ASHESI UNIVERSITY COLLEGE

ATTENDANCE MANAGEMENT SYSTEM

APPLIED

B.Sc. Management Information Systems

Adwoba Nenge Bota

2017

ASHESI UNIVERSITY COLLEGE

Attendance Management System

APPLIED PROJECT

Applied project submitted to the Department of Computer Science, Ashesi University College in partial fulfilment of the requirements for the award of Bachelor of Science degree in Management Information Systems

Adwoba Nenge Bota

April 2017

DECLARATION

I hereby declare that this Applied project is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:

……

Candidate's Name:

……

Date:

……

I hereby declare that preparation and presentation of this Applied project were supervised in accordance with the guidelines on supervision of Applied project laid down by Ashesi University College.

Supervisor's Signature:

……

Supervisor's Name:

……

Date:

……

Acknowledgement

I thank God Almighty for seeing me through four successful years and helping me to make this project a reality.

I also give thanks to my family for all their support, their love and advice, without them I would not have made it this far in life. Special thanks to my supervisor Mr David Sampah and Mr Kwadwo Osafo Maafo for their guidance, support and time. This project would not have been a reality without them. Lastly, I thank all my friends, especially my roommates, Alvin Ofori and Francis Kornu for lending a listening ear and being the support I needed.

Abstract

Attendance is an important element in face to face education. A lot of schools in advanced countries have some form of attendance management system in place to track student attendance, monitor deviances and a source for recording data. There are many methods used to take class attendance, the traditional method being the use of paper or calling out of student names. Some universities, however, use the student ID card. Over the years, people have built applications or devices to take attendance in class. These methods provide better security than the traditional methods. Some methods are better than others and are worth pursuing. In our local context, universities still depend on papers or roll calling to take student attendance. This project seeks to provide a faster and simpler way to take attendance in class with the use of a QR code. The system should combine speed with accuracy and effectiveness.

Keywords – QR code, accuracy, accuracy

Table of Contents

Table of Figures

List of Abbreviations

- API- Application Program Interface
- ASP - Active Server Pages
- AVR - Alf and Vegard's RISC processor
- CSS – Cascaded Style Sheet
- EEPROM - Electrically Erasable Programmable Read-Only Memory
- GLCD - Graphical Liquid crystal display
- HTML – Hypertext Mark-up Language
- ID – Identification
- LCD - Liquid Crystal Display
- PHP - Hypertext Preprocessor
- RFID - Radio-frequency identification
- W3C - World Wide Web Consortium

Chapter 1: Introduction

1.1 Background

One of the common ways of taking attendance is through roll call or in the case of Ashesi University and a few others; signup sheets. This method can be time-consuming and ineffective in getting students to come to class early. Over time, some universities have developed or acquired systems or devices to help in attendance taking. These include the use of barcode scanning or RFIDs to authenticate students each time they attend class. The same process can be used to control entry to restricted places or to determine the number of people who are present at some point in time.

As more universities come to understand the usefulness of class attendance, many applications have been built to make the task easier and are equally effective in preventing students from signing up attendance for absent colleagues. However, very few universities in the Ghanaian context are making use of these management systems and tools. This means that records on students' attendance are poorly kept. This project focuses on creating a simple management application that can be used by universities to make attendance taking faster and increase the accuracy with which it is done. Class attendance has been proven to increase students' chances at getting higher test grades on examination (Chen and Lin, 2008). This, therefore, creates the need to create a system that would serve as a motivation for students to come to class.

1.2 Problem Statement

Taking attendance in colleges serves as a check for students; it ensures regular participation in classes which may boost their performance as mentioned early. Many universities use barcodes scanners, image detectors or Bluetooth to successfully complete the task of taking attendance. These methods are usually expensive and can prevent

universities from taking attendance to be able to save their budget. There should be a simple way to take attendance of students without inflating schools' budget. This project focuses on creating a simple one-page application that can be used to take attendance which is both fast and affordable in comparison to the other methods. Students will realise the urgency for the class when it serves as a reward system that only the early birds enjoy. If attendance occurs only in a limited space of time, students will come to class early in other to enjoy the reward attached to being marked in the attendance. Roll calling or sign-in sheets are not successful at solving this problem. It is time-consuming and information gathered can easily be lost. This is, however, the method adopted by some universities and many other institutions. Therefore, a faster system is essential to facilitate the process of taking attendance with the aim of improving performance.

1.3 Motivation

A lot of universities want to monitor students' attendance however very few are doing so in an effective manner. This project will focus on a faster and more accurate method of attendance that can be easily adopted by universities. An attendance management system should effectively take attendance in class and generate an immediate report for viewing so that lecturers can follow up on absent students or recognise the times that students report to class. Students will always come up with excuses why they cannot attend class although they wish to successfully complete the course (Clay and Breslow, 2006). If a proper system is not in check, some students will always try to swindle the system. To improve the time of reporting to class and reduce the time spent in attendance is the main motivation for this project.

1.4 Project Objectives

This project should effectively provide a faster, more affordable system for taking attendance. It should be easily modified to suit organizations who take rigorous attendance

as part of their management or daily routine. It should also serve as a motivating factor for students to attend lectures early.

1.5 Assumptions

The system will be built and tested at Ashesi University. Ashesi University already has some form of attendance system that is currently being used. It is supported by a well-developed database for students and lecturers. The system will, therefore, make use of the database in place, therefore, all records will be obtained from the information technology department at Ashesi University. The database will, however, be extended to allow for attendance using the application that will be developed.

1.6 End Product

The system will be a web based attendance application that will utilize QR code for taking attendance. The application will have a database with both students and lecturers' data such as names, class, ID information and other relevant records.

Chapter 2: Related Works

Attendance recording is nothing new in the educational sector. Many people in the field of computer science have done research on how to improve record taking in class. This section contains some research solutions proposed.

In 2007, a group of electrical engineering students did research on a digital attendance recording system. The aim of the project was to provide a device that would relieve its users from using paper records of attendance and provide users with relevant attendance data for future use. The project was centred on building a portable device on the AVR ATmega8 microcontroller. The record is therefore stored in the microcontroller's *electrically erasable programmable read-only memory (EEPROM)* from where it is transferred to an application software residing on a computer through the serial port. This application stores the attendance data in a database and displays requested information extracted.

End users interact with the device using a "four key push buttons"; UP, DOWN, OK, CHANGE. Users will, therefore, keep pressing the buttons until attendance is completely done. The user sees what is being done through the liquid crystal display (LCD) module display. After an attendance session, the information gathered must be transferred to the database for further manipulation, this requires some user authentication. This method is effective in eradicating the paper method, it is however not very effective in its implementation. The time spent on taking attendance is greatly increased and there is the risk of data manipulation since information is stored temporarily on the microcontroller before transferred to the database (Khan, Ahmed, Abeer and Malik, 2007).

