BIOMETRIC BASED ATTENDANCE SYSTEM

**Presented by**

Kangogo Kiprotich Joel 0791299638

*A System Documentation in Partial Fulfillment of the Requirements for the Award of the Degree of* Bachelor of Science in Information Technology of Kibabii University

July, 2019

# DECLARATION

This project is our own original work, that we developed and come up by:

Kangogo Kiprotich Joel

# CERTIFICATION

The undersigned certify that has read and hereby recommend for acceptance of Kibabii University a system project entitled: “Biometric Based Attendance System”

Prof.Franklin Wabwoba

Supervisor Sign Date

Mr. MacBenjamin Wanjala Technician Sign Date

Also among the supervisors is Ms. J Shikhuyu

# ACKNOWLEDGEMENT

First and foremost, I would like to express my sincere gratitude to Almighty God. I thank Him for all the wisdom and perseverance that He has bestowed upon me since the beginning of this project. I would like also to extend my heartiest thanks with a deep sense of appreciation and respect to Prof.Franklin Wabwoba and Mr. Jackson Ng’etich for their guidance during the project design.

I would also like to thank all individuals consulted for their unfailing cooperation and sparing their valuable time to assist me on my work.

# ABSTRACT

In our vast, inter-connected world, the need for reliable identity authentication techniques has become of paramount importance. And addressing this need is the emergence of biometrics. Biometrics refers to the science of establishing individuals’ identities based on their physical and behavioral traits such as fingerprints, face, iris, voice, and gait. Compared to traditional authentication schemes that are knowledge-based (passwords) or token-based (RSA tokens), biometric-based systems are considered more convenient and secure. Users don’t have to memorize passwords or possess proof of identity such as ID cards, and impostors can be deterred or detected with increased ease. Hence, biometric systems have been deployed in numerous commercial, civilian, and forensic applications to establish identity.

Boiled down to their fundamental essence, biometric-based recognition systems rely on the comparison of a digital representation of a physical or behavioral trait with a previously recorded one of the same trait. Thus, a biometric recognition system, or simply a biometric system, is a pattern recognition system that recognizes individuals based on their biometric traits.

For the case of this system, I was able to implement fingerprint biometric based attendance system to replace the old mechanical methods of capturing student class attendance i.e. the attendance sheets were signed by students with or without the course lecturer present in class, or the class representative may give the students to sign the sheets. This could pose a lot of problems.

This system is of great importance as the student will improve his/her academic performance and also generates reports that helps the lecturers analyze the attendance pattern and compute the required percentage for student to be allowed to sit for examination.

The last few years have also seen the development of biometric technology in the banking, retail and mobile phone sectors. Today, the biometric security is a growing industry but interestingly it is not a new science. Manual fingerprints recognition studies began as early as the end of the 19th Century and the origins of iris recognition dates back to 1936.

TABLE OF CONTENTS

[DECLARATION ii](#_Toc14875199)

[CERTIFICATION ii](#_Toc14875200)

[ACKNOWLEDGEMENT iii](#_Toc14875201)

[ABSTRACT iv](#_Toc14875202)

[List of Figures vii](#_Toc14875203)

[List of Tables viii](#_Toc14875204)

[CHAPTER ONE 1](#_Toc14875205)

[INTRODUCTION 1](#_Toc14875206)

[1.1 Background Information 1](#_Toc14875207)

[1.2 Problem Statement 2](#_Toc14875208)

[1.3 Purpose of the Project 2](#_Toc14875209)

[1.4 Scope 3](#_Toc14875210)

[1.5 Justification 3](#_Toc14875211)

[1.6 Feasibility Study 4](#_Toc14875212)

[1.7 Significance 4](#_Toc14875213)

[1.8 Deliverables 4](#_Toc14875214)

[1.8.1System design and document 4](#_Toc14875215)

[1.9 System Requirements 6](#_Toc14875216)

[Chapter 2 8](#_Toc14875217)

[Literature Review 8](#_Toc14875218)

[2.1 Introduction 8](#_Toc14875219)

[CHAPTER THREE 10](#_Toc14875220)

[METHODOLOGY 10](#_Toc14875221)

[3.1 Introduction 10](#_Toc14875222)

[3.2 Data Collection tools 10](#_Toc14875223)

[CHAPTER FOUR 12](#_Toc14875224)

[SYSTEM DESIGN 12](#_Toc14875225)

[4.1 Introduction 12](#_Toc14875226)

[4.2 Database design 17](#_Toc14875227)

[4.3 Interface design 19](#_Toc14875228)

[CHAPTER FIVE 21](#_Toc14875229)

[SYSTEM CONSTRUCTIONS 21](#_Toc14875230)

[5.1 Sample code for the system 21](#_Toc14875231)

[CHAPTER SIX 32](#_Toc14875232)

[SYSTEM TESTING 32](#_Toc14875233)

[6.1 Introduction 32](#_Toc14875234)

[6.2 Module integration and testing 32](#_Toc14875235)

[6.3 Training. 33](#_Toc14875236)

[6.4 Actual handover 33](#_Toc14875237)

[USER MANUAL 34](#_Toc14875238)

[Installation guide 34](#_Toc14875239)

[Navigation guide 34](#_Toc14875240)

[Procedure for data capture 34](#_Toc14875241)

[Conclusion 35](#_Toc14875242)

[Recommendation 35](#_Toc14875243)

[References 37](#_Toc14875244)

[List of Appendixes 39](#_Toc14875245)

[Appendix I: project schedule 39](#_Toc14875246)

[Appendix II: SDLC Phases 39](#_Toc14875247)

# LIST OF FIGURES

[Figure 1 System Overview 13](#_Toc14875281)

[Figure 2 Use case diagram for Admin 14](#_Toc14875282)

[Figure 3 Lecturer use case diagram 14](#_Toc14875283)

[Figure 4 Student use case diagram 15](#_Toc14875284)

[Figure 5 Implementation process 16](#_Toc14875285)

[Figure 6 Biometric recognition 17](#_Toc14875286)

[Figure 7 A ERD diagram showing relationship between database entities 18](#_Toc14875287)

[Figure 8 Sample interface design screenshots 20](#_Toc14875288)

[Figure 9 Admin-lecturer login interface testing 32](#_Toc14875289)

[Figure 10 Student registration error checking 33](#_Toc14875290)

[Figure 11 A diagram showing phases of system design 40](#_Toc14875291)

# LIST OF TABLES

[Table 1 Project schedule 39](#_Toc14875366)

# CHAPTER ONE

# INTRODUCTION

## Background Information

A biometric system is a technological system that uses information about a person (or other biological organism) to identify that person. Biometric systems rely on specific data about unique biological traits in order to work effectively. A biometric system will involve running data through algorithms for a particular result, usually related to a positive identification of a user or other individual.

The specific nature of today’s biometric systems is associated with a specific use of the term "biometrics." In general, biometrics is any use of biological data in technology. Biometric systems focusing exclusively on the identification of humans have become the major kind of biometric system in today’s IT world.  
  
Governments, businesses and organizations can use biometric systems to get more information about individuals or about a populace as a whole. Many biometric systems are developed for security applications. An airport scanning device, a "bio-password" system, or an internal data gathering protocol is an example of a biometric system that uses identifying data for a security result.

A lot of companies and organizations today are moving towards implementing the use of biometrics because of security reasons. Various companies like banks and Telecommunication firms have tried to use Voice as password which is biometric in nature. Other systems like ERP modules have not implemented the use of biometrics. Furthermore, almost all Universities have not used biometric in their systems and therefore we decided to develop a biometric based class attendance system to automate the current system that is mechanical and manual in nature.

In Kenya, no University has adopted the Biometric Class attendance system to keep track of bona fide students who have attended class.

It was a good idea to develop such kind of a system since usernames and passwords are prone to forgetting them, lack of data integrity and therefore there was a need that we incorporated biometric identification.

The business areas that the system will apply are various Universities, Colleges, Polytechnics to keep track of those students who attended class, anther area is in companies to keep track of employees who attended the place of work.

## 1.2 Problem Statement

The current system is a manual based system commonly implemented through paper. These are the problems of this current system

Lecturers having difficulties to discover who has been genuinely attending the classes and those who do not after keeping track on the information provided by the student in the attendance sheet.

Poor storage and data duplication where the class attendance sheets need to be stored in files and they can get lost through wear and tear hence is no data consistency and integrity.

