MOBILE VIRTUAL LECTURE MANAGEMENT SYSTEM

(CASE STUDY: DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF IBADAN)

A PROJECT CARRIED OUT AS A PARTIAL REQUIREMENT FOR THE AWARD OF B.sc DEGREE IN THE DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF IBADAN

BY

ONIOSUN AYODELE ABRAHAM

MATRIC NO:192477

SUPERVISOR: DR T. OGUNTUNDE

SESSION: 2017/2018

CERTIFICATION

I certify that this work was carried out by **ONIOSUN AYODELE ABRAHAM,** with matric number: **192477** in partial fulfilment for the award of a B.sc in the department of Computer Science, University of Ibadan, under my supervision.

____________________________	_______________________
Dr T. Oguntunde (Supervisor)	**Date**
______________________________	_______________________
Dr S.O Akinola (Head of Department)	**Date**

DEDICATION

This project work is dedicated to the Spirit of the Living God, for He is the giver and the brain behind the idea of this project.

ACKNOWLEDGEMENT

Better is the end of a thing than its beginning thereof. (Ecc. 7:8)

My greatest gratitude goes to the Lord God Almighty, the one who truly knows the end from the beginning, for his goodness, mercies, favour and protection over me from my infant years till this moment and for ordering my steps in truth and wisdom throughout my academic pursuit in the University of Ibadan.

My sincere appreciation goes to my parents, Mr and Mrs O.A Oniosun, for their prayers, sacrifices and their unflinching support in every way in times of little and none.

To my ever wonderful and priceless younger ones; Oniosun Oluwafemi Joseph, Oniosun Oluwapelumi Daniel, Oniosun Temiloluwa Mary, you are really an epitome of love. I couldn't ask for better siblings.

I will also be an ingrate if I fail to appreciate my supervisor, Dr. T. Oguntunde, for his relentless guidance, encouragement and assistance during the duration I was under his supervision for this project work. His wise counsel and advice even in non-academic issues went a long way.

I deeply appreciate the effort of my wonderful lecturers, Prof. Adenike O. Osofisan, Dr A.B Adeyemo, Dr O. Osunade, Dr. B.O Akinkumi, Dr A.B.C Robert, Dr O.F.W Onifade, Dr S.O Akinola (Head of Department), Dr Angela U. Makolo, Dr Bolanle F. Oladejo, Dr Yetunde Folajimi, Dr O.P Akomolafe, Dr I.T Ayorinde, Dr T. Oguntunde (My Supervisor), Dr O.D Adeniji, Dr Mrs Ojo Adebola, Dr Mrs. C.N Woods, Dr A.O Olamiti, Dr Mrs. Sakpere Aderonke, for their impartation of knowledge, and also to all the non-teaching staff, I say Thank You.

My memory will not fail to appreciate the efforts of some of my close friends and elder ones, Adeniran Segun, Osunniyi Kolawole, Dominic Cletus Ephraim, Oyebanji Opemipo, Akanbi Yusuf Kehinde, Damisa Emmanuel Sylvester, Ekpeyong Paul Gabriel, Ojetunde John, Oyeyemi Dare, Okewole Taiwo, Adeosun Timothy, Olasupo Oluwadara, Ifedapo Anuoluwapo, Ayodeji Femi Sunday, Alabi Boluwatife, Oyebisi Adetoye, Dada Temitope, Faleye Ayodeji, Ogundokun Festus, Okegbile Julianah, Raji Opeyemi, Adedokun Funmilayo. Meeting you in life has been one of the greatest thing that happened to me apart from my salvation.

My last appreciation goes to all those who were there for me during the course of working on this project. May God bless you and perfect everything that concerns you.

Table of Contents

ABSTRACT

The conventional paper-based method of taking attendance by calling names or signing on paper register is very time consuming and insecure. A student who is present can take attendance for his colleague who is absent and this leads to falsification of attendance. Course materials and assignments do not always circulate round to the entire students due to the inefficiency on the part of the class or course representative. Assignment sheets submitted by the student are mutilated and sometimes lost.

Mobile Virtual Lecture Management System (MVLMS) was designed and implemented to provide a lasting solution to the aforementioned problems. Attendance can be taken by students using Global Positioning System (GPS), course materials and assignments are stored on the cloud. Student can submit assignment and can be graded as well.

Incremental approach of software development was adopted and modelling was done using Use case, Data Flow Diagrams (DFD), Entity Relationship Diagram (ERD) and Entity Relationship Model (ERM).

Programming technologies such as React Native, Native Base, Lumen and MySQL are used to bring this idea into fruition.

This project when fully developed will provide curb the falsification of attendance reports, improper dissemination of course materials, mutilation and outright loss of assignment.

CHAPTER ONE

INTRODUCTION

1.1 BACKGROUND OF STUDY

Students' lecture attendance, generation of attendance reports and submission of assignments in most academic institutions can be rigorous, inefficient and inaccurate using the conventional method of paper-based system. The conventional paper-based system for capturing student's lecture attendance, resources and assignment submission have been in use for many years even before the advent of technology and this is not only prone to errors, but also tedious, time-consuming and insecure. The world is gradually going mobile and having to logon to a desktop computer before performing an operation might be uncalled for at times and can be stressful. This project will harness the use of android mobile technology for solving the aforementioned problems.

The use of GPS (Global Positioning System) technology in android phones will be adopted for marking attendance of students in each course. Any student who is not within the coordinate of venue of the lecture will be disabled from taking attendance for any course, thereby eradicating the falsifications that arises when a student mark attendance for students who was absent in class. Generation of attendance reports can also be generated periodically with this platform as this will enable students to know their current attendance percentage and will help them to know if they will most likely write examinations for each course, having reached a minimum of 75% attendance.

Moreso, course materials which might not be able to circulate round in the class will be catered for in this project as students can easily logon to their portal to read and download materials uploaded by the course lecturer.

Finally, submission and grading of assignments on-the-go will also be incorporated into this project, thereby eradicating the loss of any paper assignments.

1.2 PROBLEM STATEMENT

Students' lectures attendance and assignments are prone to falsification, mutilation, misplacement and even outright loss in cases of natural disaster such as floods and fire-outbreaks.

Lecture resources do not usually get to most students through the course or class representative.

1.3 AIMS AND OBJECTIVES

The aim of this project is to design and implement a mobile and virtual lecture attendance, course materials and assignment management system.

The objectives are:

a) To design a model for a reliable, scalable GPS based attendance management system for students.

b) To implement the proposed model for an accurate report generation for class attendance.

c) To implement an efficient storage system for course materials.

d) To implement a secure and safe system for submission and keeping records of assignments.

1.4 EXPECTED OUTCOME OF THE PROJECT

Upon completion of this project, faster, reliable and efficient means of marking and tracking records of attendance of students will be ensured. It will also provide an efficient storage system for course materials and a secure means of submission and grading of assignments.

CHAPTER TWO

LITERATURE REVIEW

2.1 THEORETICAL BACKGROUND

Attendance maintenance is an important task in all the institutions to check the performance of students. Every institute has its own way to do so. Some use the old paper or file based approach and some have adopted methods of automatic attendance using different techniques. Many systems have been developed to store and maintain the attendance records. These various systems include different methods and technologies. Some attendance management systems are based on Biometric Technology (Seema Rao, Prof. K. J. Satoa, 2013) which marks the attendance using fingerprints and face of students. In some systems, attendance is stored and maintained using RFID technology (Arulogun O. T., Olatunbosun et al, February 2013) and some uses combination of RFID (Radio Frequency Identification) technology and Face Recognition (Unnati A. Patel, et al, 2014).

RFID is a technology that uses radio waves to transfer data from an electronic tag, called RFID tag or label, attached to an object, through a reader for the purpose of identifying and tracking the object. In 1945, Leon Theremin invented an espionage tool (for spy activities) for the Soviet Union which retransmitted incident radio waves with audio frequency information. In computer security, biometrics refers to authentication techniques that rely on measurable physical characteristics that can be automatically checked. There are several types of biometric identification schemes which include retina, hand geometry, vein, voice etc. The computer uses any of these biometric identification schemes to determine who you are, and based your identity. (Mary Catherine O'Connor, 2009). Fingerprint identification refers to specifying one’s identity based on his fingerprints. The fingerprints are captured without any information about the identity of the person. It is then matched across a database contain innumerous fingerprints. The identity is only retrieved when a match is found with one existing in the database.

Moreso, almost all universities include some form of assignment in their courses. The assignments are either carried out in either in groups or individually. Assignment management involves collecting, marking, and redistributing to students. A. Tregobov, 1998, breaks the process down into four stages: submission, recording, marking, and return. To effectively manage these submitted assignments, a well-designed assignment submission system is needed, hence the need for an online assignment submission system to facilitate the distribution, and collection of assignments on due dates. Many submission systems, which have alleviated the tasks of lecturers in several ways, exist. However, most of them could not meet all required criteria for an ideal submission system. The inherent problem with an email based submission system (Godfrey B (1997) lies in its security and its capacity for attaching files. Most existing submission system do not allow for structuring of files, automatic correction of automated assessment (University of Queensland Australia). Also, some cater only for one type of assignment (NYU, Undergraduate Division, Computer Science) while others (Linnaeus University Sweden) do not put emphasis on due dates and students can submit assignments even after submission date is over. All these problems have led to the development of an online submission system with desirable features.

The physical copies of files were difficult to deal with. There were a number of problems that are faced by keeping hard copies of course materials. It might be stolen, misplaced or mutilated. On top of this, there was a threat of disaster such as flood or fire outbreak. With the advent of technology, course materials can be uploaded in the cloud by the lecturer, while the student can download and read at any time.

2.2 DEFINITION OF TERMS

In a bid to understand what lecture management systems is, it is expedient that we explain the individual sub-systems that make up this system, alongside some definition of the following terms:

VIRTUAL: This simply refers to something that is simulated on a computer or on-line rather than really existing.

LECTURE: This is an oral presentation intended to present information or teach people about a particular subject. Usually, the lecturer will stand at the front of the classroom and recite information relevant to the lecturer's content.

ATTENDANCE: This refers to the concept of people, individually or as a group, appearing at a location for a previously scheduled event.

COURSE: This is a unit of teaching that typically lasts one academic term, is led by one or more instructors and has a fixed roster for students. Students receive a grade after the completion of the course.

COURSE MATERIALS: This consists of lecture notes, supplementary reading or exercises, and links to other online resources.

ASSIGNMENT: This refers to a set of tasks assigned to students by their lecturers to be completed outside the class and must be submitted on a definite date and time.

SYSTEM: A system is an interrelated set of business procedures (or components) used within one business unit, working together for some purpose (Valacich, Fulelr & George, 2008). Information on the other hand can be viewed as a type of input to an organism or system (Dusenberyy, 1992).

2.3 COMPONENTS OF LECTURE MANAGEMENT SYSTEMS

Various lecture management systems exist today that are continually updated to ensure the efficiency, accuracy and security of how various operations such as marking of attendance, dissemination of course materials and submission of assignments are performed.

The advent of mobile technology has automated a lot of manual and herculean tasks and has cancelled a lot of conventional, paper-based systems of operations.

2.3.1 ATTENDANCE MANAGEMENT SYSTEM

Attendance is a daily activity performed by every institute and college to maintain the records of student's attendance. This record is then used by the faculty of the college whenever needed. In most universities, teachers take attendance by calling out the names of students, and then marking them. This practice has its drawbacks, in the sense that, friends of absent students may write down their names and surnames. Thus practice place university teachers and their institutions at considerable disadvantages when it comes to taking attendance.

The features of many systems have been developed to store and maintain the attendance records will be discussed here.

2.3.1.1 RFID BASED ATTENDANCE MANAGEMENT SYSTEM

Radio-frequency identification (RFID) is a technology that uses radio waves to transfer data from an electronic tag, called RFID tag or label, attached to an object, through a reader for the purpose of identifying and tracking the object. RFID is a matured technology that incorporates the use of electromagnetic or electrostatic coupling in the radio frequency portion of the electromagnetic spectrum to uniquely identify an object, animal, or person. RFID chips contain a radio transmitter that emits a coded identification number when queried by a reader device. Some RFID tags can be read from several meters away and beyond the line of sight of the reader.

The tag's information is stored electronically. The RFID tag includes a small RF transmitter which transmits an encoded radio signal to interrogate the tag, and receiver which receives the message and responds with its identification information. Some RFID tags do not use a battery. Instead, the tag uses the radio energy transmitted by the reader as its energy source. RFID can be used in many applications. A tag can be affixed to any object and used to track and manage inventory, assets, people, etc. For example, it can be affixed to cars, computer equipment, books, mobile phones, etc. The Healthcare industry has used RFID to reduce counting, looking for things and auditing items. Many financial institutions use RFID to track key assets and automate compliance. Also with recent advances in social media RFID is being used to tie the physical world with the virtual world. RFID in Social Media first came to light in 2010 with Facebook's annual conference (Amirjan Bin Nawabjan, 2009).

The RFID attendance system is an automatic embedded system used in taking attendance of registered persons in a particular organization. The RFID attendance system offers an organization, the efficiency and convenience associated with RFID technology at a low cost. This method is fast as well as simple. Each employee or student uses an RFID card and the reader records the data when the employee enters or exits. RFID devices and software must be supported by a sophisticated software architecture that enables the collection and distribution of location based information in near real time. RFID system is used to record the numbers of employees' or students' attendance automatically. The ID cards of the employees is embedded with RFID tag which is read by a reader. This RFID system is interfaced to a database through a computer. This method is more effective to prevent problem encountered when getting attendance manually. Below is the block diagram of an RFID attendance system.

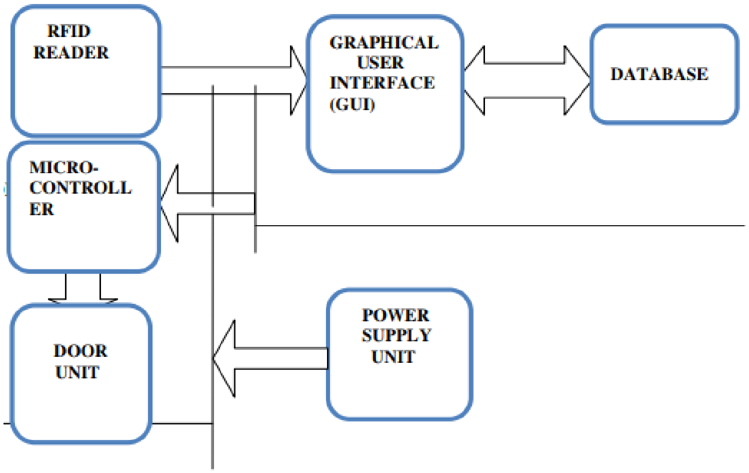


Fig 2.1 Block diagram of RFID attendance system with door unit. (Mohd F.B, 2008)

2.3.1.1.1 APPLICATIONS OF RFID TECHNOLOGY IN NIGERIA

The RFID technology has been suggested to be applied in various sectors in Nigeria to reduce corruption. These sectors include:

- The monitoring of crude oil.
- Tracking the movement of money.
- Drug agencies and monitoring containers entering the country.

An RFID technology company in Nigeria started providing anti-money laundering (AML) solutions to banks. The technology tracks the movement of money and alerts the Nigerian Financial Intelligence Unit (NFIU) of any suspicious activities. The company is providing the service to five major Nigerian banks. This technology has been used by NAFDAC to reduce the importation and distribution of counterfeit drugs. (Bermuda's RFID Vehicle Registration System, 2008).

2.3.1.2 FINGERPRINT BASED ATTENDANCE MANAGEMENT SYSTEM

Fingerprint belongs to a group that is called "Biometrics". Biometrics represents automated methods for person identification based on a physiological or behavioral characteristic. Among the features measured are: facial features, fingerprints, iris and retina features, hand veins, hand geometry, handwritten signature, keystroke dynamics and voiceprint. According to C. Saraswat, C. et al, fingerprints are represented by the pattern of ridges and furrows on the surface of a fingertip. The fingerprints are unique and the patterns remain unchanged throughout life. Fingerprints are so distinct that even the ones of identical twins are different. The prints of each finger of the same person are also different.

The fingerprint scanner captures an image of the fingerprint and uses complex fingerprint identification algorithms to either convert the image into a unique "map" of minutiae points or analyze the pattern. Fingerprint technology has been spread widely in various fields in the life. The main field historically was the criminal investigations in which, fingerprints of criminals are recorded in database in order to specify their identities later on. (Liu F and Zhao Q, 2011).

2.3.1.2.1 FINGERPRINT ACQUISITION PROCESS

The fingerprint acquisition process can be divided into three parts, namely:

- The enrolment process.

This is done once for each person. Each person would be required to register their fingerprint pattern by placing his thumb finger on the fingerprint scanner. The scanner takes the image of the finger and determines the unique characteristics of the fingerprint image. The fingerprint contains ridges and valleys which have different kinds of breaks and discontinuities. It is the various ridges and valleys that form the basis for the loops, arches, and swirls that are easily seen on fingertips (S. Pankanti, 2002). After the capturing of the ridge pattern of the fingerprint, a template is created, and the fingerprint is encrypted into series of numbers. These series of numbers will be different for each fingerprint pattern. After the process must have been completed, the fingerprint scanner sends the result of the encryption to a memory location or database.

- The verification process.

  This is the second and the most repeated process. It is done each time the user wants to make use of the fingerprint controlled device. When he places his finger on the fingerprint scanner surface, the fingerprint would be processed by the fingerprint scanner. The fingerprint pattern that has been obtained would be compared against the stored enrolment template that is already stored in the database or memory location where the enrolment process was executed. When the fingerprint pattern passes the comparison process, it shows an acknowledgement in its display and grants the user access.

- The data collection process.

  The last process that will be done is the data collection process. The data about the fingerprint device usage or record can be collected after a period of time and can be used as a form of record to know the attendance of a person or to know the number of times he or she is restricted. (A.K. Jain, 2002).

2.3.1.2.2 FINGERPRINT RECOGNITION ALGORITHM

Generally, fingerprint readers operate on one of two fingerprint recognition algorithms, which are:

- **Fingerprint identification**: This is to specify one person's identity by his fingerprint(s). Without knowledge of the person's identity, the fingerprint identification system tries to match his fingerprint(s) with those in the whole fingerprint database. It requires more processing since the applied fingerprint is to be compared with those in the whole fingerprint database. This process could take significant time, especially in the case of large databases such in case of faculty or university containing thousands of students.

- **Fingerprint verification:** This is to verify the authenticity of one person by his fingerprint. The user provides his fingerprint together with his identity information like his ID number. The fingerprint verification system retrieves the fingerprint template according to the ID number and matches the template with the real-time acquired fingerprint from the user. This requires less processing since the applied fingerprint is to be compared with a single template in the database specified by the ID. (M. Kamaraju, 2013).

2.3.1.2.3 COMPONENTS OF FINGERPRINT-BASED SYSTEM

- **SCANNER:** The most important part in any fingerprint-based system is the scanner or sensor which is the 'gate' of the fingerprint into the system.

  Two types of scanners have been viewed:

  - **Optical Scanner**: Optical scanners use a Charge Couple Device (CCD) much like the ones used in camcorders and digital cameras. The CCD makes use of photo sites that are sensitive to light. Each photo site generates an electrical signal in the presence of photons produced by light emitting diodes. These photo sites are very small they cover the entire screen where a user will place his or her finger.

    Since photo sites only emit electricity in response to the presence and the strength of light, the overall result will be an inverted image of the finger. This image is converted into digital form and checked for clarity and sharpness before being compared to the saved images of other prints in the database. (P. Verma, 2013).

  - **Capacitance scanners**: This make use of electrical current to form a fingerprint image. The sensor part of the capacitance scanner is made up of tiny plates that act as conductors. These conductors are then overlaid by an insulating film. These form the cells that make up the sensor's semiconductor chip or chips. An inverting operational amplifier is then used in conjunction with an electrical circuit and the sensor to determine the relative differences in capacitance and voltage value of the different areas of a finger. A scanner processor is then used to separate the valleys and the ridges in the resulting data. This results in a reverse image of a fingerprint. From previous context it can be shown that capacitance scanners are more secure than optical ones since they have higher ability to detect fake fingerprints. This is due to that capacitance scanners depend on the resulting capacitance of the portions of the finger that are in contact with the sensor plus the air capacitance on the ridges of a finger instead of just relying on the amount of reflected light like what the optical scanners do. A picture of a fingerprint pressed onto the sensor, can trick an optical scanner that it is processing a real fingerprint. On the other hand, optical scanners have advantages of:

1. Resistance to shock and extreme weather.
2. Lower maintenance cost.
3. Bigger sensing area leading to better resolution.

Fig 2.2 Image of a fingerprint (A. Kumar. et al, 2003)

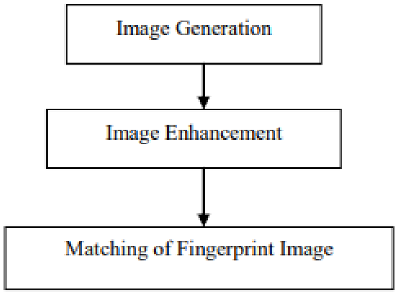


Fig 2.3 Flowchart to identify fingerprint image (Parul D. Sindha, 2012)

2.3.1.3 BARCODE BASED ATTENDANCE MANAGEMENT SYSTEM

Barcode Based Student Attendance System is a software which utilizes barcode scanner to record and maintain the attendance of the students. The main hardware that is to be used is the barcode scanner. This barcode scanner is used in order to read a barcode. A Barcode is a machine readable representation of information in a visual format. A bar code consists of a series of parallel, adjacent bars and spaces.

The types of Barcode are:

- **Linear Barcodes:** A first generation, "one dimensional" barcode that is made up of lines and spaces of various widths that create specific patterns.
- **Matrix Barcodes:** A matrix barcode, also termed as a 2D barcode or simply a 2D code, is a two-dimensional way to represent information. It is similar to a linear (1-dimensional) barcode, but can represent more data per unit area. These barcodes are read using a device called as the barcode scanner.

  There are different types of Barcode Scanners:

  - **Pen-type readers:** This consist of a light source and photodiode that are placed next to each other in the tip of a pen or wand. It is the cheapest barcode scanner available in market.
  - **CCD Reader:** CCD readers use an array of hundreds of tiny light sensors lined up in a row in the head of the reader. Each sensor measures the intensity of the light immediately in front of it.
  - **Smartphone Cameras:** Smartphone cameras have the ability to read both QR codes (which automatically take you to a website URL) as well as scanning a barcode to bring up product information such as price comparisons and user reviews.
  - **Handheld Scanner:** It is a scanner with a handle and typically a trigger button for switching on the light source.
  - **Automatic reader:** A back office equipment to read barcoded documents at high speed. (T.S.Lim, M.Kassim et al, 2009)

The barcodes would be attached at the back of each student's ID card and the information that would be embedded in the barcodes will contain unique data of the student such as matriculation number, department, faculty, year and all other important information about the student. Students are able to participate in the class attendance only by swiping their identity card through the barcode reader. They can view their attendance percentage which will motivate them to control their class attendance reputation. With real time clock capability of the system, attendance taken will be more accurate since the time for the attendance taken will be recorded (Seideman T, 2009).

