create “*Socket*” object in a separate thread. With this *Socket* object it sends request to Server Socket by server IP and the Server Port to open a connection.

```
Socket socket = new Socket(dstAddress, dstPort)
```


Socket programming at Server and Client (Manager and Attendee) sides are implemented in an inner class that extends *AsyncTask* class, of an activity. So that main functionality of the activity runs on the UI thread, these communication channel runs on different thread. This is to keep the application working, otherwise it might crash due to unresponsiveness.

3.3.6 Repeating Request for Attendance

The prime aspect of digital attendance is to keep record of attendee's presence time. To do this, attendee should inform the manager about its presence. If the attendee will press the button to request for the attendance, it will distract the on-going work in the class, meeting or organization and consumption of attendee's time. Like other systems, android has given the options of Thread technique to its developers to tackle such problems

Thread is "a process that runs separately from and simultaneously with everything else that's happening" (Felker, 2011). Any time some task takes long time to be processed, so a new Thread should be launched to perform a task in the background (Vogel, 2010b). Thread technique is used in different activities of the app. Especially at attendee side, when student needs to show his presence repeatedly at known time intervals, when meeting manager/conductor wants to know the presence time of the meeting attendees.

To accomplish this feature in the app, a *BackgroundThread* object is used as a service in the code. A time interval is defined after which the Thread will run the request Attendance Task again, and will sleep for next time interval. This process goes on till the class time finishes or meeting time ends. The sample BackgroundThread code as shown below.

```
objBackgroundThread = new BackgroundThread();
objBackgroundThread.start();

public void run() {
    try {
        // Repeat the attendance process after such time interval
        Thread.sleep(defined_Time_Interval);
    }
}
```

Other examples of Thread techniques used in this attendance app are the Asynchronous calls made during the implementation of various features in different activities, which we discussed in this section before.

3.3.7 Send Attachment in an Email

In many cases attendance record is not just for the manager of a company or instructor of a school/college/university. It can be needed by the Boss of the company/firm/organization or by the academic branch of a school/college/university. This is done by preparing a CSV file of attendance record of specific course or meeting by accessing mobile database and then attaching it in an email with the help of Intent.

An intent is declared with the Activity Action of “*ACTION_SEND*” that is to deliver multiple data to someone else, in this case it is to send to email address. Furthermore, setting the Intent Data as “*mailto:*” and type as “*text/plain*”, as the required email must be of simple text. And attaching the Universal Resource Identifier (URI) of the prepared CSV file as extra information in the Intent before start the email activity. Sample of such code is shown below.

```
Intent emailIntent = new Intent(Intent.ACTION_SEND);
Uri fileUri = Uri.fromFile(file);
emailIntent.putExtra(Intent.EXTRA_STREAM, fileUri);
```

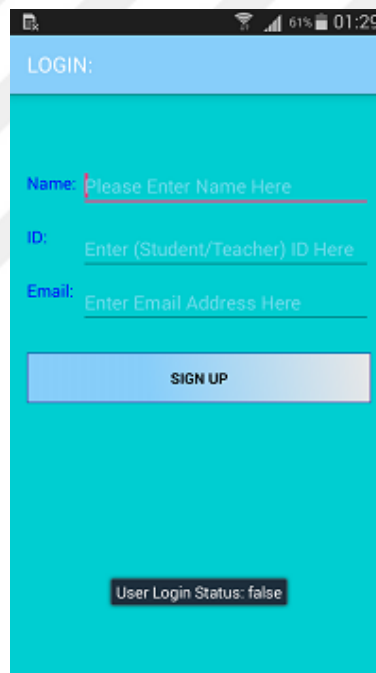
3.4 APPLICATION GUIDE FOR FEATURES CLARIFICATION

This section of the chapter clarifies the features of the app in more details by presenting the screenshots of Android Mobile Attendance app, developed in this research. It guides the reader, how app works.

3.4.1 Login

Like other apps, after installing the attendance app on mobile device, for the first time it starts with the user-creation screen as shown in Figure 3.3. This is the onetime user creation as well as user login screen. It will prompt the user that it's not logged-in and allow the user to enter Name, ID (student will enter student ID in this field, employee will enter employee ID and meeting user can enter any ID as field in this screen are compulsory) and email address.

Figure 3.3: Login



All the fields in the *Login* screen is essentials, so user should fill in correctly before pressing *SIGN-UP* button, as shown in Figure 3.4, as a sample user. If user enter email address in wrong format, for example “*sample_email@gmail*” without “.com” or “*sample_email*” without “@domainName.com”, it will prompt an Error that “Email entered is not correct” as represented in Figure 3.5.

Figure 3.4: Sample filled Login fields

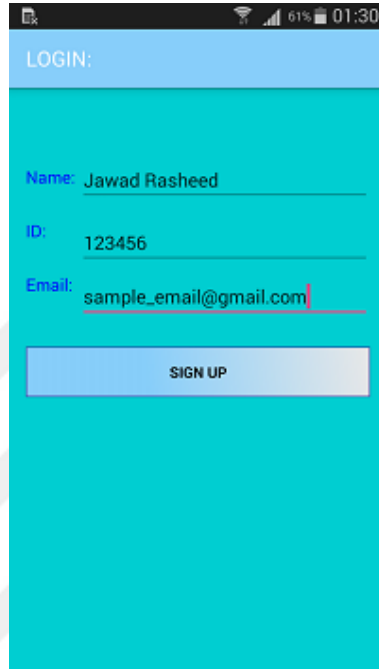
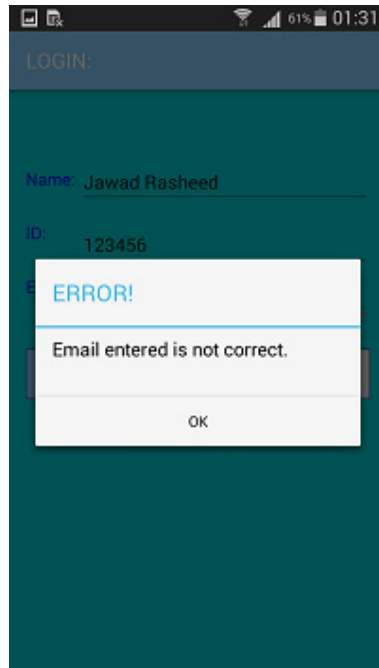
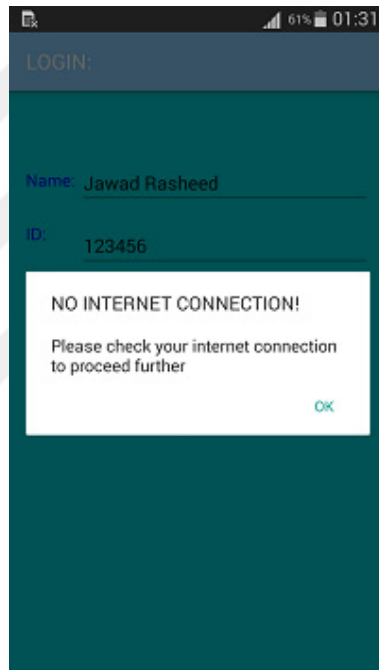


Figure 3.5: Error invalid email entered in Login



If the user proceeds by pressing the *SIGN-UP* button without having internet access, the app will show and dialog box “*NO INTERNET CONNECTION!*” which reminds the user to turn-on the internet, in order to connect with a website server for the first time to verify the email as shown in the Figure 3.6

Figure 3.6: Error no internet connection at Login



If “*NO INTERNET CONNECTION!*” dialog box appears, close and turns on the internet on the mobile device and then press the “*SIGN-UP*” button again. The Attendance app will send the user email, which is entered just before, to the web-server “*biyodata.com*”, so that web-server sends 4 digits Pass Code to this email address in order to verify the email. While the app is trying to accesses the web-server, it will be in the state as shown in Figure 3.7. After sending the email address to the web-server, the Attendance app will prompt the user, as in Figure 3.8, to enter the Pass Code that is sent to the user email address.

Figure 3.7: Sending Pass Code

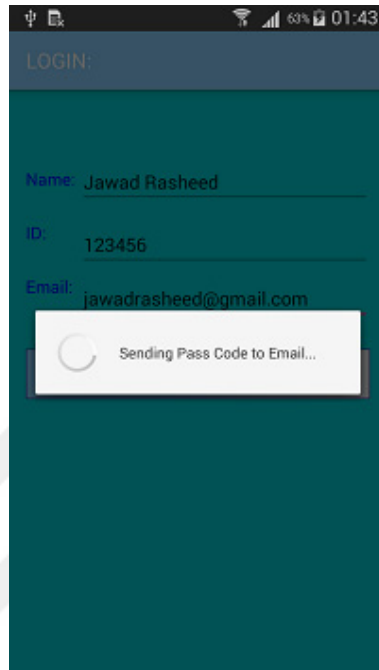
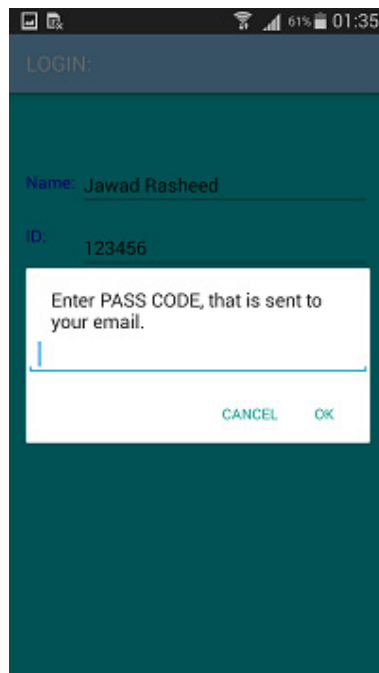
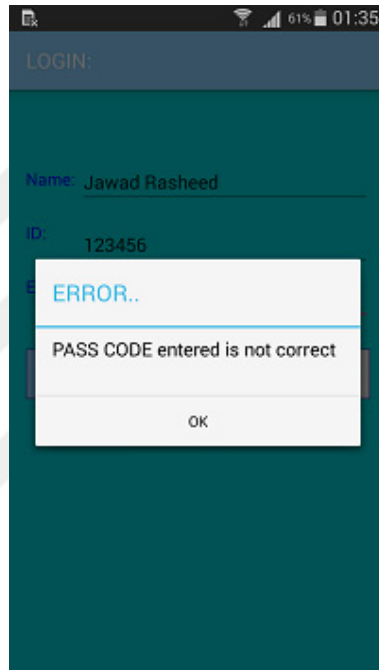