Wastage of resources i.e. a lot of paper is used, time consuming while taking signatures.

## 1.3 Purpose of the Project

The purpose is to develop a biometric based class attendance system to keep track of bona fide students.

1.3.1 Objectives

Developer’s Level Objectives

To develop a back-end module

To develop a front-end module

To link the back-end and front-end modules

To test the system’s output if it is really the desired system that delivers the expected outcome to satisfy the user needs

## 1.4 Scope

The scope of this work is within the Kibabii University environment where development and testing was implemented. The development of Fingerprint Biometric Based Attendance System is aimed to improve how attendance management is done by using fingerprint as a form of authentication for proof of attending a class within the University setting.

The expected time frame for completion of this project was 14 weeks where I managed to create time during weekends and after class sessions to ensure that I complete my project within the stipulated timeframe.

Due to complexity of the biometric integration, I had to consult and seek more support from my supervisors and technicians for biometric fingerprint integration to the system

## 1.5 Justification

The new system is going to deal with the limitations of the existing system by; keeping historical data that makes it easy for lecturers to access and grade students, providing high level of security whereby making it impossible for imposters and impersonators in making their ways to examination halls through the use of a mobile application like and Android phone where lecturer can access everywhere as a unique way. The system will allow the lecturer to monitor each student attendance, track down truants and take the appropriate action and reduces the stress in queuing up which result in delay and often time in the damage of the attendance sheet. The new system will provide user friendly interface which will help to guide each user to use it correctly without any specialized training. The main benefit of the system is that it allows the university management to meet the policy of 75% class attendance to be enabled to sit for examinations where previously were not able to be met in using the traditional system. This system aids the quality assurance department to evaluate performance also the student performance will be improved because of fear of 75% class attendance policy.

## 1.6 Feasibility Study

**Hardware**

The smartphones were available during development and testing process, the main limitation is that the fingerprint scanner was not available which made it a bit challenge thereby the project was technically not feasible.

**Software**

The Integration Development Environment (IDE), Android Studio, was used used since it is an open source software.

## 1.7 Significance

The beneficiaries of this system will be both the lecturers and students. Since it has become an academic rule that a student must attend 75% of lectures in a course for a semester in order to be eligible to be evaluated in that course. To check the attendance percentage for each student and course by using this system it will be much easier. Hence, this system will capture the biometric fingerprint of students and use it to check attendance to lectures and other activities. It also ensures physical presence unlike signatures. The fingerprint device attendance taking method is an efficient method that needs no supervision of a principle. The three parts of the system have the same concept of fingerprint devices, which are fingerprint reader, control unit and store unit, so it can be used in corporate environments easily by some modifications in software.

## 1.8 Deliverables

### 1.8.1System design and document

Interface Design Document

This system consists of three main interface design parts in order to make the system more flexible and suitable to use at any institute based on the need. It has three parts; the admin part where he has all system privileges, the lecturer part that creates the session for the class and finally the student module where he/she provides the fingerprint in on order to attend class.

Database Design Document

For the centralization of data, SQLite database was used. The database is composed of a number of tables, which are used to store different groups of records required to manage the student details including attendance. Basically there are four relational tables specifically named as; ‘attendance’, ‘lecturer’, ‘attendancesession’ and ‘student’. The ‘lecturer’ table stores the staff’s information such as the staff username, password and first name, last name, address, contact while ‘attendance’ table stores the details of class attendance.

Students are requested initially to record their fingerprint for the registration process along with other details such as first name, last name registration number and contact. They have only to place their fingers on the fingerprint reader. Then, the system will automatically submit the name of the students along with date and time to the database. The system has a user-friendly interface for fingerprint enrollment and verification. The database provides the data elements expected in the data capture phase. However, the most important are fingerprint. After initial registration, the data which has provided can be used as a lecture attendance list. The system can generate a report concerning attendance of students to each course. The graphical user interface in the mobile device allows the system administrator to monitor and produce statically reports of the students' attendance status.

Tests and Results

The system is applied to Kibabii University, Faculty of SCAI. The faculty timetable of courses is used in this test. The system can review the presence of students in the form of statistics at the University level, department or field of a student. Results of such test evaluations helped detect any risk in the system and hence allowed both users and us to make a decision about choosing the fingerprint biometric system for implementation.

Data Capture

Generally, this system is made up of the data capture system (enrollment system). Students are requested initially to record their fingerprint during the registration process. They have only to place their fingers on the fingerprint reader. Then, the system will automatically submit the name of the students along with date and time to the database where student first name, last name and registration number is populated on the interface. This system has a user-friendly interface for fingerprint enrollment and verification. The SQLite database provides the data elements expected in the data capture phase. However, the most important is fingerprint. After initial registration, the data which will be provided can be used as a lecture attendance list.

Attendance Report

This report interface is accessed under the report menu, where daily attendance of all students can be generated. The attendance of each student per lesson is stored on the database and can be retrieved. The system will be designed to allow the database administrator to view the attendance report of each student as well as a summary report on semester basis.

## 1.9 System Requirements

Functional Requirements

Some of the functional requirements for this system are:

The system is able to register students

The system is able to list the registered students

The system is able to register lecturers

The system is able to list the registered lecturers

The system is able to show attendance per student

The system is able to create session to allow students make class attendance

Non-Functional requirements

The system does not allow unregistered students to attend class

The system computes the percentage of class attendance per student and decides whether he/she sit for examination

The system requires an internet connection.

Software Development tools

Software development tools are tools required to develop a complete fingerprint attendance authentication system including all the IDEs and the tools required to run the application.

The IDEs used in development of this system include;

Android studio - for the development of the android fingerprint authentication application

Hardware

The hardware is an android mobile phone, where android 6.0 and above is supported and a fingerprint scanner for extracting fingerprints. The software consists of Android Studio, the application program (front end) and the database.

Programming Languages

The application program is developed using Android Studio IDE where we developed a mobile application in Java programming language. The advantages of Android Studio IDE are its robustness and have excellent database connectivity

# CHAPTER TWO

# LITERATURE REVIEW

## 2.1 Introduction

Literature review involves looking at what other researchers have come up before and trying to find the success and their weaknesses in coming up with such systems. Our review and research was based on systems that have been developed through various technologies relating to class attendance management systems with the use of biometrics. We found the following reviews;

Benyo et al. (2012) [1] described the design and implementation of student attendance monitoring at Budapest University of Technology and Economics in n Hungary utilizing the Near-Field Communication. The system was implemented in a highly autonomous distributed environment comprising NFC enabled contactless terminals and a scalable back-office. The terminals also support biometric identification by fingerprint reading to enhance the security aspect. In the pilot project, more than 1000 students had tested the system and they become accustomed to use the system properly.

Bueno-Delgado et al. (2012) [2] presented the overview of implementation of an

Ubiquitous computing platform based on NFC known as Smart University in Technical University of Cartagena, Spain. The two major projects development were NFC attendance registering system and NFC administrative fee payment system. Prior to the development, data collection through opinion poll was conducted to study the

Impact of the use of NFC technology in a university environment. The results showed

Promising use of NFC applications among the university community.

Benyo et al. (2012) [3] have summarized uses of NFC applications in a contactless infrastructure at the Budapest University of Technology and Economics. These

Include enrolment at the University, registering attendance during lecture, registration of end-of-semester exam and access to University resources. These contactless services have shown to be beneficial for students and lecturers.

A more recent work was reported by Ichimura & Kamada (2013) [4]. In this

study, the authors describe the functionality of Attendance Management System

(AMS) build upon NFC technology. They have developed the AMS for the University and tested using Nexus 7 devices connected to each other via peer-to-peer network.

From the implementation, they found out that the number of absentees in nearly all classes was decreased and this could be due to students’ conscientiousness that they are being observed.

Fernández et al. (2013) [5] provide description of an attendance control system based on NFC technology developed and implemented at Pontifical University of

Salamanca, Madrid, Spain. The project was carried out as Final Degree Project and developed in collaboration with Samsung Electronics. The aim was to ensure continuous assessment so that lecturers’ teaching time will not be affected by the manual way in recording attendance. Results from the survey of the pilot project indicate that students and lecturers perceived high level of satisfaction and usefulness of the project.

Shen et al. (2014) [6] presented the most recent work. In this study, the authors proposed the equipped smart classroom system to automate attendance management, to locate students, and to provide real-time feedback to students. The positioning feature of the system shown to be very useful to users; particularly for large classes.

