

Mobile Attendance System Project Documentation

Farrukh Jahangeer

2960928

**Submitted in partial fulfillment for the
degree of**

BSCH in Computing Science

**Griffith College Dublin
May, 2022**

Under the supervision of

Osama Abushama

Disclaimer

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the Degree of Bachelor of Science in Computing at Griffith College Dublin, is entirely my own work and has not been submitted for assessment for an academic purpose at this or any other academic institution other than in partial fulfilment of the requirements of that stated above.

Signed: Farrukh Jahangeer

Date: June 2022

Acknowledgements

I'd like to thank Almighty God for blessing me good health that were necessary to get through my degree and the following people, without whom I would not have been able to complete my bachelor degree from Griffith College Dublin.

First and foremost, I am extremely grateful to the computing faculty at Griffith College Dublin, especially Mr Osama Abushama, for his assistance at every stage of the final year project. And a special thanks to our programme director Mr Barry Denby, whose support enabled me to achieve the milestone.

Secondly, my heartfelt gratitude goes to my parents and family for their countless support, prayers and best wishes throughout my four-year studies in Ireland.

Finally, My best friends in Griffith College particularly Ms Khusmada who have supported me and endured my concerns and complains over the past four years of study!

Table of Contents

ACKNOWLEDGEMENTS	2
LIST OF TABLES	5
ABSTRACT	6
CHAPTER 1. INTRODUCTION	
PROPOSAL	7
MANUAL ATTENDANCE PROBLEMS	7
GOALS	7
TECHNOLOGIES	8
CHAPTER 2. BACKGROUND	9
INTRODUCTION	9
<i>Advantage of Attendance Application</i>	9
DISADVANTAGE OF ATTENDANCE APPLICATION	10
THE DISADVANTAGES OF MOBILE ATTENDANCE SYSTEM ARE:	10
EXISTING ATTENDANCE SYSTEM	10
CHAPTER 3. DESIGN	11
APPLICATION DESIGN	11
REALTIME DATABASE DESIGN	12
	12
	13
	13
	14
	14
COURSES AND SUBJECT	14
TOP LEVEL ARCHITECTURE DESIGN	15
PROJECT PLAN AND GANTT CHART	16
USE CASE DIAGRAM	17
4. IMPLEMENTATION	18
TECHNOLOGIES	18
<i>Android Application</i>	18
<i>Kotlin</i>	18
<i>Firebase Storage</i>	18
REQUIREMENTS	19
<i>Main Dashboard</i>	20
LOGIN AND REGISTER FLOWCHART	20
ADMIN FLOW DIAGRAM	22
FIGURE 4.1	22
ADMIN	23
<i>Register Student</i>	23
<i>View Student</i>	24
<i>Register Lecturer</i>	24
<i>View Lecturer</i>	25
<i>Delete Lecturer</i>	26
<i>Figure 4.24 Figure 4.25</i>	26
<i>Delete Student</i>	26
	26
<i>Figure 4.26 Figure 4.27</i>	26
<i>Authenticate New Student</i>	27
FIGURE 4.28 FIGURE 4.29	27
STUDENT FLOW DIAGRAM	27
STUDENT	28

<i>View My Attendance</i>	29
<i>Mark My Attendance</i>	29
LECTURER FLOW DIAGRAM	30
LECTURER	31
<i>Figure 4.37</i>	32
<i>View Student</i>	32
<i>Figure 4.38 Figure 4.39 Figure 4.40</i>	33
<i>Manage Attendance</i>	33
<i>Take Attendance</i>	34
<i>Figure 4.42 Figure 4.43</i>	34
<i>Manually Mark Attendance</i>	34
<i>Download Attendance</i>	35
CHAPTER 5. TESTING	36
ADDING USERNAME	36
DATE OF BIRTH	37
REGISTER STUDENT	37
REGISTER LECTURER	37
VIEW STUDENT	38
FIGURE 5.5 FIGURE 5.6	38
VIEW LECTURER	38
<i>Active Attendance</i>	39
	39
MARK ATTENDANCE	39
DOWNLOAD ATTENDANCE	40
	40
VIEW ATTENDANCE PERCENTAGE	40
APPROVAL FROM ADMIN	41
FIGURE 5.13	41
DELETING STUDENT	ERROR! BOOKMARK NOT DEFINED.
DELETING LECTURER	41
<i>Figure 5.17 Figure 5.18</i>	41
VALUES USED FOR TESTING	42
<i>Student</i>	42
<i>Lecturer</i>	42
CHAPTER 6. EVALUATION	43
LIST OF PROPOSED TASKS	43
CHAPTER 7. CONCLUSIONS AND FUTURE WORK	45
CONCLUSION	45
FUTURE WORK	45
REFERENCES	46

List Of Tables

<i>Table 1</i>	42
<i>Table 2</i>	42
<i>Table 3</i>	42
<i>Table 4</i>	43

Abstract

In this fast-developing world of technology, mobile phone is one of the devices that have been integrated in our daily life. Mobile phone has software and applications which can be accessible from anywhere and anytime. One of the applications that is being used nowadays is the mobile attendance system. This application has been taken up nearly everywhere, in schools and offices. It is great to monitor attendance of student and employee. The mobile attendance system has various advantages for example it is user-friendly, it can save time, it reduces amount of paperwork and administration work. The application is an android attendance system application. The application is designed to enable student to mark themselves attended during a lecture. Students can also see their attendance percentage. The lecturer uses the application to start the attendance timer during lecture. Lecturer can also download attendance and mark student as present manually. The faculty uses the application add new users, to manage attendance and download attendance.

Chapter 1. Introduction

Proposal

The Project is an android application, that enables student to mark themselves as present during a lecture. During the lecture, the lecturer enables the attendance system by a click of a button for a particular time on the application. The students are then to mark themselves as present only during this time. Once the lecturer, closes the enable attendance, students' attendance will not be saved.

- The lecturer can login, mark a student's attendance, see how many students are present and save the attendance if there are any changes. Moreover, the lecturer can see the modules he is teaching, see attendance list, download attendance list, and reset his password.
- The student can login, Student can register himself he can log in after admin will approve registration request. Student can mark himself as attended only when attendance is enabled. The student Can see the percentages of attendance for each module, see which modules he is currently taking, see his profile details and reset his password.
- The admin/ Faculty can login, register a student for a course, add a member of faculty, add a lecturer, manage attendance for student.

Manual Attendance Problems

The recent pandemic has shown us how important it is to have a hygienic lifestyle and taking attendance during lecture is a hassle and time consuming. Passing the attendance sheet around to take attendance is a risk to spread infection as well as it is difficult to know whether the student really attended, or someone signed them as present. Some more disadvantage of the manual attendance system is:

- Risk of human error
- Time-consuming
- More paperwork
- Inefficient and outdated in current technological era

Goals

This application sorts out these issues. As students will be using their own devices to mark themselves as present. During the lecture time, the lecturer can enable attendance anytime for a short period of time and will inform students to mark themselves as attended before the time window closes. For example, the lecturer can

leave the attendance open for student to mark themselves in for Five minites after that student cannot mark themselves as attended

The faculty and lecturer can access the attendance in real-time and does not need to check and count attendance from attendance sheet which saves their time. Moreover, it reduces the amount of paper resources needed.

Technologies

Looking at the proposed solution, Android SDK is under consideration for the development of the application. The language being considered is Kotlin language. After a lot of research, the considered storage is Firebase.

Chapter 2. Background

Introduction

Attendance monitoring is very vital for academic establishments. Manual structures involve use of paper sheets to take the attendance. Usually, the teacher or instructor calls out student names and facts their presence or an attendance sheet is passed around the room so student can mark themselves. In lots of establishments, attendance is still taken on paper. Taking attendance on paper is very time-consuming.

During the pandemic, most of the classes was moved online therefore needed easier way to make classes and attendance. Lots of companies, developed their own application and software for managing attendance and classes. It has become very trendy as most colleges and universities have started using it. Colleges and Universities still uses this application despite everything is going back to normal. The use of online attendance system will remain forever even covid 19 has ended.

Advantage of Attendance Application

The advantages of mobile attendance systems are:

❖ Quick and Responsive

The application is much quicker than manual system. As when attendance is being taken on paper, a staff member must go through all the attendance sheet for each lecture to count attendance. In the application, all the information can be retrieved within seconds.

❖ Easy to use

Nowadays, the use of mobile application has become so common that everyone can easier navigate through any application without prior knowledge. Mobile applications are designed to be user-friendly thus easy to use and navigate.

❖ Easily manage time and attendance during lecture

During manual attendance, passing around the attendance sheet is time consuming and most likely cause distractions as student tends to keep looking where the attendance sheet is. Using mobile attendance, attendance can be completed just by a click of a button.

❖ Save administration time

Administration time is consumed a lot to go through all attendance sheet and counting attendance for each student for every lecture. The application can save a lot of this time as all reports can be generated within minutes.

❖ More accurate calculation of percentage

Since all calculations are performed by the computer, it is less prone to calculation error while counting attendance. It is also less prone to human error.

❖ Scalable

Since all data is being stored on the cloud, it is easily scaled up and down depending on the number of users.

Disadvantage of Attendance Application

The disadvantages of mobile attendance system are:

❖ Equipment expense

Organizations has to spend on devices, internet and software usage.

❖ Maintenance and repair

Since it is a software, it needs to be maintained and repair regularly.

❖ Internet

Since it is real-time data being stored and accessed, it requires the internet. Without the internet an attendance would not be able to be monitored.

Existing Attendance System

Some examples of the existing attendance systems are:

❖ Griffith College Attendance System

Griffith college uses, mobile attendance monitoring system to mark a student present using the beacon in each class. The Bluetooth beacon detects the proximity of the student with the beacon and mark the student as present during the time of the lecturer.