Another piece of literature talks about an attendance management system that utilizes the Bluetooth technology to take class attendance. The aim of this project is to detect student proxy and eliminate the likelihood of students taking attendance outside the

classroom. The end users; lecturers and students will both sign in using their credentials. The lecturer starts the session by selecting the course and other relevant information. When the lecturer sends the request for the session to begin, the server initiates the session which lasts for five minutes. Students will receive this session and start the attendance process. Students take attendance by scanning Bluetooth devices present in the class. They then sign in for a few other students and themselves. The scanned Bluetooth devices and the attendance record is then sent to the server for recording. The student, based on whether he or she can capture the lecturer's Bluetooth device, may or may not be registered for attendance. This is a very interesting project which minimizes attendance taking time to only five minutes. It is also successful in eliminating cheating on the system. It is however an android based application, meaning students with different devices may not be able to take attendance as they should. (Akram and Rustagi, 2016).

A project using RFID focuses on the use of web (ASP.Net) application to monitor attendance. The project seeks to reduce attendance time to five minutes and remove human errors by reducing human interference with the system as much as possible. The system makes use of a Radio frequency identification (RFID) and microcontroller Smart Graphical Liquid crystal display (GLCD). Students make use of their ID cards, and as they scan, a motion detector captures the figure of the student. This is to prevent students from signing in for others. This method rejects scanning for cards that it does not capture motion for. The captured data is displayed using the ASP.Net framework. This project is very effective in a lot of ways; it can reduce system manipulation by reducing human interference and makes for speed in terms of reducing attendance time to about five minutes. The system is however very expensive to set up, it would require an RFID reader and a motion detector to operate (Abas, Tuck and Dahlui, 2015).

An attendance management system should be fast, reliable and not too expensive. It should also reduce student attempts to manipulate data taken. The above related works aimed at achieving one or more of these requirements using different approaches which will influence the attendance system greatly. The system will mimic the use of the RFID reader and inculcate speed and accuracy to ensure that all students are recorded with little or no interruptions.

Chapter 3: Requirements and Architecture

3.1 User requirement

Ashesi University College is one of the top universities in Africa. The current system of taking class attendance is through “Focus”, a web based application that enables users to take attendance among other functionalities it provides. However, the system has a gap, hence lecturers use papers to get names of present students for a lecture which is later transferred to focus. The gap poses a lot of problems such as;

- Loss of information; papers can easily get lost before information on it is transferred
- Students may forget to sign up
- Papers circulating around during class is very distracting
- Waste of papers
- Discourages early attendance to classes

3.2 System Design

This section describes the architecture involved in creating an attendance management system for a school. The proposed system will provide a faster and more effective way of recording class attendance and should reduce gaps in the current system. The system will be a web based application that makes use of a QR code for attendance. The application has an inbuilt timer set between five to eight minutes for which the QR code will be displayed to the class. Lecturers must sign in before they can create the QR code for display. Once the code is up, student can scan and enter their ID for quick authentication.

3.3 Component Description

Students:

Students will make use of any QR scanner device on their phones to enable them to partake in attendance. They will scan the code and enter their ID number when prompted. When this is done successfully, they will be authenticated into the system.

Lecturers:

Lecturers and Faculty Interns must sign into the application using their username and password. Once on the dashboard, a lecturer can pick a course from the list of course he teaches and create a QR code to begin the attendance. The QR code is timed and will be erased once the attendance is completed. The lecturer can view students who attended class that day on the dashboard as well as view all students who are registered to the course.

To view the number of students who attended a class in the day, the lecturer can navigate back to the dashboard which should have the list of present students. The lecturer can also search and view absent students.

3.4 Functional requirements

1. Lecturers can view all students who attended a class
2. Lecturers can view absent students
3. Lecturers can take attendance for each class
4. Lecturers can view all classes he teaches on the dashboard
5. Students can sign in by scanning QR codes

3.5 Non-functional requirements

Product

Speed is necessary to ensure that the right time is captured for each student who signs into the class. It should be fast, reliable and most of all, user friendly.

Organizational

It should not provide an extra burden to faculty or students in the process of taking class attendance.

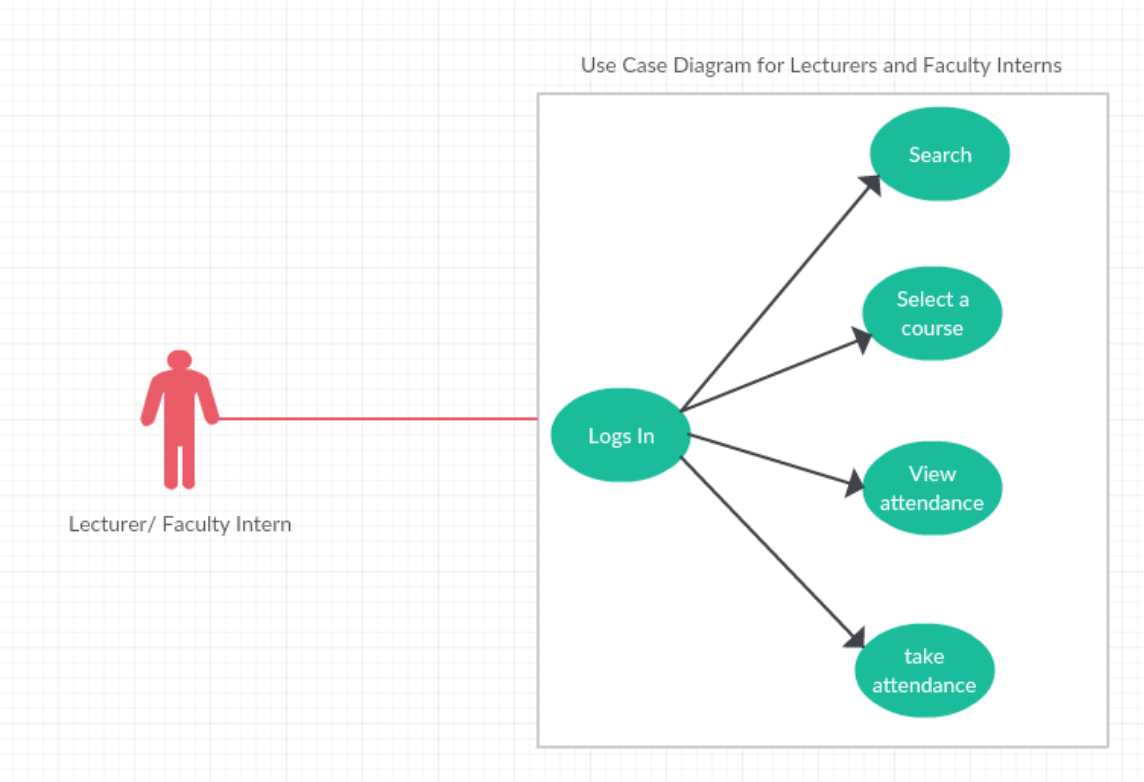
3.6 Performance requirements

The application should be available at all times for attendance to be taken. It should not delay or disrupt lectures.