Also, narrowing down to Kenya, various and many systems are moving towards to integrating biometrics in their systems for instance in banks, telecommunication services like Safaricom has now adopted voice recognition commonly known “my voice my Password” which is biometrics in nature.

Also, some ERP modules have been integrated with biometrics and works to curb the security features arising from in place. We found that most Universities has not come up with such kind of systems, most of them still use the mechanical and manual based attendance systems where students sign their attendance on paper sheets. We therefore managed to come up with a Fingerprint Biometric-based Attendance System.

# CHAPTER THREE

# METHODOLOGY

## 3.1 Introduction

Methodologyis the systematic, theoretical analysis of the methods applied to a field of study. It comprises the theoretical analysis of the body of methods and principles associated with a branch of knowledge. It can also be a general research strategy that outlines the way in which research is to be undertaken and, among other things, identifies the methods to be used in it. It defines the means or modes of data collection or, sometimes, how a specific result is to be calculated.

## 3.2 Data Collection tools

The data collection tools that we applied to gather for data in the field include interviews, questioning protocol.

1. Interviews—This is where I interviewed fellow students from various departments and tried to collect the requirement so that I could add to the system.
2. Questioning protocol—I tried to ask lecturers if the system would ease their work of waiting and avoiding interruptions while teaching.

I managed to collect requirements that enabled me to develop a prototype version of our system. Therefore, top-down approach is adopted in designing the Biometrics Attendance System whereby I decided to choose Prototyping Model as this system development methodology in which a prototype (an early approximation of a final system) is built, tested, and then reworked as necessary until an acceptable prototype is finally achieved from which the complete system is developed. It is an iterative, trial-and-error process that takes place between the developers and the users.

Therefore, when the prototype version is reworked on and a version is released as increments which employs an incremental process modelling. The first prototype is modified, based on the comments supplied by the users, and a second prototype of the new system is constructed.

# CHAPTER FOUR

# SYSTEM DESIGN

## 4.1 Introduction

During design, decisions are made about how the problem will be solved, first at a high level, then at increasingly detailed levels. System design is the first stage in which the basic approach to solving the problem is selected. During system designing the overall structure and style are decided.

The system architecture is the overall organization of the system into components called system. System design deals with transforming the customer’s requirements, as described in the SRS document, into a form that is implementable using the programming language. Certain items such as modules, relationships among identified modules, data structures, relationships between the data structures, and algorithms for implementation should be designed during this phase.

As a system designer, I tried to take following design decisions:

Organize the system into modules

Organize sub-modules for each module

Choose an approach to manage data store

Handle access to global resources

Choose implementation logic

4.2 Use case diagrams

These are system module diagrams that describe the functionality of the system. It is made up of a set of possible sequences of interactions between the users and the system.

The figure below shows System Overview

Login

View student

Add lecturer

Add student

View lecturer

View Attendance per student

Generate reports

Admin

Lecturer

Add attendance Session

Student

Finger Prints

Does it Match?

YES

NO

Generate Reports

Figure 1 System Overview

Admin

Admin is the center stage in the system.

The following admin performs on the system;

Login

Register students

Register lecturers

View registered students

View registered students

View attendance by students

Generate reports

The diagram below shows various activities undertaken by the system administrator

Admin

Add Student

View Student

Add Lecturer

View Lecturer

View Attendance Per Student

Generate reports

Logout

Login

Figure 2 Use case diagram for Admin

Diagram showing how lecturer creates class-session for student to attend the class

Lecturer

View attendance per Student

Login

Add attendance Session

View Total Attendance

Student Session

Figure 3 Lecturer use case diagram

A diagram below shows the algorithmic process for student fingerprint authentication

Begin

Waiting

Getting fingerprint

Matching

Show message

Record RegNo, date& timee

Show message

No

YES

Figure 4 Student use case diagram

Implementation process

Biometric Enrollment - To be able to recognize a person by their biometric characteristics and the derived biometric features, first a learning phase must take place. The procedure is called enrolment and comprehends the creation of an enrolment data record of the biometric data subject (the person to be enrolled) and to store it in a biometric enrolment database. The enrolment data record comprises one or multiple biometric references and arbitrary non-biometric data such as a name, address, RegNo a Personnel File Number username and password.

Biometric Characteristic

Biometric Capture Device

Biometric Enrollment Database

Biometric Feature Extraction

Figure 5 Implementation process

Biometric Recognition - For the purpose of recognition, the biometric data subject (the person to be recognized) presents his or her biometric characteristic to the biometric capture section on the application which generates a recognition fingerprint biometric sample from it. From the recognition biometric sample, the biometric feature extraction creates biometric features which are compared with one or multiple biometric templates from the biometric enrolment database. Due to the statistical nature of biometric samples there is generally no exact match possible. For that reason, the decision process only assigns the biometric data subject to a biometric template and confirm recognition if the comparison score exceeds an adjustable threshold.

Comparison & Decision

Biometric Enrollment Database

Biometric Characteristic

Biometric Capture Device

Biometric Feature Extraction

Figure 6 Biometric recognition

## 4.2 Database design

Database Design is a collection of processes that facilitate the designing, development, implementation and maintenance of enterprise data management systems

It helps produce database systems

1. That meet the requirements of the users
2. Have high performance.

Database design entails

1. Conceptual database design
2. Logical database design
3. Physical database design

**Conceptual design** involves going to the field to gather facts and findings through data collection methods such as interviews and questioning protocol and thereafter I filtered the collected data to generate concepts.

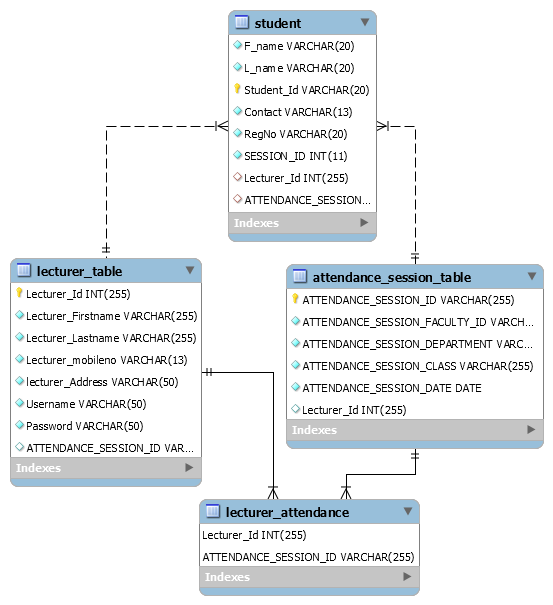
Student (student\_id, first\_name, last\_name, regNo, contact);

Lecturer (lecturer\_id, first\_name, last\_name, PFNo, password, username, address, contact);

Attendance\_Session (attendance\_session\_id, attendance\_session\_faculty\_id, attendance\_session\_department, attendance\_session\_class, attendance\_session\_date, attendance\_session\_subject, attendance\_student\_id, attendance\_status);

Lecturer\_attendance (lecturer\_id, attendance\_session\_id);

 The **logical model** concentrates on the data requirements and the data to be stored independent of physical considerations. It does not concern itself with how the data will be stored or where it will be stored physically. At this stage I carried out normalization.