❖ Mitrefinch Time and Attendance System

This system manages the student attendance as well as reducing the staff workload. This system has multiple option for the taking attendance, such as Facial Recognition ID, Contactless proximity clocking terminals, Fingerprint, GPS tracking and many more.

Chapter 3. Design

Application Design

Icon

Figure 3.1 shows the icon used for the application. This icon is also used on main page when the user enters the application. This icon is a sky-blue colour with student in white. The icon has a friendly and good impression of application.



Figure 3.1

Colours

The colour used in this application are

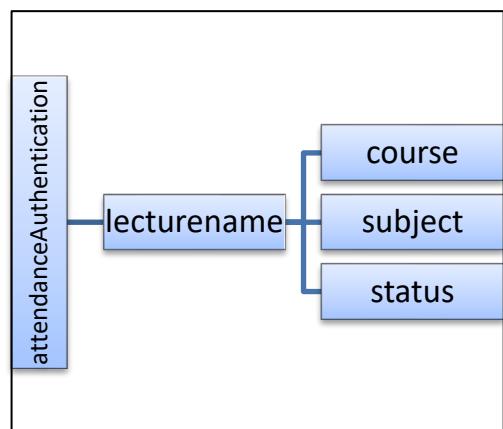
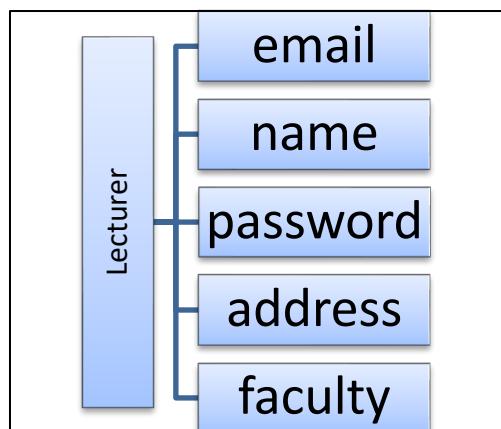
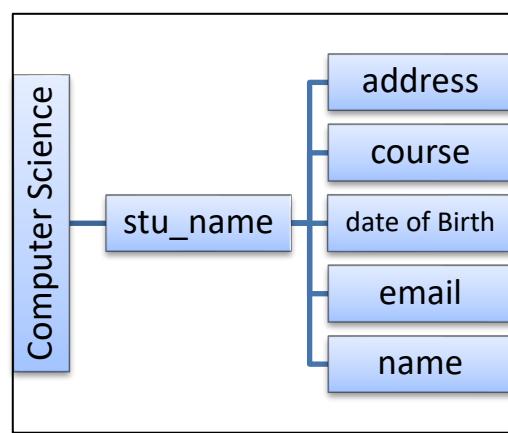
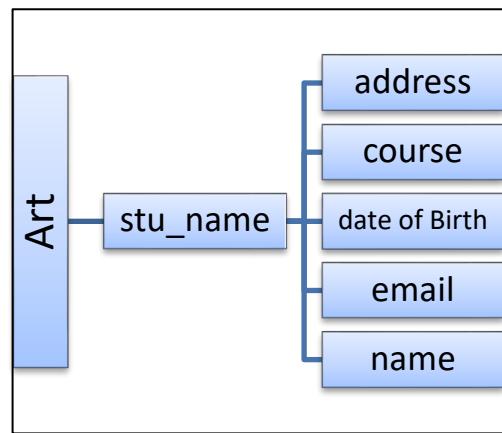
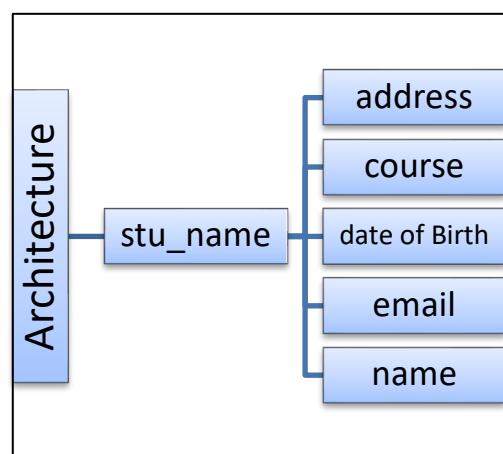
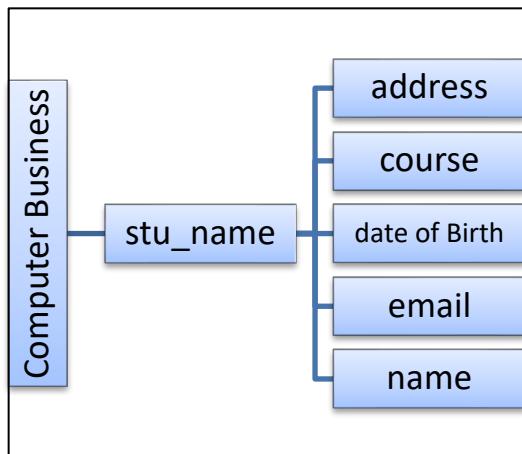
- ❖ white for backgrounds
- ❖ button on dashboard is sky-blue
- ❖ Main login, logout and submit button are white with blue borders.

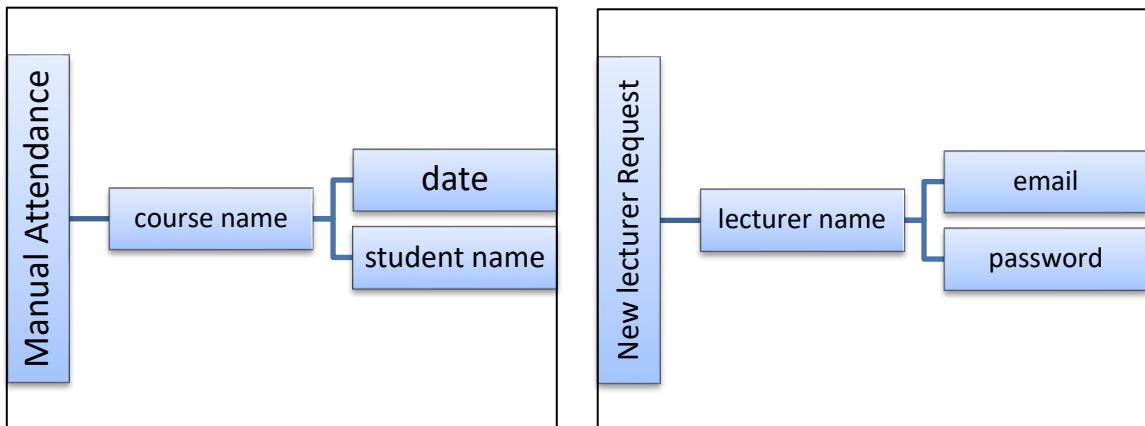
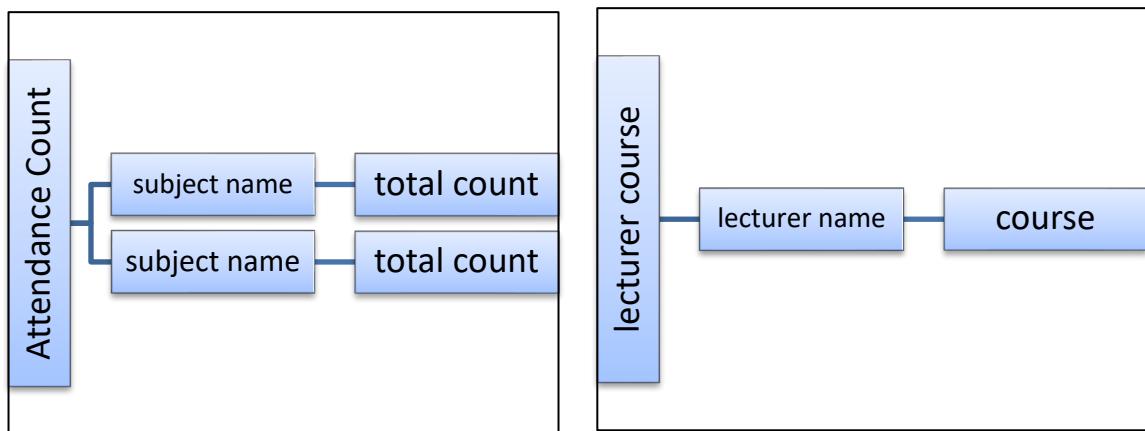
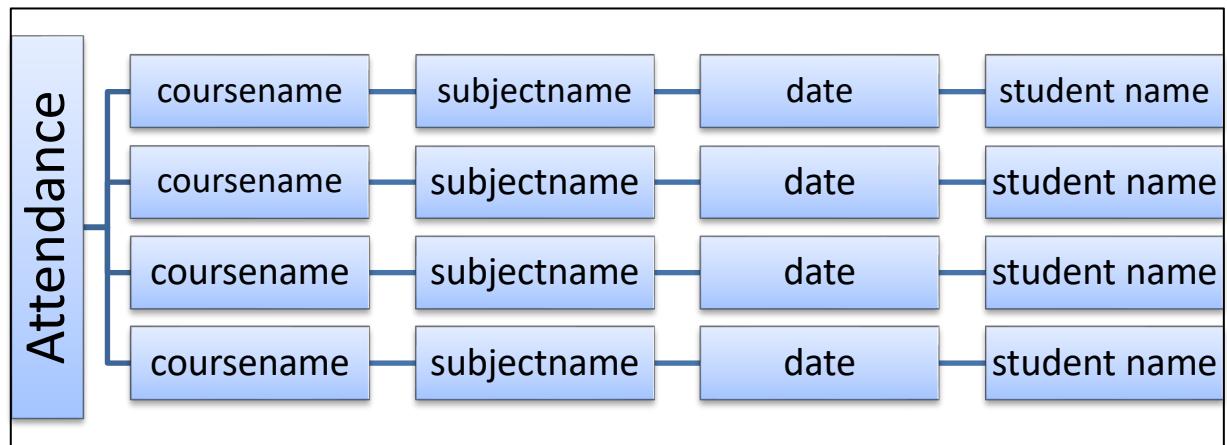
These colours used does not hurt eyes and is suitable for people with colour blindless.