2.3.1.4 FACIAL RECOGNITION BASED ATTENDANCE MANAGEMENT SYSTEM

Face recognition is a technique of biometric recognition. It is considered to be one of the most successful applications of image analysis and processing; that is the main reason behind the great attention it has been given in the past several years. The facial recognition process can be divided into two main stages: processing before detection where face detection and alignment take place (localization and normalization), and afterwards recognition occur through feature extraction and matching steps. This system uses the face recognition approach for the automatic attendance of students in the classroom without student's intervention. This attendance is recorded by using a camera that captures images of students, detect the faces in images, compare the detected faces with the database and mark the attendance. (Viola P. & Jones J, 2004)

The main elements of this technology are:

- **Face Detection:** The main function of this step is to determine whether human faces appear in a given image, and where these faces are located at. The expected outputs of this step are patches containing each face in the input image. In order to make further face recognition system more robust and easy to design, face alignment are performed to justify the scales and orientations of these patches. Besides serving as the pre-processing for face recognition, face detection could be used for region-of-interest detection, retargeting, video and image classification, etc.
- **Feature Extraction**: After the face detection step, human-face patches are extracted from images. Directly using these patches for face recognition have some disadvantages, first, each patch usually contains over 1000 pixels, which are too large to build a robust recognition system. Second, face patches may be taken from different camera alignments, with different face expressions, illuminations, and may suffer from occlusion and clutter. To overcome these drawbacks, feature extractions are performed to do information packing, dimension reduction, salience extraction, and noise cleaning. After this step, a face patch is usually transformed into a vector with fixed dimension or a set of fiducial points and their corresponding locations. In some literatures, feature extraction is either included in face detection or face recognition.
- **Face Recognition**: After formulizing the representation of each face, the last step is to recognize the identities of these faces. In order to achieve automatic recognition, a face database is required to build. For each person, several images are taken and their features

are extracted and stored in the database. Then when an input face image comes in, we perform face detection and feature extraction, and compare its feature to each face class stored in the database. There are two general applications of face recognition, one is called identification and another one is called verification. Face identification means given a face image, we want the system to tell who he / she is or the most probable identification; while in face verification, given a face image and a guess of the identification, we want the system to tell true or false about the guess. (M. A. Turk, 1991).

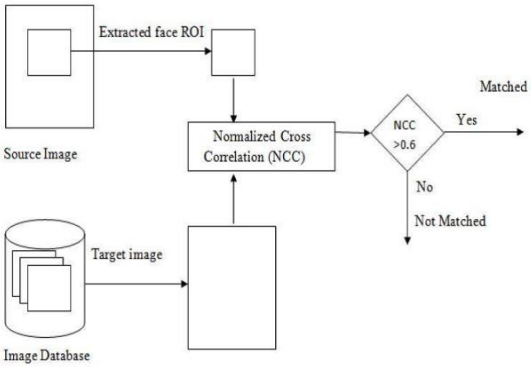


Fig 2.3 Stages of face detection (M. A. Turk,1991)

2.3.1.5 GPS BASED ATTENDANCE MANAGEMENT SYSTEM

Global Positioning System (GPS) is a satellite-enabled locating system that allows users to determine their location, regardless of topography, weather or time of day, anywhere on the globe, provided the GPS receiver is not physically obstructed. GPS was originally designed for military purposes but is now commonly used in cars and on boats. GPS receivers utilized a minimum of three of the 24 network satellites to triangulate a user's location. In the case of marine equipment, this is converted into a two dimensional display overlaid on electronic charts provided within the equipment. (Kennedy M, 2002).

The components of GPS include:

- **Space Segment**: The satellites orbiting the earth and transmitting timing and ranging messages. This consists of constellation of spacecraft and the signals broadcast by them which allow users to determine position, velocity and time.

  The basic functions of the satellites are to:

  1. Receive and store data transmitted by the control segment stations.
  2. Maintain accurate time by means of several onboard atomic clocks.

3. Transmit information and signals to users on two L-board frequencies.

- **Control Segment:** This monitors the health and position of the satellites in the space segment and transmits correct information back up to the satellites.
  This is used for maintaining the satellites and their proper functioning.
- **User Segment:** This is typically referred to a GPS receiver, processes the L-band signals transmitted from the satellites to determine PVT.

2.3.1.5.1 HOW GPS WORKS

According to Pace, Scott et.al, GPS systems in Australia rely on a combination of satellites placed in orbit around the Earth and the land based DGPS broadcasting system. Individual signals sent by each satellite and transmission station allow the GPS software to determine its relative location to each signal source and calculate a position. This relative position is then displayed using the base datum included in the GPS unit.

Usually, current software installed on a GPS unit includes geographic reference and can include aids to navigation marks and, in more advanced models, link to depth sounders to provide indications of water depth. Recent incidents involving vessel collisions with maritime or navigation infrastructure have been caused by an overreliance on GPS technology or the use of an inconsistent datum applied to the GPS unit, or the electronic chart not being updated. People are becoming more confident with road-based GPS units and are applying that same level of trust to the GPS unit onboard their vessel. On the road an inaccuracy in a GPS system can easily be recognized by the driver through referencing visual landmarks like buildings or the street. This ready reference is often not available to the vessel operator on the water and keeping a proper lookout is still vitally important. Prudent navigators use all available cues to maintain situational awareness and never rely exclusively on a single source of navigation data. It is best to remember a GPS unit is an aid to navigation not a substitute for the human eye and common sense.

2.3.1.5.2 APPLICATION OF GPS

The United States government created the system, maintains it and makes it freely accessible to anyone with a GPS receiver. The global positioning system provides critical capabilities to military, civil and commercial users around the world.

The following are the few applications of GPS:

- **Road Traffic Congestion:** A navigation device has a GPRS receiver for receiving real time information about or slow average speed on a stretch of motorway, indicating congestion. The device calculates a new itinerary to avoid the congestion, based on historically record speeds on secondary roads weighed by the current average speed in the congestion area.
- **Tectonics:** GPS enables direct fault motion measurement of earthquake between earthquake GPS can be used to measure crustal motion and deformation to estimate seismic strain build up for creating seismic hazard maps.
- **Terrorism:** GPS is very important to determine the location of terrorist attacker‟s. For example, on the Gurudaspur strike, Indian intelligence agencies had determined that the GPS sets used by the terrorist were first turned on in Sargodha a home to Pakistans largest airbase-on July 21, 2015, six days before the attack. The set were then programmed with digital waypoints, which led the attackers the border to their targets in Punjab. (The Indian Express, October 27, 2015).
- **Mining:** The use of RTK GPS has significantly improved several mining operations such as drilling, shoveling, vehicle tracking and surveying, RTK GPS provides centimeter-level positioning accuracy.
- **Tours:** Location determines what content to display, for instance, information about an approaching point of interest.
- **Navigation:** Navigators value digitally precise velocity and orientation measurements. With the help of GPS roads or paths available, traffic congestion and alternative routes, roads or paths that might be taken to get to the destination.
- **Cellular Telephony:** Clock synchronization enables time transfer, which is critical for synchronizing its spreading codes with other base stations to facilitate inter-cell hand off and support hybrid GPS / cellular position detection for nibble emergency calls and other applications.
- **Surveying:** Surveyors use absolute locations to make maps and determines property boundaries. The surveying and mapping community was one of the first to take advantage of GPS because it dramatically increased productivity and resulted in more accurate and reliable data. Today, GPS is a vital part of surveying and mapping activities around the world. (Branford, W. et all, 1996).

2.3.2 ASSIGNMENT SUBMISSION MANAGEMENT SYSTEM

Many submission systems, which have alleviated the tasks of lecturers in several ways, exist. However, most of them could not meet all required criteria for an ideal submission system. The inherent problem with an email based submission system lies in its security and its capacity for attaching files (Godfrey B., 1997). Most existing submission system do not allow for structuring of files, automatic correction of automated assessment. Also, some cater only for one type of assignment while others do not put emphasis on due dates and students can submit assignments even after submission date is over. All these problems have led to the development of an online submission system with desirable features.

An online assignment handling is a system designed to cater for the school department's needs in terms of receiving assignments from students, making them available to tutors to mark, returning grades, comments and marked work to Universities, Polytechnics, and colleges of education are considered the main provider of knowledge in various fields. Various courses of studies are taught in institutions, covering several fields including applied Sciences, Math, Computer, Human Resource, and Accounting. Most courses at universities consist of theoretical as well as practical subject matter. To evaluate the level of understanding and degree of comprehension among students, assignments are often given (Paul Darbyshire, 2013).

Assignments are submitted by students either individually or in groups. Assignment management involves collecting, marking, and redistributing to students. According to Tregobov (1998), assignment submission breaks the process down into four stages: submission, recording, marking, and return. This involves the use of internet and computers to aid this process. The most obvious advantage is that it offers faster transportation of assignments questions from the lecturers to the student and submission of their solutions from the students to the lecturers than the traditional, physical, paper-based methods.

2.3.3 COURSE MATERIALS STORAGE SYSTEM

Cloud computing, which is a medium by which files are stored and retrieved on the internet is adopted for the concept of passing across course materials from the lecturer to the student offering the courses. This eliminated the problem of course materials not been able to circulate round the whole class due to insufficient information from the class or course representative. The course material(s) are uploaded by the course lecturer and all students offering such course are notified.

2.4 SYSTEM DEVELOPMENT APPROACH

System Development Life Cycle (SDLC) is referred to a methodology used by software industry to design, develop and test high quality software. It is aimed at producing a high quality software that meets or exceeds customer satisfactions, reaches completion within times and cost estimates. It produces a consistent framework of tasks and deliverables needed to develop systems. The SDLC methodology may be condensed to include automated or manual, whether it is a new system, or an enhancement to existing system. The SDLC methodology tracks a project from an idea developed by the user through feasibility study, system analysis and design, programming, pilot testing, implementation and post implementation analysis.

The development method that intend to use for the Virtual Lecture Attendance System is the incremental development, where the software will be designed, implemented and tested incrementally until the product is finished. This approach combines the elements of waterfall model with the iterative philosophy of prototyping.

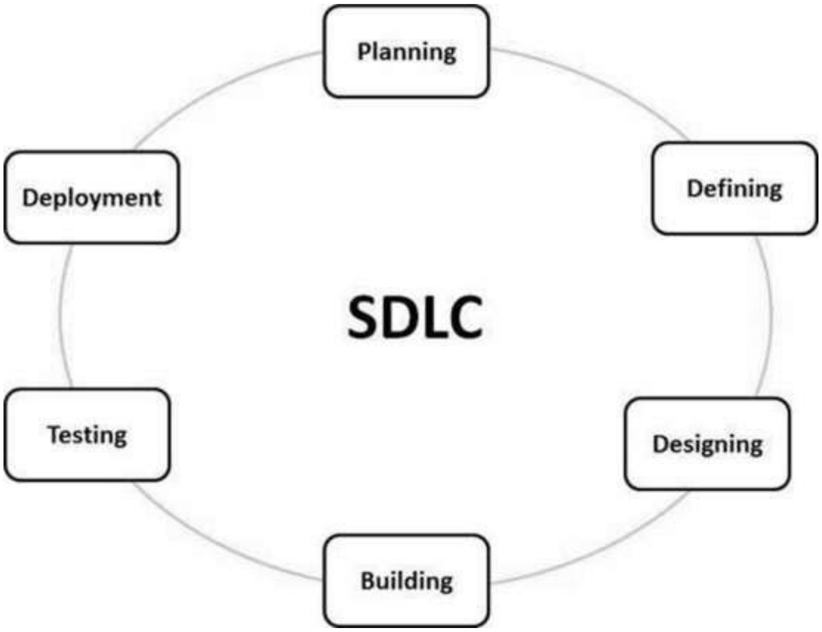


Fig 2.4 Stages of System Development Life Cycle

2.4.1 INCREMENTAL DEVELOPMENT

Incremental development is based on the idea of developing an initial implementation, exposing this to user comment and evolving it through several versions until an adequate system has been developed (Figure 2.5). Specification, development, and validation activities are interleaved rather than separate, with rapid feedback across activities.

Incremental software development, which is a fundamental part of agile approaches, is better than a waterfall approach foremost business, e-commerce, and personal systems. Incremental development reflects the way that we solve problems. We rarely work out a complete problem solution in advance but move toward a solution in a series of steps, backtracking when we realize that we have made a mistake. By developing the software incrementally, it is cheaper and easier to make changes in the software as it is being developed.

Each increment or version of the system incorporates some of the functionality that is needed by the customer. Generally, the early increments of the system include the most important or most urgently required functionality. This means that the customer can evaluate the system at a relatively early stage in the development to see if it delivers what is required. If not, then only the current increment has to be changed and, possibly, new functionality defined for later increments.

Incremental development has three important benefits, compared to the waterfall
model:

- The cost of accommodating changing customer requirements is reduced. The amount of analysis and documentation that has to be redone is much less than is required with the waterfall model.
- It is easier to get customer feedback on the development work that has been done. Customers can comment on demonstrations of the software and see how much has been implemented. Customers find it difficult to judge progress from software design documents.
- More rapid delivery and deployment of useful software to the customer is possible, even if all of the functionality has not been included. Customers are able to use and gain value from the software earlier than is possible with a waterfall process.

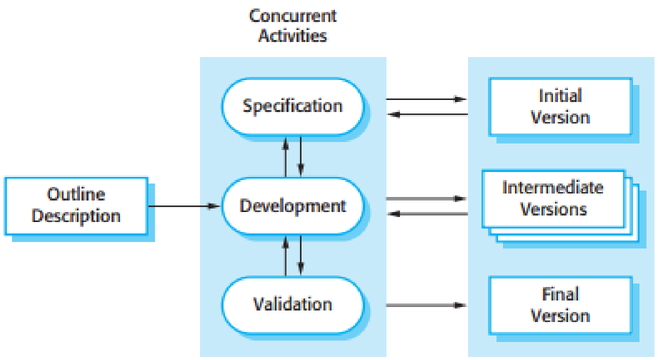


Fig 2.5 Incremental Development

2.4.2 JUSTIFICATION OF INCREMENTAL DEVELOPMENT SELECTION

The methodology selection brings many benefits towards the final delivery of the proposed system. The selected methodology ensures that there is a working model of the system at a very early stage of development, which makes it easier to find functional or design flaws. Finding issues at an early stage of development enables one to take corrective measures in a limited budget.

The selection of Incremental Development will ensure that some working functionalities can be developed quickly and early in the life cycle in order to check the requirements of the project and designs can be easily corrected at the early stage before more work are done. Progress can be measured against the increments and more increments will be agreed upon between the project supervisor and the student with a day of delivery.

2.5 WEB DEVELOPMENT TOOLS

Even though this project is a mobile based application, some web technologies are used for developing the APIs (Application Programming Interface) necessary for sending requests to and retrieval of information from the database.

The following are the web technologies that are used in this project:

2.5.1. LARAVEL

Laravel is an open-source PHP framework, which is robust and easy to understand. It follows a model-view-controller design pattern. Laravel reuses the existing components of different frameworks which helps in creating a web application. The web application thus designed is more structured and pragmatic.

Laravel offers a rich set of functionalities which incorporates the basic features of PHP frameworks like CodeIgniter, Yii and other programming languages like Ruby on Rails. Laravel has a very rich set of features which will boost the speed of web development.

Amongst the benefits of Laravel are:

- The web application becomes more scalable, owing to the Laravel framework.
- Considerable time is saved in designing the web application, since Laravel reuses the components from other framework in developing web application.
- It includes namespaces and interfaces, thus helps to organize and manage resources.

Amongst the features of Laravel are:

Modularity

Laravel provides 20 built in libraries and modules which helps in enhancement of the application. Every module is integrated with Composer dependency manager which eases updates.

Testability

Laravel includes features and helpers which helps in testing through various test cases. This feature helps in maintaining the code as per the requirements.

Routing

Laravel provides a flexible approach to the user to define routes in the web application. Routing helps to scale the application in a better way and increases its performance.

Configuration Management

A web application designed in Laravel will be running on different environments, which means that there will be a constant change in its configuration. Laravel provides a consistent approach to handle the configuration in an efficient way.

Query Builder and ORM

Laravel incorporates a query builder which helps in querying databases using various simple chain methods. It provides ORM (Object Relational Mapper) and ActiveRecord implementation called Eloquent.

Schema Builder

Schema Builder maintains the database definitions and schema in PHP code. It also maintains a track of changes with respect to database migrations.

Template Engine

Laravel uses the Blade Template engine, a lightweight template language used to design hierarchical blocks and layouts with predefined blocks that include dynamic content.

E-mail

Laravel includes a mail class which helps in sending mail with rich content and attachments from the web application.

Authentication

User authentication is a common feature in web applications. Laravel eases designing authentication as it includes features such as register, forgot password and send password reminders.

Redis

Laravel uses Redis to connect to an existing session and general-purpose cache. Redis interacts with session directly.

Queues

Laravel includes queue services like emailing large number of users or a specified Cron job. These queues help in completing tasks in an easier manner without waiting for the previous task to be completed.

2.5.2 MYSQL

There are two regions in a website design: the front-end and the back-end. The front-end is the site design and interface users see when they open the site. The back-end is the database. The database is the power horse for an application. It houses all the data and information needed by the front-end. Webmasters have several options when choosing a database, and one of these options is MySQL.

Before you understand MySQL, it helps to understand database functionality. A database is made up of tables, stored procedures, and functions. These three parts drive the back-end of your application.

Tables are the database engine's storage components. Architecture of database storage is extremely technical, but an easy way to envision a database table is to visualize a spreadsheet. Spreadsheets have columns and rows. Where these columns and rows intercept are fields. The fields contain one item of data.

MySQL is not popular just because it's free. Most free software is not associated with high-end enterprise solutions, but MySQL is an exception to this rule. MySQL can be used with small startups or high-traffic enterprise applications. Banks, social networks, universities and government agencies use MySQL as their database solution.

Because companies can go from small startup to large enterprise without changing its database solution, MySQL is scalable. Scalability is often disregarded when startups have tight budgets, but since MySQL is free, it's a good option for a startup that expects growth. It's also high performance. Performance is paramount to application stability. Small databases with only a few records will normally function well even if they are poorly designed. However, if the database tables grow to millions of records, application performance and stability can be affected, which then affects your customers and employees.

MySQL is also secure. Security is a sensitive issue, since the protection of your customer and employee records should be a major concern. With MySQL, administrators can encrypt data and set up authentication to protect all company assets.

There are several more open-source database applications that were deployed after MySQL. MySQL's success is partly due to the numerous hosting companies that offer services with unlimited MySQL database services. Oracle and SQL Server are expensive platforms, and most hosting companies only offer one SQL Server if you choose to work with it instead of an open-source database solution. Since MySQL is free and open-source, hosting companies offer unlimited databases with a lower cost than Microsoft platforms. This type of marketing gave MySQL its strong foundation in application development in global cloud and desktop applications.

MySQL is a relational database. These systems offer data integrity. They are distinct from systems such as Mongo that relies on document style records. Relational databases are used in numerous applications, but the major reason they are used to create applications is for data integrity and transactional style record manipulation.

Relational databases work on the concept of a primary and foreign key relationship. The primary key is the main unique field that identifies a record. This unique identifier is then stored in other tables to build a relationship between a main table and a secondary related table. Each customer has its own unique identifier, and then this identifier is stored in the order table. When you need to query MySQL with a list of customers and related orders, you join the two tables using specific SQL syntax.

MySQL is also a transactional database, which means that you can roll back changes to your database. For instance, suppose you want to delete a customer but you do not want to delete the customer if there are active order records associated with the customer. You attempt to delete the customer, but MySQL's relational foreign key constraints give you an error when you attempt to delete it. You can then roll back any changes you made based on commit and rollback features. You commit changes if there were no errors found, or you can roll back previous statements if an error is thrown by the database server.

Data integrity is what sets MySQL and other relational databases apart from more modern databases such as NoSQL. NoSQL databases such as Mongo are used for analytical data and capturing any number of unstructured data. MySQL requires your data to be more structured, so it's a reliable database system for people who want to secure the structure and relationship between tables.

In bigger businesses, MySQL and NoSQL databases work together. The MySQL database stores structured data such as orders and customer information, and the NoSQL database stores unstructured data such as marketing and traffic numbers. You can export data from a MySQL database to a NoSQL database to work with them both for their best features.

2.5.3 JAVASCRIPT

Javascript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JavaScript was first known as LiveScript, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name LiveScript. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

2.5.3.1 CLIENT-SIDE JAVASCRIPT

Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser.

It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content.

The JavaScript client-side mechanism provides many advantages over traditional CGI server-side scripts. For example, you might use JavaScript to check if the user has entered a valid e-mail address in a form field.

The JavaScript code is executed when the user submits the form, and only if all the entries are valid, they would be submitted to the Web Server.

JavaScript can be used to trap user-initiated events such as button clicks, link navigation, and other actions that the user initiates explicitly or implicitly.

2.5.3.2 ADVANTAGES OF JAVASCRIPT

- Less server interaction − You can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.
- Immediate feedback to the visitors − They do not have to wait for a page reload to see if they have forgotten to enter something.
- Increased interactivity − You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
- Richer interfaces − You can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.

2.6 MOBILE DEVELOPMENT TOOLS

2.6.1 ANDROID STUDIO

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as primary IDE for native Android application development.

The following features are provided in the current stable version:

- Gradle-based build support
- Android-specific refactoring and quick fixes
- Lint tools to catch performance, usability, version compatibility and other problems
- ProGuard integration and app-signing capabilities
- Template-based wizards to create common Android designs and components

- A rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations.
- Support for building Android Wear apps.
- Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine.

2.6.2 REACTNATIVE

React Native is a framework developed by Facebook that builds a hierarchy of UI components to build the JavaScript code. It has a set of components for both iOS and Android platforms to build a mobile application with native look and feel. was developed by Facebook using the same design principles, except designing interfaces.

React Native lets you build mobile apps using only JavaScript. It uses the same design as React, letting you compose a rich mobile UI from declarative components.

With React Native, you do not build a mobile web app, an HTML5 app, or a hybrid app; you build a real mobile app that is indistinguishable from an app built using Objective-C or Java. React Native uses the same fundamental UI building blocks as regular iOS and Android apps.

The features of Reactnative are:

- React: This is a Framework for building web and mobile apps using JavaScript.
- Native: You can use native components controlled by JavaScript.
- Platforms: React Native supports IOS and Android platform.