  
  
Figure 7 A ERD diagram showing relationship between database entities

 The **physical data design model** involves translating the logical design of the database onto physical media through coding. For our case the DBMS that we used is SQLite. For more details check on Appendix III

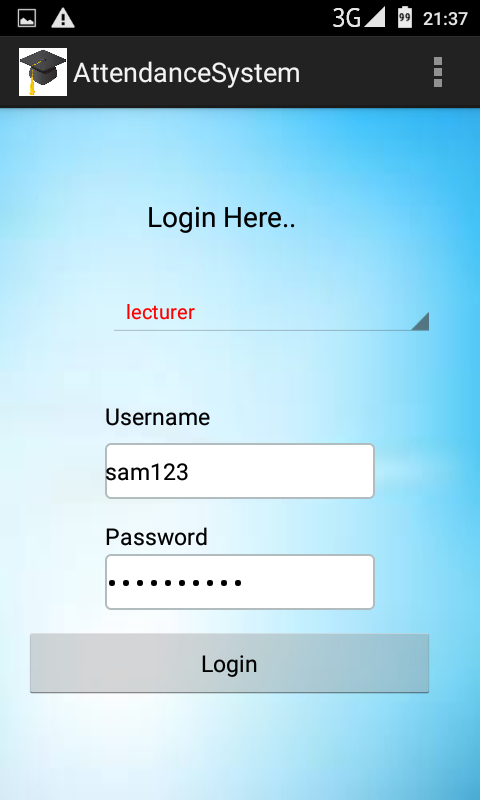
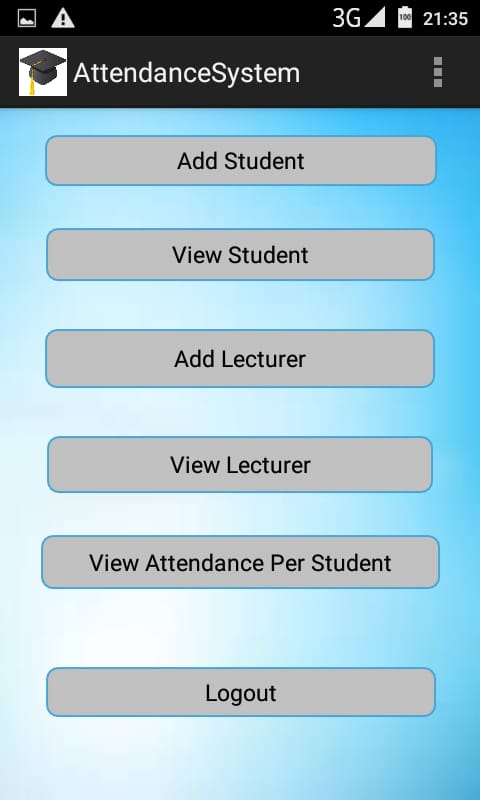
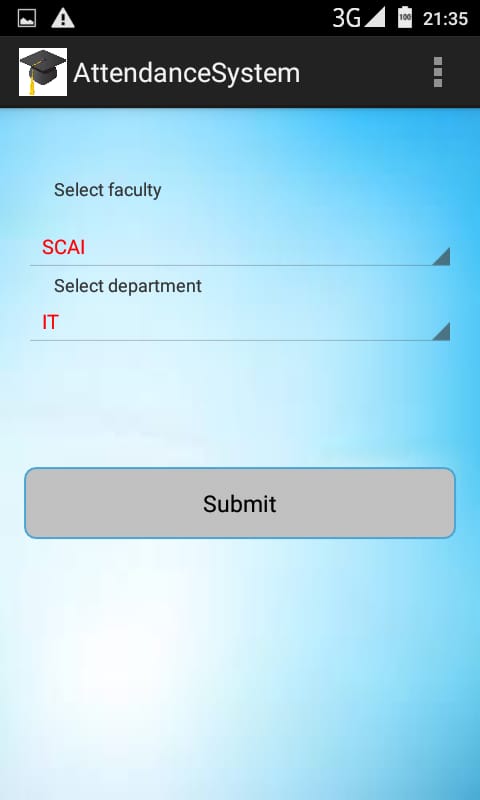
## 4.3 Interface design

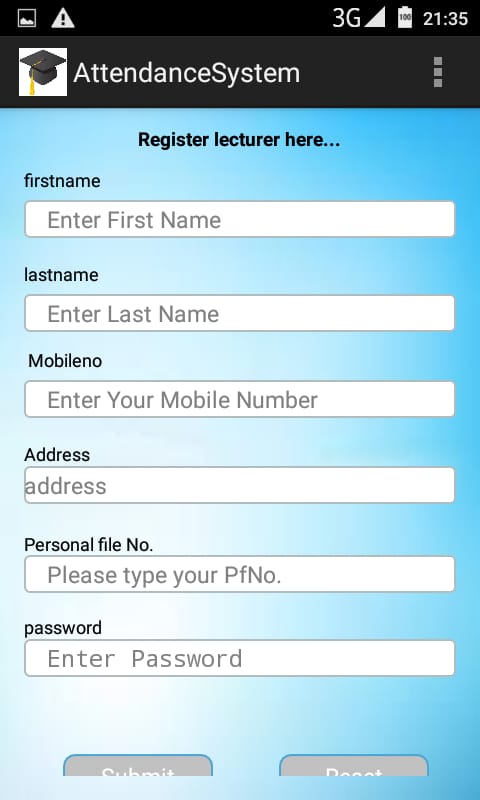
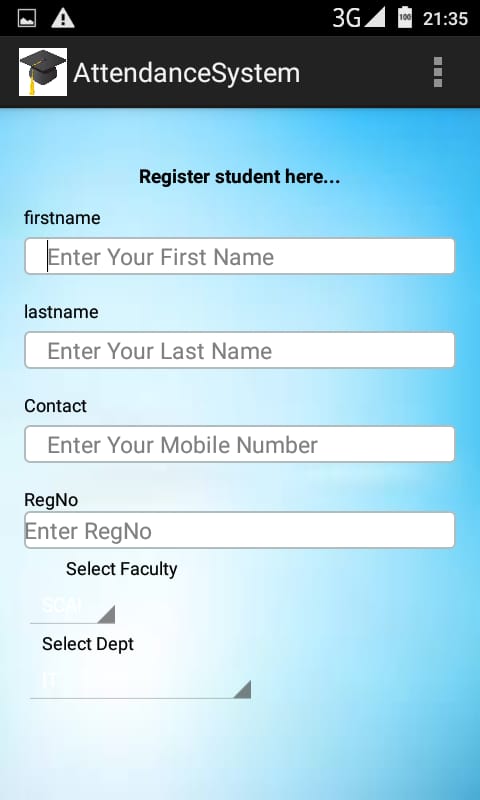
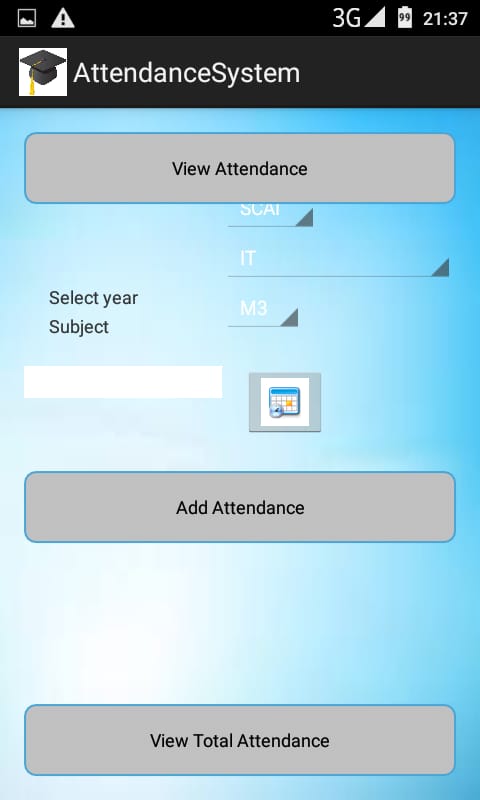
Lecturer Login Page

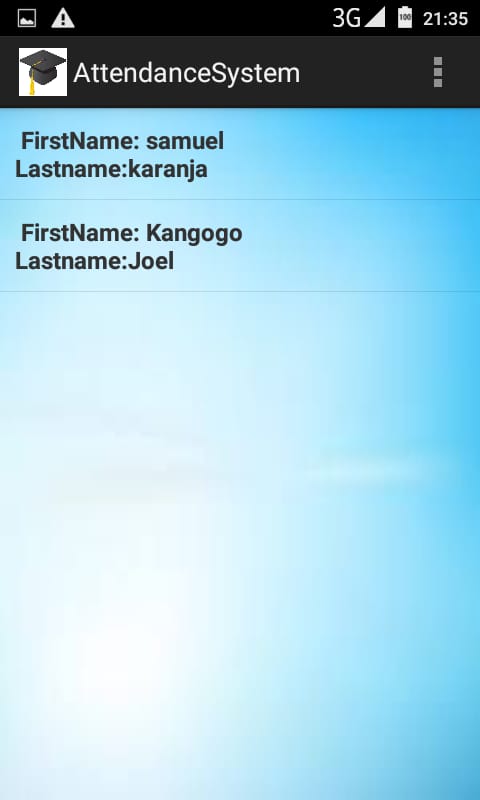
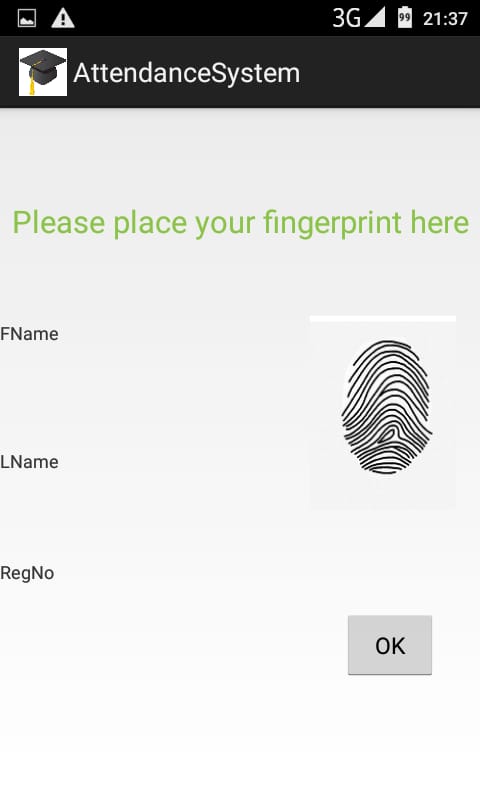
Splash Screen