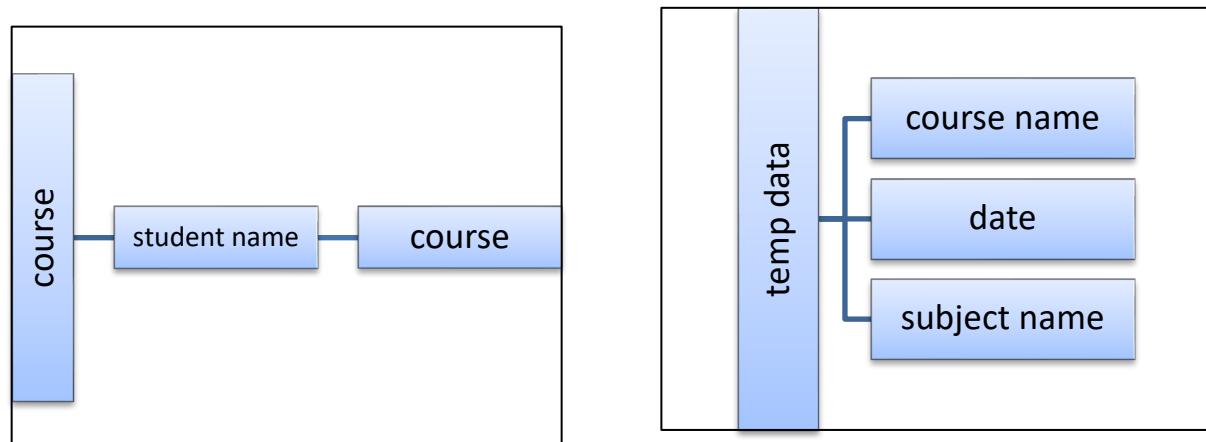
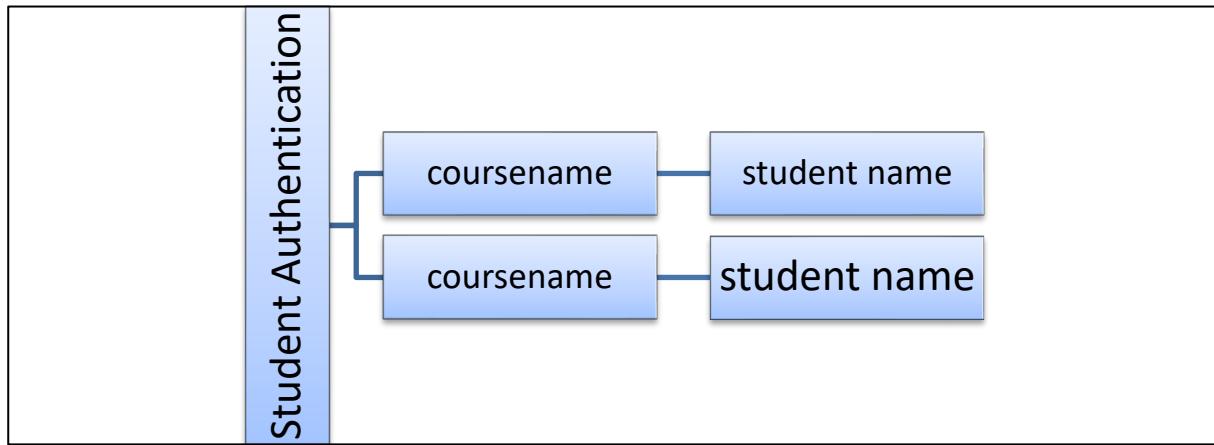
The buttons are also a suitable size, not too big, not too small to enable users to use the application without problem.

The application is design to enable three types of users, admin, lecturer, and student. This application helps the admin to register Student, View Student, Register Lecturer, View Lecturer, Delete Lecturer and Authenticate New Student. The student can mark themselves as present, view attendance for each subject, view overall attendance and mark attendance when attendance is open. The lecturer can view student, manage attendance by taking attendance, manually mark attendance and download attendance.

Realtime Database design







Courses and Subject

Courses

The courses used in this application are:

- ❖ Computer Science
- ❖ Computing Business
- ❖ Art
- ❖ Architecture

These courses are stored in the strings.xml file as a string array called course.

Subjects

The subjects used in this application are:

- ❖ For computing Science
 - Database
 - C Programming
 - OMT

- Microprocessors
- ❖ Architecture
 - Working Drawing
 - Climatology
 - Building Services
 - Theory of structures
- ❖ Art
 - Geography
 - Political Science
 - Sociology
 - Philosophy
- ❖ Computing Business
 - Accounting
 - Human Resource Management
 - Business Law
 - Marketing and Finance

Top Level Architecture Design

The Figure 3.2 shows the top-level architecture of the android application. The user's student access the android attendance application through an android mobile device. To access the data from the cloud the mobile device is connected to the internet through which the cloud and data storage can be accessed in real time. To access the data from cloud the android application has plugin which are connected to firebase cloud so when the user clicks buttons on application if communicate directly to cloud

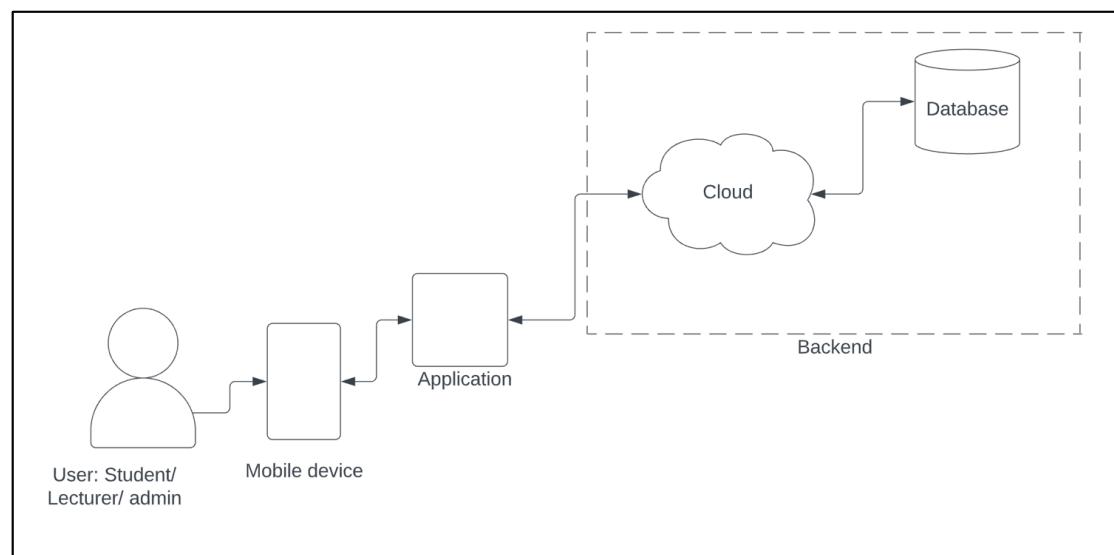


Figure 3.2

Project Plan and Gantt chart

Figure 3.3 shows the timeline that was used to build this application. The application has been developed over four months. The starting date was 31st of January and ending date was 30th May. As you can see the way I started this project I did planning I started research made documentation did meeting supervisor made changes I have mentioned in the table below with date.

Task Name	Duration/ days	Start	Finish
Start Project	2	01/31/22	
Data Analysis and Research	14	02/02/22	02/15/22
Research on application software	5	02/14/22	02/18/22
Design Architecture and UI	20	02/18/22	03/09/22
Firebase Connection, login System	14	03/10/22	03/23/22
Admin Section	9	03/24/22	04/01/22
Lecturer Section	25	04/01/22	04/25/22
Student Section	14	04/26/22	05/09/22
Testing and Bug Fixing	7	05/10/22	05/16/22
Testing	4	05/17/22	05/20/22
Documentation	12	05/19/22	05/30/22

Figure 3.3

Figure 3.4 shows the Gantt chart in terms of days taken for each task. In the diagram below it can be seen how long each task took to be completed the project pipeline is given below.

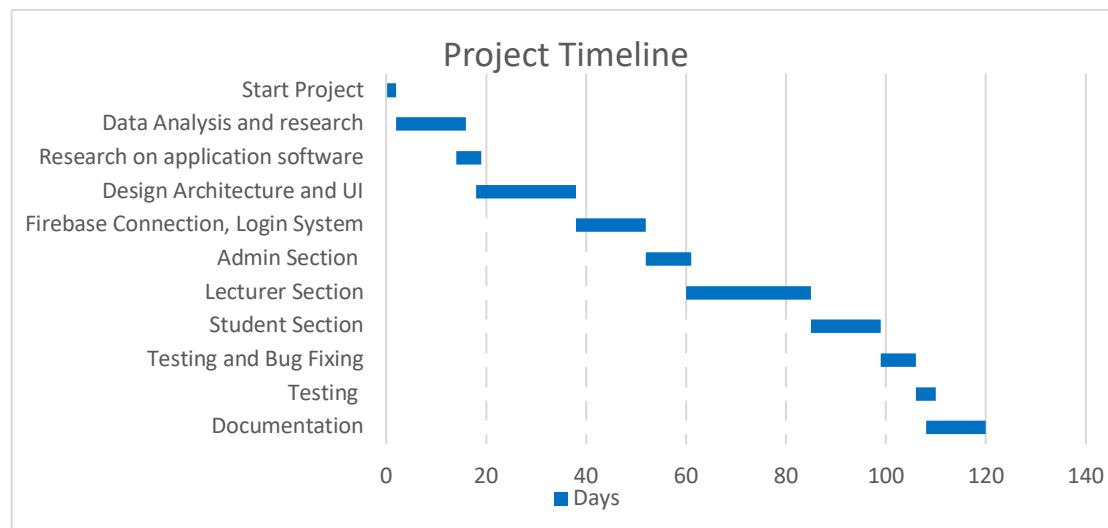


Figure 3.4

Use Case Diagram

Figure 3.5 shows a Use Case Diagram. The figure shows the role of the Admin, Student and Lecturer. All the functions available in each user app are displayed in the figure below. The admin has access to more features of the application as it controls most of the activities. The lecturer has the ability to open attendance session and mark student as present. The main activity of the student is to mark himself/herself as present when the session is active.