3.7 Use Case Diagram

The main actors who interact with the system are the lecturers or faculty interns and students. The students use their mobile devices to capture the QR codes for attendance. The lecturers can perform a few more operations within the system. Below is a use case diagram showing the operations that a lecturer performs on the system on the daily basis (Figure3.1).

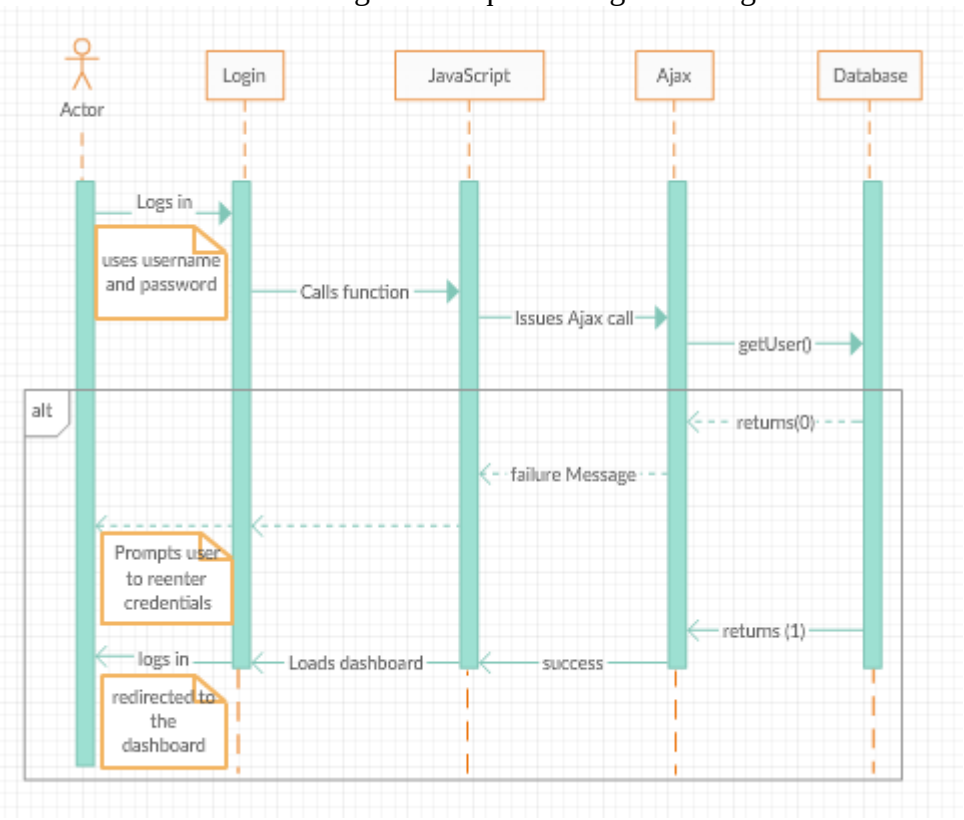
Figure3.1 Use Case Diagram for Lecturers and Faculty Interns



3.8 Sequence Diagram

The sequence diagram shows how interaction between objects occur. Figure 3.2 below shows how a lecturer is authenticated to use the application.

Figure3.2 Sequence Diagram of Login for Lecturers



3.9 Database Design

An attendance management system for a school involves a lot of modules such as the students, classes, lecturers and records on students' class attendance. The database design must therefore be well designed to prevent future crisis and conflicts. Failure in the database will be very tragic and may halt the attendance taking process. The attendance management system contains eight tables; school year, school_semester, course, course_session, course_session_student, student, class_attendance, and the lecturer.

The above table can be divided into five main components. The school year and school_semester records information on the period in discussion. The semester depends on the school year because a semester is existence in a year.

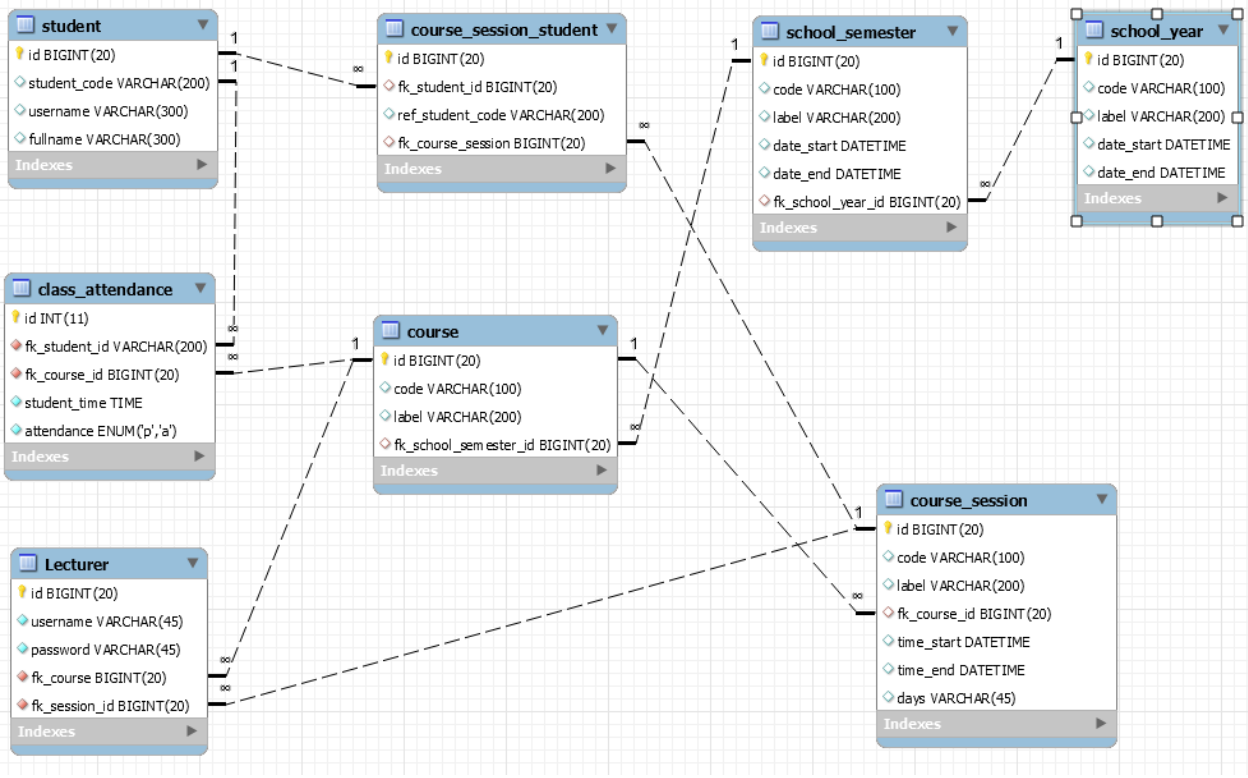
The course and course_session provide information on all the courses that run in a semester and the times at which it runs.