View Students

Admin Panel





Student fingerprints

Figure 8 Sample interface design screenshots

# CHAPTER FIVE

# SYSTEM CONSTRUCTIONS

## 5.1 Sample code for the system

**package** com.android.attendance.activity;  
  
**import** android.app.Activity;  
**import** android.content.Intent;  
**import** android.graphics.Color;  
**import** android.os.Bundle;  
**import** android.text.TextUtils;  
**import** android.view.Menu;  
**import** android.view.View;  
**import** android.view.View.OnClickListener;  
**import** android.widget.AdapterView;  
**import** android.widget.AdapterView.OnItemSelectedListener;  
**import** android.widget.ArrayAdapter;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** android.widget.Spinner;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** com.android.attendance.bean.FacultyBean;  
**import** com.android.attendance.context.ApplicationContext;  
**import** com.android.attendance.db.DBAdapter;  
**import** com.example.androidattendancesystem.R;  
  
**public class** LoginActivity **extends** Activity {  
  
 Button **login**;  
 EditText **username**,**password**;  
 Spinner **spinnerloginas**;  
 String **userrole**;  
 **private** String[] **userRoleString** = **new** String[] { **"admin"**, **"lecturer"**};  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***login***);  
  
 **login** =(Button)findViewById(R.id.***buttonlogin***);  
 **username**=(EditText)findViewById(R.id.***editTextusername***);  
 **password**=(EditText)findViewById(R.id.***editTextpassword***);  
 **spinnerloginas**=(Spinner)findViewById(R.id.***spinnerloginas***);  
  
 **spinnerloginas**.setOnItemSelectedListener(**new** OnItemSelectedListener() {  
 @Override  
 **public void** onItemSelected(AdapterView<?> arg0, View view,  
 **int** arg2, **long** arg3) {  
 *//* ***TODO Auto-generated method stub*** ((TextView) arg0.getChildAt(0)).setTextColor(Color.***RED***);  
 **userrole** =(String) **spinnerloginas**.getSelectedItem();  
  
 }  
  
 @Override  
 **public void** onNothingSelected(AdapterView<?> arg0) {  
 *//* ***TODO Auto-generated method stub*** }  
 });  
  
 ArrayAdapter<String> adapter\_role = **new** ArrayAdapter<String>(**this**,  
 android.R.layout.***simple\_spinner\_item***, **userRoleString**);  
 adapter\_role  
 .setDropDownViewResource(android.R.layout.***simple\_spinner\_dropdown\_item***);  
 **spinnerloginas**.setAdapter(adapter\_role);  
  
 **login**.setOnClickListener(**new** OnClickListener() {  
  
 @Override  
 **public void** onClick(View v) {  
 *//* ***TODO Auto-generated method stub* if**(**userrole**.equals(**"admin"**))  
 {  
  
 String user\_name = **username**.getText().toString();  
 String pass\_word = **password**.getText().toString();  
  
 **if** (TextUtils.*isEmpty*(user\_name))   
 {  
 **username**.setError(**"Please type Username"**);  
 }  
 **else if**(TextUtils.*isEmpty*(pass\_word))  
 {  
 **password**.setError(**"Enter Password!"**);  
 }  
 **else** {  
 **if**(user\_name.equals(**"admin"**) & pass\_word.equals(**"admin123"**)){  
 Intent intent =**new** Intent(LoginActivity.**this**,MenuActivity.**class**);  
 startActivity(intent);  
 Toast.*makeText*(getApplicationContext(), **"Login successful"**, Toast.***LENGTH\_SHORT***).show();  
 }**else**{  
 Toast.*makeText*(getApplicationContext(), **"Please type correct Username/Password"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 }  
 }  
   
 **else** {  
 String user\_name = **username**.getText().toString();  
 String pass\_word = **password**.getText().toString();  
  
 **if** (TextUtils.*isEmpty*(user\_name))  
 {  
 **username**.setError(**"Invalid Personal File Number"**);  
 }  
 **else if**(TextUtils.*isEmpty*(pass\_word))  
 {  
 **password**.setError(**"Enter Password"**);  
 }  
 DBAdapter dbAdapter = **new** DBAdapter(LoginActivity.**this**);  
 FacultyBean facultyBean = dbAdapter.validateFaculty(user\_name, pass\_word);  
   
 **if**(facultyBean!=**null**)  
 {  
 Intent intent = **new** Intent(LoginActivity.**this**,AddAttandanceSessionActivity.**class**);  
 startActivity(intent);  
 ((ApplicationContext)LoginActivity.**this**.getApplicationContext()).setFacultyBean(facultyBean);  
 Toast.*makeText*(getApplicationContext(), **"Login successful"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 **else** {  
 Toast.*makeText*(getApplicationContext(), **"Please type correct Username/Password"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 }  
  
  
 }  
 });  
  
  
  
 }  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 *// Inflate the menu; this adds items to the action bar if it is present.* getMenuInflater().inflate(R.menu.***main***, menu);  
 **return true**;  
 }  
  
}

**package** com.android.attendance.activity;  
  
**import** java.util.ArrayList;  
**import** java.util.Calendar;  
  
**import** com.android.attendance.bean.AttendanceBean;  
**import** com.android.attendance.bean.AttendanceSessionBean;  
**import** com.android.attendance.bean.FacultyBean;  
**import** com.android.attendance.context.ApplicationContext;  
**import** com.android.attendance.db.DBAdapter;  
**import** com.example.androidattendancesystem.R;  
  
**import** android.app.Activity;  
**import** android.app.DatePickerDialog;  
**import** android.app.Dialog;  
**import** android.content.Intent;  
**import** android.graphics.Color;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.view.View.OnClickListener;  
**import** android.widget.AdapterView;  
**import** android.widget.AdapterView.OnItemSelectedListener;  
**import** android.widget.ArrayAdapter;  
**import** android.widget.Button;  
**import** android.widget.DatePicker;  
**import** android.widget.EditText;  
**import** android.widget.ImageButton;  
**import** android.widget.Spinner;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**public class** AddAttandanceSessionActivity<AddAttandanceActivity> **extends** Activity {  
  
 **private** ImageButton **date**;  
 **private** Calendar **cal**;  
 **private int day**;  
 **private int month**;  
 **private int dyear**;  
 **private** EditText **dateEditText**;  
 Button **submit**;  
 Button **viewAttendance**;  
 Button **viewTotalAttendance**;  
 Spinner **spinnerfaculty**,**spinnerdepartment**,**spinnerSubject**;  
 String **faculty** = **"SCAI"**;  
 String **department** = **"IT"**;  
 String **subject** = **"BIT424"**;  
  
 **private** String[] **facultyString** = **new** String[] { **"SCAI"**};  
 **private** String[] **departmentString** = **new** String[] {**"IT"**,**"Computer Science"**,**"Information Science"**};  
 **private** String[] **subjectITString** = **new** String[] {**"BIT424"**,**"DIT211"**,**"CIT101"**};  
 **private** String[] **subjectComputerScienceString** = **new** String[] {**"CSC313"**,**"CSC211"**,**"CSC101"**};  
 **private** String[] **subjectInformationScienceString** = **new** String[] {**"IS122"**,**"IS211"**,**"IS422"**};  
  