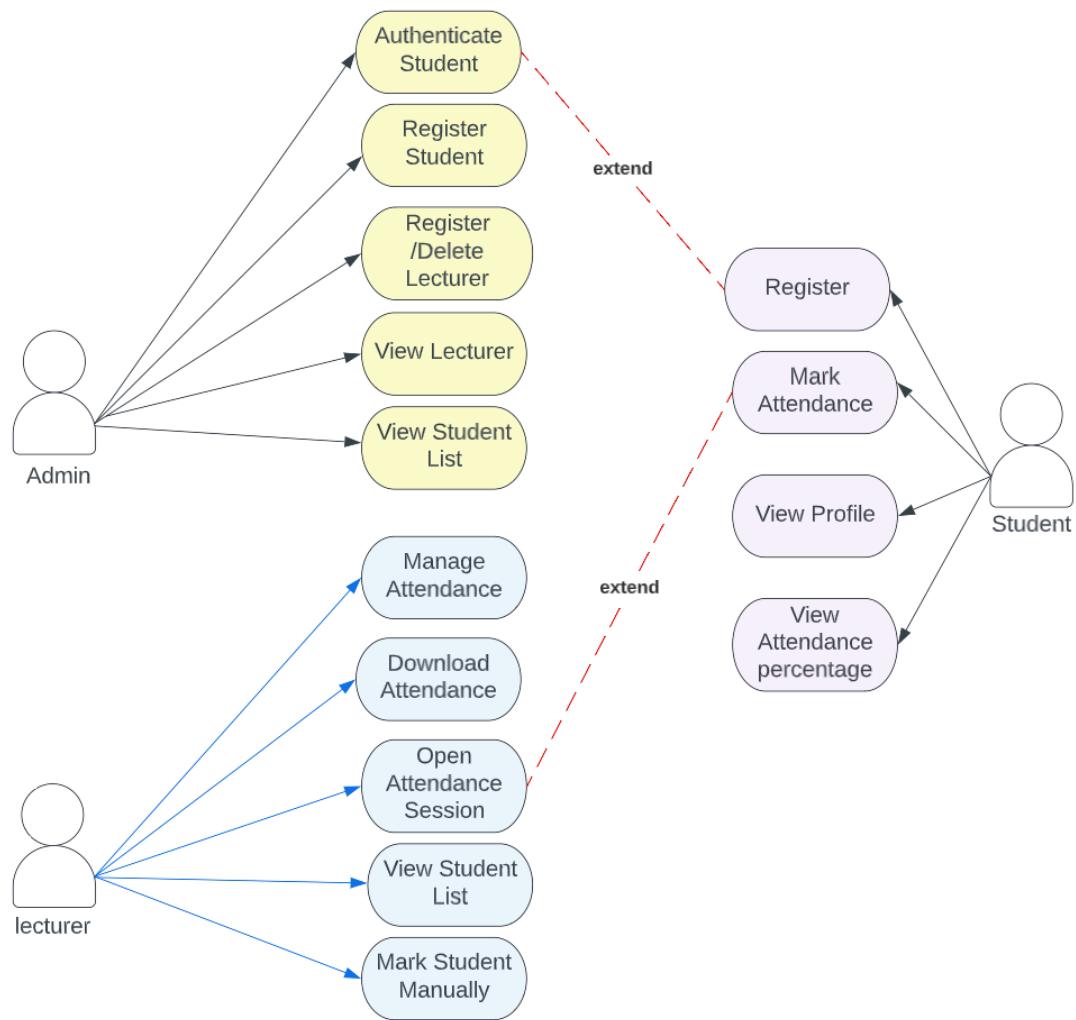


Figure 3.5

4. Implementation

Technologies

The technologies used are as follows:

Android Application

The attendance application is designed to be used in an android device. An android device is more suited for the attendance purpose as it is more user friendly. Android application development has been on the ascent for some time now. This is because of the way that Android has taken over as one of the most well-known operating system, therefore making it more suitable to make an android application.

The Android Studio is used to build the application. Android Studio provides an incorporated environment apps can be built for Android phones, tablets, and devices. Structured code modules allow project to be divided into units of functionality that can be independently build, test, and debug.

Using *Android SDK* for building frontend will be a complex to work with but as I have researched, it would good to develop using this because of following reasons:

1. Write the components once and used it on different platform.
2. Usability: As in modern world of technology, smart phones are with people all the time. So, application with usability of public scale would be beneficial.
3. Each component can have its own event handlers and functionality which can be integrated/ combined with other components to make a whole web page.
4. It creates virtual DOM which makes the working much faster than normal DOMs.
5. Code reusability is better than standard approach of building websites.

Kotlin

The language used is Kotlin as it is more convenient for coding compared to Java. Kotlin has a more streamlined and efficient design than Java, so it will perform better in situations where performance matter, especially when scaling is concerned. Kotlin supports multiple platforms and libraries which are more beneficial in the development of this application.

Firebase Storage

Firebase Cloud Storage is a service that can be used to store and download files generated directly by users. Firebase is a powerful, simple, and cost-effective object storage service built for Google scale. The storage used is Firebase Realtime Database. This allows the application to scale as well as have data to be updated in real-time.

NoSQL

NoSQL is more scalable and provide more performance. Furthermore, in comparison to the relational model, the flexibility and ease of use of NoSQL data models can speed up development, especially in the cloud computing environment. It can easily accommodate changes for future upgrades and no need to depend on SQL operation.

Requirements

Hardware Requirements

These are the physical components that the system requires to operate:

- ❖ Android Mobile device
- ❖ Smart devices
- ❖ Internet connection

Software Requirement

These are the logical component that the system requires to operate:

- ❖ Application
- ❖ Android Operating System

Other Technologies Considered

While in process of research and consultation with *Project supervisor* several technologies were considered, which included *Room Database*, *Knuth Server* and *MySQL*. MySQL doesn't fit scope of application with respect to scalability and platform. However, during the Knuth Server was down for several weeks and was delaying the progress in my work. I went to Griffith college IT department after 2 weeks of wait they said server is down and we are working on it but we don't know how long will it take to be active so While, other technologies have associated limitations with them. Upon consultation with supervisor we concluded Firebase would be the next preferred for the system.

Main Dashboard

Figure 4.1 shows the main dashboard of application. It is also the page that appears when the application is opened. It contains a picture of logo of the application and two buttons to Register or Login.



Figure 4.1

Login and Register Flowchart

Figure 4.2 shows the flow of activities for registration and for login when a user enters the application. When a user enters the system, if the user already has a login, then the user will select the Login Button. If the user needs to register itself, then he will select the Register button.

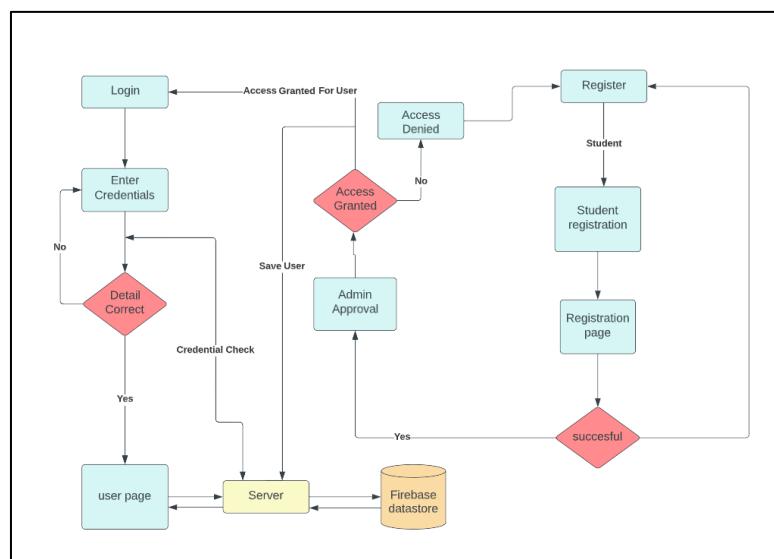


Figure 4.2

Case 1: Login

Login page has three level for Admin, Lecturer and Student. Figure 4.3 shows the choice of login for the users and reset password button. Each must enter their credentials to be able to login. When credentials are entered, the credentials are checked in the firebase datastore if it is an existing user. If the credentials do not match any user, an output is shown on screen “No Login Found”. If the credentials match an actual user, then the user page is output on screen. Most of the data coming to the user page comes from the server.

- ❖ Figure 4.4 shows the student login page. When a student logs in successfully, the student is taken to the student dashboard.
- ❖ Figure 4.5 shows the Lecturer login page. When a Lecturer logs in successfully, the lecturer is given access to the lecturer dashboard.
- ❖ Figure 4.6 shows the admin login page. When admin logs in successfully, the admin is given full access the admin dashboard where several important tasks take place.