Figure 3.8: Enter received Pass Code



As soon as the user enters the same *Pass Code* which user received in the email, the app will check whether the *Pass Code* entered is same as sent by the web-server to the user's email address. If the entered *Pass Code* is same, the user is created successfully, else it will display Error, shown in Figure 3.9

Figure 3.9: Incorrect Pass Code entered

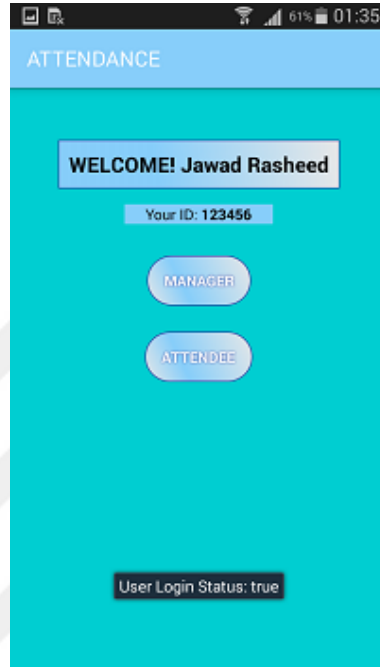


3.4.2 Main Activity

This is the launcher activity of the program. Every time the Attendance app runs, this screen will come, except for the first time, as no user is logged-in. Once the user is created, the app on the same device will start from this screen, as shown in Figure 3.10. It shows the welcome note, user name, and user ID which was entered at the time of user creation. If a user is teacher, instructor, professor, meeting manager, conference conductor, employer or the one who is responsible to take attendance, then user must click on "MANAGER" button. On the other hand, if user is student, meeting attendee,

conference audience or employee of a company, then “ATTENDEE” button is for these type of users.

Figure 3.10: Main Activity of Attendance Program



3.4.3 Manager Activity

This activity is for those users who want to take attendance. It gives three options to the user to choose. If a user is a teacher/instructor/professor “*SCHOOL ATTENDANCE*” button is for them, whereas for meeting managers, meeting holder or conference conductor has “*CASUAL MEETING*” button, and boss and employer has a button “*EMPLOYER*”. Figure 3.11 represents the screenshot of this Manager Activity.

3.4.4 Attendee Activity

Attendee activity is for those users who want to request for their take attendance. It gives three options same as in Manager activity, so that user chooses one according to the circumstances. It has three buttons as displayed in Figure 3.12. “*SCHOOL*

ATTENDANCE” button for students, *“CASUAL MEETING”* button for meeting attendees or conference audience, and *“EMPLOYEE”* for an employee of a company or firm.

Figure 3.11: Manager Activity

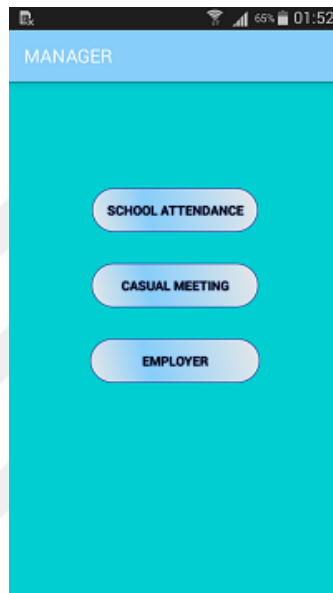
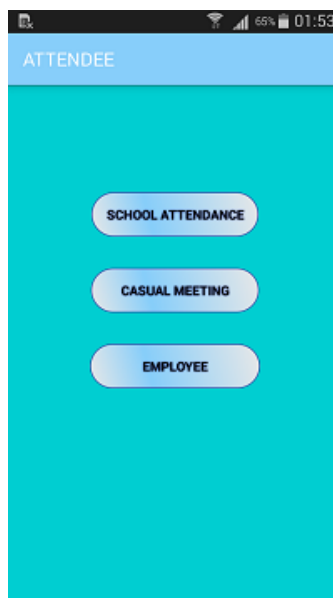


Figure 3.12: Attendee Activity



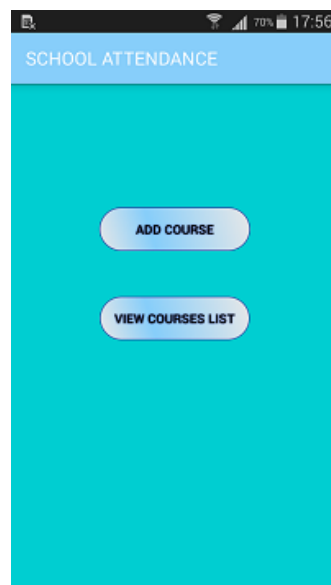
3.4.5 School Attendance Scenario

This part of sub-section explains the school attendance scenario that how a teacher can add courses and import students list in different course to take their attendance, and on Attendee side how a student can add course and instructor name in order to connect with teacher’s device over hotspot Wi-Fi connection to request for attendance.

3.4.5.1 For teachers

If a user is a teacher, instructor, professor or a person who wants to take attendance of the student, then “*MANAGER*” in “Main Activity” and “*SCHOOL ATTENDANCE*” in subsequent activity that is “Manager Activity”. Once teacher reached in “*SCHOOL ATTENDANCE*” activity of this attendance app, Teacher can add a course by clicking “*ADD COURSE*” button on this screen. But before all of this, teacher must have configured his/her dropbox account on the mobile device before adding a course, and must have placed the students list in this dropbox account before reaching out to this step. If teacher wants to see the list of added courses, he/she can press “*VIEW COURSES LIST*” button to view the course. The screen layout of this activity is shown in Figure 3.13.

Figure 3.13: School Attendance (Manager Side)



3.4.5.1.1 Add course details

In this screen teacher will fill out the course details before importing the list of students. The course code and course name are entered by the teacher in this view, and course day is selected from the dropdown menu. It is also recommended to set the starting date of the course and its ending date with the help of provided calendar by clicking “*SET START DATE*” and “*SET END DATE*” buttons respectively like in Figure 3.15. Figure 3.14 shows the screenshot of Add Course Details Activity with sample data entered.

A teacher cannot proceed to import the students list before filling-in all the fields in this Add Course Details activity, otherwise it will prompt the teacher that all fields must be entered before proceeding further. Once all required fields are entered, and teacher presses “*Import Student List*” button, screen like Figure 3.16 appears on mobile device.

Figure 3.14: Add Course Details (Manager Side)

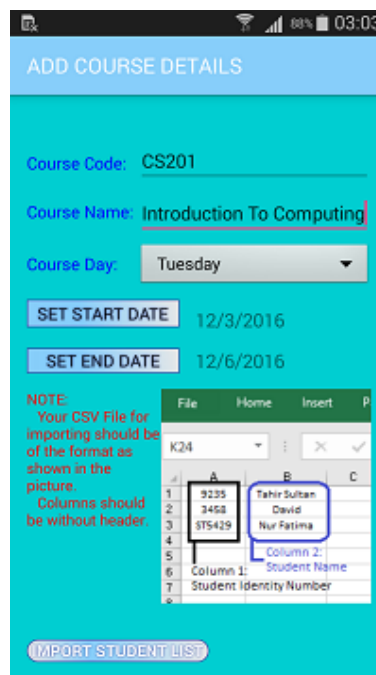


Figure 3.15: Set starting date of the course

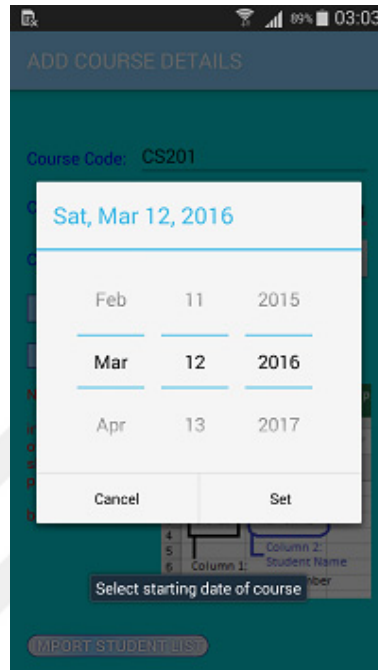
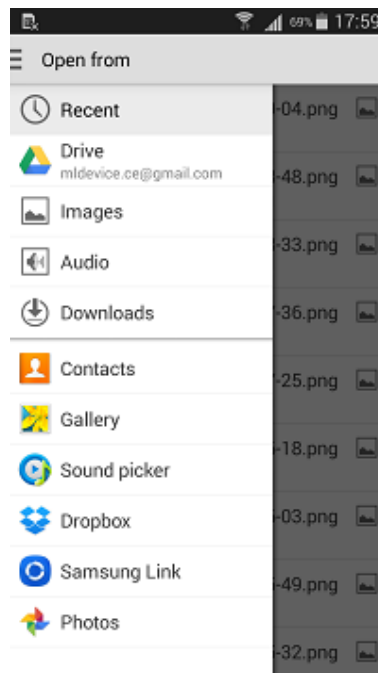


Figure 3.16: Import student list



Teacher must have configured and synchronized his/her dropbox account on the mobile device before reaching out to this step, and the students list file must be in CSV file format. Teacher can access the dropbox files from the menu. For example, teacher chooses “CSV1.csv” file as shown in Figure 3.17 as in this case. Sample file of students list to import is shown in Figure 3.18. First columns must be the Student IDs and the second column contains the full name of the student.