 **private** String[] **subjectFinal** = **new** String[] {**"BIT323"**,**"COM221"**,**"CIT110"**,**"MIS500"**,**"IS412"**,**"COM311"**};  
 AttendanceSessionBean **attendanceSessionBean**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***add\_attandance***);  
  
 *//Assume subject will be SE  
 //subjectFinal = subjectSEString;* **spinnerfaculty**=(Spinner)findViewById(R.id.***spinner1***);  
 **spinnerdepartment**=(Spinner)findViewById(R.id.***spinnerDept***);  
 **spinnerSubject**=(Spinner)findViewById(R.id.***spinnerSE***);  
  
 ArrayAdapter<String> adapter\_faculty = **new** ArrayAdapter<String>(**this**,android.R.layout.***simple\_spinner\_item***, **facultyString**);  
 adapter\_faculty.setDropDownViewResource(android.R.layout.***simple\_spinner\_dropdown\_item***);  
 **spinnerfaculty**.setAdapter(adapter\_faculty);  
 **spinnerfaculty**.setOnItemSelectedListener(**new** OnItemSelectedListener() {  
 @Override  
 **public void** onItemSelected(AdapterView<?> arg0, View view,  
 **int** arg2, **long** arg3) {  
 *//* ***TODO Auto-generated method stub*** ((TextView) arg0.getChildAt(0)).setTextColor(Color.***RED***);  
 **faculty** =(String) **spinnerfaculty**.getSelectedItem();  
 }  
  
 @Override  
 **public void** onNothingSelected(AdapterView<?> arg0) {  
 *//* ***TODO Auto-generated method stub*** }  
 });  
  
 *///......................spinner2* ArrayAdapter<String> adapter\_department = **new** ArrayAdapter<String>(**this**, android.R.layout.***simple\_spinner\_item***, **departmentString**);  
 adapter\_department.setDropDownViewResource(android.R.layout.***simple\_spinner\_dropdown\_item***);  
 **spinnerdepartment**.setAdapter(adapter\_department);  
 **spinnerdepartment**.setOnItemSelectedListener(**new** OnItemSelectedListener() {  
 @Override  
 **public void** onItemSelected(AdapterView<?> arg0, View view,  
 **int** arg2, **long** arg3) {  
 *//* ***TODO Auto-generated method stub*** ((TextView) arg0.getChildAt(0)).setTextColor(Color.***RED***);  
 **department** =(String) **spinnerdepartment**.getSelectedItem();  
 Toast.*makeText*(getApplicationContext(), **"department:"**+**department**, Toast.***LENGTH\_SHORT***).show();}  
  
 @Override  
 **public void** onNothingSelected(AdapterView<?> arg0) {  
 *//* ***TODO Auto-generated method stub*** }  
 });  
  
 ArrayAdapter<String> adapter\_subject = **new** ArrayAdapter<String>(**this**, android.R.layout.***simple\_spinner\_item***, **subjectFinal**);  
 adapter\_subject.setDropDownViewResource(android.R.layout.***simple\_spinner\_dropdown\_item***);  
 **spinnerSubject**.setAdapter(adapter\_subject);  
 **spinnerSubject**.setOnItemSelectedListener(**new** OnItemSelectedListener() {  
 @Override  
 **public void** onItemSelected(AdapterView<?> arg0, View view,  
 **int** arg2, **long** arg3) {  
 *//* ***TODO Auto-generated method stub*** ((TextView) arg0.getChildAt(0)).setTextColor(Color.***RED***);  
 **subject** =(String) **spinnerSubject**.getSelectedItem();  
  
 }  
  
 @Override  
 **public void** onNothingSelected(AdapterView<?> arg0) {  
 *//* ***TODO Auto-generated method stub*** }  
 });  
  
  
 **date** = (ImageButton) findViewById(R.id.***DateImageButton***);  
 **cal** = Calendar.*getInstance*();  
 **day** = **cal**.get(Calendar.***DAY\_OF\_MONTH***);  
 **month** = **cal**.get(Calendar.***MONTH***);  
 **dyear** = **cal**.get(Calendar.***YEAR***);  
 **dateEditText** = (EditText) findViewById(R.id.***DateEditText***);  
 **date**.setOnClickListener(**new** OnClickListener() {  
  
 @Override  
 **public void** onClick(View arg0) {  
 showDialog(0);  
  
 }  
 });  
  
 **submit**=(Button)findViewById(R.id.***buttonsubmit***);  
 **submit**.setOnClickListener(**new** OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 Intent intent = **new** Intent(AddAttandanceSessionActivity.**this**,Studentfingerprint0.**class**);  
 startActivity(intent);  
 }  
 });  
  
 **viewTotalAttendance**=(Button)findViewById(R.id.***viewTotalAttendanceButton***);  
 **viewTotalAttendance**.setOnClickListener(**new** OnClickListener() {  
  
 @Override  
 **public void** onClick(View arg0) {  
 AttendanceSessionBean attendanceSessionBean = **new** AttendanceSessionBean();  
 FacultyBean bean=((ApplicationContext)AddAttandanceSessionActivity.**this**.getApplicationContext()).getFacultyBean();  
  
 attendanceSessionBean.setAttendance\_session\_faculty\_id(bean.getFaculty\_id());  
 attendanceSessionBean.setAttendance\_session\_department(**faculty**);  
 attendanceSessionBean.setAttendance\_session\_class(**department**);  
 attendanceSessionBean.setAttendance\_session\_subject(**subject**);  
  
 DBAdapter dbAdapter = **new** DBAdapter(AddAttandanceSessionActivity.**this**);  
  
 ArrayList<AttendanceBean> attendanceBeanList = dbAdapter.getTotalAttendanceBySessionID(attendanceSessionBean);  
 ((ApplicationContext)AddAttandanceSessionActivity.**this**.getApplicationContext()).setAttendanceBeanList(attendanceBeanList);  
  
 Intent intent = **new** Intent(AddAttandanceSessionActivity.**this**,ViewAttandanceActivity.**class**);  
 startActivity(intent);  
  
 }  
 });  
 }  
 @Override  
 @Deprecated  
 **protected** Dialog onCreateDialog(**int** id) {  
 **return new** DatePickerDialog(**this**, **datePickerListener**, **dyear**, **month**, **day**);  
 }  
 **private** DatePickerDialog.OnDateSetListener **datePickerListener** = **new** DatePickerDialog.OnDateSetListener() {  
 **public void** onDateSet(DatePicker view, **int** selectedYear,  
 **int** selectedMonth, **int** selectedDay) {  
 **dateEditText**.setText(selectedDay + **" / "** + (selectedMonth + 1) + **" / "** + selectedYear);  
 }  
 };  
  
}

# CHAPTER SIX

# SYSTEM TESTING

## 6.1 Introduction

System Testing is the testing of a complete and fully integrated software product. Usually, software is only one element of a larger computer-based system. Ultimately, software is interfaced with other software/hardware systems. System Testing is actually a series of different tests whose sole purpose is to exercise the full computer-based system.

## 6.2 Module integration and testing

Both Admin module and lecturer module are on the same login page where there is a combo box that you select to Login as Admin or Lecturer but the modules were tested separately. Each module was tested to ensure that they work correctly to generate the output required for designing it without issuing any error.