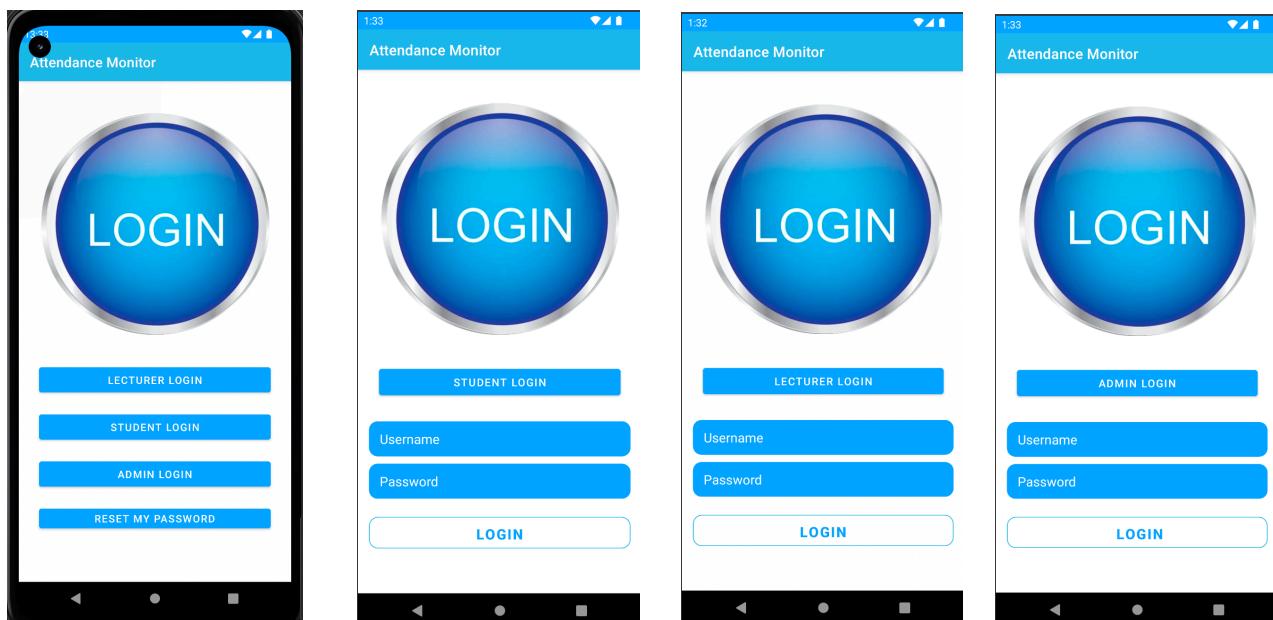


Figure 4.3

Figure 4.4

Figure 4.5

Figure 4.6

Case 2: Register

When a new user comes to use the application, the user must choose the button register and click for student registration, as shown in Figure 4.7. After clicking on the specific category of user, the registration form activity appears on the screen as shown in Figure 4.8. After successfully entering all details, the details are saved, and information is sent to admin for approval as shown in Figure 4.9. When the admin receives the request, the request can be accepted if the user its truly part of the college. After a request is accepted the user can use the credential entered during registration for login.



Figure 4.7

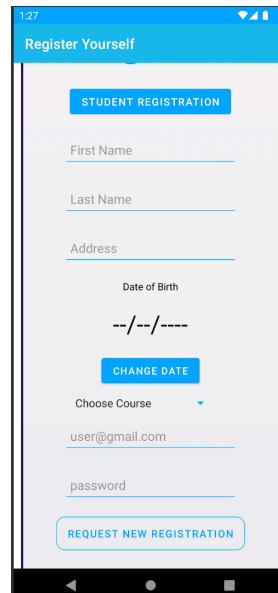


Figure 4.8

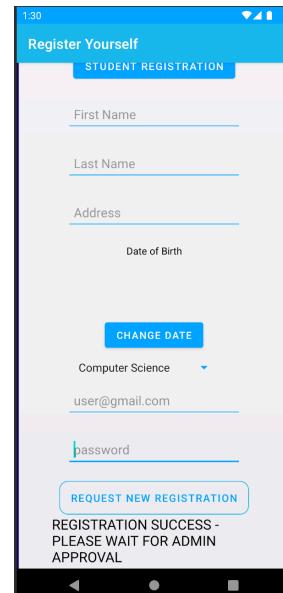


Figure 4.9

Admin Flow diagram

Figure 4.10 shows the admin flow diagram which shows the activities the admin can perform. The processes and decision of each activity is also demonstrated.

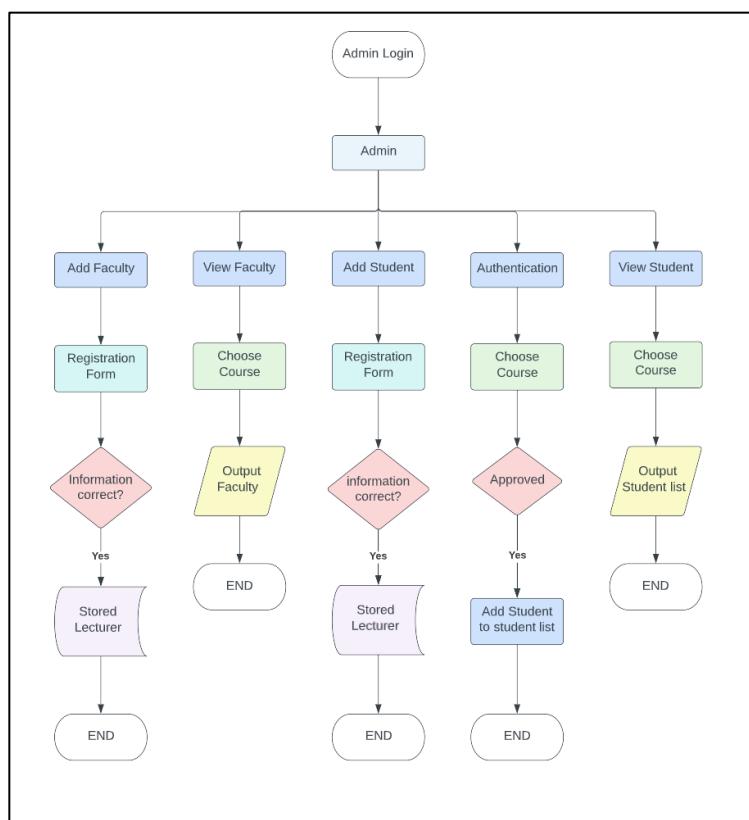


Figure 4.1

Admin

There is only one admin login for the application. The admin can be a member of the faculty or administration that look after the administrative work. The admin has login “admin_me@gmail.com”. Figure 4.11 shows the admin dashboard. The admin of the application can

- Register Student
- View Student
- Register Lecturer
- View Lecturer
- Delete Lecturer
- Authenticate New User

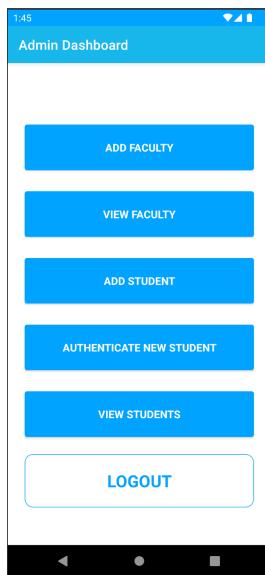


Figure 4.11

The screenshot shows the 'Add Student' form. It includes fields for FIRSTNAME (Firstname), LASTNAME (Lastname), EMAIL (stu_email@gmail.com), ADDRESS (Address), DATE OF BIRTH (06/02/2004), COURSE (Choose Course), and Password. A 'SUBMIT' button is at the bottom.

Figure 4.12

The screenshot shows the 'Add Student' form with filled fields: FIRSTNAME (keshika), LASTNAME (R), EMAIL (stu_keshika@gmail.com), ADDRESS (Dublin), DATE OF BIRTH (06/02/2004), COURSE (Computing Business), and Password. A 'SUBMIT' button is at the bottom. A success message 'REGISTRATION SUCCESS' is displayed at the bottom.

Figure 4.13

The screenshot shows a success message 'REGISTRATION SUCCESS' displayed prominently on the screen.

Figure 4.14

Register Student

Admin can register student by entering the details of the student. In registration of student admin will require the following detail such as first name, last name, email, address, Date of Birth, Course and Password. Figure 4.12 shows the add student page. When an admin registers a student like in figure 4.13, the student does not require any further approval and the student can login successfully. A message of success also appears at the bottom of the screen once a student has been registered successfully as shown in figure 4.14. The email for the student starts with “stu_” followed by a name and “@gmail.com”.

View Student

The admin can view the list of students that has been registered for a particular course. Figure 4.15 after clicking the view student button it will shows the spinner where the course can be selected which student is registered for as it shown in figure 4.16. After selecting a course and the submit button is clicked, the students registered for that field appears on screen as shown in Figure 4.17 where admin can see the student details and can delete the student as well just by clicking the button shown under student details.