The course_session_student and student contains information on all the students in the school and their respective courses and course time.

The class attendance table is used to keep record on attendance that happens in all the courses. The lecturer table allows the lecturers to be authenticated to take attendance in their various courses.

The figure below shows the database structure for the attendance management system.

Figure3.3 Database Structure for the Attendance Management System



3.10 Architecture

The application is built using the three-tier architecture layer; the presentation layer, application layer and the data access layer. The presentation layer shows the user interface of application. This is the client side of the application and is what the user sees once the page is loaded in the browser. This side is modified using mostly HTML and CSS which will be discussed in detail in later chapters.

The application layer is the server side of the application. It performs actions related to user clicks on the page. This layer serves as a medium between the presentation layer and the database layer by returning from the database all the user requests performed on the user interface. This layer is modified in JavaScript, PHP, Ajax and supported with a QR code API. Below in Figure3.4 and Figure3.5 are snapshots of JavaScript functions and queries used to respond to user actions.

Figure3.4 List of queries

```php
<?php
/**
 * @Author: Adwoba Bota
 * This contains the function queries that communicate with the database
 */
include_once("wrapper.php");
class functions extends wrapper
{
function functions()
{
# code...
}
function login($name,$pword){
$strQuery="select id, username, password from Lecturer where username= '$name' and password ='$pword'";
return $this-> query($strQuery);
}
function attendance($cid,$time,$code,$sid){
$strQuery ="insert into class_attendance set fk_student_id='$sid', fk_studentSession_id='$cid', student_time='$time', code='$code', attendance='p'";
return $this-> query($strQuery);
}
function viewTable(){
$strQuery=" select class_attendance.student_time, (student.fullname)
from class_attendance,student where class_attendance.fk_student_id=student.student_code";
return $this-> query($strQuery);
}
// selects the courses and the times they occur
function courses($id){
$strQuery="select course.label, course_session.id, cast(course_session.time_start as time)
as times from course, course_session, Lecturer, course_lecturer
where Lecturer.id=$id=course_lecturer.fk_lect_id and course.id=course_session.fk_course_id
and course.id=course_lecturer.fk_course_id and course.id=course_session.fk_course_id";
return $this-> query($strQuery);
}
function sessions($id){
$strQuery="select id, time_start, fk_course_id from course_session where fk_course_id= $id";
return $this-> query($strQuery);
}
function courseNames($id){
$strQuery="select course_session.id, course.label, cast(course_session.time_start as time)
as times from course, course_session where course_session.fk_course_id=course.id and course_session.id=$id";
return $this-> query($strQuery);
}
function classList($id){
    $strQuery="select student.fullname from student, course_session_student
where student.id=course_session_student.fk_student_id and fk_course_session=$id";
return $this->query($strQuery);
```

Figure3.5 List of JavaScript functions

```javascript
// script page contains functions to make operations on the dashboard faster
function list() {
  var xmlhttp = new XMLHttpRequest();
  xmlhttp.onreadystatechange = function () {
    if (this.readyState == 4 && this.status == 200) {
      myFunction(this);
    }
  };
  xmlhttp.open("GET", "allAjax.php?cmd=1", true);
  xmlhttp.send();
}

function myFunction(xml) {
  var i;
  var xmlDoc = $.parseJSON(xml.responseText);
  var table = "<table><tr><th>Student</th><th>Entry Time</th></tr>";
  for (i in xmlDoc.views) {
    table += "<tr><td>" +
      xmlDoc.views[i]["fullname"] +
      "</td><td>" +
      xmlDoc.views[i]['student_time'] + "</tr></table>";
  }

  document.getElementById("table").innerHTML = table;
}

// retrieves the list of courses
/* When the user clicks on the button,
toggle between hiding and showing the dropdown content */
function course() {
  // console.log('inside course');
  document.getElementById("myDropdown").classList.toggle("classshow");
}

// Close the dropdown if the user clicks outside of it
window.onclick = function (event) {
    if (!event.target.matches('.dropbtn')) {

      var dropdowns = document.getElementsByClassName("classdropdown-content");
      var i;
      for (i = 0; i < dropdowns.length; i++) {
        var openDropdown = dropdowns[i];
        if (openDropdown.classList.contains('classshow')) {
          openDropdown.classList.remove('classshow');
        }
      }
    }
  }
  // lecture login
function login() {
  var username = $("#username").val();
  var password = $("#password").val();

  var theUrl = "allAjax.php?cmd=2&username=" + username + "&password=" + password;
  $.ajax(theUrl, {
    async: true,
    complete: loginComplete
  });
}

function loginComplete(xhr, status) {
  if (status != "success") {
    alert("error");
```