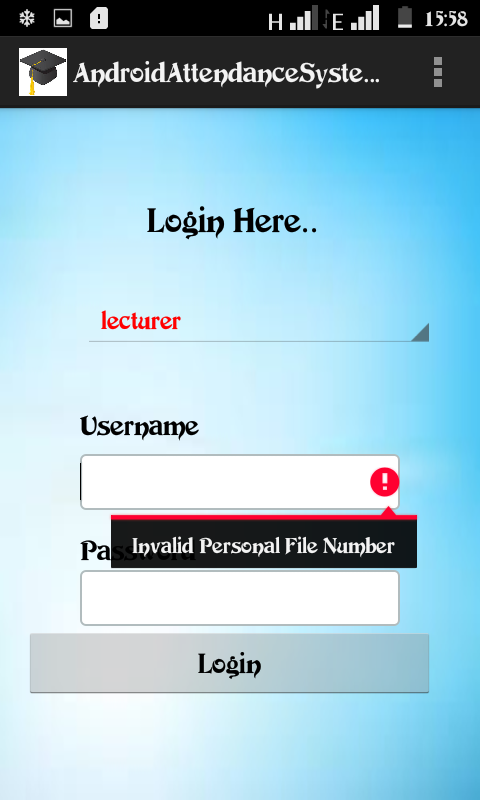
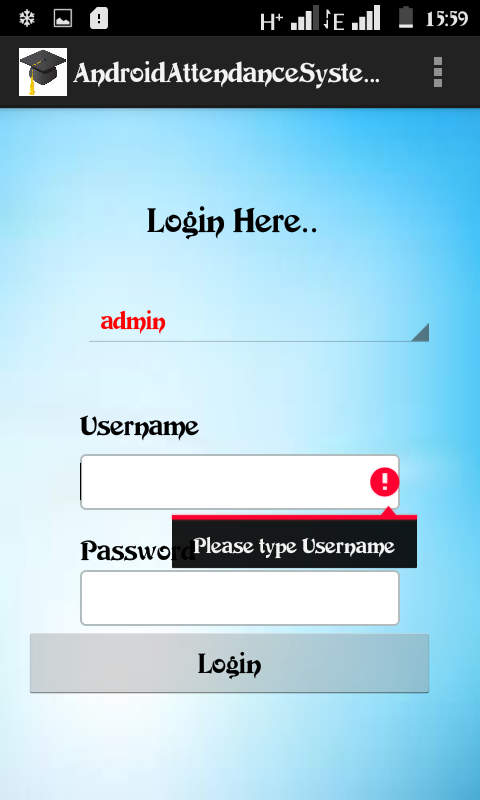
 

Figure 9 Admin-lecturer login interface testing

The software was tested with different inputs of different data types. The system was found to work correctly because in places where wrong data type was input, the system displayed error messages requesting the user to input correct data. This was aided by use of text validation rules coded in the program.

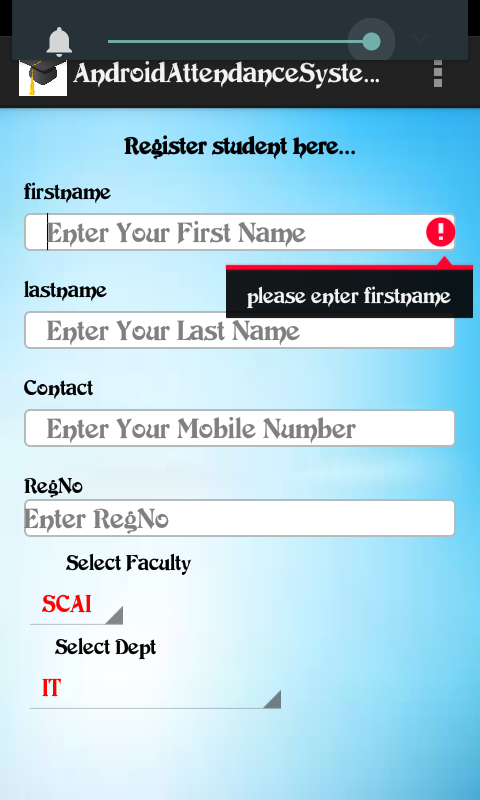


Figure 10 Student registration error checking

## 6.3 Training.

Just like the normal system, training on the use of the system has been given enough materials to enable the user to easily use the system. The user installation manual with the user help system has been part of education and training.

## 6.4 Actual handover

It Involves demonstration of various system functionalities. The first time the project proposal was presented to the panel for acceptance to continue with developing the system which there after overseen through its development period by the supervisor and later presented on full development to the same panel of assessors for verification.

# USER MANUAL

## Installation guide

The system is an android mobile based that needs to be installed on phones

## Navigation guide

When the user clicks on the application on the phone, a splash screen for welcome is launched automatically where it takes you to Login Page where you select user to Login and if there is any help needed, you contact the Admin for more clarification.

## Procedure for data capture

Admin or Lecturer is required to sign in for an account by providing a username and a password (Username or PFNo for lecturers). The lecturers and students are added to the system by the Administrator. Upon logging in, the user can do what they are intended to do.

Procedure for generating reports;

Once class attendance has been made, Admin views the attendance and generate a report of the attended class automatically since when the lecturer creates sessions for student to provide their fingerprints to attend class, the same is updated in Admin Panel where he/she can generate the reports.

# Conclusion

I believe that research will continue to bloom and fork on various paths in this area of biometrics. The developing idea and prototype of Biometric-based Student’s Attendance System using Biometric technology is on the verge of introducing worldwide institutions into an automated way of capturing student’s attendance. It will, by far, improve the current manual process of tracking and recording student’s attendance.

This system promotes a way for students to ‘sign’ their attendance sheets on a digital form by just providing their fingerprints. For lecturers, implementation of this kind of system would ease their effort in tracking students who are missing from class as well as reducing effort to verify students’ attendance and to also provide authentic data that is not falsified through identity as opposed to signing of sheets that may tempt others students to be dishonest for signing to their fellow colleagues.

Therefore, in my system, I partially managed to implement fingerprint biometric-based attendance system due to time and cost-resource limitations.

# Recommendation

The system took a period of fourteen weeks (14) where I managed and tried my best to come up with a prototype version of the system. I managed to utilize my free time, weekends working tirelessly through day and night so that I could create time during the system design because also, within that timeframe I had to attend for classes and other activities during week days.

I managed to come up with three modules i.e. for lecturer, admin and student modules and because the time was limited I will have to improve on the functionalities of the system with time as the prototyping methodology accepts that which will become increments versions of the system.

Another challenge also is that resources needed on my system were missing for example the fingerprint scanner for registering students was a problem to acquire. Finding with technicians and supervisors and getting in touch during the development phase was a little bit challenging and exchange of ideas, pieces of advice was not well coordinated.

I therefore recommend the following;

1. Students should be given time to present and come up with ideas while they are in first year and present proposals while in second year so that they are given time to build their concept well and better to avoid last minute rush.
2. The University should avail materials needed on time so that we work well and professionally ranging from hardware to software.
3. Supervisors should work in touch and very close so that any issues are sought earlier before the last minute so that we head on right direction.
4. The important dates set during the project development should be kept fixed and adhered to since procrastination brings a sought of reluctance, morale and the commitment of students is lowered.

Otherwise if the above issues are sorted out then there is no doubt that professional and quality projects will be delivered.

# 

# References

Soewito, B., Gaol, F. L., Simanjuntak, E., & Gunawan, F. E. (2016, July). Smart mobile attendance system using voice recognition and fingerprint on smartphone

Apoorv, R., & Mathur, P. (2016, November). Smart attendance management using Bluetooth Low Energy and Android.

Shanbhag, G., Jivani, H., & Shahi, S. (2014). Mobile Based Attendance Marking System using Android and Biometrics. *IJIRST-International Journal for Innovative Research in Science & Technology*, *1*(1), 87-90.

Patel, U. A., & Priya, S. (2014). Development of a student attendance management system using rfid and face recognition: A review. *International Journal of Advance Research in Computer Science and Management Studies*, *2*(8), 109-119.

Noor, S. A. M., Zaini, N., Latip, M. F. A., & Hamzah, N. (2015, December). Android-based attendance management system.

Srinidhi, M. B., & Roy, R. (2015, January). A web enabled secured system for attendance monitoring and real time location tracking using Biometric and Radio Frequency Identification (RFID) technology.

Chew, C. B., Mahinderjit-Singh, M., Wei, K. C., Sheng, T. W., Husin, M. H., & Malim, N. H. A. H. (2015). Sensors-enabled smart attendance systems using NFC and RFID technologies. *International Journal of New Computer Architectures and their Applications (IJNCAA)*, *5*(1), 19-28.

Anand, S., Bijlani, K., Suresh, S., & Praphul, P. (2016, December). Attendance monitoring in classroom using smartphone & Wi-Fi fingerprinting

Chawhan, S. S., Girhale, M. P., & Mankar, G. (2013). Mobile phone based attendance system. *IOSR journal of computer engineering (IOSR-JCE)*, *10*(3), 48-50.

Islam, M. M., Hasan, M. K., Billah, M. M., & Uddin, M. M. (2017, December). Development of smartphone-based student attendance system.

Sarker, D. K., Hossain, N. I., & Jamil, I. A. (2016, December). Design and implementation of smart attendance management system using multiple step authentication. In *2016 International Workshop on Computational Intelligence (IWCI)*(pp. 91-95).