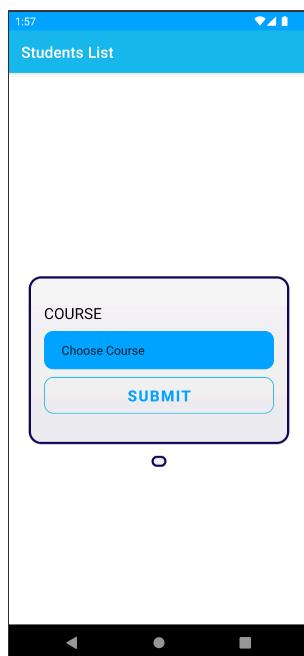


Figure 4.15

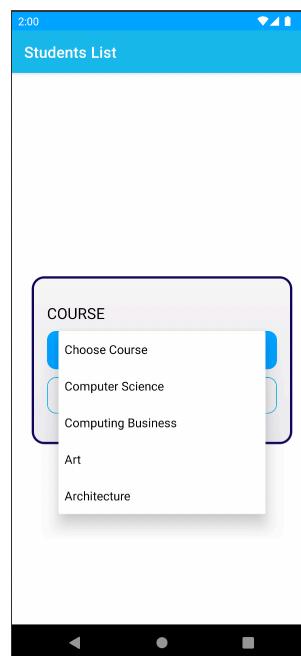


Figure 4.16

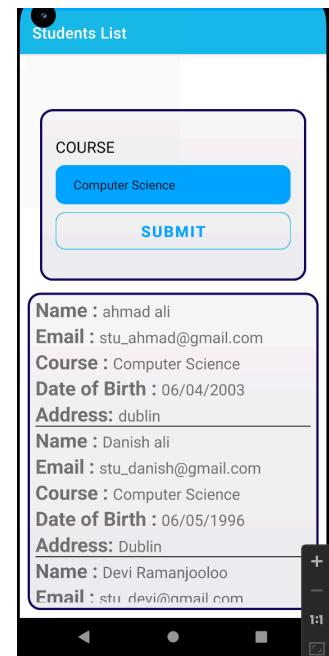


Figure 4.17

Register Lecturer

Admin can register lecturer by entering the details of the lecturer. Figure 4.18 In registration of lecturer admin will require the following details of lecturer such as first name, last name, email, address, faculty, and Password. Figure 4.19 shows the add lecturer page. When an admin registers a lecturer, the lecturer does not require any further approval and the lecturer can login successfully. A message of success also appears at the bottom of the screen once a lecturer has been registered successfully as you can see under figure 4.20. The email for the lecturer starts with "lec_" followed by a name and "@gmail.com".

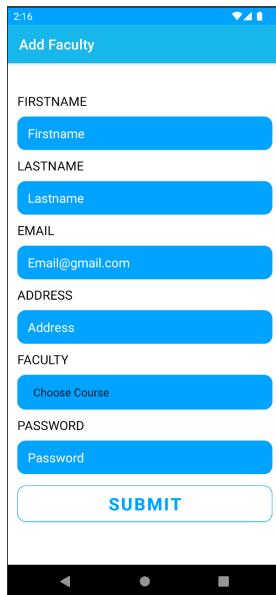


Figure 4.18

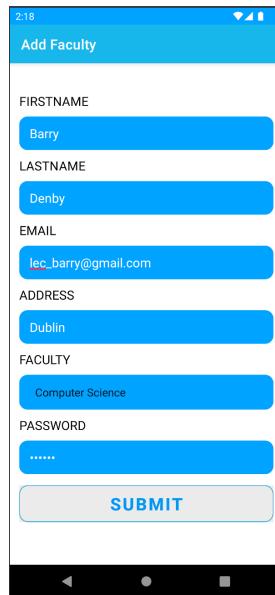


Figure 4.19

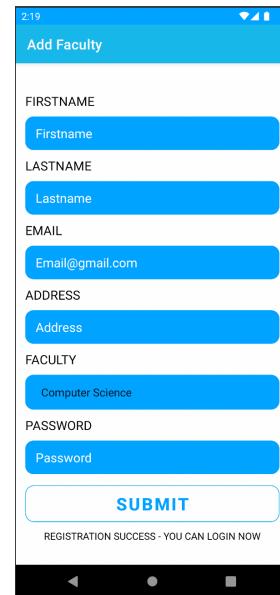


Figure 4.20

View Lecturer

The admin can view the list of lecturers that has been registered for a particular course. After clicking the view lecturer button admin will get a choose course option as shown in figure 4.21. In Figure 4.22 shows the spinner where the course can be selected which lecturer is teaching once. After selecting a course and the submit button is clicked, the details of lecturers registered for that field appears on screen as shown in Figure 4.23 where admin can view details of lecturer and can delete the lecturer.

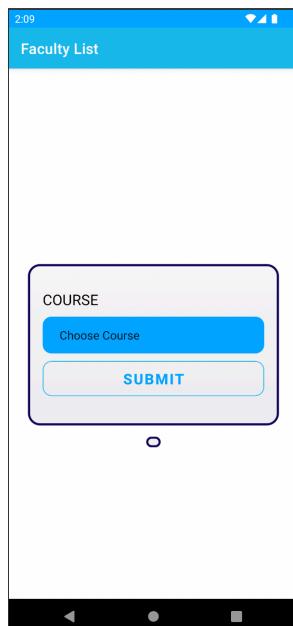


Figure 4.21

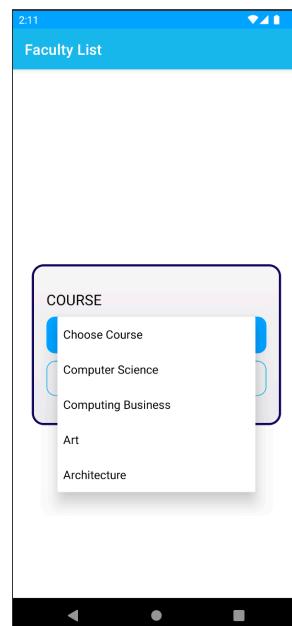


Figure 4.22

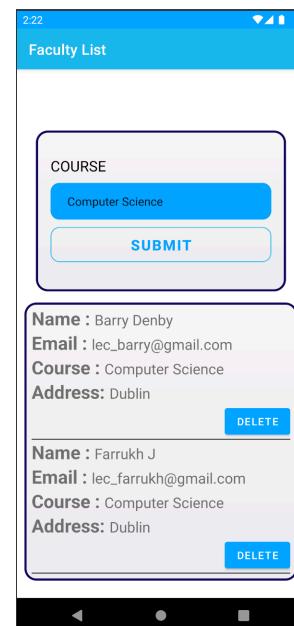


Figure 4.23

Delete Lecturer

The admin can view the list of lecturers that has been registered for a particular course. The course is selected from the spinner. After selecting a course and the submit button is clicked, the details of lecturers registered for that field appears on screen with the delete button as it shown in figure 4.24. When a delete button is clicked the lecturer is deleted as shown in figure 4.25.

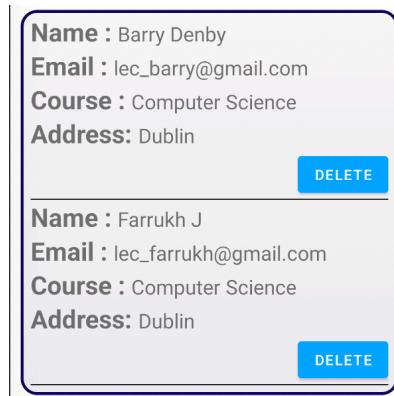


Figure 4.24

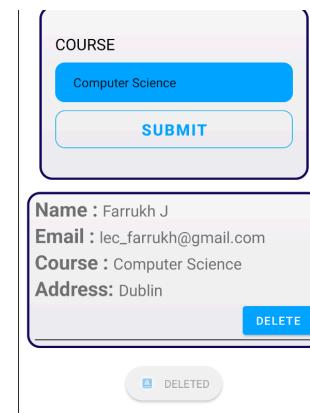


Figure 4.25

Deactivate Student

The admin can view the list of students that has been registered for a particular course. The course is selected from the spinner. After selecting a course and the submit button is clicked, the details of student registered for that field appears on screen with the Deactivate button as shown in figure 4.26. When the Deactivate button is clicked the student is deactivated from the database as shown in figure 4.27.



Figure 4.26

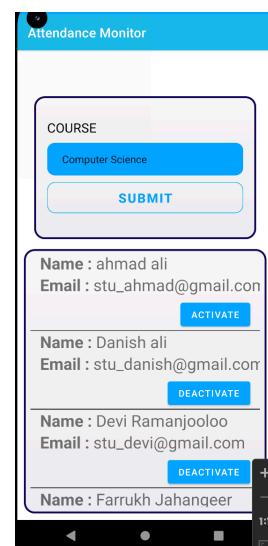


Figure 4.27

Authenticate New Student

If a student has completed the registration form by himself / herself. That student would not be able to login just after registration after registration is done it will come to admin authentication page where admin will select the course spinner as shown in figure 4.28 after clicking the button activate then the student login account will be activated, and student can login successfully. The admin must approve the student registration first then the student would be able to login. Figure 4.29 shows the page where an admin can approve a student.

The spinner is to select the course of the student, as soon as the course id selected and the submit button is clicked, the names and email address of all students that are yet to be authenticate come on the screen. To give approval to a particular student the “Activate” button is clicked. This allows a student to use that email address. But admin also has authority to deactivate the account once he clicked on activate student log in will be activated then in view student under active student details there will deactivate button once admin will click on deactivate student will not be able to login.



Attendance Monitor

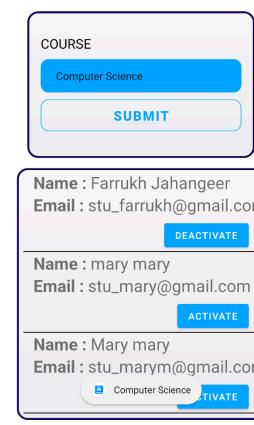


Figure 4.28

Figure 4.29

Student Flow Diagram

Figure 4.30 shows the student flow diagram which depicts the main activities and their processes for process as you can see below all the functionality linked with student log in is given below.