Figure 3.17: Select the desired file from dropbox

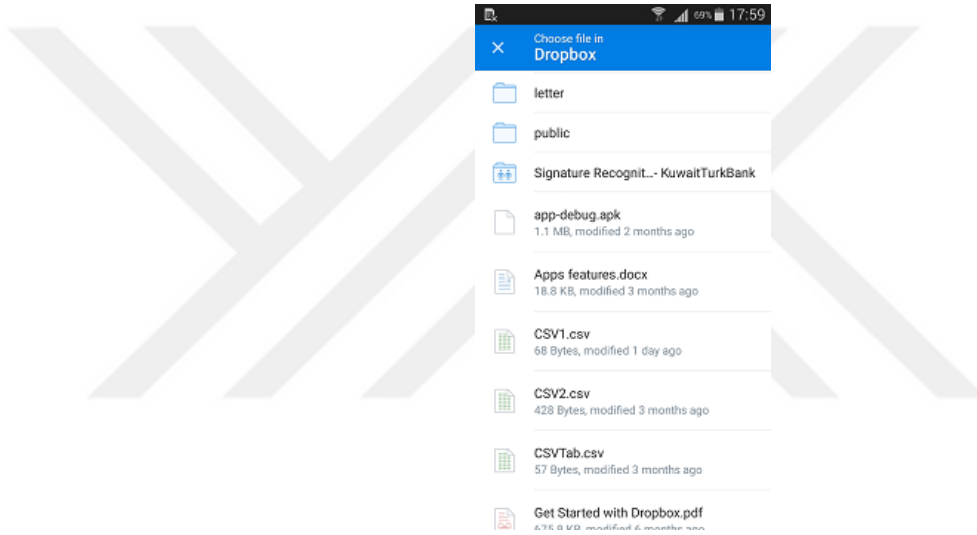
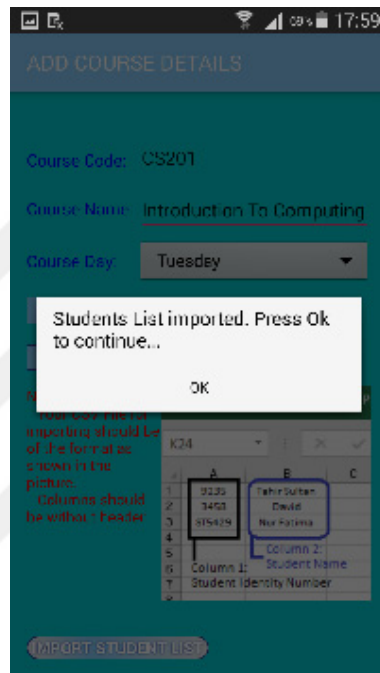


Figure 3.18: Sample Student List format to Import

	A	B	C	D	E	F	G	H
1	1357	Sara						
2	1240	Khalid						
3	1282	Nur						
4	1078	Hamza						
5	1067	Omer						
6	1462	Ayca						
7								
8								
9								
10								

If the selected file from the dropbox to import into Attendance app is in correct format, it will create the course in the local database of mobile device and adds the students in the created course by displaying the confirmation message as shown in Figure 3.19

Figure 3.19: Student list imported successfully, confirmation message



In the same way a teacher can create as many courses as desired, and these courses are displayed in the form of a list, where teacher can do further activities.

3.4.5.1.2 View courses list

If a teacher presses “VIEW COURSES LIST” button in the “SCHOOL ATTENDANCE” activity. This shows all the courses added by this teacher in a list pattern like Figure 3.20. By clicking on the name of desired course from the list, the teacher can further do the operations like taking automatic attendance, mark manual attendance, view students who are registered in this course, view the attendance record of the students and export the attendance sheet of the selected course.

3.4.5.1.3 Delete course

If a teacher wants to delete any course from the database, it's very simple and one step action. The teacher will just click long (press and hold) on the name of the course from the list shown in the “*View Courses List*” activity, and it will take the teacher to delete course activity as shown in Figure 3.21, the teacher wants to delete “*Math*” course by entering *Course Code* as “*MT209*”. Or a teacher can use “*DELETE COURSE*” button in the “*View Courses List*” activity to manually enter the course name for deletion. After deletion the list will be courses list will be updated on the screen and looks like Figure 3.22

Figure 3.20: View courses list

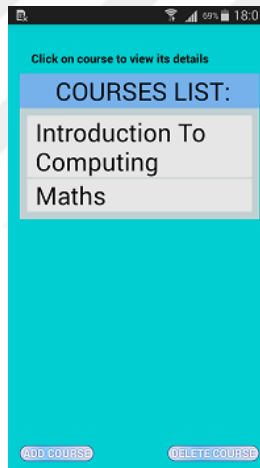


Figure 3.21: Delete Course

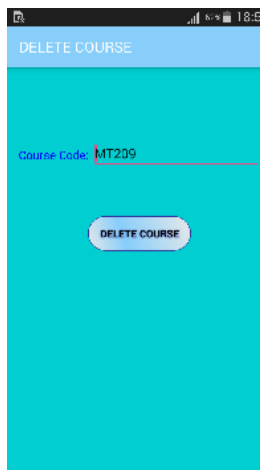
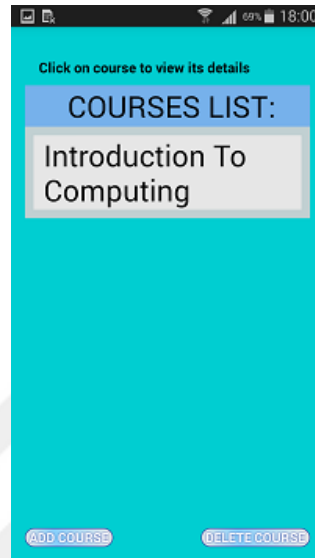


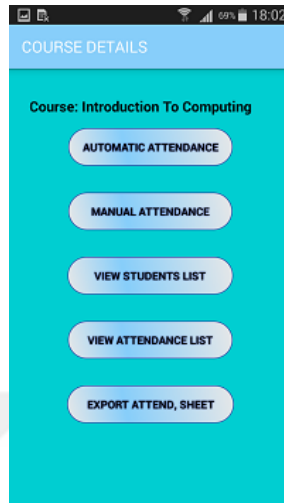
Figure 3.22: Updated view courses list after course deletion



3.4.5.1.4 Course selected

Any course can be selected from the list of courses, as an example the teacher pressed on “*Introduction To Computing*” course and the resulting screen looks like Figure 3.23. On the top, it shows the name of the course selected and below it will provide the buttons “*AUTOMATIC ATTENDANCE*” button to start the automatic attendance process, “*MANUAL ATTENDANCE*” button in case teacher wants to enter attendance of any student manually, “*VIEW STUDENTS LIST*” button to view the registered students and add or delete student, “*VIEW ATTENDANCE*” button to view the attendance record of this course and “*EXPORT ATTENDANCE SHEET*” button to export the attendance record in a sheet to teacher’s email address.

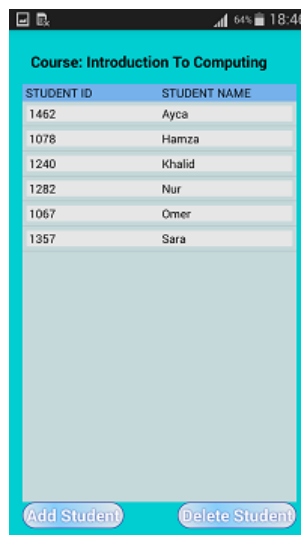
Figure 3.23: Course selected



3.4.5.1.5 View students List

Before proceeding to *Automatic Attendance* activity, it is better to look at *VIEW STUDENTS LIST* activity. In this activity teacher can view the list of registered (added) students, can add or delete student. Figure 3.24 shows the list of registered students in course “*Introduction To Computing*” (course selected in previous activity).

Figure 3.24: View students list of selected course



3.4.5.1.6 Add student

This option is for a scenario such as one student joins the course in the middle of the semester or year. By the help of this activity the teacher can add the student manually by entering student's name and ID. For example, teacher enters student name as "David Beckham" and student ID as "69010", as shown in Figure 2.25. The app will update the students list database and the new list of students in "Introduction To Computing" course is like Figure 2.26.