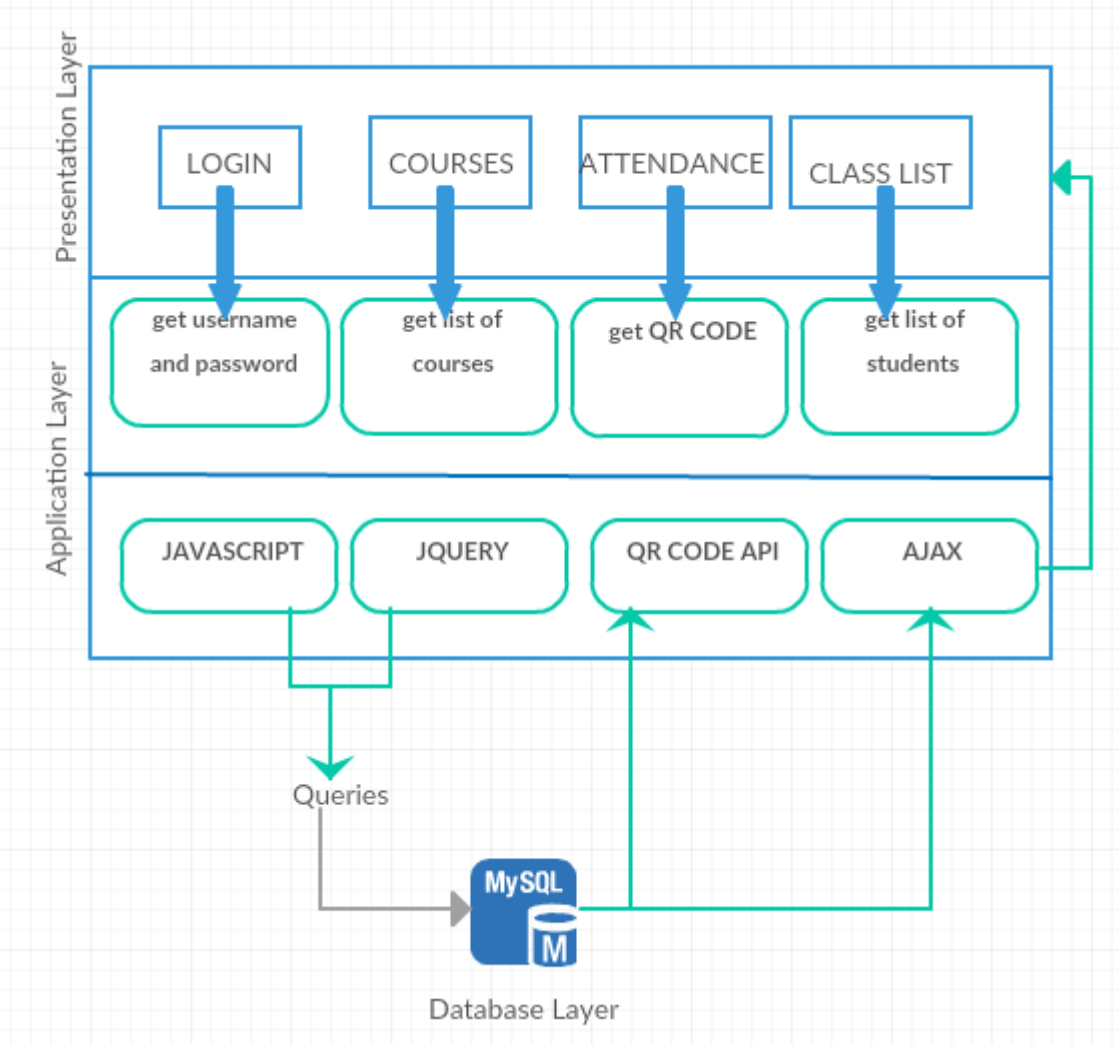
The data access layer contains all user related data. This layer is modified using SQL language. Figure3.6 shows connection to the database.

Fig3.6 Database Connection

```php
<?php
/**
*Database connection
*/
include_once("settings.php");

class wrapper{
    var $db=null;
    var $result=null;

    function wrapper(){
    }
    /**
    *Connect to database
    *@return boolean true if connected else false
    */

    function connect(){
        //connect
        $this->db=new mysqli(DB_HOST,DB_USERNAME,DB_PASSWORD,DB_NAME);
        if($this->db->connect_errno){
            //no connection, exit
            return false;
        }
        return true;
    }
    /**
    *Query the database
    *@param string $strQuery sql string to execute
    */
    function query($strQuery){
        if(!$this->connect()){
            return false;
        }
        if($this->db==null){
            return false;
        }
        $this->result=$this->db->query($strQuery);
        if($this->result==false){
            return false;
        }
        return true;
    }
    /*
    * Fetch from the current data set and return
    *@return array one record
    */
    function fetch(){
        //Complete this funtion to fetch from $this->result
        if($this->result==null){
            return false;
        }

        if($this->result==false){
            return false;
        }

        return $this->result->fetch_assoc();
```

A lecturer may log in to take attendance. Once logged in, the lecturer will be redirected to the dashboard where he or she can perform the functional requirement as mentioned in section 3.4. Below in Figure3.7 is a diagram showing how the three layers communicate to send user the required feedback necessary.

Figure3.7 A three tier architecture of Roll Call Application



Chapter 4: Implementation

This section describes the various tools and techniques used in creating an attendance management system for the school. The section includes a description on the various layers; presentation, application and database layer. It also contains tasks performed by the system.

4.1 Tools

Presentation Layer
HTML

Hypertext Mark-up Language (HTML) was used to model the client side of web pages. It can easily be used together with other languages to create dynamic user applications. It is widely used by programmers and is recommended by World Wide Web Consortium (W3C) (Rouse, 2005).

CSS

Cascading Style Sheets is used together with HTML to create web pages. CSS provides the framework or layout for how text, images and videos appear on a website. it also used to describe how colours and fonts are used on webpages and is primary responsible for making responsive pages. CSS is widely used by many developers and therefore has wide support from its community.

Materialize Framework

Materialize framework is a CSS framework for creating responsive pages. It works well with other scripting languages, google fonts and font awesome. It can easily be modelled for use on web pages and application. Materialize framework has its own default CSS styling, JavaScript files, icons and fonts. It can be used to build both web and mobile

applications. Materialize framework is however relatively new and support on its use is limited to a smaller community than other frameworks

Font Awesome

Font Awesome provides a wide range of icons which can be modified using CSS. It is an open source tool and can be used for commercial purposes. It also works well on all browsers (Gandy,n.d.).

Application Layer
PHP

Hypertext Preprocessor is a server side scripting language which is integrated with HTML to create dynamic web pages. PHP also manages request to and from the database with structure query language (SQL). PHP easily be integrated with other web languages which makes it suitable for creating web pages.

JavaScript

JavaScript is a scripting language used for creating dynamic web pages and applications. It is easier to create user response functions or actions using JavaScript. It is also well supported on many browsers and does not need a plugin to function effectively.

QR code API

This is a simple API used to generate QR codes (Tewolde, & Arase, 2011).

Database Layer
This layer shows where data on attendance is kept and how it is kept.