The benefits of using Reactnative are:

- JavaScript: You can use the existing JavaScript knowledge to build native mobile apps.
- Code sharing: You can share most of your code on different platforms.
- Community: The community around React and React Native is large, and you will be able to find any answer you need.

CHAPTER THREE

METHODOLOGY

3.1 INTRODUCTION

This chapter gives an insight into the various stages of the analysis and design of the proposed system, i.e. Virtual Lecture Management System.

The design phase of a system development process concentrates on how a system should work and the development of procedures to make it work that way. System design can also be seen as the process of transforming a problem into resulting product known as solution. A design module or process is a functional entity with a well-defined set of inputs and outputs. Therefore, each module or process can be viewed as a component of the whole system. Valacich et al. (2012).

The system design method (often called methodology) is an orderly and integrated collection of various methods, tools and notations that impose a structure on the development process, by:

- Providing a number of well-defined and complementary ways in which information about important aspects of the system is represented.
- Providing recommended and suggested series of tasks to be carried out so that the methods fit together as a whole. Sommerville (2010).

Basically, this project is committed to developing a virtual lecture management system, taking the department of Computer Science, University of Ibadan as a case study. The existing and proposed systems are evaluated in the system analysis of this project so as to realize and structure requirements for the proposed system. The outcome of the system analysis determines the success of the proposed system.

3.1.1 DESCRIPTION OF EXISTING SYSTEM

As at the time of developing this system, the conventional method for capturing student's lecture attendance and assignment submission is manual and this has resulted into mutilation and an outright loss of the submitted attendance and assignments in cases of natural disasters such as floods and fire out-breaks. Generation of attendance reports are also falsified by students marking attendance for another classmate who is absent.

Moreso, course materials which are delivered by the lecturers to the class representative do not reach all students, leaving some students uninformed. These manual methods are not only unreliable, but also not secure.

3.1.2 PROBLEMS OF EXISTING SYSTEM

Capturing of students' lecture attendance and submission of assignment using the paper approach has been in use for long in most Nigeria tertiary institutions, of which the University of Ibadan is not an exception. However, a number of problems are identified with this kind of manual approach, among which are the following:

- Falsification of attendance report.
- Misplacement and outright loss of attendance reports and submitted assignments by negligence by the class representatives.
- Lecture materials to circulate to all students, leaving some students uninformed.

3.1.3 ANALYSIS OF THE PROPOSED SYSTEM

The proposed system will be an android application that uses the GPS to specify students' location as at the time a lecture is going on. Any student who is not within the coordinate of venue of the lecture will be disabled from taking attendance for any course, thereby eradicating the falsifications that arise when a student mark attendance for students who was absent in class. Generation of attendance reports can also be generated periodically with this platform as this will enable students to know their current attendance percentage and will help them to know if they will most likely write examinations for each course, having reached a minimum of 75% attendance.

Moreso, course materials which might not be able to circulate round in the class will be catered for in this project as students can easily logon to their portal to read and download materials uploaded by the course lecturer.

Finally, submission and grading of assignments on-the-go will also be incorporated into this project, thereby eradicating the loss of any paper-based assignments.

3.2 REQUIREMENTS ANALYSIS

The requirements for the proposed system were gathered by studying existing paper based system for taking attendance and submission of assignments. Also, various forms of interviews were conducted with the student of the department of Computer Science, University of Ibadan, as regards the inability of the course materials to circulate round.

3.2.1 USER REQUIREMENTS

The system will be used by the students, lecturers and an administrator. The system can be accessed after a successful login by any of these entities.

3.2.2 FUNCTIONAL REQUIREMENTS

The system requirements of the proposed system can be divided into two, which are:

1. Functional requirements.
2. Non-functional requirements.

3.2.2.1 FUNCTIONAL REQUIREMENTS

The functional requirements analysis describes how the system is meant to solve the problem at hand. It describes an interaction between the system and its environment. It captures the services that the system should provide which include; its behaviors in particular situations and its reactions to inputs. Below are the functional requirements of the system:

Inputs: The system shall collect inputs from the students, lecturer and administrator. The inputs include the following:

- User login details.
- Forgotten password details.
- Adding new and updating existing records.
- Searching for existing records.
- Upload of materials and assignments.
- Changing user password.
- Editing user profile.

Processing: This refers to the way inputs are being processed by the system, how it reacts to certain inputs. The system undergoes the following processes:

- Verification and authentication of login details.
- Checking for the existence of a record before adding a new one and updating an existing one.
- Verification of the current password of the user during the process of changing the user password.
- Checking if the new password matches with the confirm new password during the process of changing the user password.
- Getting the latitude and the longitude of a venue for the purpose of determining students' location per time.
- Validating the student's location before taking a new attendance.
- Notifications of newly uploaded course materials and assignments for the students.

Outputs: This refers to the result of the processes performed on the inputs. The system shall produce the following outputs:

- Generation of attendance reports for students and lecturers.
- List of all records – students, lecturers, courses, venues, course materials, assignments.
- Report on qualified students to write particular examinations.

3.2.2.2 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements are constraints on the functions performed by the system. The functions include:

User friendliness: The system shall be user friendly to allow easy usage for the user.

Response time: The system shall respond quickly to user requests as being processed.

Security: The system allows only authorized user to have access to the system.

3.3 SYSTEM MODELLING

This is a part of the system analysis phase where we present conceptual representations of the system. This is important as it is the ability to present a clear and concise set of requirements that can be redefined to suit the needs of the system.

3.3.1 SYSTEM MODELLING APPROACH

This project will be carried out using the incremental development approach. Incremental development is based on the idea of developing an initial implementation, exposing this to user comment and evolving it through several versions until an adequate system has been developed.

We rarely work out a complete problem solution in advance but move toward a solution in a series of steps, backtracking when we realize that we have made a mistake. By developing the software incrementally, it is cheaper and easier to make changes in the software as it is being developed.

Each increment or version of the system incorporates some of the functionality that is needed by the customer. Generally, the early increments of the system include the most important or most urgently required functionality. This means that the customer can evaluate the system at a relatively early stage in the development to see if it delivers what is required. If not, then only the current increment has to be changed and, possibly, new functionality defined for later increments.

Incremental development has three important benefits:

- The cost of accommodating changing customer requirements is reduced. The amount of analysis and documentation that has to be redone is much less.
- It is easier to get customer feedback on the development work that has been done. Customers can comment on demonstrations of the software and see how much has been implemented. Customers find it difficult to judge progress from software design documents.
- More rapid delivery and deployment of useful software to the customer is possible, even if all of the functionality has not been included.

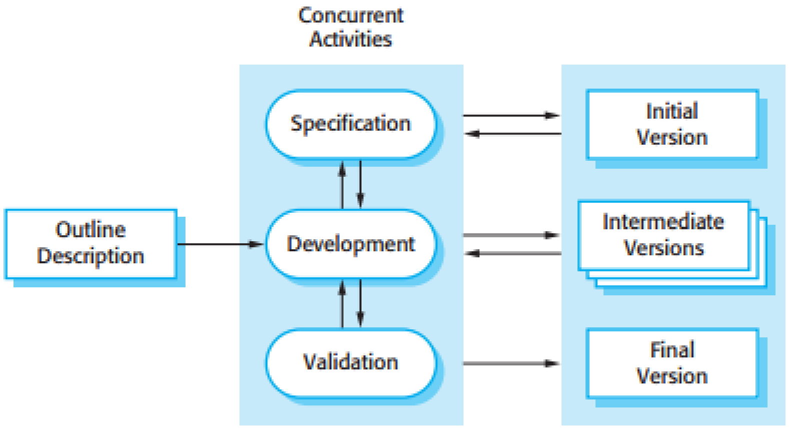


Fig. 3.1 The incremental development

3.3.2 SYSTEM MODULES

The components of the system are created in modules that can work independently and also interact with other components as a whole system. The modules in the system include the following:

ADMINISTRATOR

The administrator is saddled with the responsibility of management of records in the system. Highlighted below are the functions of an administrator:

- **Login:** For security purpose, the user needs to be authenticated before access can be granted to the system. The authentication is via the validation of the email address and password provided.
- **Forgot password:** The administrator can reset his password via this module by supplying his correct and existing email address.
- **Add new records:** The administrator can add new lecturers, students, courses, venues.
- **Update existing records:** The administrator can make changes to already existing lecturers, students, courses, venues and his profile as well.
- **Search existing records:** The administrator can perform an easy search on lecturers, students, courses, venues.
- **Change password:** For security reasons, it is safe that the administrator changes his password for security reasons.
- **Logout:** This is used to exit the system by ending the session of the user.

LECTURERS

The lecturer is responsible performing the following functions:

- **Login:** For security purpose, the user needs to be authenticated before access can be granted to the system. The authentication is via the validation of the email address and password provided.
- **Forgot password:** The lecturer can reset his password via this module by supplying his correct and existing email address.
- **Generation of attendance reports:** The lecturer can generate attendance report for his courses.
- **Upload new records:** The lecturer can upload new materials and assignments.
- **Update existing records:** The lecturer can make changes to already his existing materials, assignments and his profile as well.
- **Search existing records:** The lecturer can perform an easy search on his existing materials, assignments
- **Grading of students' assignment:** Upon downloading the submitted assignments, the lecturer can grade students' assignment and the student will be instantly notified.
- **Change password:** For security reasons, it is safe that the administrator changes his password for security reasons.
- **Logout:** This is used to exit the system by ending the session of the user.

STUDENTS

The student is responsible performing the following functions:

- **Login:** For security purpose, the user needs to be authenticated before access can be granted to the system. The authentication is via the validation of the matric and password provided.
- **Forgot password:** The student can reset his password via this module by supplying his correct and existing email address.

- **Taking of attendance:** The student can mark attendance via the system, only when he is in the location of the venue of the ongoing lectures.
- **View attendance reports:** The student can generate attendance report for his courses.
- **Download of resources:** The student can download materials and assignments sent by lecturers of different courses.
- **Submission of assignments:** Upon the completion of the solution of assignment, the student can submit the assignment for grading.
- **Update profile:** The student can make changes with his profile details.
- **Change password:** For security reasons, it is safe that the user changes his password for security reasons.
- **Logout:** This is used to exit the system by ending the session of the user.

3.4 PHYSICAL MODELING

This represents the design of the physical entities of the system. It deals with the partitioning of entire system into modules. It includes the following diagrams:

3.4.1 USE CASE MODELING

It is a means to graphically or pictorially capture the functional requirements of a system, (Sommerville, 2011). It represents a discrete interaction between a user and the machine. The interaction between a user and the system include: registration, login in, document activities, etc. The use case modeling is an effective technique for requirement elicitation, (Sommerville, 2011). The use case model includes the following components:

- **Use case:** A sequence of related actions initiated by an actor; it represents a specific way to use the system. A use case is always initiated by an actor.
- **Actor:** A use case is always initiated by an actor. An actor is an external entity that interacts with the system (similar to an external entity in data-flow diagramming).
- **Connectors:** These are different kinds of lines which include arrow headlines, dashed arrow headlines and simple thick lines each of which represents different types of interactions and constraints that specify the conditions necessary for a particular use case to be performed.

- **System boundary:** This houses all the use cases and the interactions that happen within the system. It has a rectangular shape.

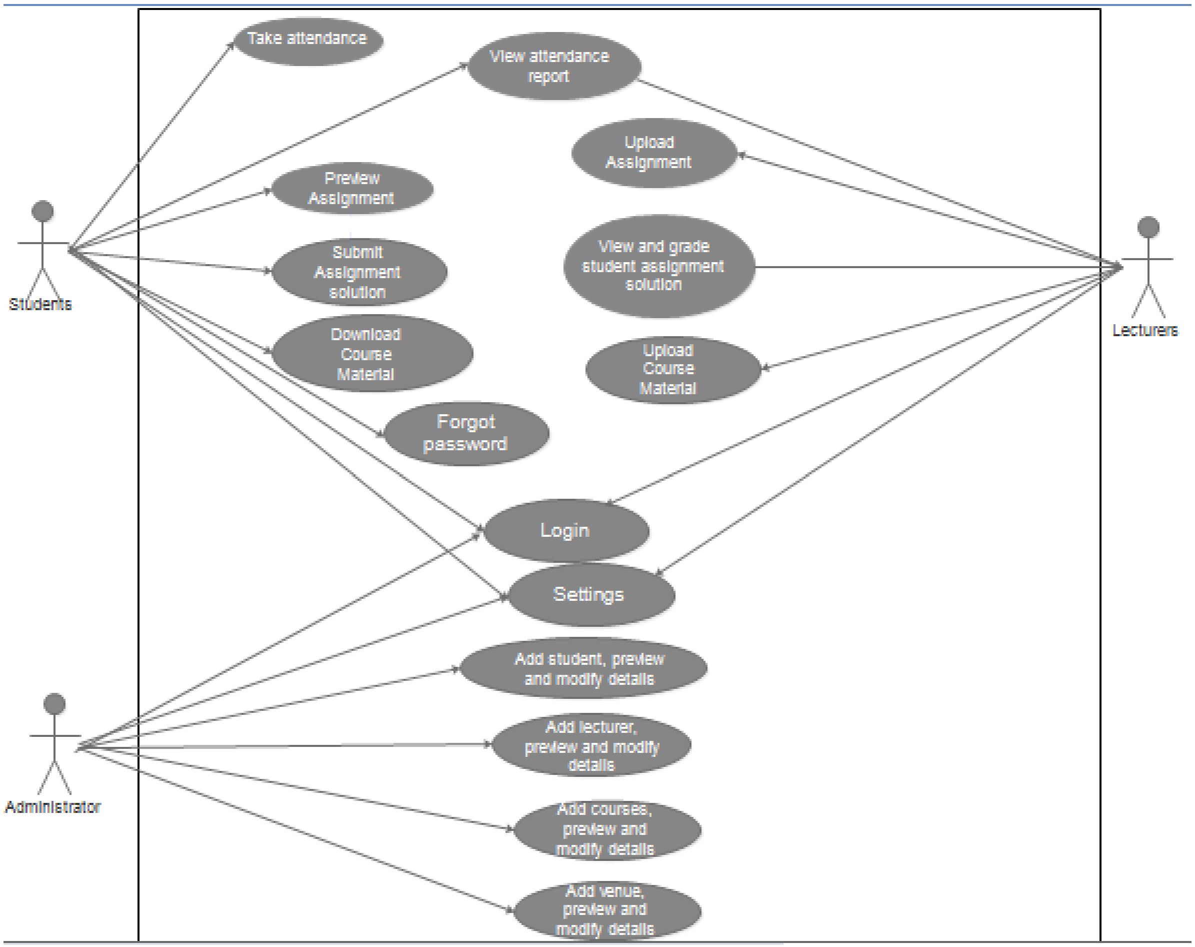

Take attendance
View attendance report
Upload Assignment
Preview Assignment
Submit Assignment solution
View and grade student assignment solution
Download Course Material
Upload Course Material
Forgot password
Login
Settings
Add student, preview and modify details
Add lecturer, preview and modify details
Add courses, preview and modify details
Add venue, preview and modify details
Students
Lecturers
Administrator

Figure 3.4: UML Use Case Diagram for the Virtual Lecture Management System

Brief Description of the Use Case Diagram

STUDENT USE CASE

- **Login:** A student is required to provide the accurate matric number and password to enable him/her has access to the system.
- **Settings:** This use case comprises of profile, edit profile, change password and logout.
- **Forgot password:** This process enables the students to reset their forgotten password via their email addresses.
- **Take attendance:** This allows the students to take attendance and they must be in the venue for the lecture before they can take attendance.
- **Attendance report:** Students can view the report of their attendance, download the report and sort on weekly, monthly and semester basis.
- **Download course materials:** Students can download the course materials uploaded by their lecturers.
- **Preview assignment:** Students can view the details of the assignment uploaded by their lecturers.
- **Submit assignment solution:** After solving the assignment, students can submit their solutions and the lecturer will be notified.

LECTURER USE CASE

- **Login:** A lecturer is required to provide the accurate email address and password to enable him/her has access to the system.
- **Settings:** This use case comprises of profile, edit profile, change password and logout.
- **Forgot password:** This process enables the lecturers to reset their forgotten password via their email addresses.
- **Attendance report:** Lecturers can view the report of their attendance, download the report and sort on weekly, monthly and semester basis.
- **Upload and view uploaded course materials:** Lecturers can upload their course materials and make necessary corrections to it.
- **Upload and view uploaded assignments:** Lecturers can upload their assignments and make necessary corrections to it.

- **Grade assignment solution:** Lecturers can view the details of the assignment uploaded by the students and assign appropriate grades to them.

ADMINISTRATOR USE CASE

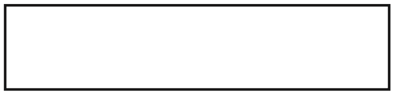
An administrator of the system is the HOD and the following are the processes of his use case:

- **Login:** The administrator is required to provide the accurate email address and password to enable him/her has access to the system.
- **Logout:** This logs out the administrator from the system.
- **Forgot password:** This process enables the administrator to reset his forgotten password via his email addresses.
- **Add new and update existing records:** The administrator can add new records such as students, lecturers, courses, venues and timetable details and make changes to these records as well.
- **Search for existing records:** Existing records can be searched for easy access.

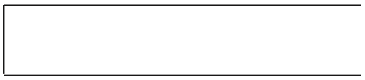
3.4.2 DATAFLOW DIAGRAMS

Data-flow diagrams (DFDs) are system models that show a functional perspective where each transformation represents a single function or process. DFDs are used to show how data flows through a sequence of processing steps (http://www.SoftwareEngineering-9.com/Web/DFDs). There are four (4) symbols used in the representation of dataflow diagram; they are:

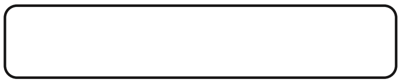
- **Source/sink:** This refers to origin and/or destination of data.



- **Data store:** This refers to data at rest. It represents any location of data e.g. database



- **Process:** This refers to the actions performed on data so that they can be transformed



- **Data flow:** This refers to data in motion from one place to another in the system.

3.4.1.1 CONTEXT DATAFLOW DIAGRAM

A context diagram is a data flow diagram that shows the system boundaries, external entities that interact with the system, and the major information flows. A context diagram has only one process which represents the entire system itself. The source/sink represents its environmental boundaries. The figure below represents the context diagram of the system.

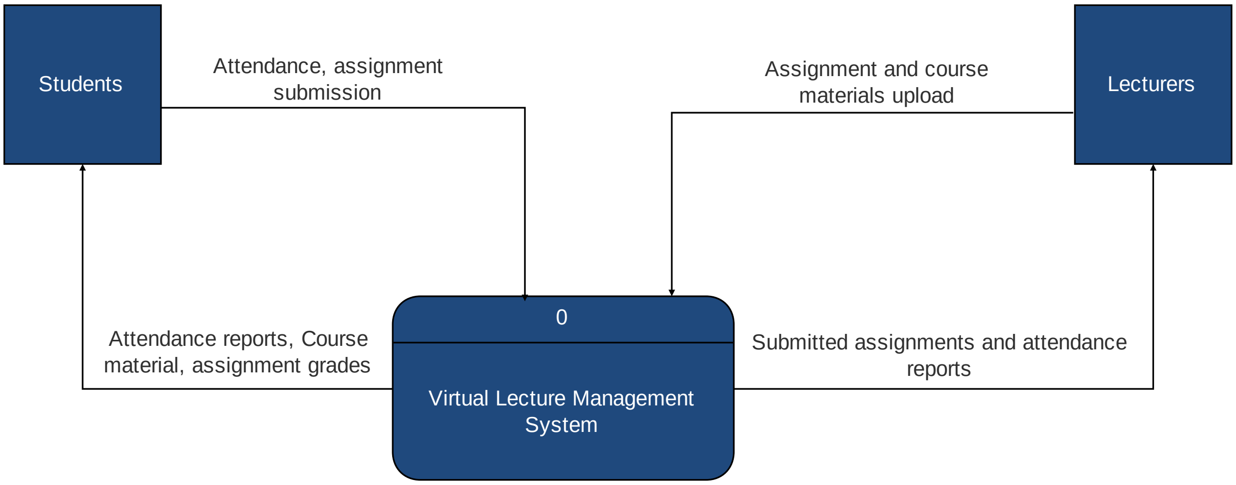

Students
Attendance, assignment submission
Assignment and course materials upload
Lecturers
0
Attendance reports, Course material, assignment grades
Virtual Lecture Management System
Submitted assignments and attendance reports

Figure 3.2: Context Dataflow Diagram of Virtual Lecture Management System

Brief description of the Context Dataflow Diagram

- **Process:** The context dataflow diagram of the system contains only one process named; Virtual lecture management system. This process represents the entire process of the system.
- **Data flows:** The arrows in the system represent the data flows. The data flows are labeled and depicts the kind of data that go into and out of the system.
- **Data source / sink:** The data sources/sinks in this project are: lecturers and students These data sources/sinks can send data to, and receive information from the system.

3.4.1.2 LEVEL 0 DATAFLOW DAIGRAM

The four (4) symbols of dataflow diagram are used in the level 0 dataflow representation of this project. From the diagram below, communication begins when a user logs in. The login details are then processed to ensure validity and authenticity of the user in order to prevent unauthorized access into the system.