Figure 3.25: Add student example

ADD STUDENT

Course: Introduction To Computing

Student ID: 69010

Student Name: David Beckham

ADD STUDENT

Figure 3.26: Updated view students list after adding student

Course: Introduction To Computing

STUDENT ID	STUDENT NAME
1462	Ayca
69010	David Beckham
1078	Hamza
1240	Khalid
1282	Nur
1067	Omer
1357	Sara

Add Student Delete Student

3.4.5.1.7 Delete student

If a teacher wants to delete the record of student along with student details, he/she can press the “Delete Student” button or click long (press and hold) on the student name in the list of *VIEW STUDENTS LIST*. New screen appears which looks like Figure 3.27. For example, in this screen teacher wants to delete the record of student named “Hamza”, teacher will just press on Hamza name in the list from *VIEW STUDENTS LIST* activity and it will get the Hamza’s ID automatically, as shown in Figure 3.27. After deletion of this student the new students list is shown in Figure 3.28

Figure 3.27: Delete student

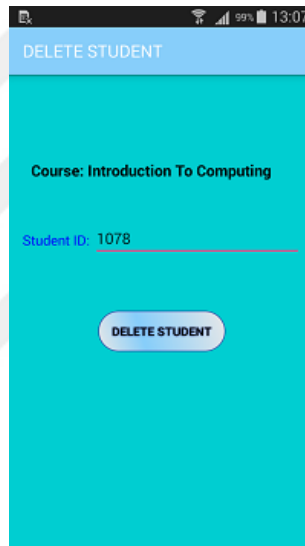


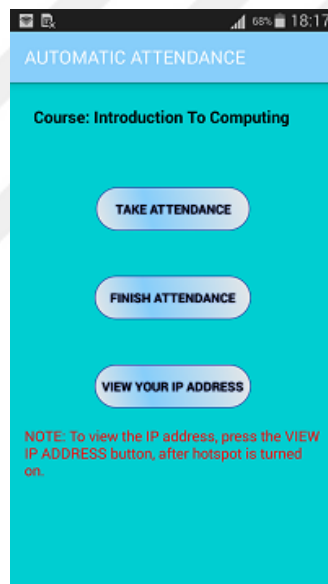
Figure 3.28: Updated view students list after deleting student



3.4.5.1.8 Automatic attendance

This activity will turn-on the user’s device Wi-Fi Hotspot and will wait for the user to press “*TAKE ATTENDANCE*” button. The mobile device screen is like Figure 3.29. Once the user presses the “*TAKE ATTENDANCE*” button, this button disappears and the mobile device opens the data communication channel in order to allow the attendees (in this case students) to send their attendance request. Figure 3.30 represents the mobile device screen state in this condition.

Figure 3.29: Automatic Attendance



If any attendee (student, in this case) sends an attendance request, screen displays attendee’s ID (student’s ID, in this case. In case of casual meeting, it will display meeting attendee name. And in case of Employer-Employee relation it will display employee’s ID), as shown in Figure 3.31 that teacher receives message from student whose Student ID is “1462”.

Figure 3.30: Take Attendance

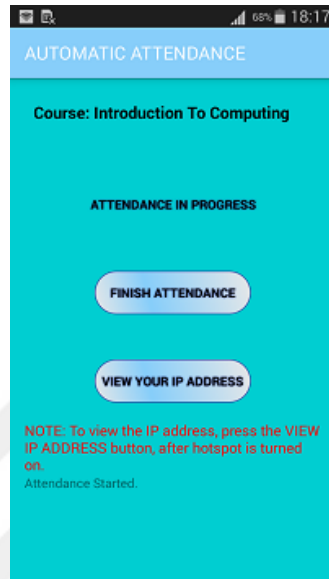
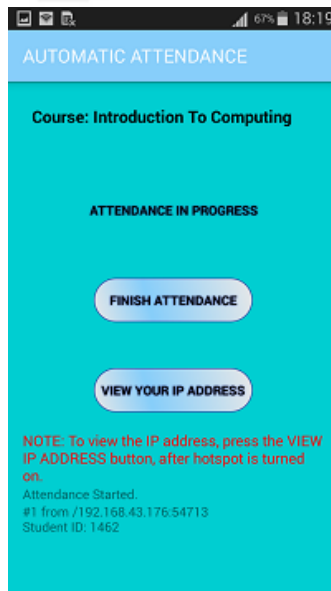


Figure 3.31: Attendee's attendance request received



In the same way this process keeps going on unless the lecture finishes, or the manager (teacher, in this case) presses the “*FINISH ATTENDANCE*” button. While in process of “*TAKE ATTENDANCE*” the app will maintain a temporary text file and saves the

received messages in it. Once the “*FINISH ATTENDANCE*” button is pressed or attendance process finishes anyhow, the app will read the temporary text file and update the attendance of the attendees (students/meeting-attendees/employees) accordingly.

There is one button “*VIEW YOUR IP ADDRESS*”, it is to show the IP address of the device on his/her screen. This Internet Protocol (IP) address must be provided to each attendee who is going to connect with manager’s device. Usually this IP address is “192.168.43.1”.

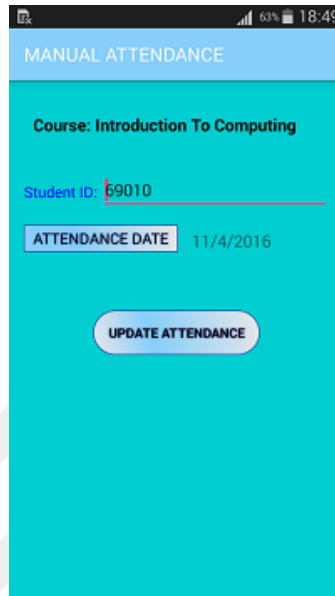
The manager (teacher in this case) cannot go back from this screen unless finishes the attendance process, but can minimize the app to do other work, while attendance process keeps running in the back.

3.4.5.1.9 *Manual attendance*

This option is useful in the condition like attendee (student) forgot mobile phone or mobile phone is not working. Student can request teacher to mark attendance manually. Teacher enters the student ID and selects the date of which attendance has to be marked, and press “*UPDATE ATTENDANCE*”, as shown in Figure 3.32 the teacher enters “69010” as Student ID and presses the button. The app will check whether the student ID exists in the system or not. If it exists, it updates the attendance record, else it shows an Error.

In case teacher forgot the student ID, then he can simply press the student name in the list of “*VIEW STUDENTS LIST*” activity, which also initiates the *Manual Attendance* activity and then the teacher does not need to enter the student ID for manual attendance.

Figure 3.32: Manual attendance

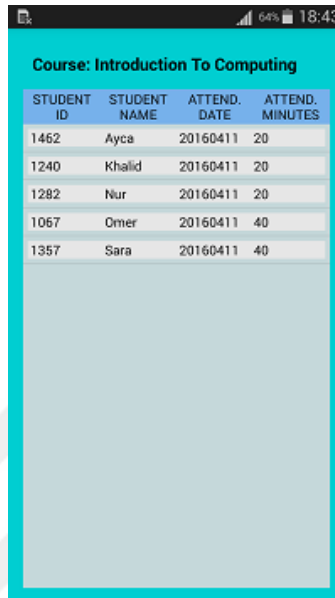


3.4.5.1.10 View attendance

In this view the teacher can see the attendance taken of the selected course till now. For example, if a teacher wants to see the attendance of course “Introduction To Computing” the screen will look like Figure 3.33. It displays the registered student’s names along with the attendance dates and time the student remains during the lecture of each day.

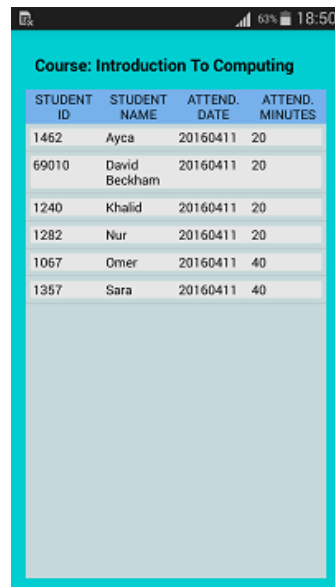
View in Figure 3.33 is the before marking the manual attendance of a student with ID “69010”, as described in the previous sub-section the teacher marks the manual attendance of one student. After marking the attendance, the app will update the record and the attendance record will look like Figure 3.34

Figure 3.33: View attendance record



STUDENT ID	STUDENT NAME	ATTEND. DATE	ATTEND. MINUTES
1462	Ayca	20160411	20
1240	Khalid	20160411	20
1282	Nur	20160411	20
1067	Omer	20160411	40
1357	Sara	20160411	40

Figure 3.34: View attendance record after marking attendance manually



STUDENT ID	STUDENT NAME	ATTEND. DATE	ATTEND. MINUTES
1462	Ayca	20160411	20
69010	David Beckham	20160411	20
1240	Khalid	20160411	20
1282	Nur	20160411	20
1067	Omer	20160411	40
1357	Sara	20160411	40

Above screen clearly shows that 4 students sat 20 minutes in the class while 2 students remained 40 minutes during the lecture on date “11-04-2016”.

3.4.5.1.11 Export attendance sheet

Exporting the attendance to someone or own email address is just one click away. The manager (in this scenario, teacher) clicks on “*EXPORT ATTEND. SHEET*” button and a process starts like shown in Figure 3.35. Attendance app generates a CSV file containing the attendance record of the attendees (students), asks the user to choose the email service through which he/she want to send an email. For example, in this case teacher wants to export the attendance record of course “*Introduction To Computing*”, he pressed on “*EXPORT ATTEND. SHEET*” and chooses *Gmail* as emailing service, as Figure 3.36 shows.