MySQL

MySQL is widely used by many people and is compatible with a wide variety of platforms (Linux, Windows, Unix). MySQL is open source database system and has support

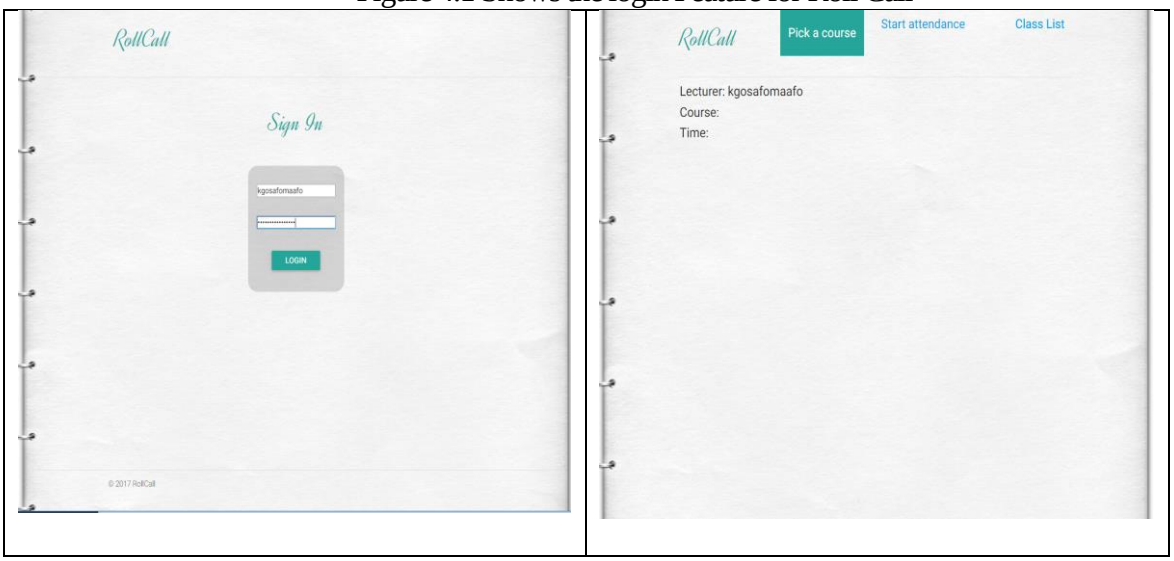
from a dedicated community. It has high performance and is suitable for both small and big data storage. It is also easy to manage and the cost of ownership is very low. In comparison to MS server, MySQL is friendlier and easy to manipulate. It can host multiple servers at a time and the syntax for use is easier to learn.

4.2 Procedures

Login

Lecturers log into the system before they begin attendance for the day. They do this by entering their credentials such as username and password. Below in figure 4.1 is are pictures of the login feature on the application. When a lecturer logs into the application. The second picture (showing the dashboard) is loaded. The lecturer can then pick a course and precede with attendance for that day.

Figure 4.1 Shows the login Feature for Roll Call

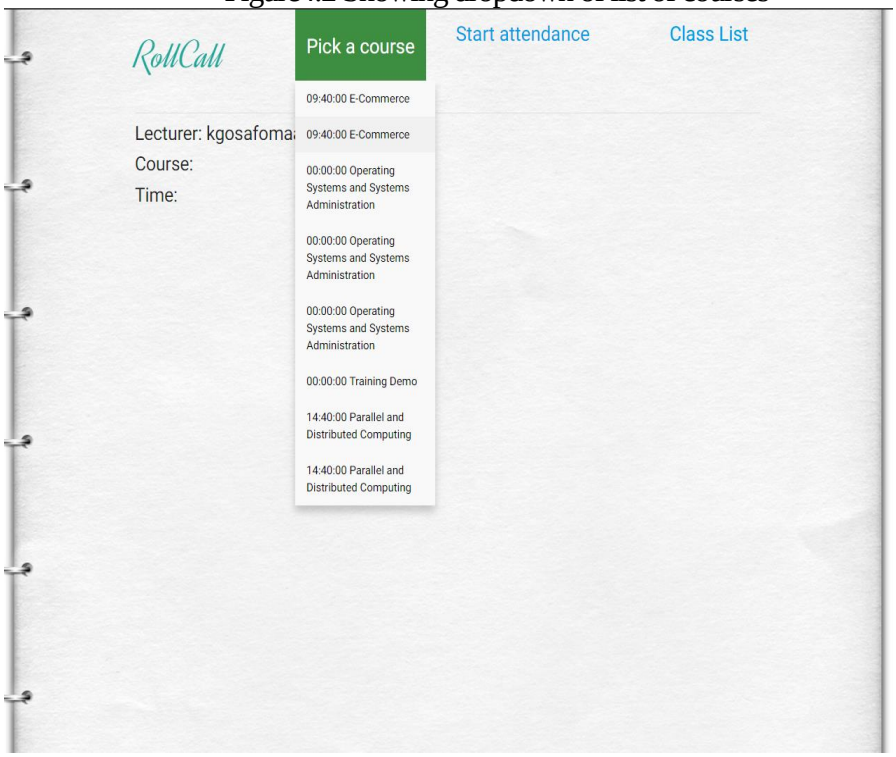


Course Selection

Lecturers who teach multiple courses can click on the “pick a course” button on the dashboard. This will trigger a dropdown of all the courses the lecturer teaches. Based on the course that the lecturer wants to make attendance for, he or she makes her selection

and proceeds with the attendance session. Figure 4.2 shows a dropdown of courses by a lecturer.

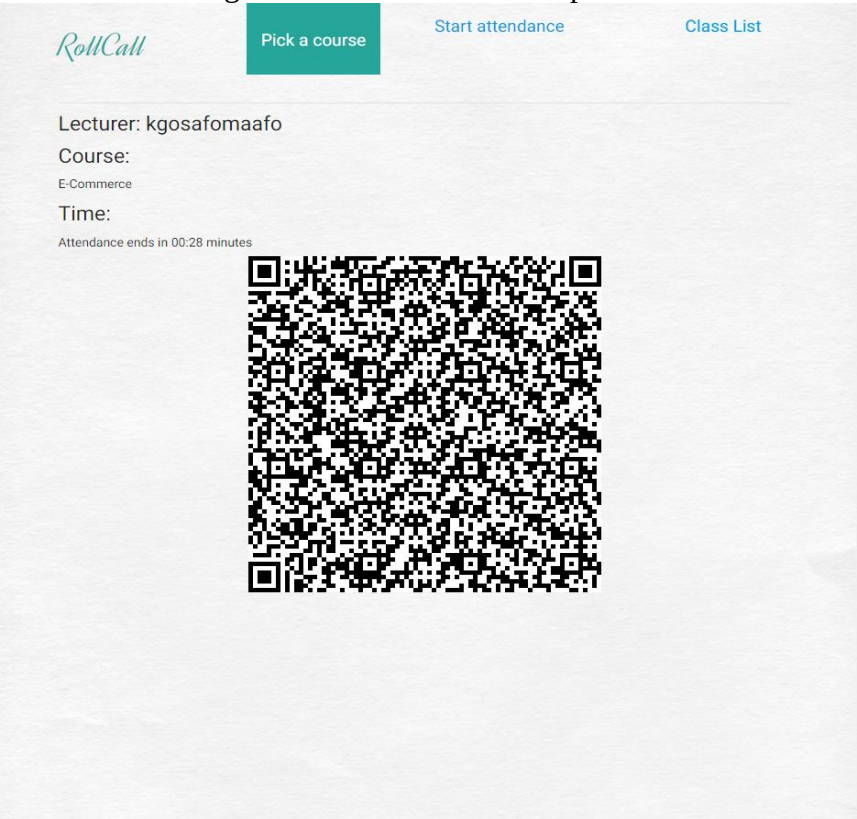
Figure4.2 Showing dropdown of list of courses



Attendance

Lecturers start attendance by clicking on “start attendance” on the dashboard. This will result in the creation of a QR code. The QR code is timed and will be unavailable for scanning once the timer has elapsed. Below is a figure showing attendance in progress.