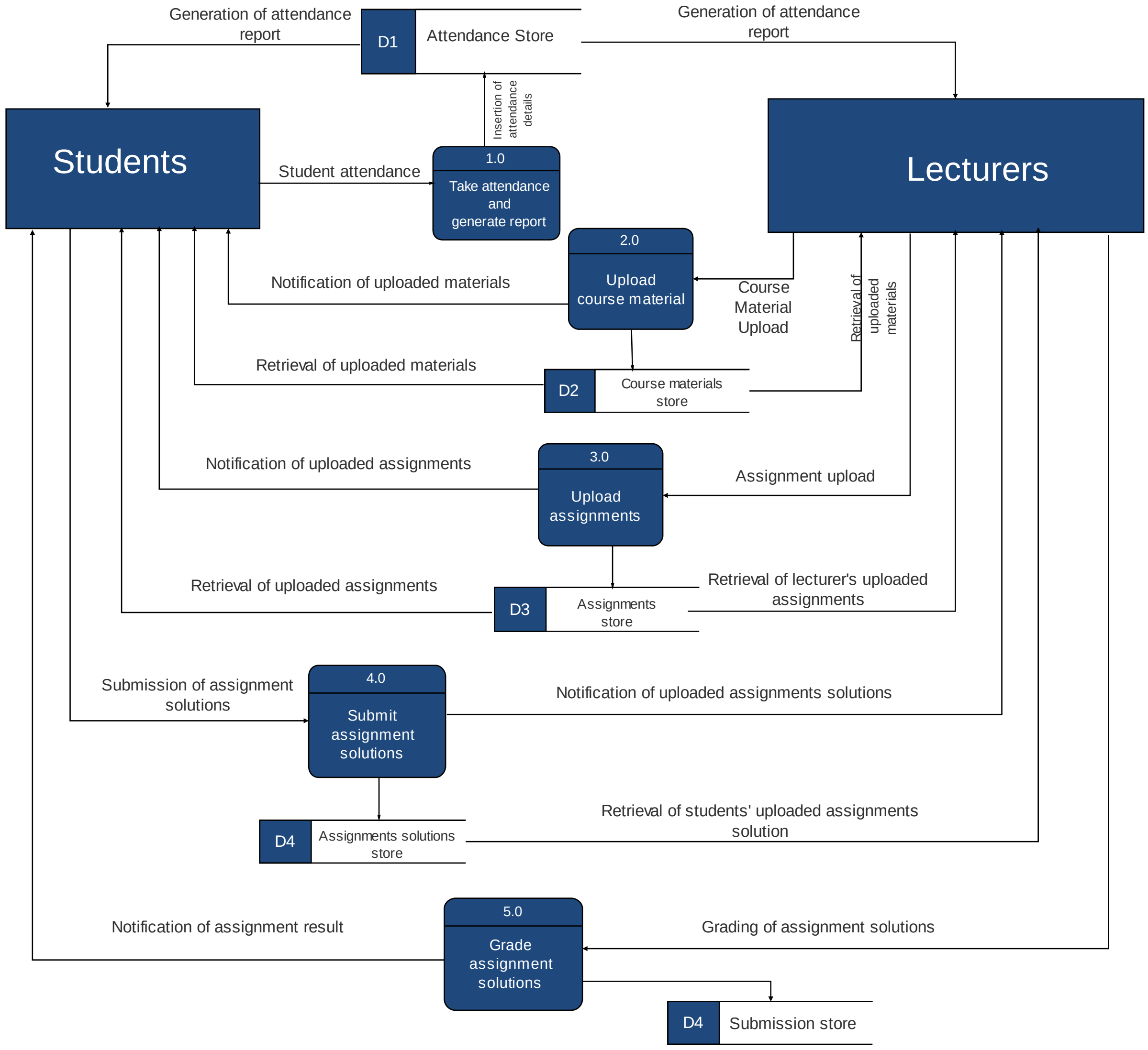

Generation of attendance report
D1
Attendance Store
Generation of attendance report
Insertion of attendance details
Students
Student attendance
1.0
Take attendance and generate report
Lecturers
2.0
Upload course material
Course Material Upload
Retrieval of uploaded materials
Notification of uploaded materials
Retrieval of uploaded materials
D2
Course materials store
Notification of uploaded assignments
3.0
Upload assignments
Assignment upload
Retrieval of uploaded assignments
D3
Assignments store
Retrieval of lecturer's uploaded assignments
Submission of assignment solutions
4.0
Submit assignment solutions
Notification of uploaded assignments solutions
D4
Assignments solutions store
Retrieval of students' uploaded assignments solution
Notification of assignment result
5.0
Grade assignment solutions
Grading of assignment solutions
D4
Submission store

Figure 3.3: Level-0 Dataflow Diagram of the Virtual Lecture Management System

Brief description of the Level-0 Dataflow Diagram

Data stores: There are eight (8) data stores in this project. These data stores represent the database tables where data are stored and retrieved. The data stores are:

- Attendance store.
- Assignments store.
- Course store.
- Lecturers store.
- Materials store.
- Students store.
- Assignment Submission store.
- Timetable store.
- Venues store.

Processes: This project is broken down into four (4) processes in the level-0 dataflow diagram. The processes are:

- **Attendance:** This process enables a student to mark attendance and generate reports periodically.
- **Upload course materials:** This process enables a lecturer to upload course materials and student can download.
- **Upload assignments:** This process enables a lecturer to upload assignments and student can download.
- **Submit assignment solution:** This process enables students to submit solutions to assignments.
- **Grade assignments:** This process enables a lecturer to grade submitted assignment.

Data source / sink: They are used for data storage and retrieval. The data source / sink is the librarian.

Data flows: This indicates the direction of data and the kind of data that goes in and out of the system.

3.4.3. ENTITY RELATIONSHIP MODEL

The entity-relationship (E-R) data model was developed to facilitate database design by allowing specification of an enterprise schema that represents the overall logical structure of a database. Its E-R model also has an associated diagrammatic representation, which we will be considered after this part. The table below shows the E-R model for this project.

Table. 3.1: Entity Relationship Model for the proposed project

Entities	**Attendance:** This entity represents the details of attendance taken by students. **Assignments:** This entity represents assignments uploaded by lecturers. **Courses:** This entity represents the courses added by the administrator. **Lecturers:** This entity represents the lecturers added by the administrator. **Materials:** This entity represents the materials uploaded by lecturers. **Students:** This entity represents the students added by the administrator. **Submission:** This entity represents the assignment solutions submitted by students. **Timetable:** This entity represents the timetable details made by the administrator. **Venues:** This entity represents the venue details added by the administrator.
Attributes	***Attendance attributes:*** id, student_id, course_id, date, venue, created_at, updated_at ***Assignments attributes:*** id, title, course_id, lecturer_id, deadline, path, size, note, created_at,updated_at. ***Courses attributes:*** id, code, title, unit, level, type, created_at,updated_at. ***Lecturers attributes:*** id, title, surname, firstname, middlename, email, salt, password, phone, activation, reset_code, role, created_at,updated_at. ***Materials attributes:*** id, lecturer_id, course_id, note, path, size, created_at,updated_at. ***Students attributes:*** id, surname, firstname, middlename, salt, password, matric, email, phone, gender, level, dept, activation, reset_code, pix, finger, created_at,updated_at.

	Submission attributes: id, ass_id, student_id, score, path, created_at,updated_at. ***Timetable attributes:*** id, course_id, venue_id, lecturers_id, time, created_at,updated_at. ***Venues attributes:*** id, name, longitude, latitude, created_at, updated_at.
Relationships	**Administer relationship:** This is the relationship that exists between the lecturer and the assignment. A lecturer can administer zero or more assignments and many assignments can belong to a lecturer. **Has relationship:** This is the relationship that exists between a course and a material. A course can have one or more materials while a material can be used for a single course. **Marks relationship:** This is the relationship that exists between the student and attendance. A student can mark one or more attendances and many attendances can only belong to a student. **Submit relationship:** This is the relationship that exists between the student and assignment. A student can submit one or more assignment and one or more assignment belongs to a student. **Takes relationship:** This is the relationship that exists between the lecturer and course. A lecture can teach one or more courses and a course can be taught by many lecturers.
Constraints	**Mapping Cardinalities:** A many-to-many relationship exists between lecturer and assignment. A one-to-many relationship exists between course and material. A many-to-many relationship exists between student and attendance. A many-to-many relationship exists between student and assignment. A many-to-many relationship exists between lecturer and courses.

3.4.5 ENTITY RELATIONSHIP DIAGRAM

An entity-relationship diagram (ERD) is a graphical representation of an information system that shows the relationship between people, objects, places, concepts or events within that system.

Components of Entity-Relationship diagram

- Entity: a thing or object in the real world that is distinguishable from, other object. It is represented by a rectangle
- Relationship: an association among several entities. It is represented by a diamond shape.
- Single lines link attributes to relationship sets and entity sets to relationship sets.
- Double lines indicate total participation of an entity in a relationship set

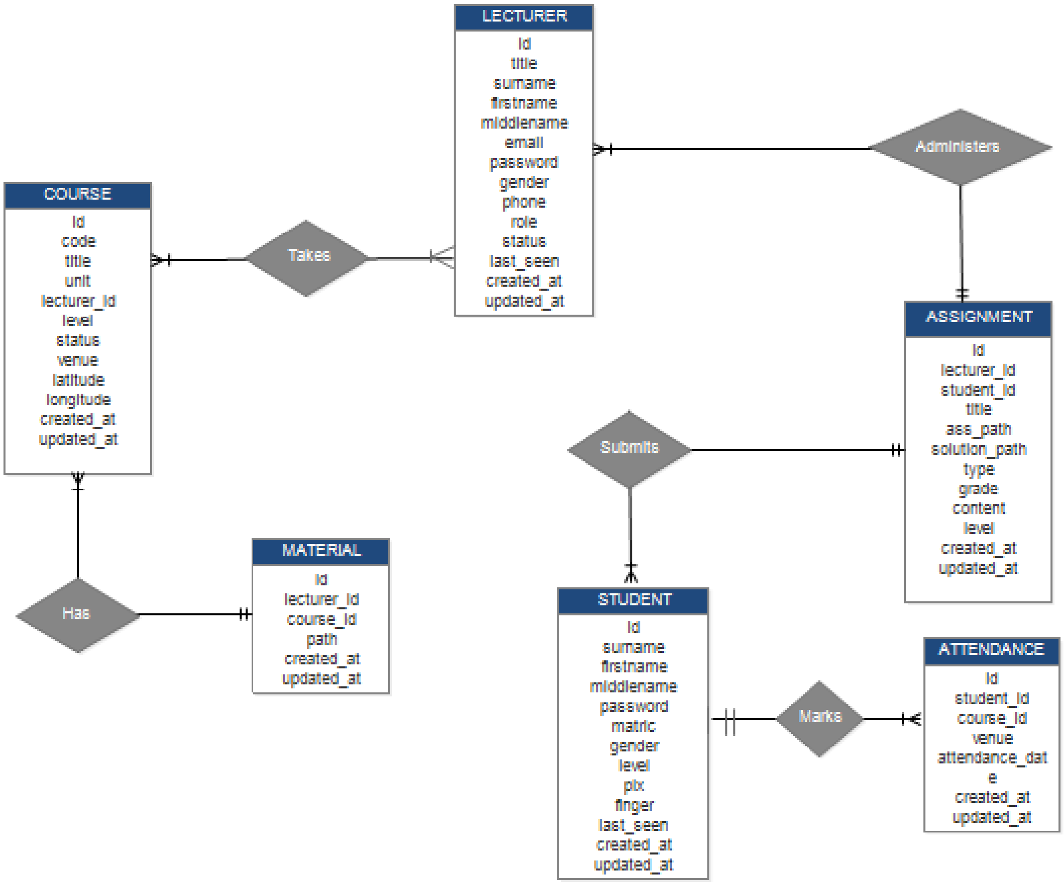


Fig. 3.5: Entity-Relationship Diagram for the project

3.4.6 DESIGNING THE DATABASE SCHEMA

System design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. System design could be seen as the application of systems theory to product development. It aims at identifying and describing fundamental system abstractions and their relationships.

Transformation of E-R Diagram into Relational schema

Transforming an entity relationship diagram into a relational database schema can be accomplished in the following steps. The steps are described below

- **Represent entities:** Each entity type in the E-R diagram becomes a relation. The identifier of the entity type becomes the primary key of the relation, and other attributes of the entity type become non-primary key attributes of the relation.
- **Represent relationships:** Each relationship in an E-R diagram must be represented in the relational database design.
- **Normalize the relations:** The relations created in steps 1 and 2 may have unnecessary redundancy. So we need to normalize these relations to make them well structured
- **Merge the relations:** Redundant relations may need to be merged and renormalized to remove redundancy.

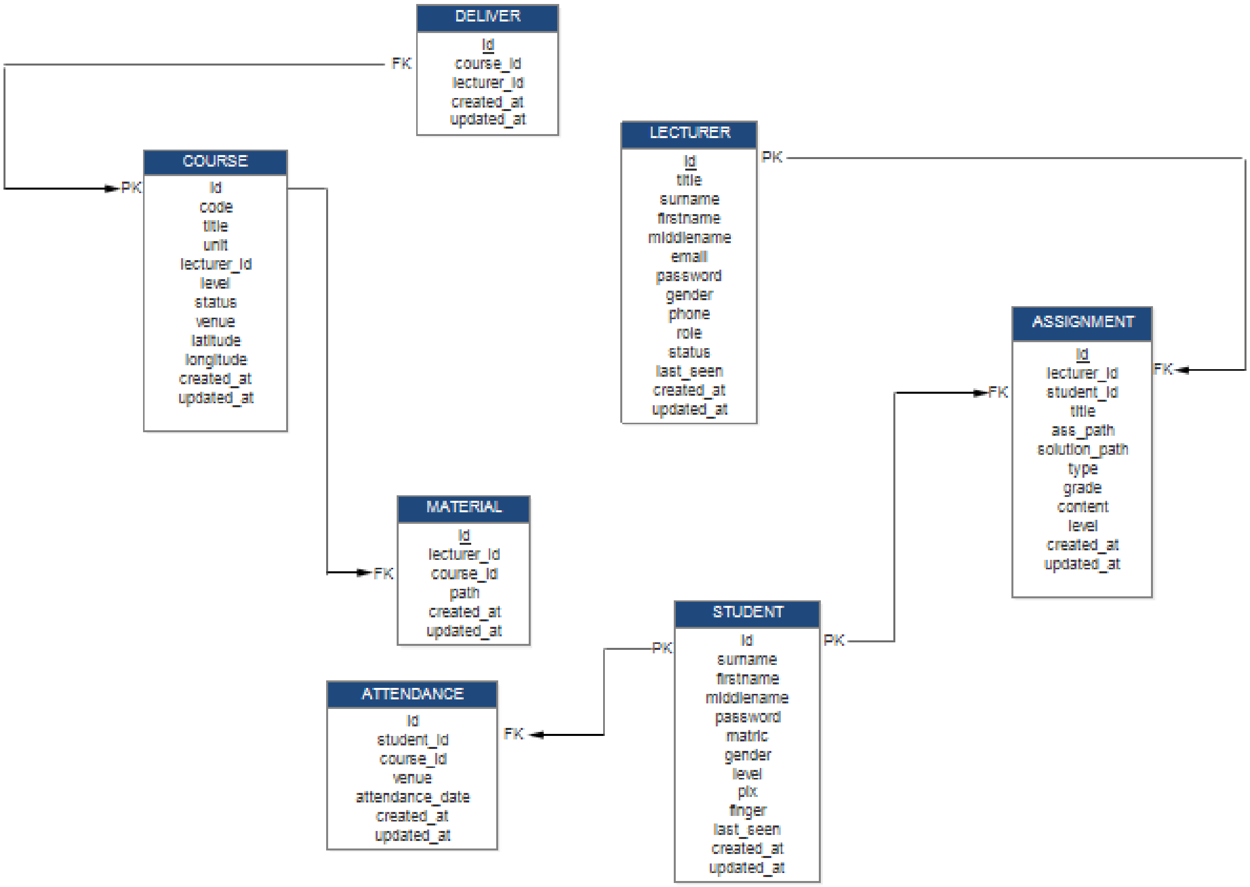


Fig. 3.6: Relational model for the proposed system

CHAPTER FOUR

IMPLEMENTATION, TESTING AND DOCUMENTATION

4.1 INTRODUCTION

This chapter presents the implementation and testing overview of the application. The mobile virtual lecture management system uses a couple of technologies which are discussed in this chapter. Moreso, implementation details of the system, requirements for the system, documentation and the various techniques are discussed in this chapter as well.

4.2 CHOICE OF PROGRAMMING LANGUAGES

Varieties of programming languages were used for the implementation of this project, the most prominent being React Native. React Native is a framework developed by Facebook that builds a hierarchy of UI components to build the JavaScript code. It has a set of components for both iOS and Android platforms to build a mobile application with native look and feel.

Lumen, a microframework of Laravel was used for developing the APIs (Application Programming Interface) for querying the database. MySQL is used for storage of data, while NativeBase is used for the UI/UX design.

4.3 HARDWARE REQUIREMENTS

The hardware requirements of the application include the following:

- Monitor: 1280 x 800 minimum screen resolution.
- CPU: Minimum of 1.5 GHz processor.
- Hard Disk: Minimum of 120 GB of available disk.
- RAM: Minimum of 4 GB.

4.4 SOFTWARE REQUIREMENTS

The software requirements of the application are as follows:

- Operating System: Windows XP and above for systems running Windows Operating System, and Ubuntu 10.10 and above for systems running Linux Operating System.
- The Java Development Kit (JDK).
- The Android Studio 2.0 IDE and above.
- Node Package Module.

4.5 USER INSTRUCTION MANUAL AND INSTALLATION

React Native is used for developing cross-platform mobile applications. i.e the same codebase can be used for developing both android and ios application. However, this project is limited to android version only due to the inability of getting a MAC system and a standby iPhone for testing the ios version.

To start using this application, you must get the apk installed on your android phone.

4.5.1 HOME SCREEN

This is the first screen that will be displayed to the user upon launching the app.

To login as a student, click on student button.

To login as a lecturer, click on lecturer button.

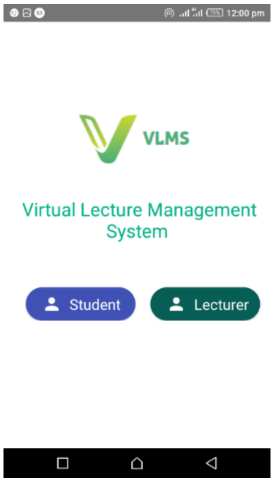


Fig. 4.1 Home Screen

4.5.2 LECTURERS’ LOGIN SCREEN

For any lecturer to login via this panel, his details must have been added by the administrator to the database. A lecturer can login by doing the following:

- Supply your email address and password.
- Click on Login button.

- Wait for your details to be authenticated.
- If the authentication is successful, you will be directed to the dashboard, else, an error message “Incorrect login details” will be displayed.

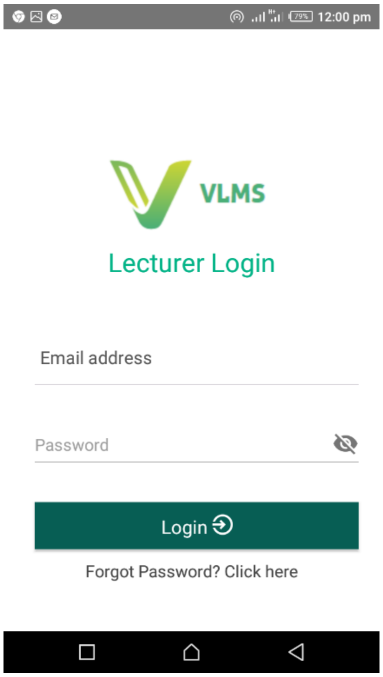
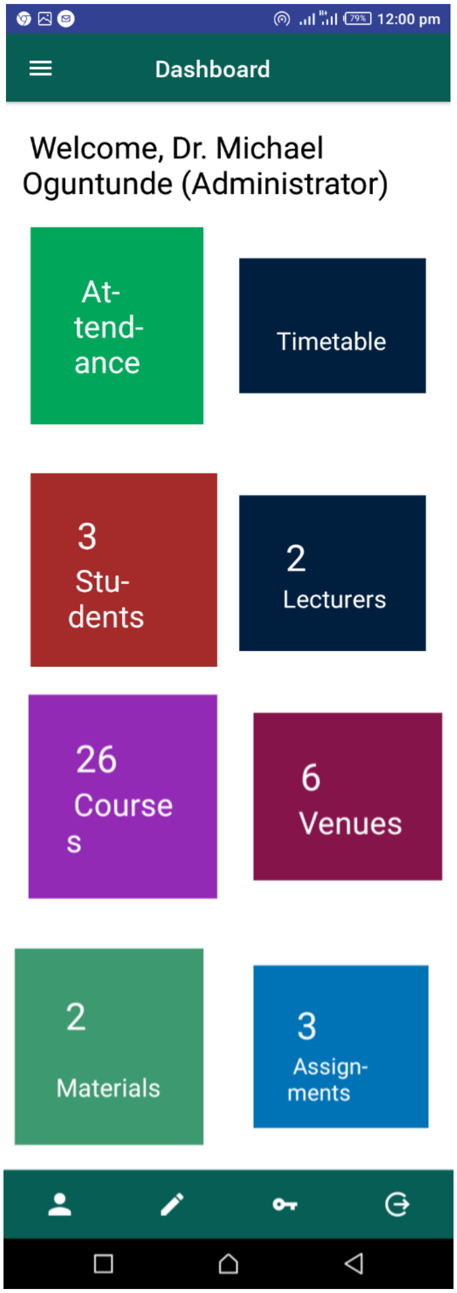


Fig. 4.2 Lecturer’s Login Screen

4.5.3 LECTURER'S DASHBOARD SCREEN

This is the screen that will be displayed to the lecturer upon a successful login. This dashboard is displayed in two different forms, one for the administrator and the other for the lecturer without administrator role.

If you are logged in as an administrator, your dashboard will be displayed like this

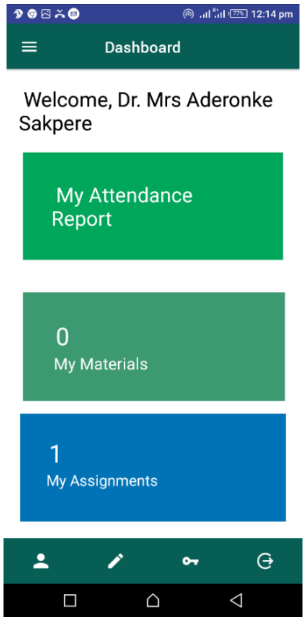


Welcome, Dr. Michael Oguntunde (Administrator)

Fig. 4.3 Administrator's Dashboard Screen

- If you want to view the attendance reports for your courses, click on **Attendance.**
- If you want to view timetable details, click on **Timetable.**
- If you want to view the list of all students, click on **Students.**
- If you want to view the list of all lecturers, click on **Lecturers.**
- If you want to view the list of all courses, click on **Courses.**
- If you want to view the list of all venues, click on **Venues.**
- If you want to view the list of all your uploaded materials, click on **Materials.**
- If you want to view the list of all your uploaded assignments, click on **Assignments.**

If you are logged in as non-administrator, your dashboard will be displayed like this:



12:14 pm

Dashboard

Welcome, Dr. Mrs Aderonke Sakpere

My Attendance Report

0
My Materials

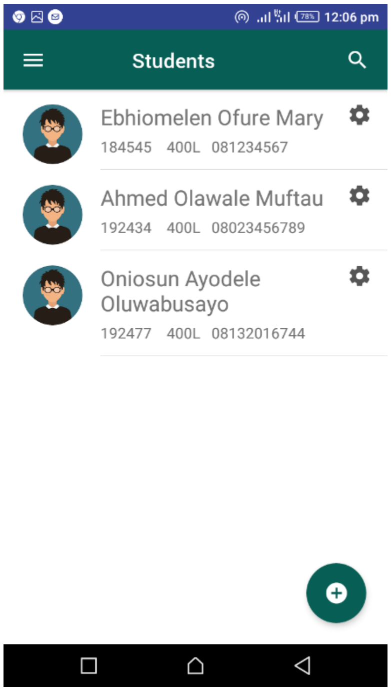
1
My Assignments

Fig. 4.4 Non-administrator's Dashboard Screen

- If you want to view the attendance reports for your courses, click on **My Attendance Reports**.
- If you want to view the list of all your uploaded materials, click on **My Materials.**
- If you want to view the list of all your uploaded assignments, click on **My Assignments.**

4.5.4 STUDENTS SCREEN

This is the screen where the list of all students can be viewed.