Figure 3.35: Choose email service for exporting attendance file

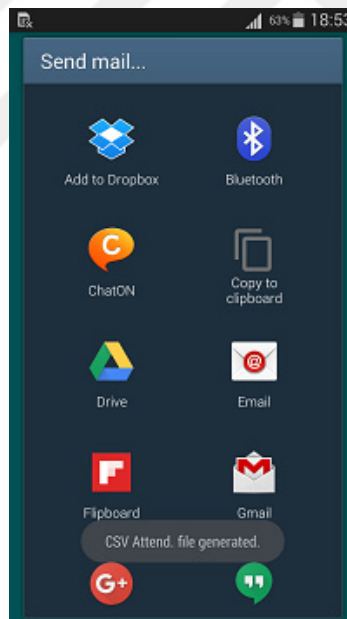
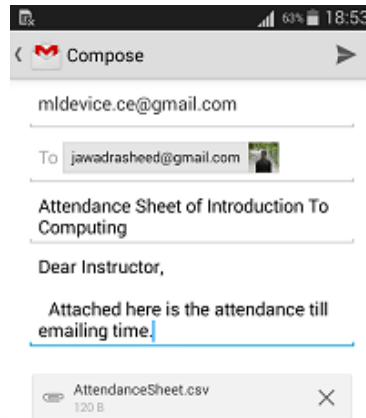


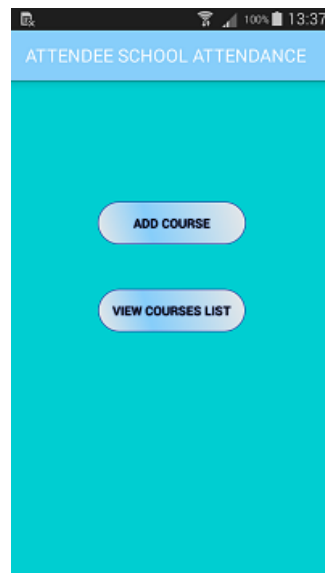
Figure 3.36: Send attendance sheet to email



3.4.5.2 For students

If a user is a student or a person who wants to request teacher for attendance, then student should use “*ATTENDEE*” in “*Main Activity*” and “*SCHOOL ATTENDANCE*” in subsequent activity that is “*Attendee Activity*”. Once student reaches in “*SCHOOL ATTENDANCE*” activity of this attendance app, as shown in Figure 3.37, student can add a course by clicking “*ADD COURSE*” button on this screen and view the course he/she added by clicking “*VIEW COURSES LIST*”.

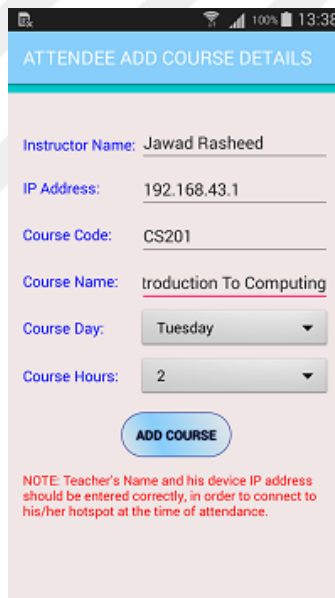
Figure 3.37: Attendee School Attendance



3.4.5.2.1 Add course details

In this screen student fills out the course details and instructor's name and instructor's device IP address, to create a course in order to send request for its attendance. The student should fill-in the “*Instructor Name*” field very carefully as this is the device hotspot name through which student connects with the instructor. As an example, suppose students want to add a course “*Introduction To Computing*”, which is of two hours duration and course took place on Tuesday by instructor “*Jawad Rasheed*”, as shown in Figure 3.38.

Figure 3.38: Add course details (Attendee Side)



ATTENDEE ADD COURSE DETAILS

Instructor Name: Jawad Rasheed

IP Address: 192.168.43.1

Course Code: CS201

Course Name: Introduction To Computing

Course Day: Tuesday

Course Hours: 2

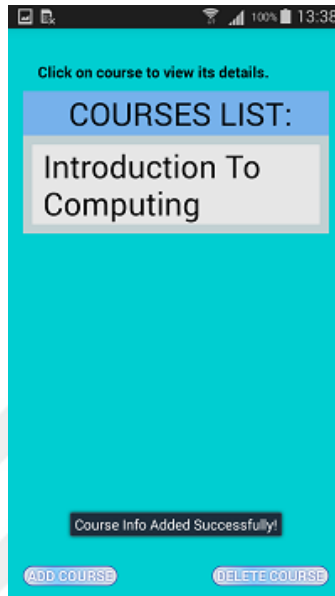
ADD COURSE

NOTE: Teacher's Name and his device IP address should be entered correctly, in order to connect to his/her hotspot at the time of attendance.

3.4.5.2.2 View courses list (Attendee side)

If a student presses “*VIEW COURSES LIST*” button in the “*SCHOOL ATTENDANCE*” activity. This shows all the courses added by the student in a list pattern like Figure 3.39. By clicking on the name of desired course from the list, the student requests the teacher for attendance.

Figure 3.39: View courses list added by a student



3.4.5.2.3 Delete course

A student can delete course in the same way as described in subsection 3.4.5.1.3 of this chapter.

3.4.5.2.4 Request to mark attendance

After the manager (teacher) turns-on “*TAKE ATTENDANCE*” process, attendee (student) can request to mark attendance by just clicking on the name of the course. It will take some time to first connect with the teacher’s device hotspot and after connecting, the student’s device send student’s ID automatically to the teacher’s device. The teacher will record the student’s ID and sends back the confirmation message. After getting the confirmation the student screen looks like Figure 3.40. For example, a student (whose name is Nur and ID is 1282) pressed on “Introduction To Computing” course in the list, the screen shown in Figure 3.40 while connecting with the Hotspot of the Teacher. After establishing the connection screen like Figure 3.41 appears which shows the confirmation message (including student’s ID) sent by the teacher, otherwise it will show the error if occurred while communicating with the teacher’s device.

Figure 3.40: Attendee connecting with manger's device hotspot

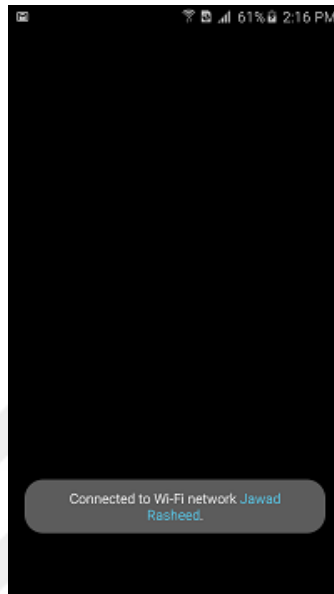
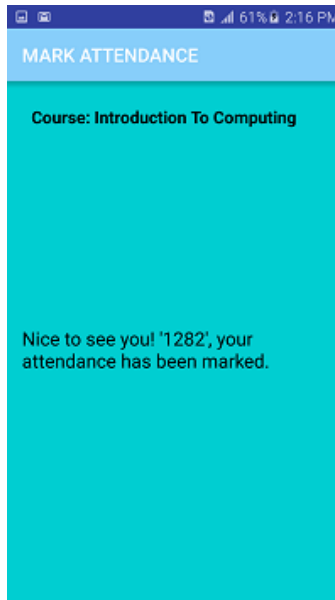


Figure 3.41: Attendance confirmation message



Right after getting the confirmation message from the teacher's device, the student device disconnects from teacher's device hotspot and a background service will start this process again after every 20 minutes till the lecture ends.

3.4.6 Casual Meetings Scenario

This sub-section explains the attendance process for casual meetings scenario that how a meeting manager/conductor can add a meeting and take its audience attendance automatically or manually, and on Attendee side how a meeting attendee can connect with the meeting manager to request for attendance over hotspot Wi-Fi connection.

3.4.6.1 For meeting manager or conductor

If a user is a meeting manager or conductor and wants to take attendance of the meeting attendee, then he/she should press "*MANAGER*" in "Main Activity" and "*CASUAL MEETINGS*" in subsequent activity that is "Manager Activity". Once meeting manager reached in "*CASUAL MEETINGS*" activity of this attendance app, he/she can add a meeting by clicking "*ADD MEETING*" button on this screen, and to view the meetings list, there is button "*VIEW MEETINGS LIST*" in this activity, as shown in Figure 3.42.

3.4.6.1.1 Add casual meeting

This screen is for meeting manager to add meeting by entering the meeting name and selecting the its date from the calendar view. By default, the meeting name is set on the current date, but manager can change it, if he/she wants, as shown in Figure 3.43

Figure 3.42: Casual meetings

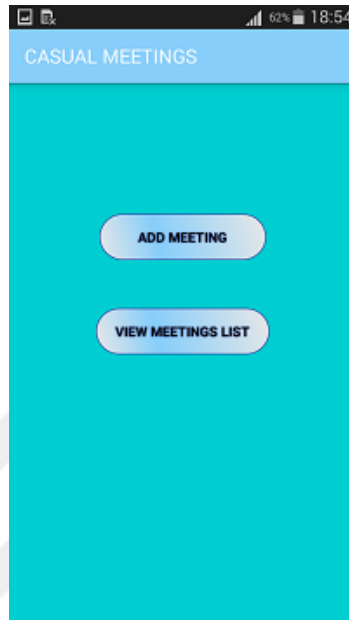
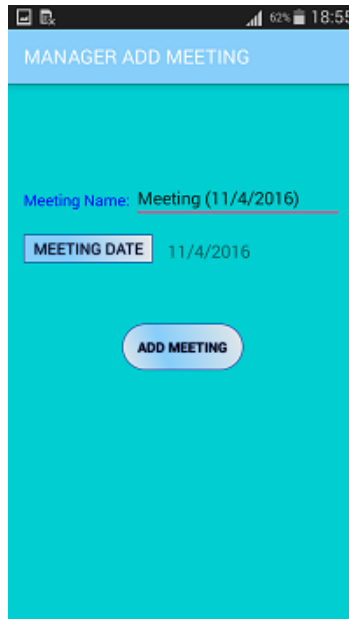


Figure 3.43: Add casual meeting details (Manager Side)



3.4.6.1.2 *View meetings list*

If a meeting manager presses “*VIEW MEETINGS LIST*” button in the “*CASUAL MEETINGS*” activity. This shows all the meetings added by this meeting manager in a list pattern like Figure 3.44, for example manager created a meeting with name “*Meeting (11/04/2016)*”. By clicking on the name of desired meeting from the list, the manager can further do the operations like taking automatic attendance, mark manual attendance, view the attendance record of the meeting attendees and export the attendance sheet of the selected meeting.

3.4.6.1.3 *Delete casual meeting*

Casual meetings are deleted in the same way as deleting the course defined in section 3.4.5.1.3 of this chapter. The manager will click long (press and hold) on the meeting name shown in the list to delete it. Screen like Figure 3.45 appears.