# List of Appendixes

## Appendix I: project schedule

Table 1 Project schedule

|  |  |  |  |
| --- | --- | --- | --- |
| S/NO. | TASK DESCRIPTION | START | END |
|  | Requirement Analysis | 16/06/2019 | 26/06/2019 |
|  | System Analysis And Design | 29/06/2019 | 24/07/2019 |
|  | System Implementation | 25/07/2019 | 28/07/2019 |
|  | System Testing | 29/07/2019 | 31/07/2019 |
|  | Documentation | 02/08/2019 | 04/08/2019 |
|  | Submission |  | - |

## Appendix II: SDLC Phases

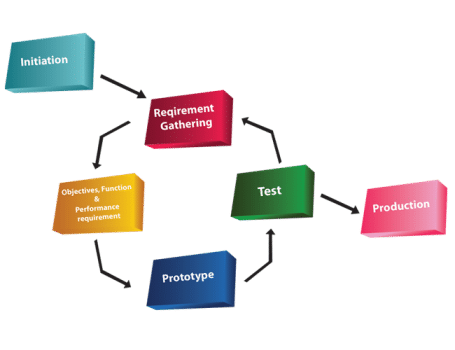


Figure 11 A diagram showing phases of system design

Appendix III: Database sample code

private static final int *DATABASE\_VERSION* = 1;  
  
 *// Database Name* private static final String *DATABASE\_NAME* = "Attendance";  
  
 *// Contacts table name* private static final String *LECTURER\_INFO\_TABLE* = "lecturer\_table";  
 private static final String *STUDENT\_INFO\_TABLE* = "student\_table";  
 private static final String *ATTENDANCE\_TABLE* = "attendance\_table";  
 private static final String *ATTENDANCE\_SESSION\_TABLE* = "attendance\_session\_table";  
  
  
 *// Contacts Table Columns names* private static final String *KEY\_LECTURER\_ID* = "lecturer\_id";  
 private static final String *KEY\_LECTURER\_FIRSTNAME* = "lecturer\_firstname";  
 private static final String *KEY\_LECTURER\_LASTNAME* = "lecturer\_Lastname";  
 private static final String *KEY\_LECTURER\_MO\_NO* = "lecturer\_mobilenumber";  
 private static final String *KEY\_LECTURER\_ADDRESS* = "lecturer\_address";  
 private static final String *KEY\_LECTURER\_USERNAME* = "lecturer\_username";  
 private static final String *KEY\_LECTURER\_PASSWORD* = "lecturer\_password";  
  
 private static final String *KEY\_STUDENT\_ID* = "student\_id";  
 private static final String *KEY\_STUDENT\_FIRSTNAME* = "student\_firstname";  
 private static final String *KEY\_STUDENT\_LASTNAME* = "student\_lastname";  
 private static final String *KEY\_STUDENT\_MO\_NO* = "student\_mobilenumber";  
 private static final String *KEY\_STUDENT\_ADDRESS* = "student\_address";  
 private static final String *KEY\_STUDENT\_DEPARTMENT* = "student\_department";  
 private static final String *KEY\_STUDENT\_CLASS* = "student\_class";  
 private static final String *KEY\_STUDENT\_FINGERPRINT* = "student\_fingerprint";  
  
 private static final String *KEY\_ATTENDANCE\_SESSION\_ID* = "attendance\_session\_id";  
 private static final String *KEY\_ATTENDANCE\_SESSION\_FACULTY\_ID* = "attendance\_session\_faculty\_id";  
 private static final String *KEY\_ATTENDANCE\_SESSION\_DEPARTMENT* = "attendance\_session\_department";  
 private static final String *KEY\_ATTENDANCE\_SESSION\_CLASS* = "attendance\_session\_class";  
 private static final String *KEY\_ATTENDANCE\_SESSION\_DATE* = "attendance\_session\_date";  
 private static final String *KEY\_ATTENDANCE\_SESSION\_SUBJECT* = "attendance\_session\_subject";  
  
 private static final String *KEY\_SESSION\_ID* = "attendance\_session\_id";  
 private static final String *KEY\_ATTENDANCE\_STUDENT\_ID* = "attendance\_student\_id";  
 private static final String *KEY\_ATTENDANCE\_STATUS* = "attendance\_status";  
  
  
 public DBAdapter(Context context) {  
 super(context, *DATABASE\_NAME*, null, *DATABASE\_VERSION*);  
 }

Appendix IV: Interface Design Sample Code

**public class** LoginActivity **extends** Activity {  
  
 Button **login**;  
 EditText **username**,**password**;  
 Spinner **spinnerloginas**;  
 String **userrole**;  
 **private** String[] **userRoleString** = **new** String[] { **"admin"**, **"lecturer"**};  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***login***);  
  
 **login** =(Button)findViewById(R.id.***buttonlogin***);  
 **username**=(EditText)findViewById(R.id.***editTextusername***);  
 **password**=(EditText)findViewById(R.id.***editTextpassword***);  
 **spinnerloginas**=(Spinner)findViewById(R.id.***spinnerloginas***);  
  
 **spinnerloginas**.setOnItemSelectedListener(**new** OnItemSelectedListener() {  
 @Override  
 **public void** onItemSelected(AdapterView<?> arg0, View view,  
 **int** arg2, **long** arg3) {  
 *//* ***TODO Auto-generated method stub*** ((TextView) arg0.getChildAt(0)).setTextColor(Color.***RED***);  
 **userrole** =(String) **spinnerloginas**.getSelectedItem();  
  
 }  
  
 @Override  
 **public void** onNothingSelected(AdapterView<?> arg0) {  
 *//* ***TODO Auto-generated method stub*** }  
 });  
  
 ArrayAdapter<String> adapter\_role = **new** ArrayAdapter<String>(**this**,  
 android.R.layout.***simple\_spinner\_item***, **userRoleString**);  
 adapter\_role  
 .setDropDownViewResource(android.R.layout.***simple\_spinner\_dropdown\_item***);  
 **spinnerloginas**.setAdapter(adapter\_role);  
  
 **login**.setOnClickListener(**new** OnClickListener() {  
  
 @Override  
 **public void** onClick(View v) {  
 *//* ***TODO Auto-generated method stub* if**(**userrole**.equals(**"admin"**))  
 {  
  
 String user\_name = **username**.getText().toString();  
 String pass\_word = **password**.getText().toString();  
  
 **if** (TextUtils.*isEmpty*(user\_name))   
 {  
 **username**.setError(**"Please type Username"**);  
 }  
 **else if**(TextUtils.*isEmpty*(pass\_word))  
 {  
 **password**.setError(**"Enter Password!"**);  
 }  
 **else** {  
 **if**(user\_name.equals(**"admin"**) & pass\_word.equals(**"admin123"**)){  
 Intent intent =**new** Intent(LoginActivity.**this**,MenuActivity.**class**);  
 startActivity(intent);  
 Toast.*makeText*(getApplicationContext(), **"Login successful"**, Toast.***LENGTH\_SHORT***).show();  
 }**else**{  
 Toast.*makeText*(getApplicationContext(), **"Please type correct Username/Password"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 }  
 }  
   
 **else** {  
 String user\_name = **username**.getText().toString();  
 String pass\_word = **password**.getText().toString();  
  
 **if** (TextUtils.*isEmpty*(user\_name))  
 {  
 **username**.setError(**"Invalid Personal File Number"**);  
 }  
 **else if**(TextUtils.*isEmpty*(pass\_word))  
 {  
 **password**.setError(**"Enter Password"**);  
 }  
 DBAdapter dbAdapter = **new** DBAdapter(LoginActivity.**this**);  
 FacultyBean facultyBean = dbAdapter.validateFaculty(user\_name, pass\_word);  
   
 **if**(facultyBean!=**null**)  
 {  
 Intent intent = **new** Intent(LoginActivity.**this**,AddAttandanceSessionActivity.**class**);  
 startActivity(intent);  
 ((ApplicationContext)LoginActivity.**this**.getApplicationContext()).setFacultyBean(facultyBean);  
 Toast.*makeText*(getApplicationContext(), **"Login successful"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 **else** {  
 Toast.*makeText*(getApplicationContext(), **"Please type correct Username/Password"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 }  
 }  
 });

}  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 *// Inflate the menu; this adds items to the action bar if it is present.* getMenuInflater().inflate(R.menu.***main***, menu);  
 **return true**;  
 }  
}