Ebhiomelen Ofure Mary
184545 400L 081234567

Ahmed Olawale Muftau
192434 400L 08023456789

Oniosun Ayodele Oluwabusayo
192477 400L 08132016744

Fig. 4.5 All Students Screen

To add new student:

- Click on the "+" icon and a screen to add new student will be displayed ⊕
- Fill all the fields appropriately and click on the **Add Student** button.

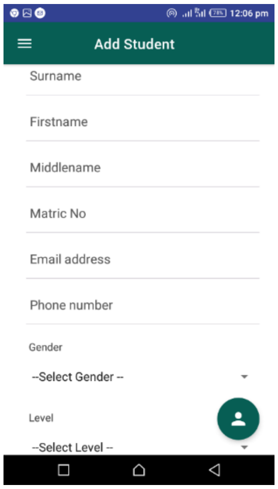
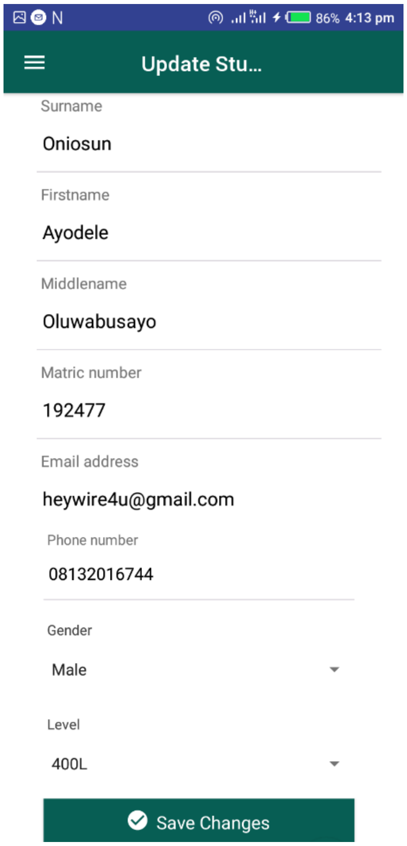


Fig. 4.5 Add Student Screen

- After the student has been added, the screen that displays the list of all students will be displayed.
- To view a student’s profile, just click on the name of the student.

To modify an already existing student:

- Click on the gear icon beside the name of the student ⚙
- Two options will be displayed – **update** and **delete.**
- Click on update.
- A screen called update student that contains the details of the students will be displayed.
- Effect changes to any of the field and then click on **Save Changes**.



Surname
Oniosun

Firstname
Ayodele

Middlename
Oluwabusayo

Matric number
192477

Email address
heywire4u@gmail.com

Phone number
08132016744

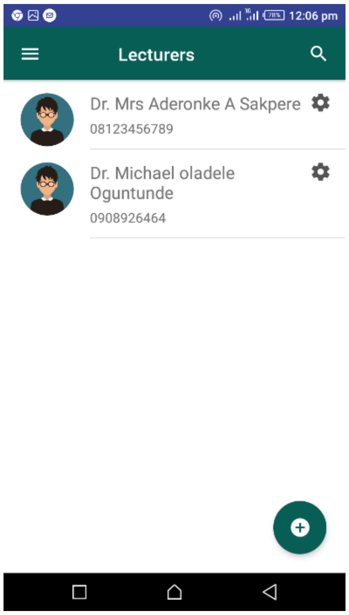
Gender
Male

Level
400L

Fig. 4.6 Update Student Screen

4.5.5 LECTURERS SCREEN

This is the screen where the list of all lecturers can be viewed.



Dr. Mrs Aderonke A Sakpere
08123456789

Dr. Michael oladele Oguntunde
0908926464

Fig. 4.7 All Lecturers Screen

To add new lecturer:

- Click on the “+” icon and a screen to add new lecturer will be displayed.
- Fill all the fields appropriately and click on the **Add Lecturer** button.

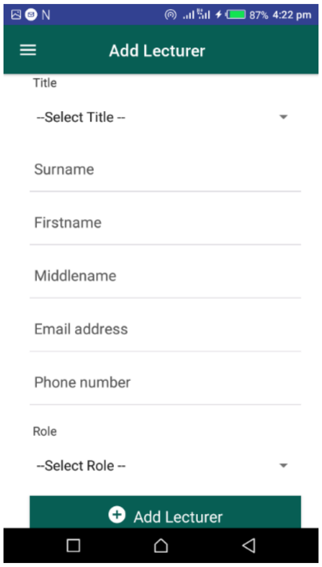
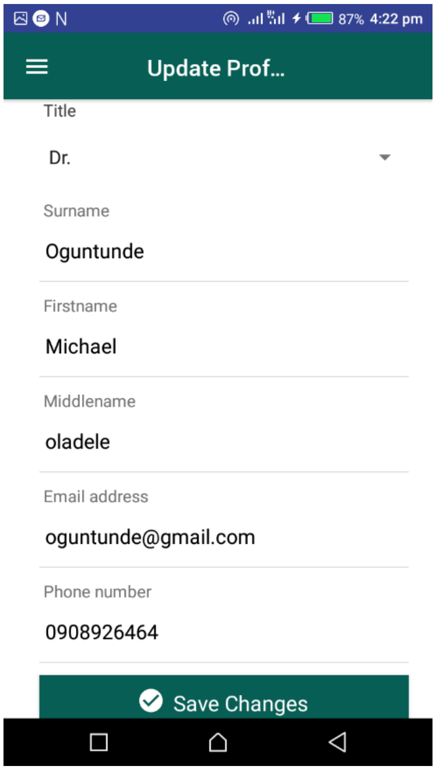


Fig. 4.8 Add Lecturer Screen

- After the lecturer has been added, the screen that displays the list of all lecturers will be displayed.
- To view a lecturer's profile, just click on the name of the lecturer.

To modify an already existing lecturer:

- Click on the gear icon beside the name of the lecturer ⚙
- Two options will be displayed – **update** and **delete.**
- Click on update.
- A screen called update lecturer that contains the details of the lecturers will be displayed.
- Effect changes to any of the field and then click on **Save Changes**.



Title
Dr.

Surname
Oguntunde

Firstname
Michael

Middlename
oladele

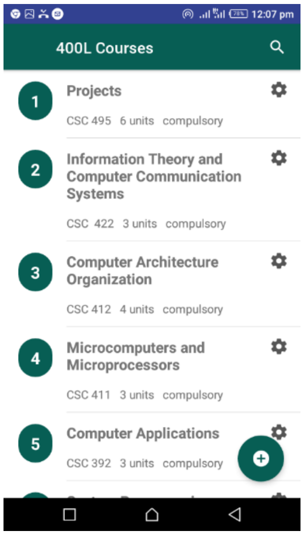
Email address
oguntunde@gmail.com

Phone number
0908926464

Fig. 4.9 Update Lecturer Screen

4.5.6 COURSES SCREEN

This is the screen where the list of all courses can be viewed.



12:07 pm

400L Courses

1 Projects
CSC 495 6 units compulsory

2 Information Theory and Computer Communication Systems
CSC 422 3 units compulsory

3 Computer Architecture Organization
CSC 412 4 units compulsory

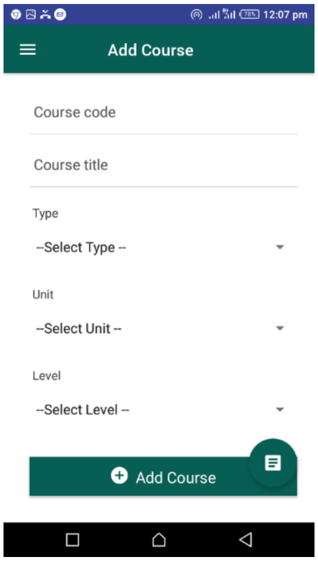
4 Microcomputers and Microprocessors
CSC 411 3 units compulsory

5 Computer Applications
CSC 392 3 units compulsory

Fig. 4.10 All Courses Screen

To add new course:

- Click on the “+” icon and a screen to add new course will be displayed.
- Fill all the fields appropriately and click on the **Add Course** button.



12:07 pm

Add Course

Course code

Course title

Type
--Select Type --

Unit
--Select Unit --

Level
--Select Level --

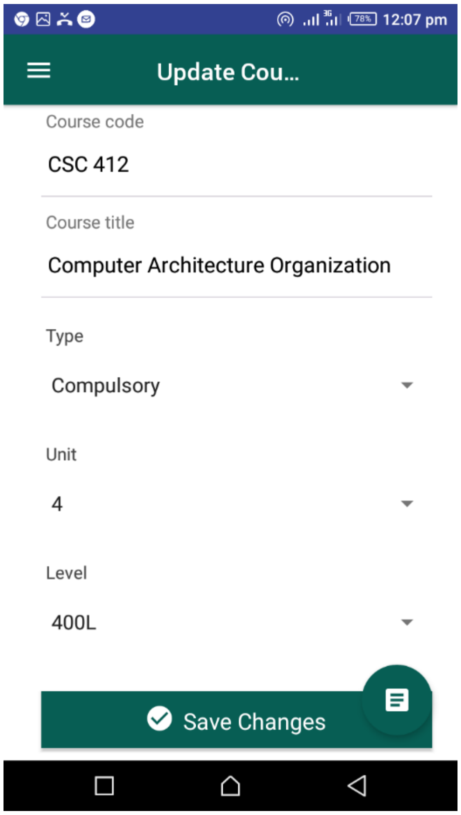
Add Course

Fig. 4.11 Add Course Screen

- After the course has been added, the screen that displays the list of all courses will be displayed.

To modify an already existing course:

- Click on the gear icon beside the name of the course ⚙
- Two options will be displayed – **update** and **delete.**
- Click on update.
- A screen called update course that contains the details of the course will be displayed.
- Effect changes to any of the field and then click on **Save Changes**.



Course code
CSC 412

Course title
Computer Architecture Organization

Type
Compulsory

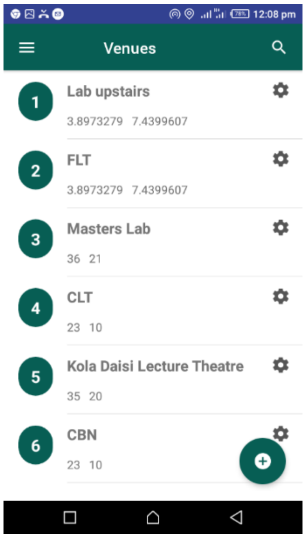
Unit
4

Level
400L

Fig. 4.12 Update Lecturer Screen

4.5.7 VENUES SCREEN

This is the screen where the list of all venues will be displayed.



12:08 pm

Venues

1 Lab upstairs
3.8973279 7.4399607

2 FLT
3.8973279 7.4399607

3 Masters Lab
36 21

4 CLT
23 10

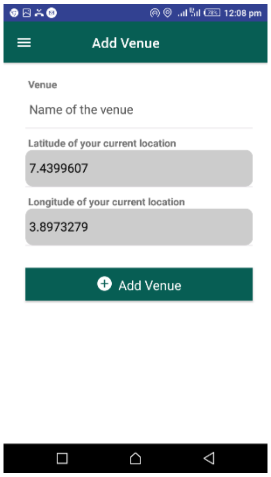
5 Kola Daisi Lecture Theatre
35 20

6 CBN
23 10

Fig. 4.13 All Courses Screen

To add new venue:

- Ensure that the location feature of your android phone is turned on.
- Click on the "+" icon and a screen to add new venue will be displayed. ⊕
- The latitude and the longitude of your location will be automatically gotten, so no need to input that.
- Input the name of the venue and click on the **Add Venue** button.



12:08 pm

Add Venue

Venue
Name of the venue

Latitude of your current location
7.4399607

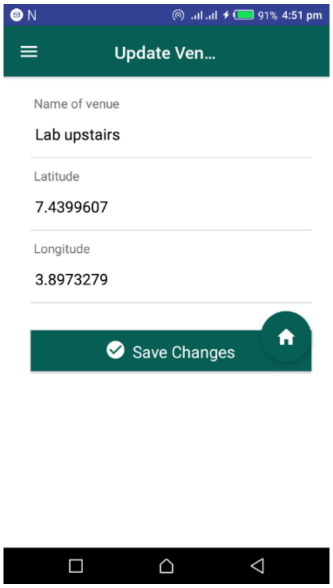
Longitude of your current location
3.8973279

Fig. 4.14 Add Venue Screen

- After the venue has been added, the screen that displays the list of all venues will be displayed.

To modify an already existing venue:

- Click on the gear icon beside the name of the venue ⚙
- Two options will be displayed – **update** and **delete.**
- Click on update.
- A screen called update venue that contains the details of the venue will be displayed.
- Effect changes to any of the field and then click on **Save Changes**.



Name of venue: Lab upstairs

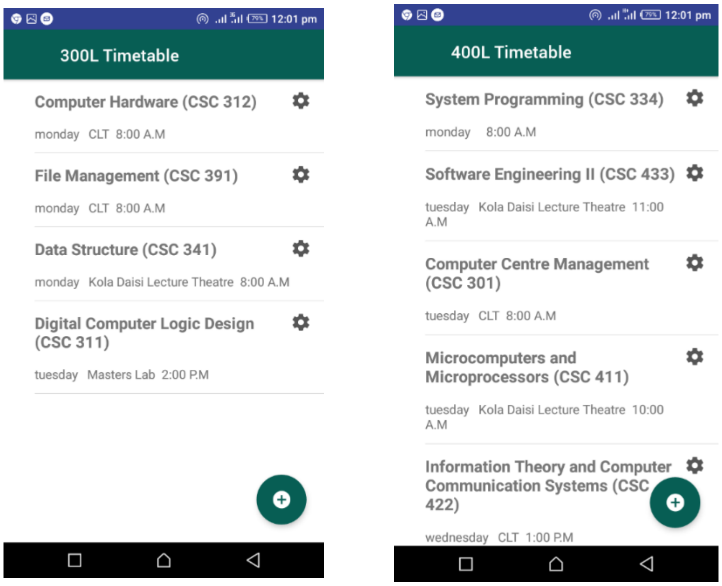
Latitude: 7.4399607

Longitude: 3.8973279

Fig. 4.15 Update Venue Screen

4.5.8 TIMETABLES SCREEN

This is the screen where the list of all timetables for a level will be displayed.



300L Timetable

Computer Hardware (CSC 312)
monday CLT 8:00 A.M

File Management (CSC 391)
monday CLT 8:00 A.M

Data Structure (CSC 341)
monday Kola Daisi Lecture Theatre 8:00 A.M

Digital Computer Logic Design (CSC 311)
tuesday Masters Lab 2:00 P.M

400L Timetable

System Programming (CSC 334)
monday 8:00 A.M

Software Engineering II (CSC 433)
tuesday Kola Daisi Lecture Theatre 11:00 A.M

Computer Centre Management (CSC 301)
tuesday CLT 8:00 A.M

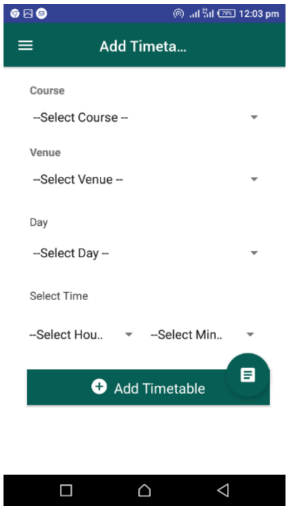
Microcomputers and Microprocessors (CSC 411)
tuesday Kola Daisi Lecture Theatre 10:00 A.M

Information Theory and Computer Communication Systems (CSC 422)
wednesday CLT 1:00 P.M

Fig. 4.16 Timetable Screen

To add new timetable:

- Click on the "+" icon and a screen to add new timetable will be displayed.
- Fill all the fields appropriately and click on **Add Timetable** button.



Course
--Select Course --

Venue
--Select Venue --

Day
--Select Day --

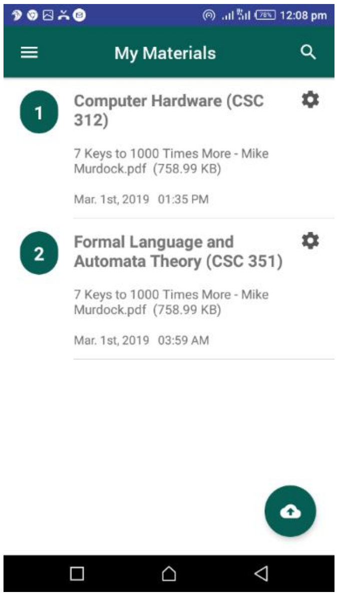
Select Time
--Select Hou.. --Select Min..

Fig. 4.17 Add Timetable Screen

- After the timetable has been added, the screen that displays the list of all timetables will be displayed.

4.5.9 LECTURER MATERIALS SCREEN

This is the screen where the list of all materials will be displayed.



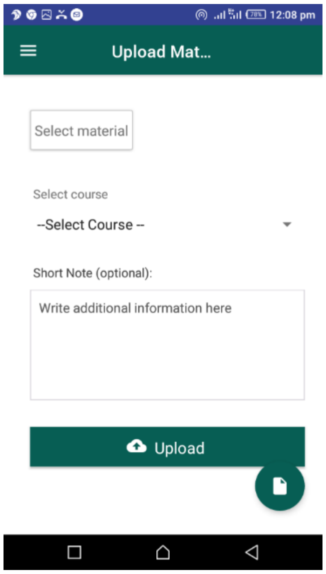
1 Computer Hardware (CSC 312)
7 Keys to 1000 Times More - Mike Murdock.pdf (758.99 KB)
Mar. 1st, 2019 01:35 PM

2 Formal Language and Automata Theory (CSC 351)
7 Keys to 1000 Times More - Mike Murdock.pdf (758.99 KB)
Mar. 1st, 2019 03:59 AM

Fig. 4.18 My Materials Screen

To add upload new material:

- Click on the file icon and a screen to upload new material will be displayed.
- Click on **select material** button to select the material to be uploaded.
- Fill other fields appropriately and click on **upload** button.



Select course
--Select Course --

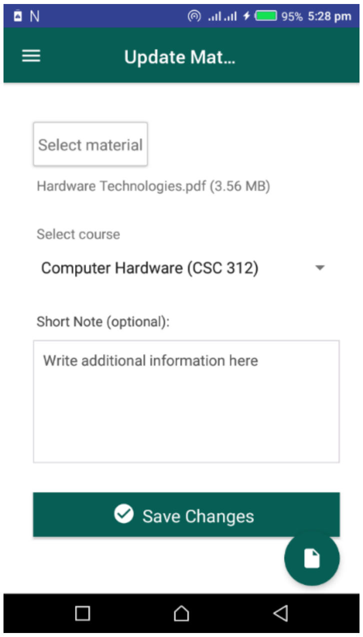
Short Note (optional):
Write additional information here

Fig. 4.19 Upload Material Screen

- After the new material has been uploaded, the screen that displays the list of all uploaded materials will be displayed.

To modify an already existing material:

- Click on the gear icon beside the name of the material ⚙
- Two options will be displayed – **update** and **delete.**
- Click on update.
- A screen called update material that contains the details of the material will be displayed.
- Effect changes to any of the field and then click on **Save Changes**.



Hardware Technologies.pdf (3.56 MB)

Select course

Computer Hardware (CSC 312)

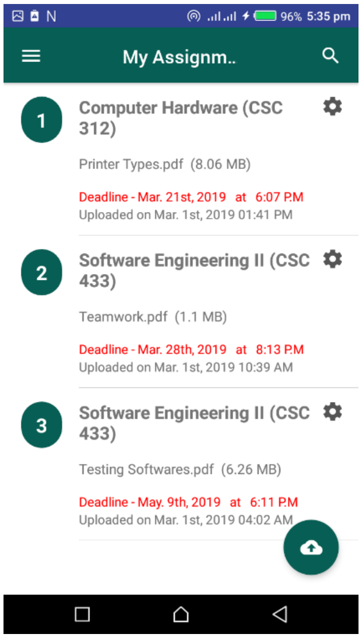
Short Note (optional):

Write additional information here

Fig. 4.20 Update Material Screen

4.5.10 LECTURER ASSIGNMENTS SCREEN

This is the screen where the list of all assignments will be displayed.



96% 5:35 pm

My Assignm..

1 Computer Hardware (CSC 312)

Printer Types.pdf (8.06 MB)

Deadline - Mar. 21st, 2019 at 6:07 PM
Uploaded on Mar. 1st, 2019 01:41 PM

2 Software Engineering II (CSC 433)

Teamwork.pdf (1.1 MB)

Deadline - Mar. 28th, 2019 at 8:13 PM
Uploaded on Mar. 1st, 2019 10:39 AM

3 Software Engineering II (CSC 433)

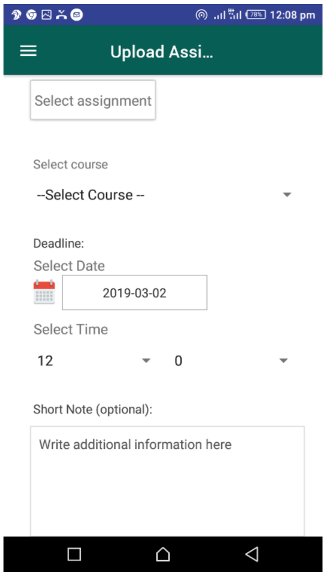
Testing Softwares.pdf (6.26 MB)

Deadline - May. 9th, 2019 at 6:11 PM
Uploaded on Mar. 1st, 2019 04:02 AM

Fig. 4.21 My Assignments Screen

To add upload new assignment:

- Click on the cloud icon and a screen to upload new material will be displayed.
- Click on **select assignment** button to select the assignment to be uploaded.
- Fill other fields appropriately and click on **upload** button.



12:08 pm

Upload Assi...