Figure 3.44: View casual meetings list (Manager side)

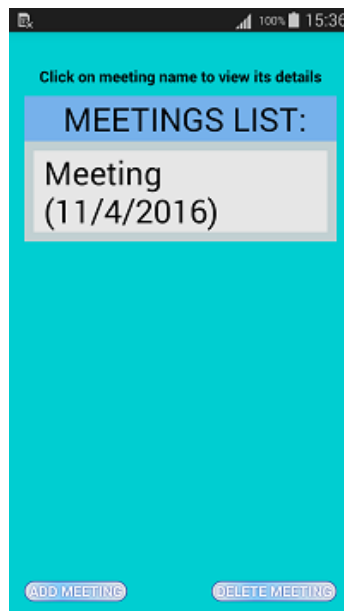
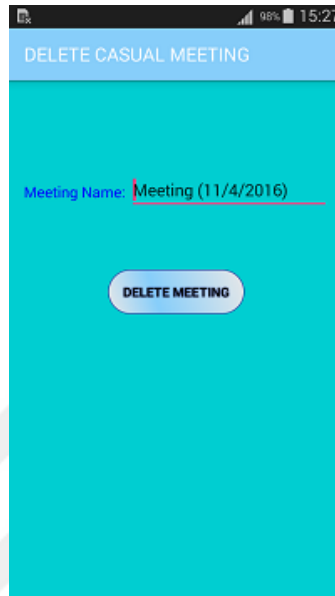


Figure 3.45: Delete Casual Meeting (Manager side)



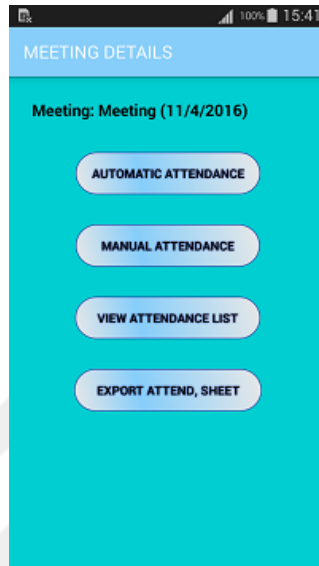
3.4.6.1.4 Meeting selected

Any meeting is selected from the list of meetings, as an example the manager pressed on “*Meeting (11/04/2016)*” meeting and the resulting screen looks like Figure 3.46. On the top, it shows the name of the meeting selected and below it will provide the buttons “*AUTOMATIC ATTENDANCE*” button to start the automatic attendance process, “*MANUAL ATTENDANCE*” button in case manager wants to enter attendance of any attendee manually, “*VIEW ATTENDANCE*” button to view the attendance record of this meeting and “*EXPORT ATTENDANCE SHEET*” button to export the attendance record in a sheet to manager’s email address.

3.4.6.1.5 Automatic attendance

Works same as automatic attendance described in section 3.4.5.1.8 of this chapter, except that it receives the meeting attendee name instead of ID and update the attendance record of the selected meeting when “*TAKE ATTENDANCE*” process finishes.

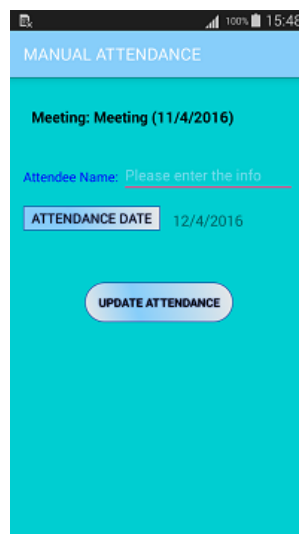
Figure 3.46: Meeting selected



3.4.6.1.6 Manual attendance

Works same as manual attendance described in section 3.4.5.1.9 of this chapter, except that it receives the meeting attendee name to mark the attendance as shown in Figure 3.47

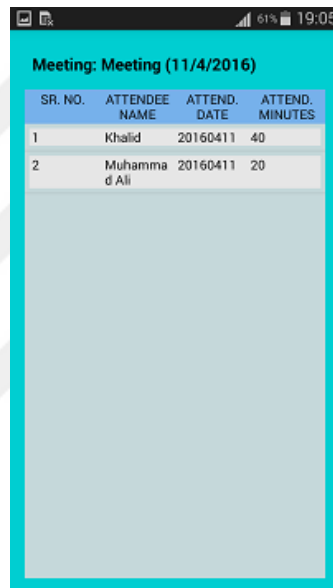
Figure 3.47: Manual attendance of casual meeting attendee



3.4.6.1.7 *View attendance*

View attendance is same as view attendance in school attendance scenario in section 3.4.5.1.10. It shows the attendance of all the meeting attendees who attended the meeting and also displays the presence time of each attendee, as shown in Figure 3.48.

Figure 3.48: View attendance of casual meeting attendees



SR. NO.	ATTENDEE NAME	ATTEND. DATE	ATTEND. MINUTES
1	Khalid	20160411	40
2	Muhamma d Ali	20160411	20

3.4.6.1.8 *Export attendance sheet*

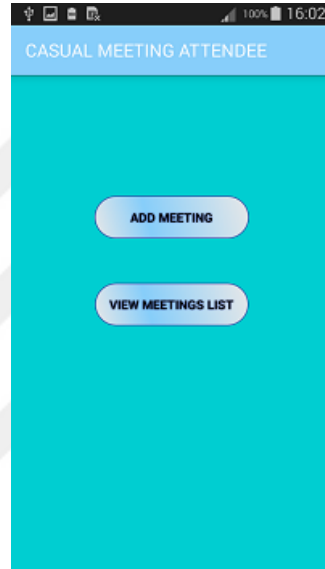
Export attendance sheet works in the same as export attendance described in section 3.4.5.1.11 of this chapter.

3.4.6.2 **For meeting attendee**

If a user is a meeting attendee or conference audience who wants to request meeting manager for attendance, then user should use “*ATTENDEE*” in “*Main Activity*” and

“CASUAL MEETING” in subsequent activity that is “Attendee Activity”. Once attendee reaches in “CASUAL MEETING” activity of this attendance app, as shown in Figure 3.49, attendee can add a meeting by clicking “ADD MEETING” button on the screen and view the meeting he/she added by clicking “VIEW MEETINGS LIST”.

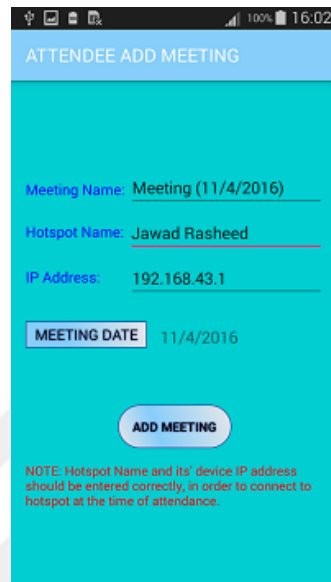
Figure 3.49: Casual meeting attendee



3.4.6.2.1 Add casual meeting details

In this screen meeting attendee fills out the meeting details and meeting’s device hotspot name and device IP address, to create a meeting in order to send request for its attendance. The attendee should fill-in the “Hotspot Name” field very carefully as this is the device hotspot name through which attendee connects with the manager. As an example, suppose attendee want to add a meeting “Meeting (11/04/2014)”, and hotspot name as “Jawad Rasheed”, as shown in Figure 3.50

Figure 3.50: Add casual meeting attendee



ATTENDEE ADD MEETING

Meeting Name: Meeting (11/4/2016)

Hotspot Name: Jawad Rasheed

IP Address: 192.168.43.1

MEETING DATE 11/4/2016

ADD MEETING

NOTE: Hotspot Name and its device IP address should be entered correctly, in order to connect to hotspot at the time of attendance.

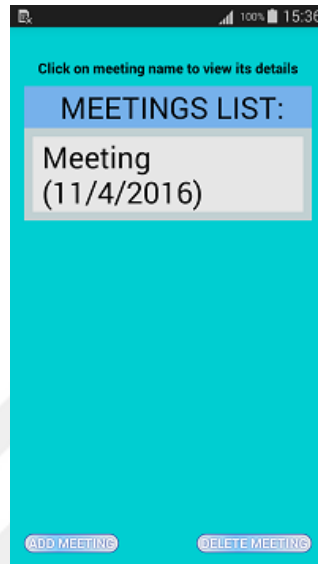
3.4.6.2.2 View meetings list (Attendee side)

If an attendee presses “*VIEW MEETINGS LIST*” button in the “*CASUAL MEETING*” activity. This shows all the meetings added by the attendee in a list pattern like Figure 3.51. By clicking on the name of desired meeting from the list, the attendee requests the manager for attendance.

3.4.6.2.3 Delete casual meeting

Works in the same way as described in section 3.4.6.1.3 of this chapter. By long click (press and hold) on the name of the meeting.

Figure 3.51: View casual meetings list (Attendee side)



3.4.6.2.4 Request to mark attendance

After the manager turns-on “*TAKE ATTENDANCE*” process, attendee can request to mark attendance by just clicking on the name of the meeting. It will take some time to first connect with the manager’s device hotspot and after connecting, the attendee’s device sends attendee’s name automatically to the manager’s device. The manager will record the attendees name and sends back the confirmation message. After getting the confirmation the attendee’s device will disconnect from the Wi-Fi of manager device and wait for next 20 minutes. Right after getting the confirmation message from the manager’s device, the attendee’s device disconnects from manager’s device hotspot and a background service will start this process again after every 20 minutes till the meeting ends. This process is same as described in section 3.4.5.2.4 of this chapter.