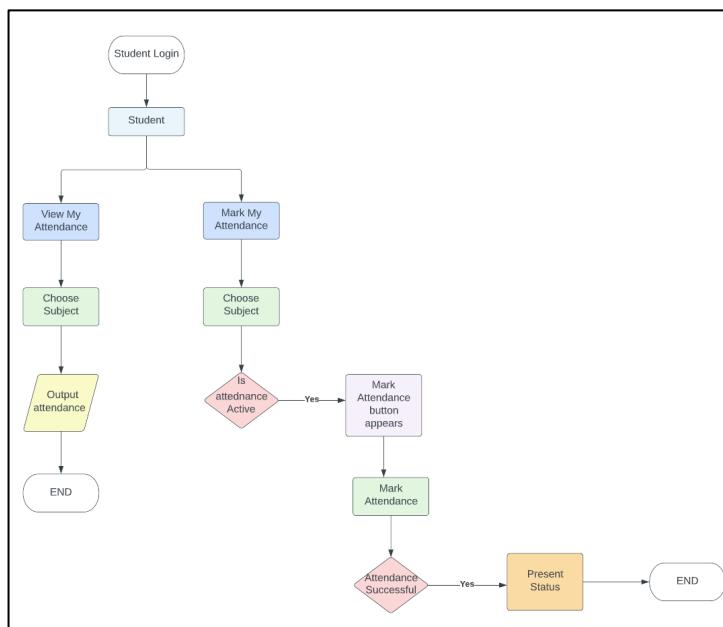


Figure 4.30

Student

When student will open the attendance app. A student will see two options. Firstly, there is a registration option where the student can register by himself using the register button but after filling the registration form, student will see message of authentication. Then that student will pop up in admin view student page where the admin must confirm the registration by clicking on authenticate button after that the student can have access to the application. Secondly, admin can register the student and can give log in details to student and therefore no further admin approval is required. The student login starts with “stu_” followed by the student’s name and “@gmail.com”.

After logging in successfully the student will be able view The student dashboard contains the detail of the student followed by three buttons, namely View attendance, Mark My Attendance and Log out. Figure 4.31 shows a student dashboard.



Figure 4.31

View My Attendance

Figure 4.32 and 4.33 shows the view attendance page. On the top of page there is spinner where the student can select its module for which he wants to see the attendance. Afterwards, the percentage attendance for that course is displayed followed by the date and the time he marked himself present and all previous dates with time will be displayed below and the overall attendance percentage will be displayed of the specific subject which student choose to see attendance and overall attendance will be given as well as it is shown in the figures below.

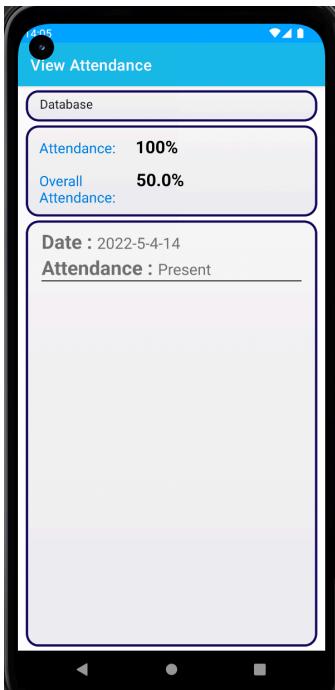


Figure 4.32

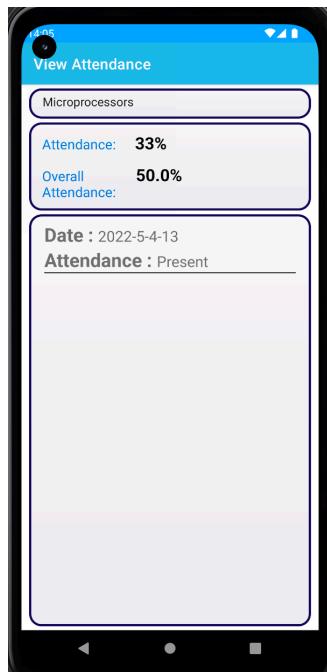


Figure 4.33

Mark My Attendance

This activity contains a status text on the top of the page which shows whether attendance marking is Active or Inactive. As soon as student click on view my attendance button student will status on the top of app either active or not active. If the status is in active, it will be written active in green colour and student will be able to see spinner where can select subjects and can see which subject attendance status is active

The next feature is the spinner that the student uses to select the module that needs to be mark present. Next comes, a button called "Mark me", which is click by the student to mark himself present. At the bottom of the page, is another status text that shows whether attendance is successful or not. Figure 4.34 and 4.35 shows a student Mark My Attendance Activity.

To be able to mark one present the student must first select the module, Afterwards the status will be shown whether it is Active or Inactive. The Mark me Button is only accessible if the lecturer has opened the attendance timer. The students have five minutes to mark themselves as present when attendance is open by lecturer. If the attendance status is inactive subject spinner will be disable student will not be able to click or select any subject and mark my attendance will be disable as well

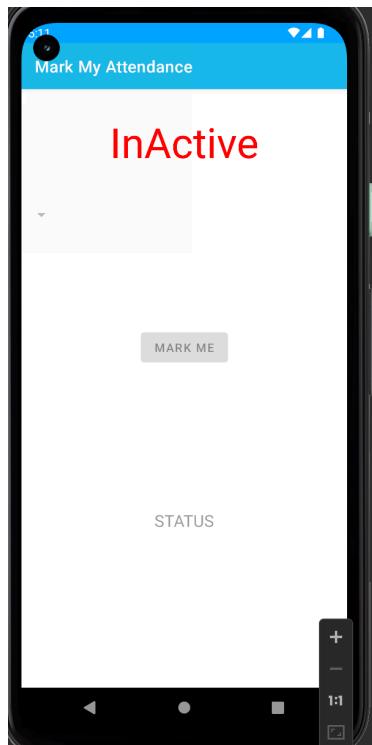


Figure 4.34



Figure 4.35

Lecturer Flow Diagram

Figure 4.36 shows the flow diagram of the lecturer. The flow diagram shows all the main activities and their process and theirs functions from start to end in the figure below.

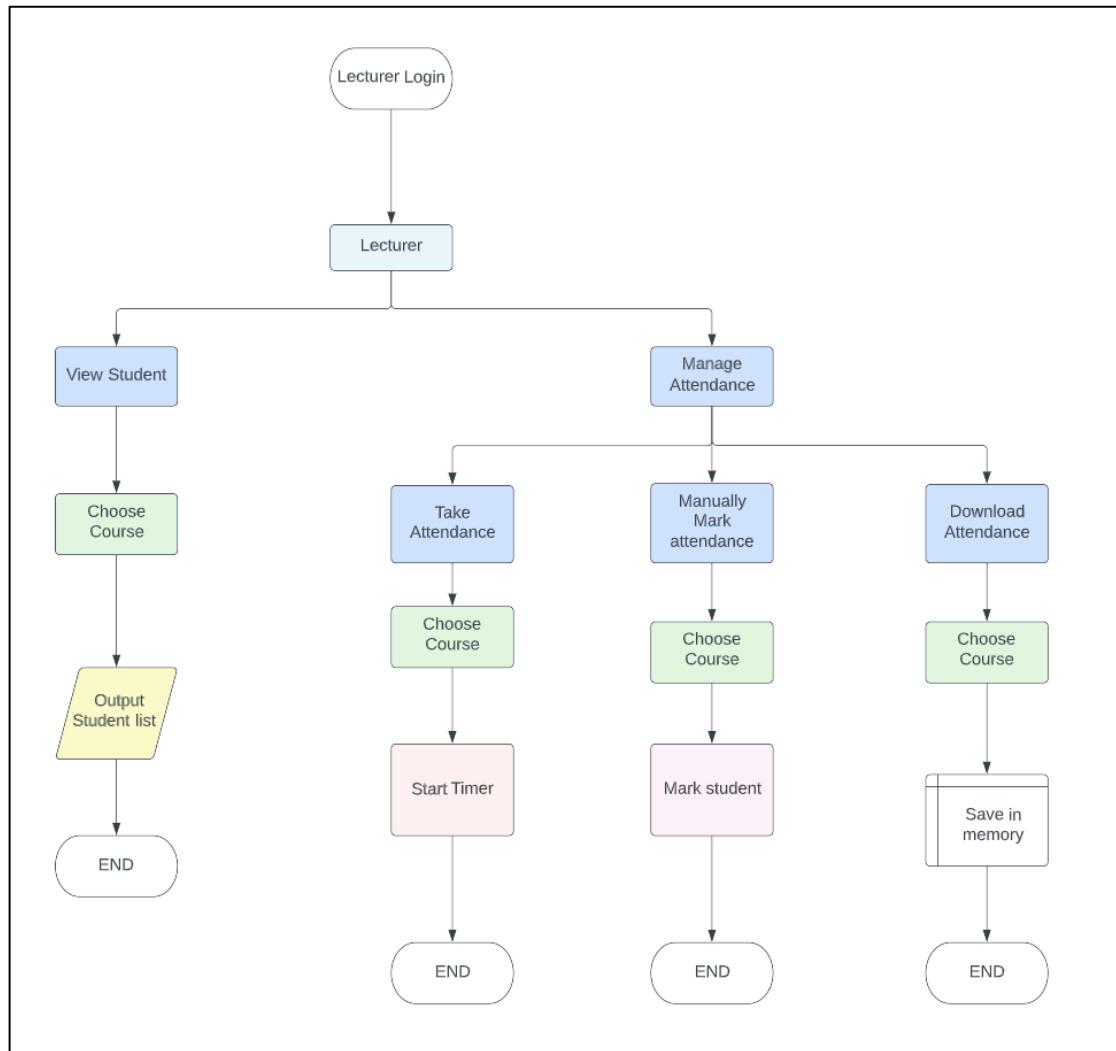


Figure 4.36

Lecturer

A lecturer can View Students and Manage Attendance. Figure 4.37 shows a lecturer dashboard. When a lecturer logs in, he can view his name on top of page, followed by three buttons namely View Student, Manage Attendance, and logout. The lecturer login starts with "lec_" followed by the lecturer's name and "@gmail.com". in the figure below you can see all the functions of lecturer.