Select assignment

Select course

--Select Course --

Deadline:

Select Date

2019-03-02

Select Time

12 0

Short Note (optional):

Write additional information here

Fig. 4.22 Upload Assignment Screen

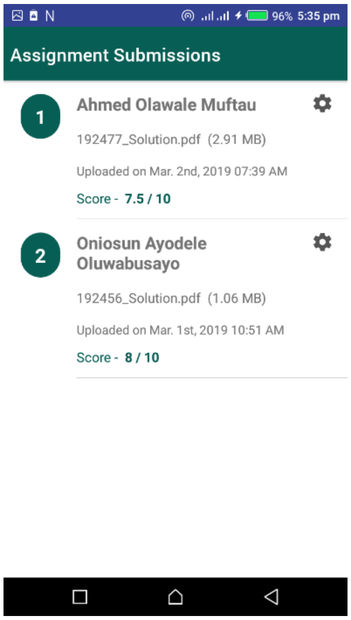
- After the new assignment has been uploaded, the screen that displays the list of all uploaded assignments will be displayed.

To modify an already existing assignment:

- Click on the gear icon beside the name of the assignment ⚙
- Two options will be displayed – **update, view submission** and **delete.**
- Click on update.
- A screen called update assignment that contains the details of the assignment will be displayed.
- Effect changes to any of the field and then click on **Save Changes**.

To view student's submission to an already existing assignment:

- Click on the gear icon beside the name of the assignment ⚙
- Two options will be displayed – **update, view submission** and **delete.**
- Click on view submission.
- A screen called assignment submission that contains the list of students who have submitted assignment as well as their grades will be displayed.



Assignment Submissions

1 Ahmed Olawale Muftau ⚙
192477_Solution.pdf (2.91 MB)
Uploaded on Mar. 2nd, 2019 07:39 AM
Score - 7.5 / 10

2 Oniosun Ayodele Oluwabusayo ⚙
192456_Solution.pdf (1.06 MB)
Uploaded on Mar. 1st, 2019 10:51 AM
Score - 8 / 10

Fig. 4.23 Update Material Screen

To grade a student’s assignment:

- Click on the gear icon beside the name of the student ⚙
- An option **Assign Grade** will be displayed.
- A modal will be popped up to input the score of the student.
- Input the grade and then click on the **update.**

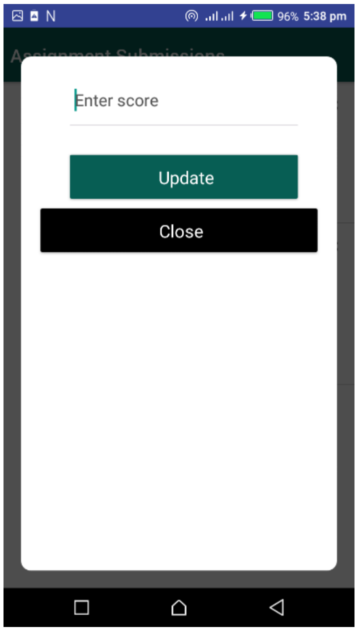
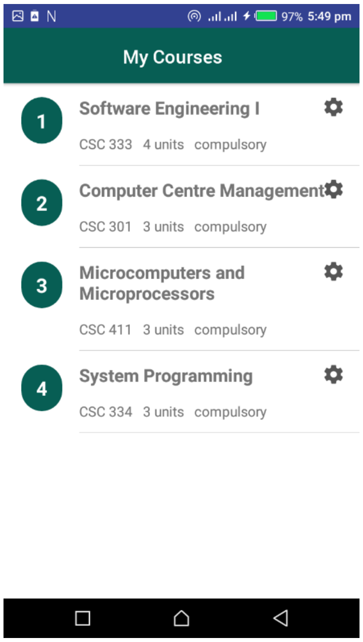


Fig. 4.23 Grade Student Screen

4.5.11 LECTURER ATTENDANCE SCREEN

This is where the records of the attendance taken by students per each course is being displayed.

The list of all the courses taken by the lecturer will be displayed first.



My Courses

1 Software Engineering I
CSC 333 4 units compulsory

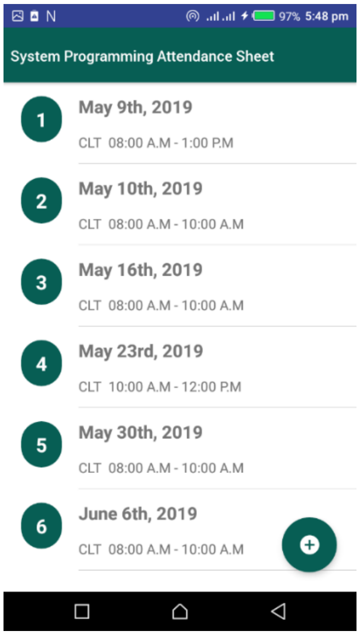
2 Computer Centre Management
CSC 301 3 units compulsory

3 Microcomputers and Microprocessors
CSC 411 3 units compulsory

4 System Programming
CSC 334 3 units compulsory

Fig. 4.24 My Courses Screen

Click on the name of the course to view the attendance details.



System Programming Attendance Sheet

1 May 9th, 2019
CLT 08:00 A.M - 1:00 P.M

2 May 10th, 2019
CLT 08:00 A.M - 10:00 A.M

3 May 16th, 2019
CLT 08:00 A.M - 10:00 A.M

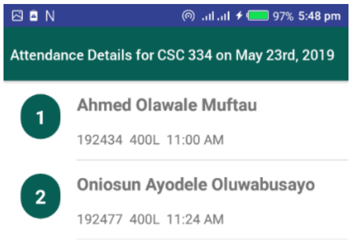
4 May 23rd, 2019
CLT 10:00 A.M - 12:00 P.M

5 May 30th, 2019
CLT 08:00 A.M - 10:00 A.M

6 June 6th, 2019
CLT 08:00 A.M - 10:00 A.M

Fig. 4.25 Attendance sheet for a course

Then, click on any date to view the list of students that marked attendance that day.



Attendance Details for CSC 334 on May 23rd, 2019

1 Ahmed Olawale Muftau
192434 400L 11:00 AM

2 Oniosun Ayodele Oluwabusayo
192477 400L 11:24 AM

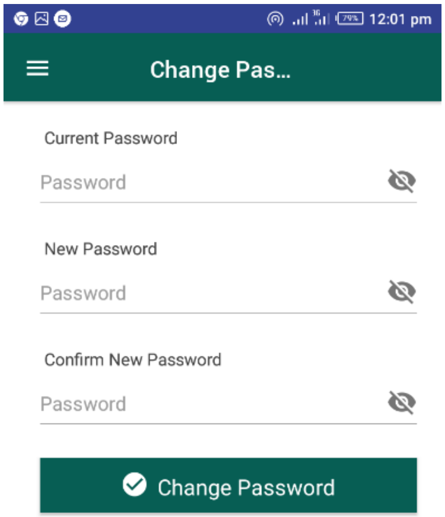
Fig. 4.26 List of students that marked attendance

4.5.12 CHANGE PASSWORD SCREEN

For security purpose, it is important that the lecturer change his password periodically.

To change your password:

- Ensure that your current password you used to log in is appropriately supplied.
- Enter new password and retype the new password in confirm new password field.
- Click on change password.
- You will be logged out and asked to login again.



Change Pas...

Current Password
Password

New Password
Password

Confirm New Password
Password

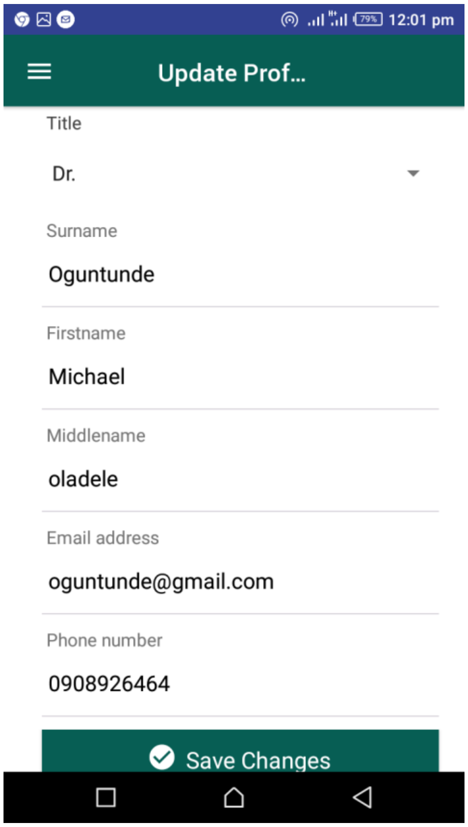
Fig. 4.27 Change password

4.5.13 UPDATE PROFILE SCREEN

Sometimes, the lecturer might feel the need to change details like his email address, phone number and any other necessary details.

To update a lecturer’s profile:

- Navigate to the dashboard and click on the pencil icon at the bottom part ✎
- A screen to update the details will be displayed.
- Effect the changes and click on **Save Changes.**



Title
Dr.

Surname
Oguntunde

Firstname
Michael

Middlename
oladele

Email address
oguntunde@gmail.com

Phone number
0908926464

Fig. 4.28 Update Profile

4.5.14 STUDENTS' LOGIN SCREEN

For any student to login via this panel, his details must have been added by the administrator to the database. A student can login by doing the following:

- Supply your matric number and password.
- Click on Login button.

- Wait for your details to be authenticated.
- If the authentication is successful, you will be directed to the dashboard, else, an error message "Incorrect login details" will be displayed.

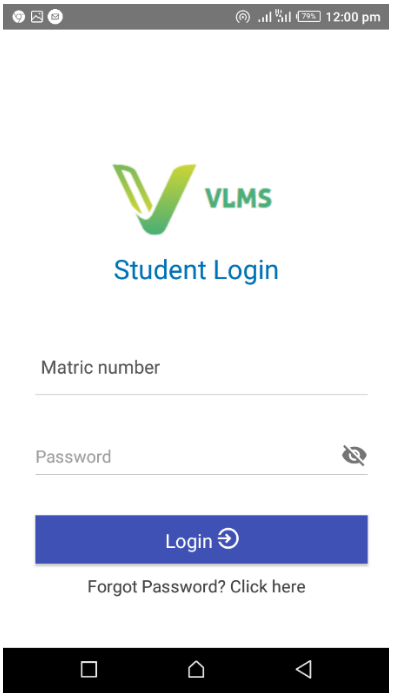
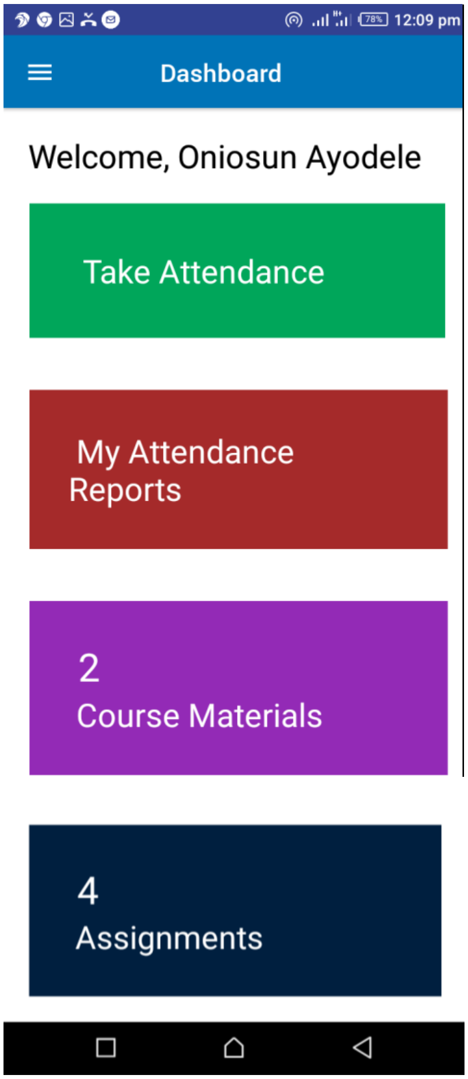


Fig. 4.29 Student's Login Screen

4.5.15 STUDENTS' DASHBOARD SCREEN

This is the screen that will be displayed to the student upon a successful login.



Welcome, Oniosun Ayodele

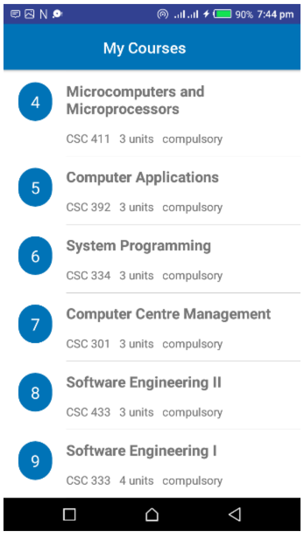
Fig. 4.30 Student's Dashboard Screen

- If you want to take attendance, click on **Take Attendance.**
- If you want to view your attendance report, click on **My Attendance Reports.**
- If you want to view the list of uploaded materials, click on **Course Materials.**
- If you want to view the list of uploaded assignments, click on **Assignments.**

4.5.16 STUDENTS’ ATTENDANCE REPORT SCREEN

This is the screen where the attendance report for each course offered by the student is displayed.

The list of all the courses you are offering will be displayed first.



90% 7:44 pm

My Courses

4 Microcomputers and Microprocessors
CSC 411 3 units compulsory

5 Computer Applications
CSC 392 3 units compulsory

6 System Programming
CSC 334 3 units compulsory

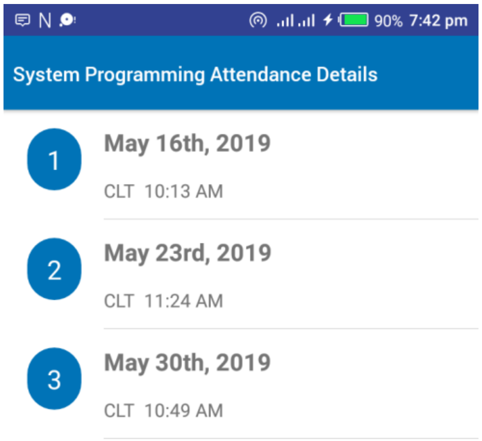
7 Computer Centre Management
CSC 301 3 units compulsory

8 Software Engineering II
CSC 433 3 units compulsory

9 Software Engineering I
CSC 333 4 units compulsory

Fig. 4.31 My Courses Screen

Click on a course and the attendance details of the student in that course will be displayed.



90% 7:42 pm

System Programming Attendance Details

1 May 16th, 2019
CLT 10:13 AM

2 May 23rd, 2019
CLT 11:24 AM

3 May 30th, 2019
CLT 10:49 AM

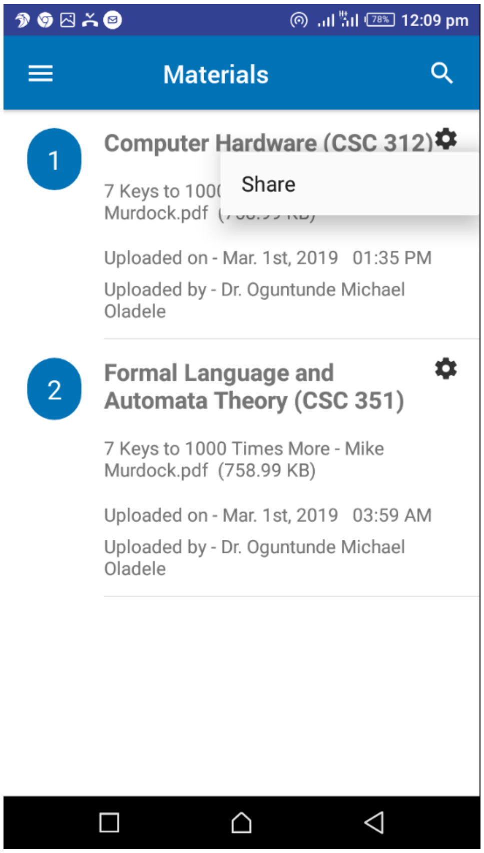
Fig. 4.32 Attendance details of the student

4.5.17 STUDENTS’ MATERIALS SCREEN

This is the screen where the uploaded course materials can be downloaded and shared by the student.

To download the course material, click on the name of the material.

To share the course material, click on the gear icon and click on **share** link.



Computer Hardware (CSC 312)

7 Keys to 1000
Murdock.pdf (

Uploaded on - Mar. 1st, 2019 01:35 PM

Uploaded by - Dr. Oguntunde Michael Oladele

Formal Language and Automata Theory (CSC 351)

7 Keys to 1000 Times More - Mike Murdock.pdf (758.99 KB)

Uploaded on - Mar. 1st, 2019 03:59 AM

Uploaded by - Dr. Oguntunde Michael Oladele

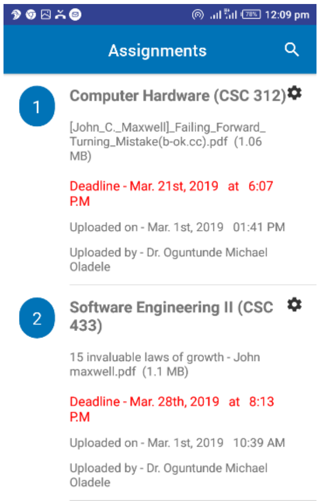
Fig. 4.33 Student’s Materials Screen

4.5.18 STUDENTS’ ASSIGNMENTS SCREEN

This is the screen where the uploaded assignments can be downloaded and shared by the student.

To download the assignment, click on the name of the assignment.

To share the assignment, click on the gear icon and click on **share** link.



1 Computer Hardware (CSC 312)

[John_C._Maxwell]_Failing_Forward_Turning_Mistake(b-ok.cc).pdf (1.06 MB)

Deadline - Mar. 21st, 2019 at 6:07 P.M

Uploaded on - Mar. 1st, 2019 01:41 PM

Uploaded by - Dr. Oguntunde Michael Oladele

2 Software Engineering II (CSC 433)

15 invaluable laws of growth - John maxwell.pdf (1.1 MB)

Deadline - Mar. 28th, 2019 at 8:13 P.M

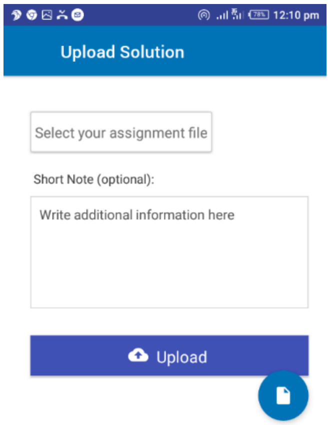
Uploaded on - Mar. 1st, 2019 10:39 AM

Uploaded by - Dr. Oguntunde Michael Oladele

Fig 4.34 Student’s Assignments

To submit solution to the assignment:

- Click on the gear icon and click on **submit solution** link.
- Click on **select your assignment file** to select your solution to the assignment.
- Fill in a short note, if any.
- Then click on **upload** button.



Short Note (optional):

Write additional information here

Fig. 4.33 Student’s Submit Assignment Solutions Screen

4.6 TESTING

Testing of the software is necessary in order to check the existence of fault in the software so as to make changes that will remove the faults. Testing is necessary to prevent system failure.

The major testing done include:

4.6.1 Unit testing

Unit testing breaks down the software into components and verifies the functionalities of each individually within the programming environment. Also known as component testing, it tests if each component works as it is supposed to, why it gets the required input and returns the required output.

The components present in MVLMS include:

- Login.
- Attendance.
- Assignment.
- Courses.
- Lecturers.
- Students.
- Materials.
- Timetables.
- Venues.
- Profile.

4.6.2 Test Case Specification

The test cases specify the inputs, predicted results and execution conditions. Each test case aims to evaluate the operation of a service element of the system.

Test Items: The items to be tested are all of the voice enabled mobile airline reservation services.

Test case pass/fail criteria: These includes criteria to identify decision rules that are used to determine whether a test case passes or fails a test.

Pass Criteria: The test cases executed on the system will pass if they meet all specific requirements of the system.

Fail Criteria: A test case is said to fail, if the desired functionality is not satisfied by the system.

Testing Environment: An individual not involved in the project development has conducted this test on mobile phones to make sure successful operation of services.

Valid Data: Data is valid only when it is consistent with the defined data types in the data contract.

Invalid Data: Data which is not consistent with the one defined in the data contract to use the services operation. Both valid and invalid data are provided in the test in order to evaluate the service's response in both cases.