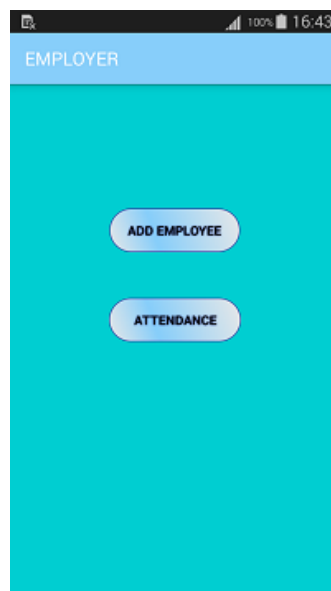
3.4.7 Business Environment Scenario

This sub-section explains the attendance process for business environment such as firms, organizations or company. It allows the employer to add the employees and take the attendance automatically by keeping the Check-In and Check-Out time of each employee over hotspot Wi-Fi connection.

3.4.7.1 For employer

If a user is an employer who is responsible of taking the attendance in a firm, organization or company then he/she should press “*MANAGER*” in “Main Activity” and “*EMPOLOYER*” in subsequent activity that is “Manager Activity”. Once employer reached in “*EMPOLOYER*” activity of this attendance app, he/she can add an employee by clicking “*ADD EMPLOYER*” button on this screen, and to view the employees’ attendance detail there is button “*ATTENDANCE*” in this activity, as shown in Figure 3.52.

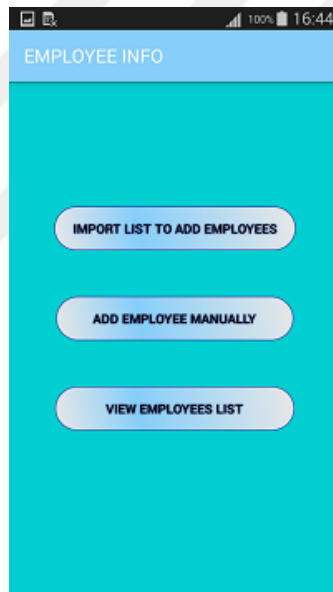
Figure 3.52: Employer activity



3.4.7.1.1 Add employee by importing list

In this activity an employer imports the list of employees working in the firm or company by one button “*IMPORT LIST TO ADD EMPLOYEES*”. Importing of Employees list is same as import of students list described the section 3.4.5.1.1. The format of the CSV file is also the same, the first column must contain the *Employee ID* and second column should be of *Employee Name*. Add employee screen looks like Figure 3.53.

Figure 3.53: Add employee and employee info



3.4.7.1.2 Add employee manually

An employer can also add employee one by one, without import the list of employees. Employer will add the employee name and employee ID here. For example, the employer enters “Hassan Nawaz” as employee name and “emp203” as employee ID in the Figure 3.54. After pressing “*Add Employee*” button the app will update the employees list database.

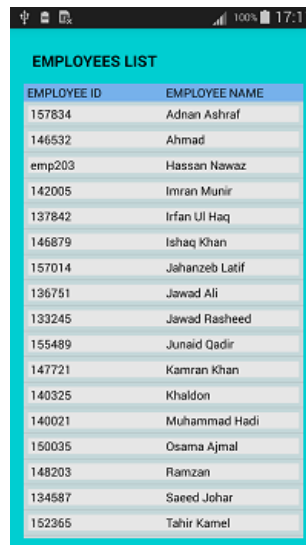
Figure 3.54: Add employee manually



3.4.7.1.3 *View employees list*

An employer can view the list of employees working in the firm or organization by clicking on “VIEW EMPLOOYES LIST”. As Figure 3.55 shows, that employee named “Hassan Nawaz” is added in the database as a result of adding employee manually.

Figure 3.55: View employee list

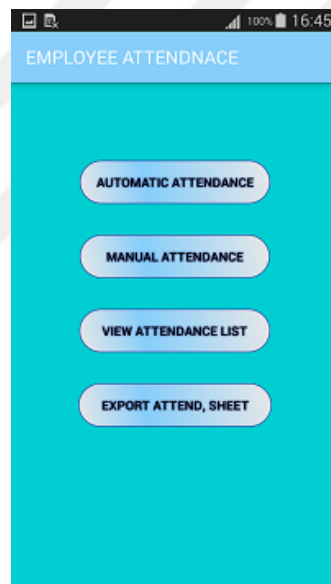


EMPLOYEE ID	EMPLOYEE NAME
157834	Adnan Ashraf
146532	Ahmad
emp203	Hassan Nawaz
142005	Imran Munir
137842	Irfan Ul Haq
146879	Ishaq Khan
157014	Jehanzeb Latif
136751	Jawad Ali
133245	Jawad Rasheed
155489	Junaid Qadir
147721	Kamran Khan
140325	Khalid
140021	Muhammad Hadi
150035	Osama Ajmal
148203	Ramzan
134587	Saeed Johar
152365	Tahir Kamel

3.4.7.1.4 *Employees attendance*

In this section of *EMPLOYER*, an employer takes the automatic attendance with the help of “*AUTOMATIC ATTENDANCE*” button, mark manual attendance of employees with “*MANUAL ATTENDANCE*” button, view the attendance of employees of desired date by using “*VIEW ATTENDANCE*” button and also exports the attendance of all the employees in one file in an email attachment from “*EXPORT ATTEND. SHEET*” button as shown in Figure 3.56.

Figure 3.56: Employees attendance



3.4.7.1.5 *Automatic attendance*

Works same as automatic attendance described in section 3.4.5.1.8 of this chapter, except that it receives the employee id and update the attendance record of the selected employer when “*TAKE ATTENDANCE*” process finishes.

3.4.7.1.6 Manual attendance

To mark the attendance manually of an employee this app requires employee ID, working date, Check-In time and Check-Out time as shown in Figure 3.57. If employer wants to record the Check-In time of an employee, he/she should select the date and the time and press “*UPDATE CHECK-IN TIME*” and if wants to record the Check-Out time then select the date and check-out time and press “*UPDATE CHECK-OUT TIME*”. App will inform the employer after updating the required tables in the database. If employee’s check-in time is not recorded and employer wants to set check-out time, the app alerts the employer to update the employee check-in time first. On the other hand, if employer entered wrong employee ID, it will show error. This employee can also start this activity by clicking once on the employee name in the list of “*VIEW EMPLOYEES LIST*”

Figure 3.57: Employees attendance manually

The screenshot shows a mobile application interface for manual attendance tracking. The title bar is blue and reads "MANUAL ATTENDANCE". The main content area is teal. At the top, there is a label "Employee ID" followed by a text input field containing "Enter employee ID" with a red underline. Below this, there are two main sections. The first section has a label "ATTENDANCE DATE" with a value of "12/4/2016" and a "SET CHECK-IN TIME" label with a value of "16:46". Below these is a rounded button labeled "UPDATE CHECK IN TIME". The second section has a "SET CHECK-OUT TIME" label with a value of "16:46" and a rounded button labeled "UPDATE CHECK-OUT TIME". The status bar at the top shows signal strength, 100% battery, and the time 16:46.

3.4.7.1.7 View attendance

An employer can view the attendance of employees who worked on specific date, by selecting the date with the help of button on top right corner of the screen in the Figure 3.58. For example, employer wants to see the working hours of the employees who worked on March 11, 2016. Figure 3.59 shows how employer sets the date with the help of “ATTENDANCE DATE” button to view its attendance. After selecting the date, the app displays the list of employees, their check-in and check-out time and the hours they worked, as shown in Figure 3.60.

Figure 3.58: View employees' attendance

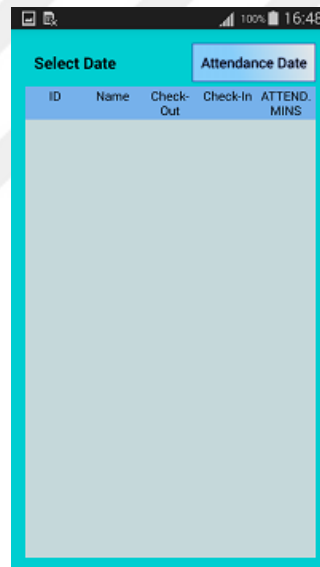


Figure 3.59: View employees' attendance set date

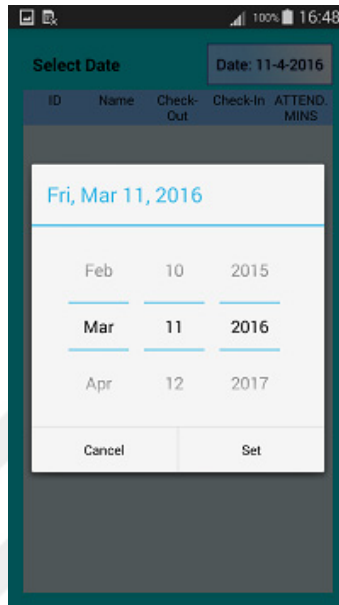
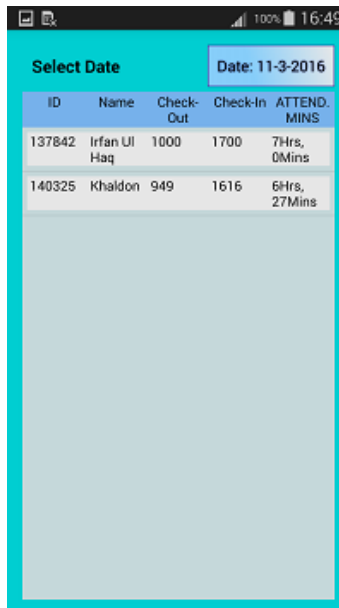


Figure 3.60: View employees' attendance of selected date



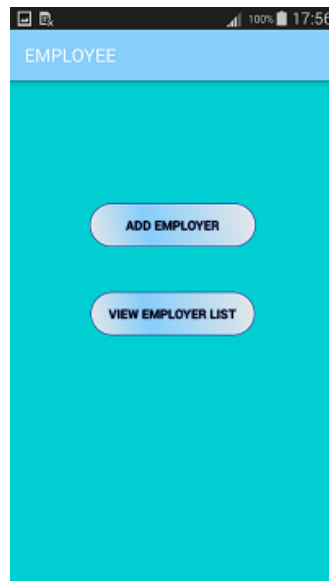
3.4.7.1.8 Export attendance sheet

Export attendance sheet works in the same as export attendance described in section 3.4.5.1.11 of this chapter.