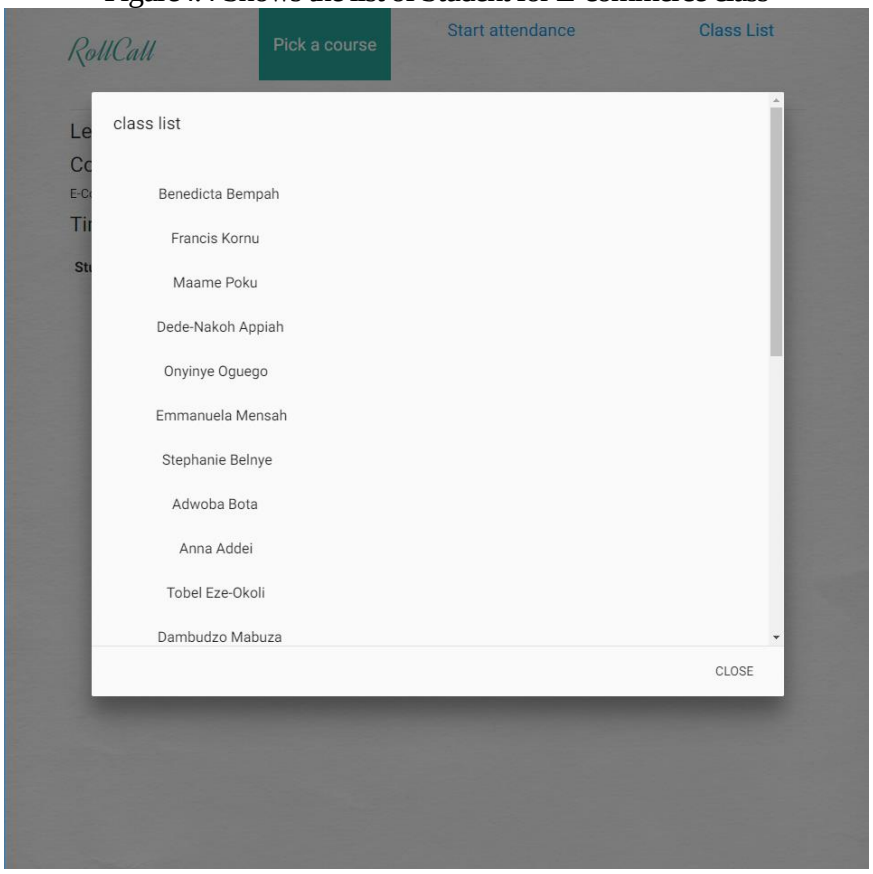
Figure 4.3 Shows attendance in process



Class List

This feature gives the lecturer an overview of all students in his or her class. To view the list of students in a lecturer's class, he or she can click on the "class list" on the dashboard. This will trigger a dropdown of all students in the class. Below is a figure showing the class list of students registered in the class of a lecturer.

Figure4.4 Shows the list of Student for E-commerce class



Chapter 5: Testing and Result

5.1 Development Testing

This section contains all tests conducted during the development stages of the application. The various units were tested before they were integrated with the rest of the application.

Unit testing

Unit testing is focused on testing functionality of the various methods in application. The test conducted included tests on search and login functionalities. In both functionalities, users can successfully login and search as required. All the buttons respond accurately when clicked. Users can select the appropriate course from the list of courses presented.

Component testing

Users use their phone cameras to capture barcode for attendance. The student list loads automatically unto the page after attendance. Users can remove students from attendance list where necessary.

System Testing

This is the final test done on the system before deploying for test with users. With system testing, the entire system is tested against user requirements. In the case this application (Roll Call). The system must satisfy five functional requirements as specified in section 3.4. To meet these requirements, the database is loaded with data of some classes, lecturers and students. (Before this, dummy data was being used for testing). The system was then tested for response speed and performance. The system satisfied all functional requirements listed.

User Testing

Some students in fourth year with QR code scanners on their phones were used for the testing. They were required to scan and enter their student ID to complete the attendance process. Both iPhone users and android phone users could successfully scan the QR codes.

Limitations

Some QR code scanners are slower than others and are unable to capture the code quick enough. This will therefore influence the time reported for student presence in class.

At further distances, the code captured from the QR code can be wrong. This is a limitation on the QR code scanners rather than on the QR code generator (API for generating the QR code). This is because at further distances the scanner is not able to capture all the distinct dots in the QR code

Another limitation is that the quality of the projector used in projecting can affect the kind of feedback read by the scanner. This will reduce the extent to which any projector can be used to take attendance.

Below in figure 5.1 and 5.2 are the user interface and code for taking student attendance.