4.6.2.1 TEST CASE SPECIFICATION FOR LOGIN MODULES

Test name: Login_test				
Test module: Login component				
Assumptions: Navigates to User's dashboard				
Purpose: to ensure that the user get access to the platform				
Test data: **Student:** Matric number – 192477 Password – 192477 **Lecturer:** Email address – oguntunde@gmail.com Password - JesusChristIsLord				
Steps executed	**Expected results**	**Actual data**	**Results obtained**	**Pass/fail**
Insert a valid value for each test data and click on login	Navigates to user dashboard	Valid data for both fields	Dashboard displayed	pass
Insert an invalid value for each test data and click on login	Display "incorrect login details"	Invalid data for both fields	Display "incorrect login details"	pass

4.6.2.2 TEST CASE SPECIFICATION FOR ATTENDANCE MODULES

Test name: Take_attendance_test

Test module: Take attendance component				
Assumptions: Take attendance				
Purpose: to ensure that the student take the attendance when he gets to the venue of the lecturer				
Test data: Course: System Programming, Level: 400L, Day = Monday, Time = 09:00, Date: 2019-03-07, Student ID = 1				
Steps executed	**Expected results**	**Actual data**	**Results obtained**	**Pass/fail**
Insert a valid value for each test data and take attendance	Student attendance is submitted	Valid data for all fields	Attendance successfully taken	Pass
Inability to automatically retrieve some details	Display “All fields are compulsory”	All fields are compulsory	Display “All fields are compulsory”	Pass

4.6.2.3 TEST CASE SPECIFICATION FOR ADD NEW STUDENT

Test name: Add_student_test				
Test module: Add student component				
Assumptions: Add new student				
Purpose: to ensure that the student details is added to the system.				
Test data: Surname = Oniosun, Firstname = Ayodele, Middlename = Abraham, Matric no = 192477, Email address = heywire4u@gmail.com, Phone number = 08132016744, Gender = Male, Level = 400				
Steps executed	**Expected results**	**Actual data**	**Results obtained**	**Pass/fail**
Insert a valid value for each test data and click on add	Student gets added	Valid data for all fields	Student successfully added	Pass
Insert an empty value for any of each test data and click on add	Display “Please fill all fields”	Fill all fields	Display “Please fill all fields”	Pass

Insert an already existing value for any of matric no, phone number field and click on add	Display "Details has been added before"	Existing details for any of the field	Display "Details has been added before"	Pass

4.6.2.4 TEST CASE SPECIFICATION FOR ADD NEW LECTURER

Test name: Add_lecturer_test				
Test module: Add lecturer component				
Assumptions: Add new lecturer				
Purpose: to ensure that the lecturer details is added to the system.				
Test data: Title = Dr, Surname = Oguntunde, Firstname = Michael, Email address = oguntunde@gmail.com, Phone number = 09033965925, Role = Administrator				
Steps executed	**Expected results**	**Actual data**	**Results obtained**	**Pass/fail**
Insert a valid value for each test data and click on add	Lecturer gets added	Valid data for all fields	Lecturer successfully added	Pass
Insert an empty value for any of each test data and click on add	Display "Please fill all fields"	Fill all fields	Display "Please fill all fields"	Pass
Insert an already existing value for any of email address, phone number and click on add	Display "Details has been added before"	Existing details for any of the field	Display "Details has been added before"	Pass

4.6.2.5 TEST CASE SPECIFICATION FOR ADD NEW COURSE

Test name: Add_course _test				
Test module: Add course component				
Assumptions: Add new course				
Purpose: to ensure that the course details is added to the system.				
Test data: Course Code = CSC 301, Course title = Computer Centre Management, Type = Required, Unit = 3, Level = 400				
Steps executed	**Expected results**	**Actual data**	**Results obtained**	**Pass/fail**
Insert a valid value for each test data and click on add	Course gets added	Valid data for all fields	Course successfully added	Pass
Insert an empty value for any of each test data and click on add	Display "Please fill all fields"	Fill all fields	Display "Please fill all fields"	Pass
Insert an already existing value for any of course code and title and click on add	Display "Details has been added before"	Existing details for any of the field	Display "Details has been added before"	Pass

4.6.2.6 TEST CASE SPECIFICATION FOR ADD NEW VENUE

Test name: Add_venue _test				
Test module: Add venue component				
Assumptions: Add new venue				
Purpose: to ensure that the venue details is added to the system.				
Test data: Name = CLT, Latitude and Longitude is automatically generated by the source code by switching on the location feature on your android phone.				
Steps executed	**Expected results**	**Actual data**	**Results obtained**	**Pass/fail**
Insert a valid value for each test data and click on add	Venue gets added	Valid data for all fields	Venue successfully added	Pass

Insert an empty value for any of each test data and click on add	Display "Please fill all fields"	Fill all fields	Display "Please fill all fields"	Pass
Insert an already existing value for venue name and click on add	Display "Details has been added before"	Existing details for any of the field	Display "Details has been added before"	Pass

4.6.2.7 TEST CASE SPECIFICATION FOR ADD NEW TIMETABLE

Test name: Add_timetable _test				
Test module: Add timetable component				
Assumptions: Add new timetable				
Purpose: to ensure that the timetable details is added to the system.				
Test data: Course = System Programming, Venue = CLT, Day = Monday, Hour = 13, Minutes = 30				
Steps executed	**Expected results**	**Actual data**	**Results obtained**	**Pass/fail**
Insert a valid value for each test data and click on add	Timetable gets added	Valid data for all fields	Timetable successfully added	Pass
Insert an empty value for any of each test data and click on add	Display "Please fill all fields"	Fill all fields	Display "Please fill all fields"	Pass
Insert an already existing value and click on add	Display "Details has been added before"	Existing details for any of the field	Display "Details has been added before"	Pass

4.6.2.8 TEST CASE SPECIFICATION FOR UPLOAD MATERIAL

Test name: Upload_material _test
Test module: Upload material component
Assumptions: Upload new material
Purpose: to ensure that the new course material is uploaded to the system.

Test data: Select material = Testing techniques.pdf, Course = Software engineering, Short note: Kindly peruse through this document.				
Steps executed	**Expected results**	**Actual data**	**Results obtained**	**Pass/fail**
Insert a valid value for each test data and click on add	Material gets uploaded	Valid data for all fields	Course material successfully uploaded	Pass
Insert an empty value for any of each test data and click on add	Display "Please fill all fields"	Fill all fields	Display "Please fill all fields"	Pass

4.6.2.9 TEST CASE SPECIFICATION FOR UPLOAD ASSIGNMENT

Test name: Upload_assignment _test				
Test module: Upload assignment component				
Assumptions: Upload new assignment				
Purpose: to ensure that the new course assignment is uploaded to the system.				
Test data: Select assignment = Testing techniques.pdf, Course = Software engineering, Deadline date: 2019-04-01, Time: 12:00, Short note: Kindly peruse through this document.				
Steps executed	**Expected results**	**Actual data**	**Results obtained**	**Pass/fail**
Insert a valid value for each test data and click on add	Assignment gets uploaded	Valid data for all fields	Course assignment successfully uploaded	Pass
Insert an empty value for any of each test data and click on add	Display "Please fill all fields"	Fill all fields	Display "Please fill all fields"	Pass

4.8 Integration Testing

Integration testing was done after successfully testing each component of the mobile application. Interfaces were created in form of components and communication was done with the database efficiently and output were produced based on the inputs that were supplied.

4.9 Acceptance Testing

Acceptance testing was done after the implementation of the system. The acceptance testing will check if the system works correctly in the user environment and if all user specified functionalities are present. It also tests if the system adheres to the policies and qualities standard.

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATION

5.1 SUMMARY

Over the years, a lot of operations have been done the manual way in many sectors in Nigeria, especially the educational sector. Taking of attendance has always been done using papers and this makes the attendance to be falsified by students as student who is present in a class can take attendance for another student who was absent. Course materials do not circulate round the students in a class as a result of negligence on the part of the course or class representative. The mutilation and outright loss of assignments is not left out as well.

Mobile Virtual Lecture Management System (MVLMS) was designed and implemented to provide a lasting solution to the aforementioned problems.

It was designed for taking attendance using Global Positioning System (GPS), generating attendance report, usage of cloud storage for course materials and assignments, thereby making it safer and eradicating any form of loss as compared to the paper-based ones. Students offering can download the uploaded materials and assignments and they can also make submission to the assignment and can be graded by lecturers.

This project is an improvement over the usage of Biometric Technology (Seema Rao, Prof. K. J. Satoa, 2013) which marks the attendance using fingerprints and face of students and RFID technology (Arulogun O. T., Olatunbosun et al, February 2013). The usage of Biometric and RFID for taking attendance is not bad and not to be condemned but the advantage this project has over them is its easy accessibility on mobile phones. Moreso, MVLMS incorporated uploads and downloads of course materials and assignments, and submission and grading of assignments, thereby giving it an edge over the previous ones.

The requirements for MVLMS were gathered by studying existing paper based system for taking attendance and submission of assignments. Also, various forms of interviews were conducted with the student of the department of Computer Science, University of Ibadan, as regards the inability of the course materials to circulate round.

Incremental approach of software development was adopted and modelling was done using Use case, Data Flow Diagrams (DFD), Entity Relationship Diagram (ERD) and Entity Relationship Model (ERM).

React Native and Native Base were used for the frontend. The backend was done using Lumen, a micro-framework of Laravel. MySQL database was adopted as well for the storage of data.

The mobile application was installed on different versions of android phones and each components was successfully tested, integrated and deployed for usage.

5.2 CONCLUSION

The deployment of this project automated the process of marking attendance, dissemination of course materials and submission of assignments in the department of Computer Science, University of Ibadan.

Students can have access to the course materials and assignments at the palm of their hands. It also helps lecturers to view the attendance reports of the courses they teach, without any stress or worrying over the inaccuracy and inefficiency of the report, as compared to the manual ones.

5.3 RECOMMENDATIONS

Though this project was implemented using React Native, which is a cross-platform framework for developing mobile apps, but due to time constraint, non-availability of iPhone and Mac OS for testing, the implementation of this project only runs on android operating system. Therefore, there is a need to improve upon the codebase for this project to make it suitable for iPhones.

Moreso, there is need for facial recognition of students during the process of marking attendance in order to make the attendance module more secured and efficient.

REFERENCES

1. Seema Rao, Prof. K. J. Satoa: An Attendance Monitoring System Using Biometrics Authentication, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 4, April 2013.
2. Arulogun O. T., Olatunbosun. A., Fakolujo O. A., and Olaniyi. O. M.: RFID-Based Students Attendance Management System International Journal of Scientific & Engineering Research Volume 4, Issue 2, February2013.
3. Unnati A. Patel, Dr. Swaminarayan Priya: 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, Volume 2, Issue 8, August 2014.
4. A. Tregobov: “The Web-Based Assignment Submission Systems”, presented at NAWeb’98, University of New Brunswick, Canada, 1998.
5. Godfrey B (1997): Problems with Email Assignments, Department of Information Systems, University of Tasmania[online] Available at http://www.infosys.utas.edu.au/people/individuals/bob_godfrey/seminars/dec97/emailagn.html.
6. University of Queensland Australia [online] Available at: http://submit.itee.uq.edu.au/. [5] NYU, Undergraduate Division, Computer Science [Online]Available at: http://www.oracle-base.com.
7. Linnaeus University Sweden [online] Available at: http://lnu.divaportal.org.
8. Mc’ Oswel Jamin Sibin (2010): RFID Based Attendance System. University of Technology Malaysia: Degree Thesis.
9. Bermuda's RFID Vehicle Registration System Could Save $2 Million/Year http://www.rfidjournal.com/article/view/3321.
10. Mohd F.B, 2008: RFID attendance system.
11. T.S.Lim, S.C.Sim, and M.M.Mansor, “RFID based attendance system,” 2009 IEEE Symosium on Industrial Electronics and Applications, no. Isiea, pp. 778–782, Oct. 2009.
12. M.Kassim, H.Mazlan, N.Zaini, and M.K.Salleh, “Webbased student attendance system using RFID technology,” 2012 IEEE Control and System Graduate Research Colloquium, no. Icsgrc, pp. 213–218, Jul. 2012.

13. Seideman, T. “Barcodes Sweep the World. Wonder of Modern Technology”. http://www.barcoding.com/information/barcode_history.shtml.
14. C. Saraswat, C. et al, “An Efficient Automatic Attendance System using Fingerprint Verification Technique”. International Journal on Computer Science and Engineering. 2(02):264-269, 2010
15. Liu F, Zhao Q, Zhang D.A novel hierarchical fingerprint matching approach. Pattern Recogn 2011;
16. S. Pankanti, S. Prabhakar, and A.K. Jain, “On the Individuality of Fingerprints”. IEEE Transaction on Pattern Analysis and Machine Intelligence.24(8), 2002
17. M. Kamaraju, P. A. Kumar, B. A. Krishna and B Rajasekhar, “Embedded Fingerprint Recognition System”, Recent Researches in Telecomunications, Informatics, Electronics and Signal Processing, 2013
18. P. Verma and N. Gupta, “Fingerprint Based Student Attendance System Using GSM”, International Journal of Science and Research (IJSR), Vol. 2, Issue 10, 2013, pp. 128-131.
19. A. Kumar, D.C Wong, H.C. shen, and A.K jain, “personal verification using palmprint and hand geometry Biometric”, In Proc. of 4th international Conference on Audio-and video based biometric person Authentication, Guildford, UK, jun. 2003.
20. “Minutiae Based Fingerprint Recognition System”, Parul D. Sindha INDIAN JOURNAL OF RESEARCH, Volume : 1 | Issue : 12 | December 2012 ISSN - 2250-1991
21. Viola P. & Jones J.: Robust Real-Time Face Detection. International Journal of Computer Vision 57(2), 137–154, 2004.
22. M. A. Turk and A. P. Pentland, “Face Recognition Using Eigenfaces,”in Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 586–591. 1991.
23. Pace, Scott, et.al, 1995. The Global Positioning System: Accessing National Policies. Santa Monica: RAND.
24. Rafael .C. Gonzalez, Digital Image Processing, Pearson Publications.
25. Kennedy, M. (2002). The Global Positioning System and GIS.
26. Kaplan, E.D. (2006). Understanding GPS principles and application.
27. http://www.tmr.qld.gov.au
28. Branford, W., Parkinson James. & Spilker, J. „Global Positioning System: Theory and Applications‟, volume.1, Amer Inst of Aeronautics, Lincoln, NE, U.S.A. 1996.

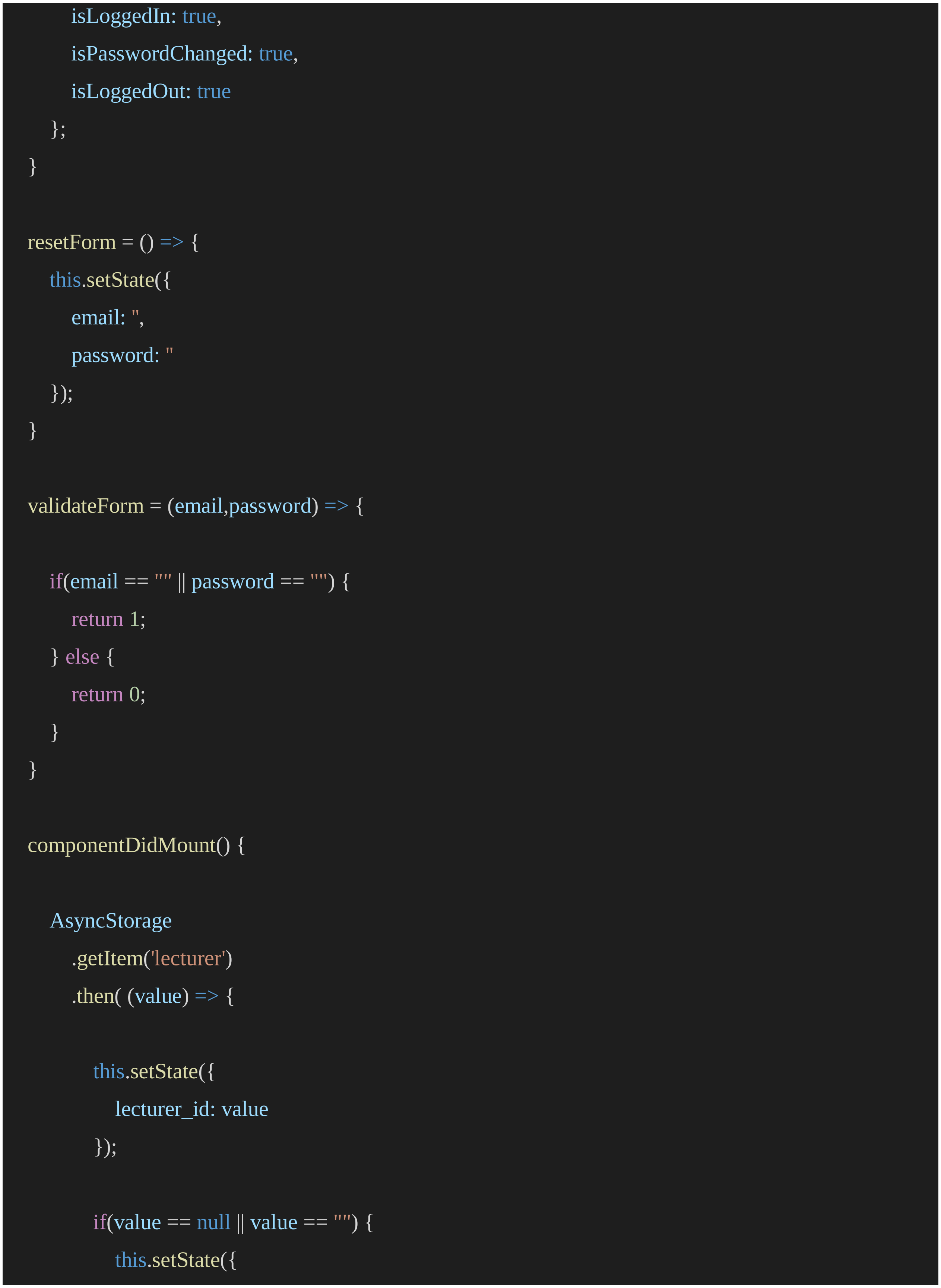
29. Paul Darbyshire, Distributed Web-Based Assignment and Management, Web-Based Learning and Teaching Technologies: Opportunities and Challenges, Anil Aggarwal (ed) pp 198-215.

30. Godfrey B (1997). Problems with Email Assignments, Department of Information Systems,University of Tasmania [online] Available at http://www.infosys.utas.edu.au/people/individuals/bob_godfrey/seminars/dec97/emailagn.html.

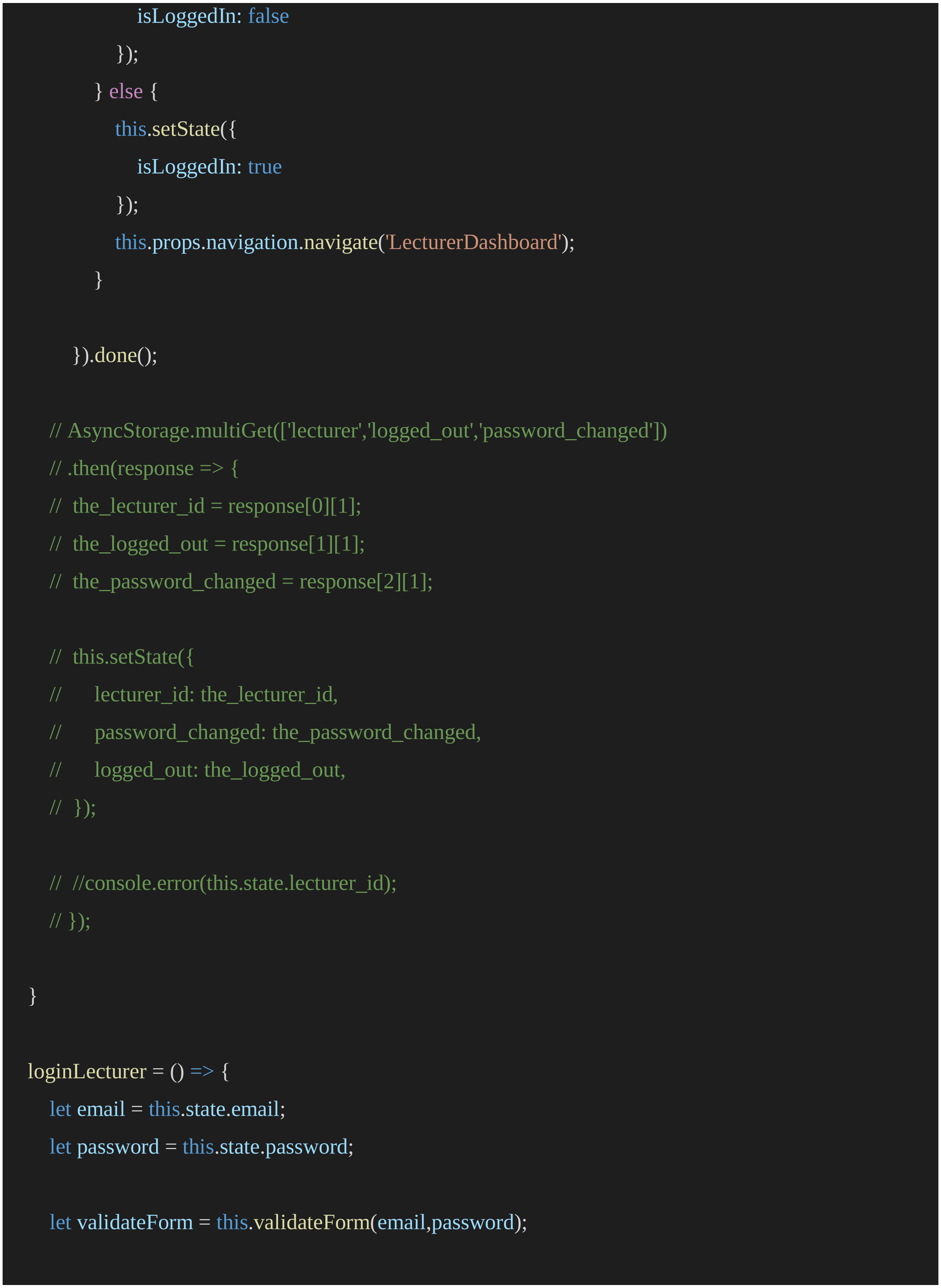
APPENDIX

LOGIN:

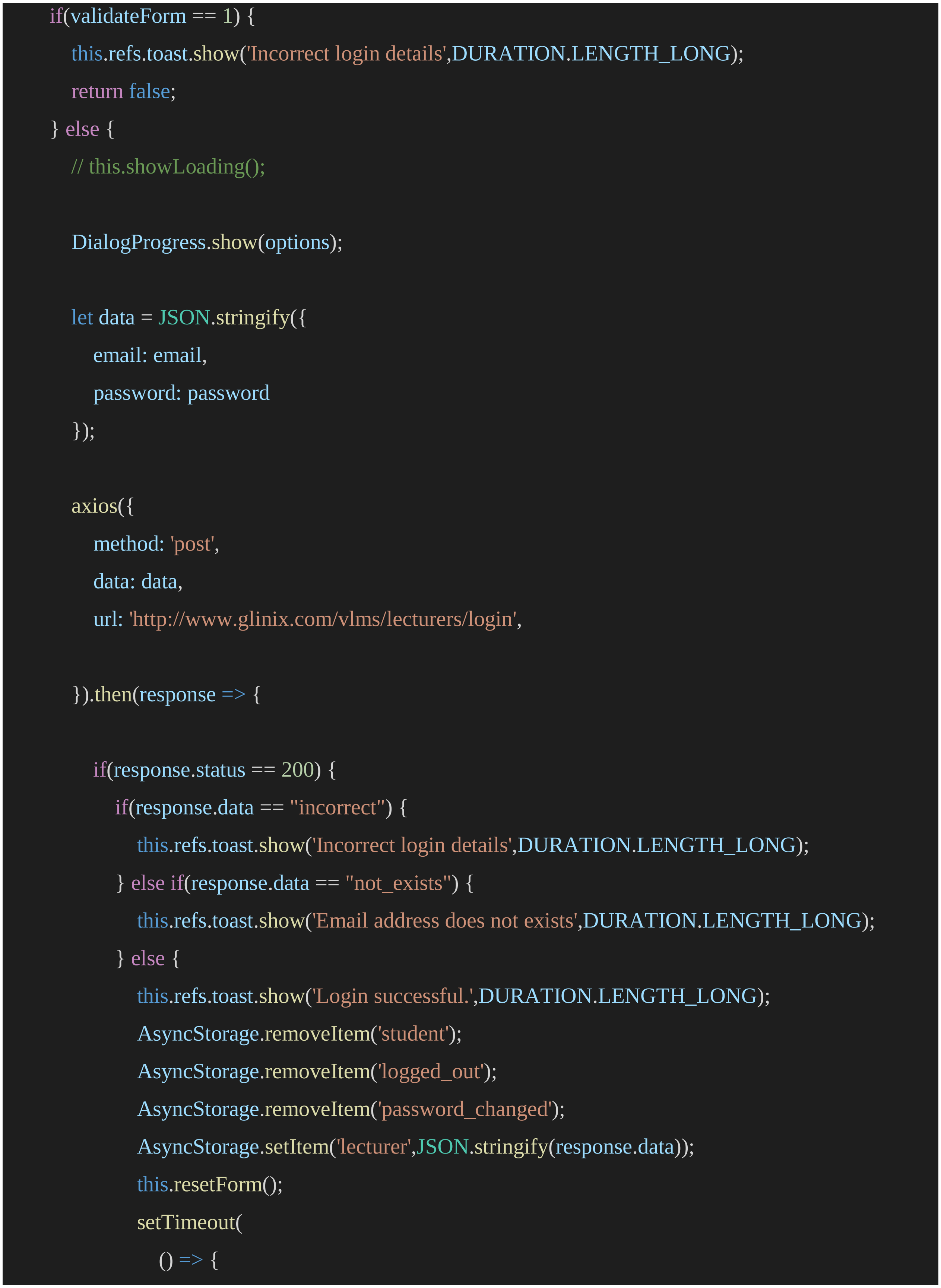
```
import React, { Component } from 'react';
import { Text, View, Image, ActivityIndicator,AsyncStorage } from 'react-native';
import { Container, Header, Content, Item, Input, Icon, Button, Picker } from 'native-base';
import {styles} from '../../css/Designs';
import axios from 'axios';
import Toast, {DURATION} from 'react-native-easy-toast';
import PasswordInputText from 'react-native-hide-show-password-input';
import DialogProgress from 'react-native-dialog-progress';

const options = {
  title:"",
  message:"Verifying Login Details..",
  isCancelable:true
}

export class LecturerLogin extends Component {

  static navigationOptions = {
     header: null
  }

  constructor(props) {
    super(props);
    this.state = {
      logged_out: '',
      password_changed: '',
      email: '',
      password: '',
      lecturer_id: '',
      showToast: false,
      loading: false,
```



```
        isLoggedIn: true,
        isPasswordChanged: true,
        isLoggedOut: true
    };
}

resetForm = () => {
    this.setState({
        email: '',
        password: ''
    });
}

validateForm = (email,password) => {

    if(email == "" || password == "") {
        return 1;
    } else {
        return 0;
    }
}

componentDidMount() {

    AsyncStorage
        .getItem('lecturer')
        .then( (value) => {

            this.setState({
                lecturer_id: value
            });

            if(value == null || value == "") {
                this.setState({
```



```
          isLoggedIn: false
        });
      } else {
        this.setState({
          isLoggedIn: true
        });
        this.props.navigation.navigate('LecturerDashboard');
      }

    }).done();

  // AsyncStorage.multiGet(['lecturer','logged_out','password_changed'])
  // .then(response => {
  //  the_lecturer_id = response[0][1];
  //  the_logged_out = response[1][1];
  //  the_password_changed = response[2][1];

  //  this.setState({
  //      lecturer_id: the_lecturer_id,
  //      password_changed: the_password_changed,
  //      logged_out: the_logged_out,
  //  });

  //  //console.error(this.state.lecturer_id);
  // });

}

loginLecturer = () => {
  let email = this.state.email;
  let password = this.state.password;

  let validateForm = this.validateForm(email,password);
```