3.4.7.2 For employee

If a user is an employee in any firm or company and wants to request employer for attendance, then user should use “*ATTENDEE*” in “*Main Activity*” and “*EMPLOYEE*” in subsequent activity that is “*Attendee Activity*”. Once attendee reaches in “*EMPLOYEE*” activity of this attendance app, as shown in Figure 3.61, attendee can add an employer by clicking “*ADD EMPLOYER*” button on the screen and view the list of employers he/she added by clicking “*VIEW EMPLOYERS LIST*”.

Figure 3.61: Employee activity



3.4.7.2.1 Add employer

In this screen employee fills out the employer details and employer’s device hotspot name and device IP address, to create a employer in order to send request for its attendance. The attendee should fill-in the “*Hotspot Name*” field very carefully as this is the device hotspot name through which employee connects with the employer. As an example, suppose attendee want to add a employer “*Systems Ltd*”, and hotspot name as “*Jawad Rasheed*”, as shown in Figure 3.62

Figure 3.62: Add employer

ADD EMPLOYER

Employer Name: Systems Ltd

Hotspot Name: Jawad Rasheed

IP Address: 192.168.43.1

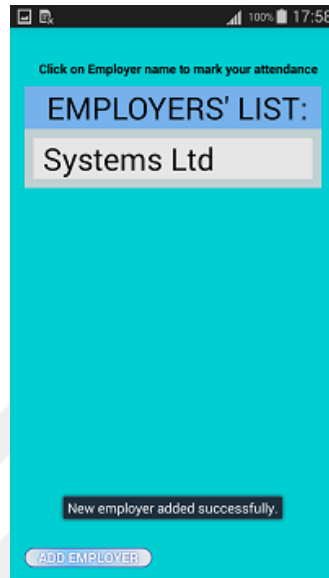
ADD EMPLOYER

NOTE: Hotspot Name and its device IP address should be entered correctly, in order to connect to hotspot at the time of attendance.

3.4.7.2.2 View employers list (Attendee side)

If an attendee presses “*VIEW EMPLOYERS LIST*” button in the “*EMPLOYEE*” activity. This shows all the employers added by the employee in a list pattern like Figure 3.63. By clicking on the name of desired employer from the list, the attendee requests the employer for attendance.

Figure 3.63: View employers list



3.4.7.2.3 Delete casual meeting

Works in the same way as described in section 3.4.6.1.3 of this chapter. By long click (press and hold) on the name of the employer.

3.4.7.2.4 Request to mark attendance

After the employer turns-on “*TAKE ATTENDANCE*” process, attendee can request to mark attendance by just clicking on the name of the meeting. It will take some time to connect with the manager’s device hotspot and after connecting, the employee’s device sends his/her employee ID automatically to the employer’s device. The employer will record the attendance and sends back the confirmation message. After getting the confirmation message the employee’s device will show the message on the screen and disconnect from the Wi-Fi hotspot of employee device. In the same way when an employee wants to leave the work place, he/she clicks on the employer name in the list to mark the Check-out time.

4. FINDINGS

This chapter states the finds of this research and results achieved after its implementation. It also compares this app with other apps in the market and states the difference of performance among these. Below mentioned finding and results are based on the experiments carried out on eleven (11) different android mobile devices with the attendance app installed.

- a. **No infrastructure required;** Many attendance apps in the android market such as “Hello Office” involves infrastructure like external router, a server or other thing to mark the attendance time. But this app does not require any external support for establishing a connection between the devices or storing the database.
- b. **Coverage area;** As per our experiments, this app work in a huge hall (20 meters long and 14 meters wide, with some furniture and persons who are carrying out the experiment in the hall). During the experiment the attendee stands in on corner and manager stands in the diagonally opposite corner, and the attendance was recorded successfully. By applying “Pythagorean Theorem”, the distance between Manager and Attendee at the time of attendance was approximately 24.5 meters.

Same experiment was carried out in open air, with no obstacle in between, and the maximum connectivity range at which this app worked was approximately 33 meters.

Previous app that takes attendance over Bluetooth connection, has a maximum range of 9 meters,

- c. **Number of connectivity with other device at one time;** Number of connections here means number of persons (attendees) who can connect with the manager device at one time to mark their attendance.

In this proposed and implemented data communication procedure, experiments showed that ten attendees can connect with the manager device at the same time, which is much less than the number of connections at in “Hello Office” (due to

the infrastructure involved), but it is higher than the attendance apps implemented with Bluetooth technology which handle five to seven connections at one time.

- d. **Flexibility;** Unlike other apps, the developed attendance app is not just for school attendance, it also has the communication procedure that incorporates the feature which allows business companies, firms, and conferences to use.
- e. **Ease to use;** The implemented app is easy to use, has a clean interface that requires less human interaction, which allows teacher and student to focus on their lecture instead spending time on attendance process. It makes the employee to worry less over the payroll related issue.

Table 4.1: Comparisons of Results with other attendance apps

	Android attendance apps (manual attendance only)	Android attendance apps (over Bluetooth connectivity)	Android attendance apps (with external Wi-Fi and server)	Attendance app (developed in this research)
Infrastructure Required	No	NO	Yes	No
Coverage Area (in meters)	Not applicable	< 9	0 – 40	0 – 24.5
Number of connections at one time (N)	Not applicable	5 – 7	Unlimited as compared to others	Max 10
Human Interaction	Much	Normal	Much Less	Less

5. CONCLUSION

The focus of the research reported in this thesis is on development of communication procedure for mobile application. The aim of the research is to achieve a better way for conducting human attendance via mobile device, for every field of life.

The important aspect of this research is that it optimizes the performance of taking attendance. This is achieved by introducing the Hotspot Wi-Fi idea and implementing it successfully. The research and its experiment shows that this app works over the coverage area of maximum 24.5 meters' radius, which is quite better than the apps with Bluetooth connectivity.

Even though the app developed in this research does not require any infrastructure and it can be used for general purpose. Unlike other cheap and common android apps, it is built in such a way that captures most of the market, from school attendance to employees' attendance.

Some more work is needed to enhance the layouts to introduce some visuals and graphical representation, that shows a real time update.

The app developed for this research is part of the property belongs to Biyodata Teknoloji San. Tic. Ltd. Şti. All the copy rights of this App belongs to Gökmen Altay on behalf of Biyadata Teknoloji.

REFERENCES

Books

Abrahamson, P. (2007). *Agile Software Development of Mobile Information Systems*. In *Advanced Information Systems* (pp. 1-4). Berlin: Springer

Felker, D. (2011). *Android Application Development for Dummies*. Hoboken, NJ. Wiley Publishing, Inc.

Hayes, I. S. (2003). *Just enough wireless computing*. Prentice Hall.

Meier, R. (2010). *Professional Android 2 Application Development*. Indianapolis, IN. Wiley Publishing, Inc

Periodicals

- Ahsan K., Shah H. & Kingston P., 2010. *RFID Applications: An Introductory and Exploratory Study*, IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 1, No. 3, January 2010, ISSN (Online): 1694-0784, Staffordshire University - UK
- Elgin, Ben (August 17, 2005). *Google Buys Android for Its Mobile Arsenal*. Bloomberg Businessweek. Bloomberg. Archived from the original on February 24, 2011. Retrieved March 16, 2016. In what could be a key move in its nascent wireless strategy, Google (GOOG) has quietly acquired startup Android, Inc., ...
- Mandal B. K., Bhattacharyya D. & Kim T., 2014. *An Architecture Design for Wireless Authentication Security in Bluetooth Network*. International Journal of Security and Its Applications Vol.8, No.3 (2014), pp.1-8
- Patel C., Trivedi S. & Patel S., 2012. *Biometrics in IRIS Technology: A survey*, International Journal of Scientific and Research Publications, Volume 2, Issue 1, January 2012. ISSN: 2250-3153
- Pawe T. Wojciechowski. N. P., *Language and Infrastructure Design for Mobile Computation*. University of Cambridge, June 2000.

Other Sources

- Activity 2012. [online]
<http://developer.android.com/reference/android/app/Activity.html> [accessed on January 28, 2016].
- Android 6.0 Changes. 2016. [online] <http://www.developer.android.com> [accessed on February 02, 2016]
- Bishop, T. (2012, March 6). *Google Play replaces Android Market, consolidates Google's media marketplaces*. [online] <http://www.geekwire.com/2012/google-playreplaces-android-market-consolidates-googles-media-marketplaces> [accessed on December 02, 2015]
- Developer Economics Q3 2013* analyst report. [accessed on December 11, 2015].
- Eddy, N. (2011, December 8). *Cloud Computing: Cloud, Mobile Apps, Public Storage Are Top IT Trends for 2012 and Beyond*: Gartner. [online] <http://www.eweek.com/c/a/Cloud-Computing/Cloud-Mobile-Apps-Public-Storage-Are-Top-IT-Trends-for-2012-and-Beyond-Gartner-130060/> [accessed on December 15, 2015]
- Google shows off new version of Android, announces 1 billion active monthly users. Techspot. [accessed on January 05, 2016]
- Systems Permission. 2016. [online].
<http://developer.android.com/guide/topics/security/permissions.html> [accessed on December 22, 2015].
- Vogel, L. (2010b). *Android Threads, Handlers and AsyncTask - Tutorial*. <http://www.vogella.de/articles/AndroidPerformance/article.html> [accessed February 03, 2016]