Figure 5.1 Shows a prompt for student ID for attendance in class

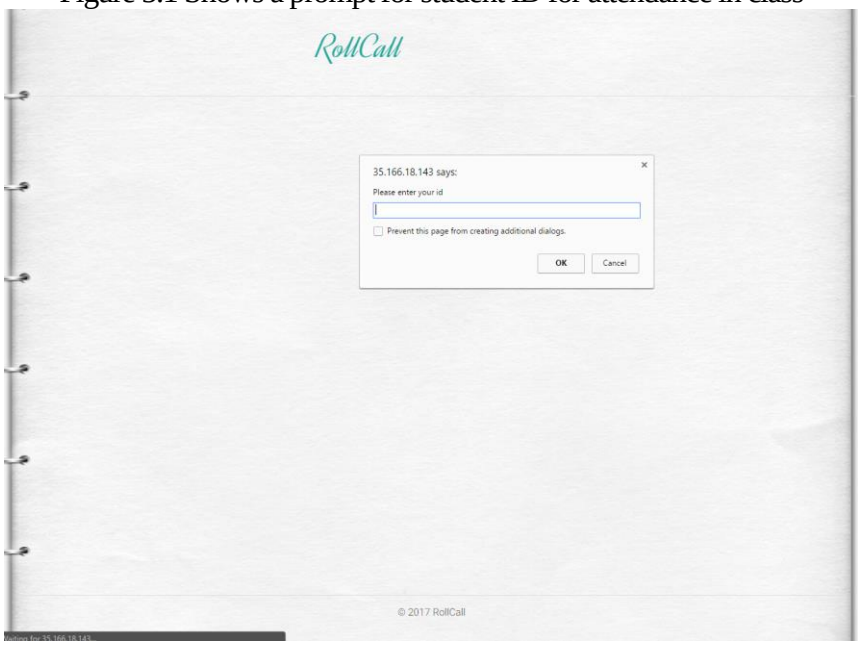


Figure 5.2 shows the code for generating the student side of the attendance

student.php

```
<!DOCTYPE html>
<html>
  <head>
    <!--Import Google Icon Font-->
    <link href="http://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
    <link href="http://fonts.googleapis.com/icon?family=Dynalight" rel="stylesheet">
    <!--      use materialised-->
    <link type="text/css" rel="stylesheet" href="css/materialize/css/materialize.css"  media="screen,projection"/>
    <!--    font awesome-->
    <link rel="stylesheet" href="css/font-awesome-4.6.3/css/font-awesome.min.css">
    <!-- my css page -->
    <link rel="stylesheet" type="text/css" href="css/myCSS.css">
    <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
    <style type="text/css" media="screen">
      body { text-align:center; }
      body {
      background: url("images/background.jpg");
      background-size: 100% 100%;
      background-repeat:   no-repeat;
      top: 0;
      }
      body {
      display: flex;
      min-height: 100vh;
      flex-direction: column;
      }
      main {
      flex: 1 0 auto;
      }
    </style>
  </head>
  <body onload="studentId()">
    <header>
      <div class="row">
        <div class="col s6 offset-s2">
          <h1 style="font-family: 'Dynalight';color:#26a69a">RollCall</h1>
        </div>
      </div>
      <div class="divider"></div>
    </header>
    <main>
      <p id="demo"></p>
      <script>
        function studentId() {
            var stu_id = prompt("Please enter your id");
            student(stu_id);
        }

        function student(stu_id){
          var courseId=sessionStorage.courseId;
          var code=sessionStorage.random;
          var xmlhttp = new XMLHttpRequest();
          xmlhttp.onreadystatechange = function() {
            if (this.readyState == 4 && this.status == 200) {
              studentComplete(this);
            }
          };
          xmlhttp.open("GET", "allAjax.php?cmd=4&student="+stu_id+"&courseId="+courseId+"&code="+code, true);
          xmlhttp.send();
        }

        function studentComplete(xml){
          var xmlDoc=$.parseJSON(xml.responseText);
          if(xmlDoc.result==0){
            document.getElementById(studentId)="Your attendance could not be completed";
          }
          else if(xmlDoc.result==1){
              document.getElementById(studentId)="Attendance is completed";
          }
        }

      </script>
    </main>
    <footer class="page-footer" style="padding:25px;">
      <div class="divider"></div>
      <div class="footer-copyright">
        <div class="container">
          © 2017 RollCall
        </div>
      </div>
    </footer>
  </body>
</html>
```

5.2 Compatibility Testing

The application is built and tested on Google Chrome and is therefore the only known browser to be compatible to the web application now.

5.3 Challenges

The main challenge was the use of JavaScript functions to make the web application dynamic. JavaScript is very new to me and so its use means learning how to implement and debug for easy use. This slowed down the progress of the application. The result of using JavaScript is however worthwhile.

Another challenge was building a well-equipped database. Although the database was built on an already existing database, it involved modifying to be able to add an

attendance module. This was not easy because of the restrictions placed on the database by the already existing one.

Chapter 6: Conclusion and Recommendations

6.1 Conclusions

This application is made for taking attendance in class, it however has features that makes it suitable for taking attendance anywhere record keeping on attendance is valued. Organisations can therefore easily integrate it into their work to ensure effective record keeping.

6.2 Recommendations

An attendance management system should be simple, fast and user friendly. Attendance taking should not eat into the lesson time; to ensure this, the method used should be easy to apply and quick in recording attendance during a class session. It will also be interesting to make the method of attendance fun for those involved.

In the near future, to make the application faster and simpler, the attendance feature should allow attendance to be completed using facial recognition. Here, a picture of students present in the class during attendance will be captured and authenticated in the database as required. This will reduce the attendance process to about a minute.

6.3 Future works

This project was emphatically targeted at reducing attendance time hence to make it even faster the application can be remodelled to use facial recognition. The lecturer or faculty intern will take a snapshot of students present in the class during attendance. This will be authenticated in the database as required. Using facial recognition will reduce the time spent on attendance to about a minute. It will also reduce human interference on the system and hence eliminate chances of students signing up for others who may not be present. It will also be effective in ensuring promptness of students to class.

Bibliography

1. Abas, M., Tuck, T., & Dahlui, M. (2015). *Attendance Management System (AMS) with fast track analysis - IEEE Xplore Document*. *Ieeexplore.ieee.org*. Retrieved 15 October 2016, from http://ieeexplore.ieee.org/document/7042597/

2. Akram, F. & Rustagi, D. (2015). *An efficient approach towards privacy preservation and collusion resistance attendance system - IEEE Xplore Document*. *Ieeexplore.ieee.org*. Retrieved 18 October 2016, from http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7375285

3. Chen, J., & Lin, T. F. (2008). Class attendance and exam performance: A randomized experiment. *The Journal of Economic Education*, *39*(3), 213-227

4. Clay, T., & Breslow, L. (2006). *Why Students Don't Attend Class*. Web.mit.edu. Retrieved 15 October 2016, from http://web.mit.edu/fnl/volume/184/breslow.html

5. Gandy, D. *Font Awesome, the iconic font and CSS toolkit*. *Fontawesome.io*. Retrieved 12 April 2017, from http://fontawesome.io/

6. Khan, A., Abeer, A., Afzal, A., Malik, K., & Ahmed, S. (2007). *Digital Attendance Recording System - IEEE Xplore Document*. *Ieeexplore.ieee.org*. Retrieved 5 October 2016, from http://ieeexplore.ieee.org/document/4381320/

7. Rouse, M. (2005). *What is HTML (Hypertext Markup Language)? - Definition from WhatIs.com*. *SearchMicroservices*. Retrieved 8 April 2017, from http://searchmicroservices.techtarget.com/definition/HTML-Hypertext-Markup-Language

8. Tewolde, A., & Arase, K. (2011). *amanuel/JS-HTML5-QRCode-Generator*. *GitHub*. Retrieved 8 April 2017, from https://github.com/amanuel/JS-HTML5-QRCode-Generator