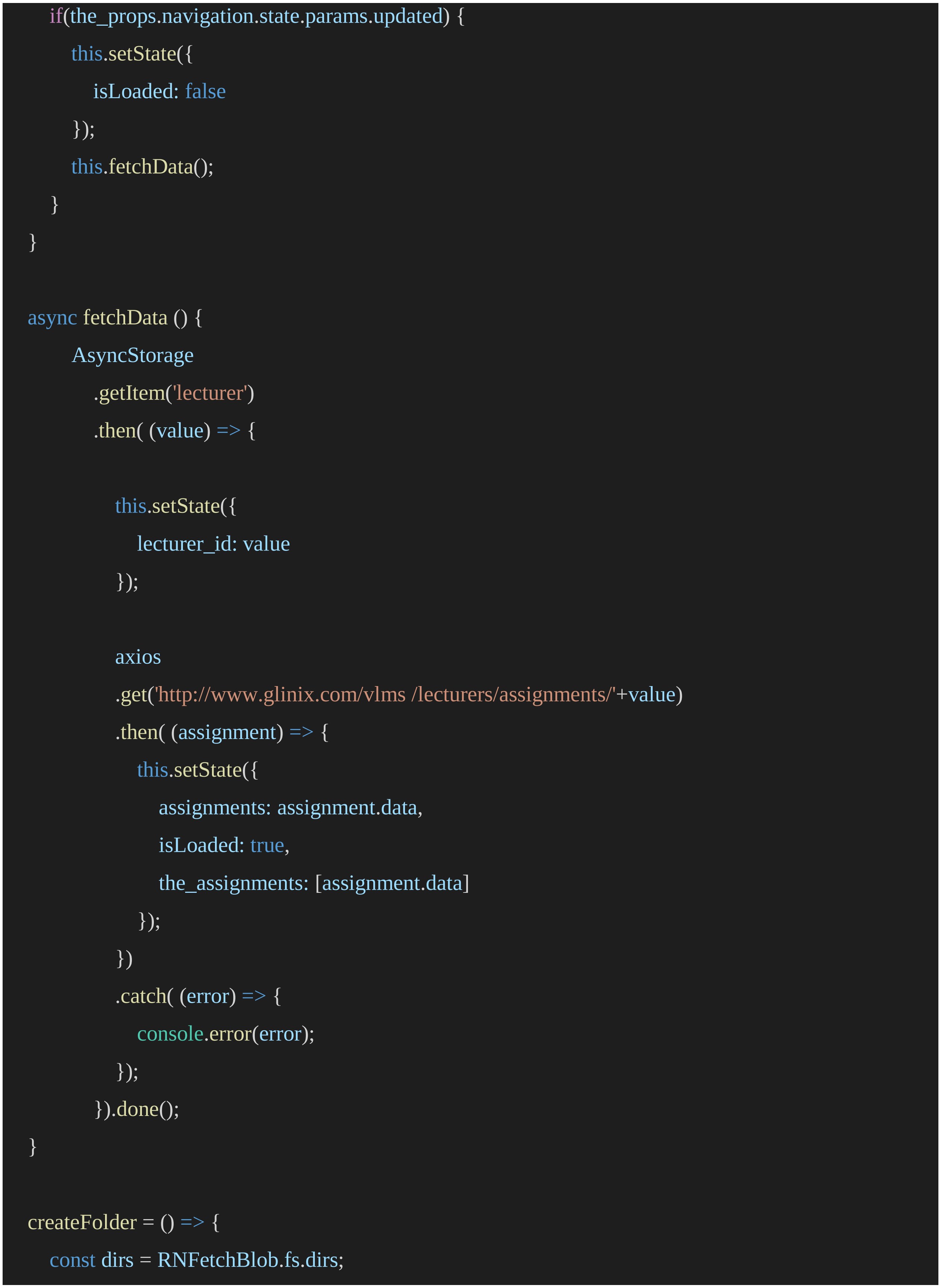


```
if(validateForm == 1) {
    this.refs.toast.show('Incorrect login details',DURATION.LENGTH_LONG);
    return false;
} else {
    // this.showLoading();

    DialogProgress.show(options);

    let data = JSON.stringify({
        email: email,
        password: password
    });

    axios({
        method: 'post',
        data: data,
        url: 'http://www.glinix.com/vlms/lecturers/login',

    }).then(response => {

        if(response.status == 200) {
            if(response.data == "incorrect") {
                this.refs.toast.show('Incorrect login details',DURATION.LENGTH_LONG);
            } else if(response.data == "not_exists") {
                this.refs.toast.show('Email address does not exists',DURATION.LENGTH_LONG);
            } else {
                this.refs.toast.show('Login successful.',DURATION.LENGTH_LONG);
                AsyncStorage.removeItem('student');
                AsyncStorage.removeItem('logged_out');
                AsyncStorage.removeItem('password_changed');
                AsyncStorage.setItem('lecturer',JSON.stringify(response.data));
                this.resetForm();
                setTimeout(
                    () => {
```

```
                this.props.navigation.navigate('LecturerDashboard');
              }
            , 100);
          }
          //this.hideLoading();
          DialogProgress.hide();
        } else {
          this.refs.toast.show('Invalid request. Try again.',DURATION.LENGTH_LONG);
          //this.hideLoading();
          DialogProgress.hide();
        }
      }).
      catch( (error) => {
        this.refs.toast.show('Invalid request. Try again.',DURATION.LENGTH_LONG);
        //this.hideLoading();
        DialogProgress.hide();
      });
  }
}

render() {

  let display_login = null;
  let password_changed_text = null;
  let logged_out_text = null;

  if(this.state.isPasswordChanged) {
    password_changed_text = "Your password was successfully changed. Kindly relogin below";
    AsyncStorage.removeItem('password_changed');
  }

  if(this.state.isLoggedOut) {
    logged_out_text = "You've been logged out.";
    AsyncStorage.removeItem('logged_out');
    }

    return(
      <Container>

        {!this.state.isLoggedIn ?

          <Content>

            <View>

              <View
                style={styles.the_container}>
                <Image
                source ={require('../../img/logo.png')}
                />
                <Text
                style={styles.loginText}>Lecturer Login
                </Text>

              </View>

              <View style={{marginTop:20}}>

                <Item style={styles.Inputs}>
                  {/* <Icon name='md-mail'/> */}
                  <Input placeholder='Email address' onChangeText = { (text) =>
this.setState({email: text}) } value={this.state.email}/>
                </Item>

                <View style={styles.Inputs}>
                  <PasswordInputText
                    value={this.state.password}
                    onChangeText={ (password) => this.setState({ password }) }
                  />
                </View>

                <Button full success style={styles.greenButtonWithoutBottom} onPress={() =>
this.loginLecturer()}>
                  <Text style={{color:'#fff',fontSize:18}}>Login <Icon name='log-in'/></Text>
                </Button>

                <Text style={{textAlign:'center', color:'#444',fontSize:16}} onPress={() =>
{this.props.navigation.navigate('LecturerForgotPassword')}}> Forgot Password? Click here </Text>

                <Toast
                  ref="toast"
                  style={{backgroundColor:'black'}}
                  position='bottom'
                  positionValue={500}
                  fadeInDuration={200}
                  fadeOutDuration={3000}
                  opacity={0.8}
                  textStyle={{color:'white'}}
                />
              </View>
            </View>
          </Content> : display_login
        }
      </Container>
    );
  }
}
```

ALL ASSIGNMENTS

```
import React, { Component } from 'react';
import {Text, View, ActivityIndicator, AsyncStorage, ScrollView} from 'react-native';
import { Container, Header,  Icon, Button, Left, Right, Body, Title, Fab} from 'native-base';
import {styles} from '../../../css/Designs';
import axios from 'axios';
import { TheAssignment } from './TheAssignment';
import RNFetchBlob from 'rn-fetch-blob';
import RNFS from 'react-native-fs';

const DocumentDir = RNFetchBlob.fs.dirs.DocumentDir;

export class AllAssignments extends Component {

    static navigationOptions = {
        header: null
    }

    constructor(props) {
      super(props);

        this.state = {
                assignments: [],
                the_assignments: [],
                lecturer_id: '',
                isLoaded: false
        };

        this.fetchData = this.fetchData.bind(this);
        this.createFolder = this.createFolder.bind(this);
    }

    componentDidMount() {
        this.fetchData();
    }

    componentWillReceiveProps(the_props) {
```



```
  if(the_props.navigation.state.params.updated) {
    this.setState({
      isLoaded: false
    });
    this.fetchData();
  }
}

async fetchData () {
    AsyncStorage
      .getItem('lecturer')
      .then( (value) => {

        this.setState({
          lecturer_id: value
        });

        axios
        .get('http://www.glinix.com/vlms /lecturers/assignments/'+value)
        .then( (assignment) => {
          this.setState({
            assignments: assignment.data,
            isLoaded: true,
            the_assignments: [assignment.data]
          });
        })
        .catch( (error) => {
          console.error(error);
        });
      }).done();
}

createFolder = () => {
  const dirs = RNFetchBlob.fs.dirs;
```

```
    console.error(dirs.DCIMDir);

    // RNFetchBlob.fs.mkdir(`${RNFetchBlob.fs.dirs.PictureDir}/vlms`)
    // .then( () => {
    //  console.error("done");
    // })
    // .catch( (error) => {
    //  console.error(error);
    // });
  }

  render() {

    let display_assignments = null;

    if(this.state.assignments.length == 0) {
      display_assignments = (
        <Text style={styles.theDangerText}> You have not uploaded any assignment </Text>
      )
    } else {
      display_assignments = this.state.assignments.map( (assignment,key) => {
        return <TheAssignment key={key} serial={key+1} id={assignment.id}
course_title={assignment.course_title} course_code={assignment.course_code} note={assignment.note}
deadline={assignment.deadline}  hour={assignment.hour} minutes={assignment.minutes}
meridian={assignment.meridian} path={assignment.path} size={assignment.size}
uploaded_on={assignment.uploaded_on}  navigation={this.props.navigation}/>
      });
    }

    return(

      <Container>

        <Header style={styles.tabColor}>
```

```
          <Left style={{flex:1}}>
            <Button transparent>
             <Icon name='menu'/>
            </Button>
          </Left>
          <Body style={{flex:1}}>
            <Title>My Assignments</Title>
          </Body>
          <Right style={{flex:1}}>
            <Button transparent>
             <Icon name='search' />
            </Button>
          </Right>
        </Header>

        <ScrollView>
          {!this.state.isLoaded ?
            <View style={styles.theLoading}>
              <ActivityIndicator size="large" />
            </View> : display_assignments
          }
        </ScrollView>

        {/* <Button full success style={styles.greenButtonWithTopBottom} onPress={() =>
this.createFolder()}>
          <Text style={{color:'#fff',fontSize:18}}>Create folder </Text>
        </Button> */}

        <Fab
          direction="up"
          style={{ backgroundColor: '#075E54' }}
          position="bottomRight"
          onPress={() => this.props.navigation.navigate('AddAssignment')}>
          <Icon name="md-cloud-upload"/>
```

```
            </Fab>


          </Container>
        );
    }
}
```

ALL MATERIALS

```
import React, { Component } from 'react';
import {Text, View, ActivityIndicator, AsyncStorage,ScrollView, RefreshControl} from 'react-native';
import { Container, Header,  Icon, Button, Left, Right, Body, Title, Fab} from 'native-base';
import {styles} from '../../../css/Designs';
import axios from 'axios';
import { TheMaterial } from './TheMaterial';

export class AllMaterials extends Component {

    static navigationOptions = {
        header: null
    }

    constructor(props) {
      super(props);

        this.state = {
            materials: [],
            the_materials: [],
            lecturer_id: '',
            isLoaded: false,
            refreshing:false,
        };

        this.fetchData = this.fetchData.bind(this);
```

```
}

componentDidMount() {
    this.fetchData();
}

componentWillReceiveProps(the_props) {
    if(the_props.navigation.state.params.updated) {
        this.setState({
            isLoaded: false
        });
        this.fetchData();
    }
}

_onRefresh = () => {
    this.setState({
        isLoaded: false,
        refreshing:true
    });
    this.fetchData();
}

fetchData = () => {
        AsyncStorage
            .getItem('lecturer')
            .then( (value) => {

                this.setState({
                    lecturer_id: value
                });

                axios
                .get('http://www.glinix.com/vlms /lecturers/materials/'+value)
```

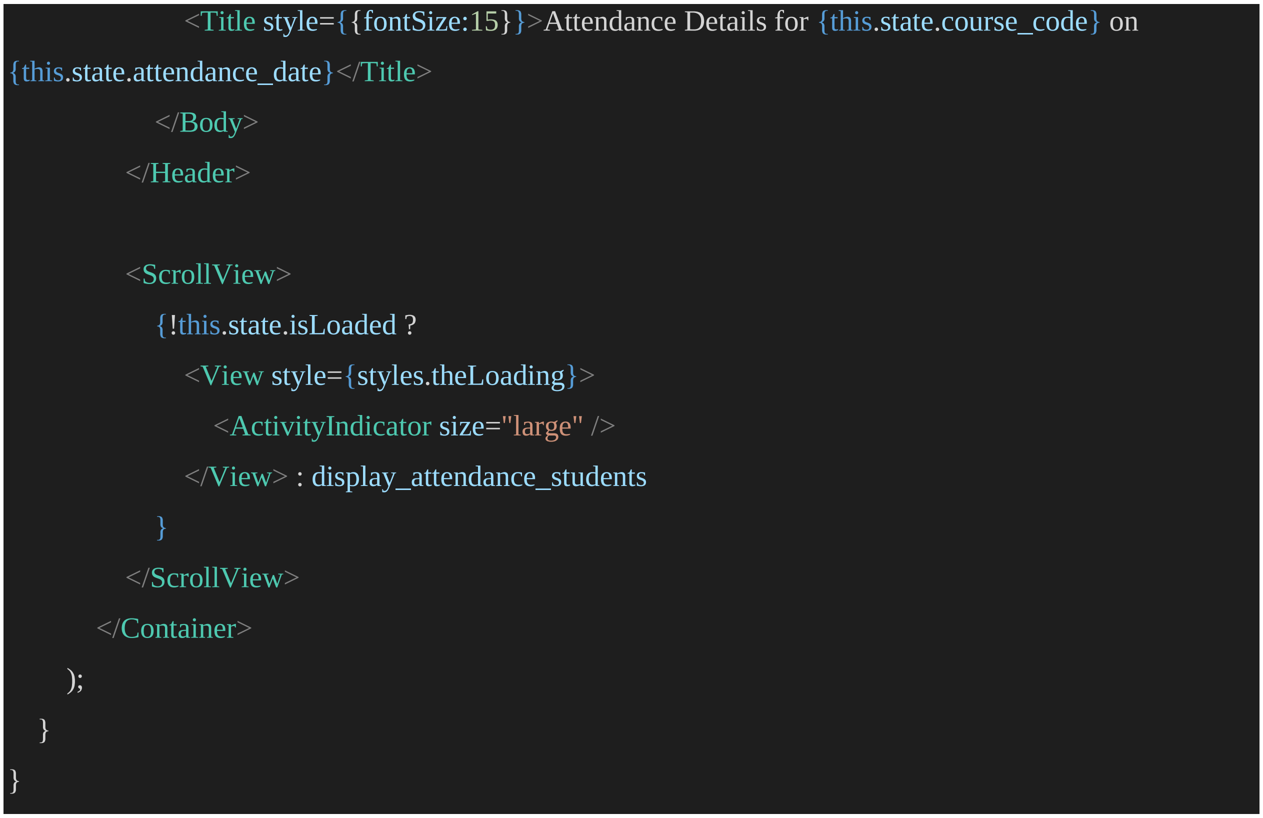
```
                    .then( (material) => {
                        this.setState({
                            materials: material.data,
                            isLoaded: true,
                            the_materials: [material.data]
                        });
                    })
                    .catch( (error) => {
                        console.error(error);
                    });
                }).done();
    }

    render() {

        let display_materials = null;

        if(this.state.materials.length == 0) {
            display_materials = (
                <Text style={styles.theDangerText}> You have not uploaded any material </Text>
            )
        } else {
            display_materials = this.state.materials.map( (material,key) => {
                return <TheMaterial key={key} serial={key+1} id={material.id}
course_title={material.course_title} course_code={material.course_code} note={material.note}
path={material.path} size={material.size} uploaded_on={material.uploaded_on}
navigation={this.props.navigation}/>
            });
        }

        return(

            <Container>
```

```
<Header style={styles.tabColor}>
  <Left style={{flex:1}}>
    <Button transparent>
      <Icon name='menu'/>
    </Button>
  </Left>
  <Body style={{flex:1}}>
    <Title>My Materials</Title>
  </Body>
  <Right style={{flex:1}}>
    <Button transparent>
      <Icon name='search' />
    </Button>
  </Right>
</Header>

<ScrollView
  refreshControl={
    <RefreshControl
      refreshing={this.state.refreshing}
      onRefresh={this._onRefresh}
    />
  } >
  {!this.state.isLoaded ?
    <View style={styles.theLoading}>
      <ActivityIndicator size="large" />
    </View> : display_materials
  }
</ScrollView>

<Fab
  active="true"
  direction="up"
  style={{ backgroundColor: '#075E54' }}
```

```
                position="bottomRight"
                onPress={() => this.props.navigation.navigate('AddMaterial')}>
                <Icon name="md-cloud-upload"/>
            </Fab>

        </Container>
      );
    }
}
```

STUDENT ATTENDANCE

```
import React, { Component } from 'react';
import {Text, View, ActivityIndicator, AsyncStorage, ScrollView } from 'react-native';
import { Container, Header,  Icon, Button, Left, Right, Body, Title, Fab} from 'native-base';
import {styles} from '../../../css/Designs';
import axios from 'axios';
import { TheAttendanceStudent } from './TheAttendanceStudent';

export class ListAttendanceStudents extends Component {

  static navigationOptions = {
    header: null
  }

  constructor(props) {
    super(props);

    this.state = {
      lecturer_id: '',
      attendance_id: props.navigation.state.params.attendance_id,
      attendance_date: props.navigation.state.params.attendance_date,
      attendance_time: props.navigation.state.params.attendance_time,
      course_code: props.navigation.state.params.course_code,
      attendance_students: [],
      isLoaded: false
    };
  }

  componentDidMount() {
    AsyncStorage
      .getItem('lecturer')
      .then( (value) => {

        this.setState({
          lecturer_id: value
        });

        this.fetchData();
      }).done();
  }

  componentWillReceiveProps(the_props) {
    if(the_props.navigation.state.params.updated) {
      this.setState({
        isLoaded: false
      });
      this.fetchData();
    }
  }

  fetchData = () => {
    axios
    .get("http://www.glinix.com/vlms /lecturers/attendance-students/'+this.state.attendance_id).
    then(response => {

      this.setState({
        attendance_students: response.data,
        isLoaded:true
      });
    }).
    catch( (error) => {

    });
  }



  render() {

    let display_attendance_students = null;

    if(this.state.attendance_students.length == 0) {
      display_attendance_students = (
        <Text style={styles.theDangerText}> No attendance details yet</Text>
      )
    } else {
      display_attendance_students = this.state.attendance_students.map( (student,key) => {
        return <TheAttendanceStudent key={key} id={student.id} serial={key+1}
name={student.student_name} matric={student.matric} level={student.level} time={student.time}
count_taken={this.state.attendance_students.length} count_total={student.count_total}
navigation={this.props.navigation}/>
      });
    }

    return(

      <Container>

        <Header style={styles.tabColor}>
          <Body style={{flex:1}}>
```



```
                        <Title style={{fontSize:15}}>Attendance Details for {this.state.course_code} on
{this.state.attendance_date}</Title>
                    </Body>
                </Header>

                <ScrollView>
                    {!this.state.isLoaded ?
                        <View style={styles.theLoading}>
                            <ActivityIndicator size="large" />
                        </View> : display_attendance_students
                    }
                </ScrollView>
            </Container>
        );
    }
}
```