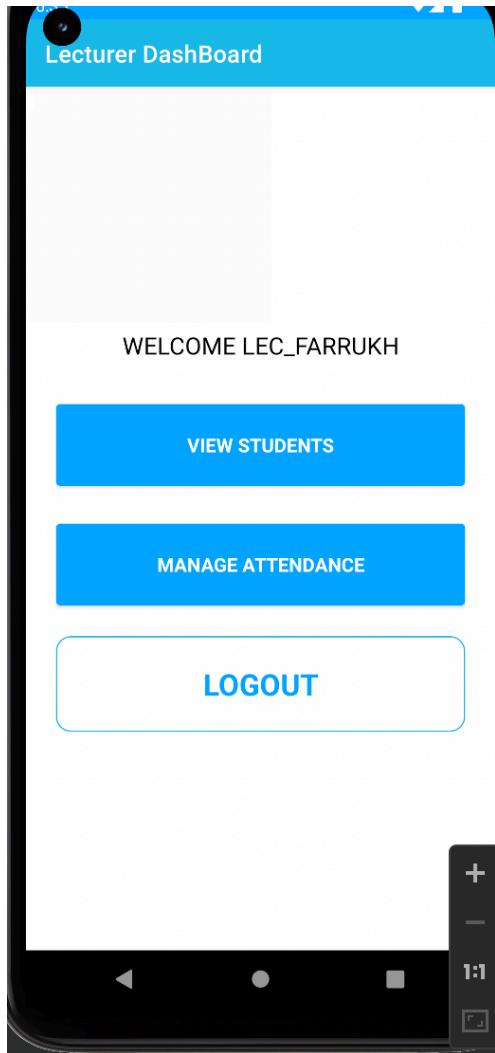


Figure 4.37

View Student

The lecturer can view the list of students that has been registered for a particular course. Figure 4.38 shows the lecturer will click on view student button he will see spinner where the course can be selected which student is taking. After selecting a course and the lecturer will click submit button as shown in figure 4.39. the students registered for that course appears on screen as shown in Figure 4.40 lecturer can see the details of student.

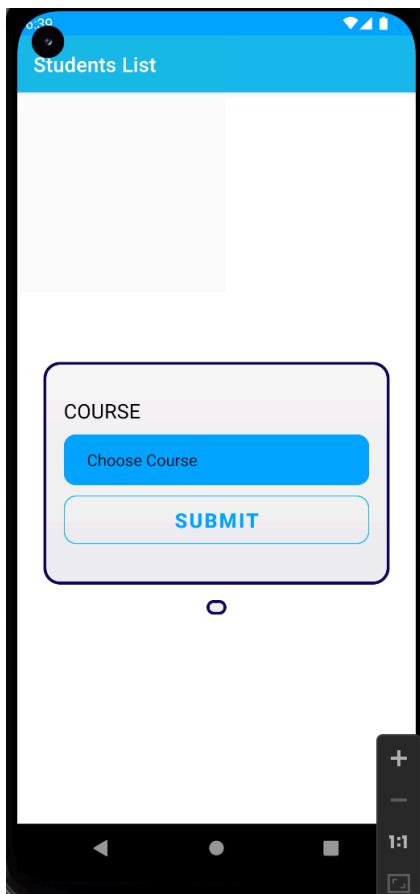


Figure 4.38

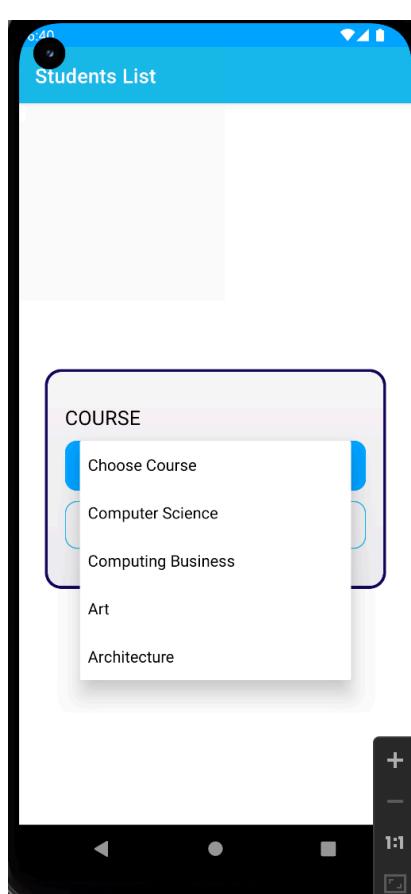


Figure 4.39

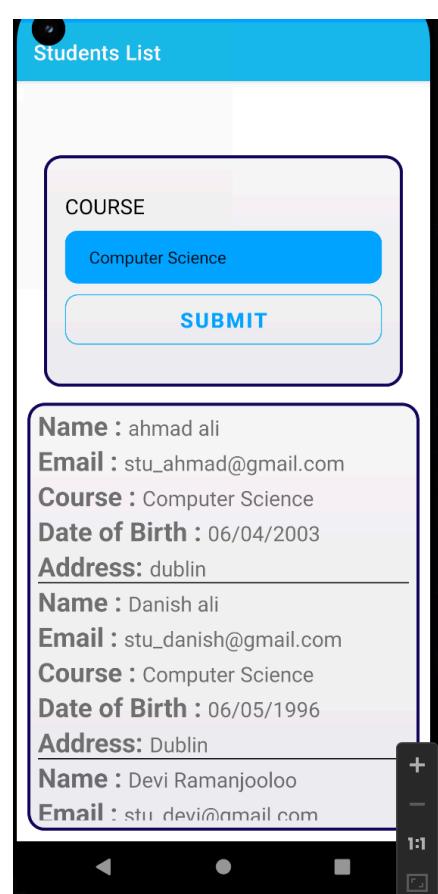


Figure 4.40

Manage Attendance

Manage Attendance Activity has three buttons namely Take Attendance, Manually Mark Attendance and Download Attendance. Figure 4.41 shows the Manage Attendance Activity.

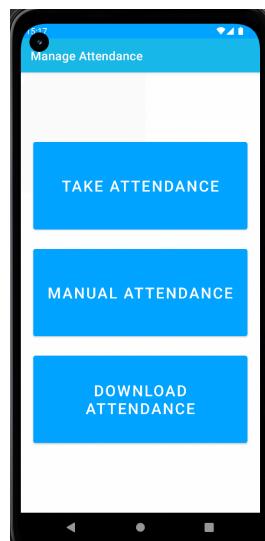


Figure 4.41

Take Attendance

Figure 4.42 shows the Take Attendance Activity. It contains a heading where course and subject is selected from a spinner. When the submit button is clicked after selecting course and subject, the timer button for 5 minutes is visible and can then be clicked. The countdown can also be seen, and the timer can be paused which as well will make attendance inactive for a while as shown in Figure 4.43. Underneath the timer button there are two buttons count the Attendance and Update Previous Attendance.

The count the Attendance button is used to save the attendance. The Update Previous Attendance Button is used in cases where the lecturer wants to open the attendance for a second five minutes then he will do update previous attendance after timer then the two attendances will merge.

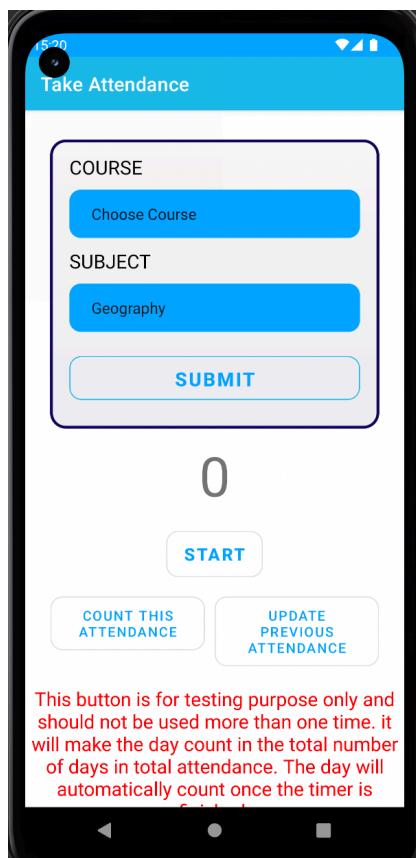


Figure 4.42

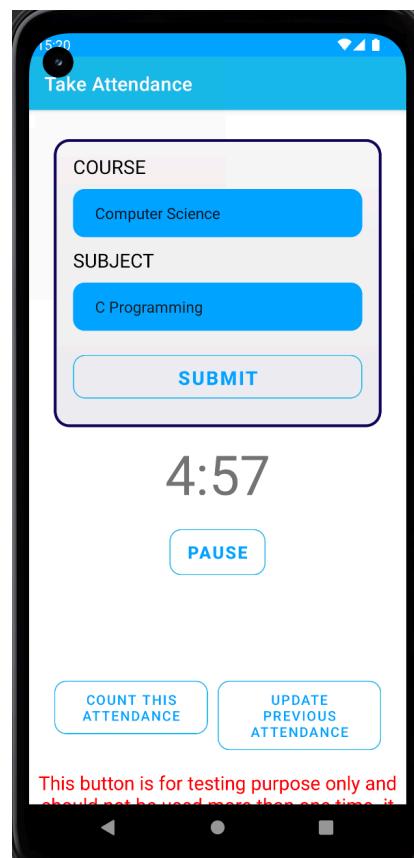


Figure 4.43

Manually Mark Attendance

This Activity enable the lecturer to select course, subject, and date from spinners and by clicking the Submit button, it will render all the students from that course. The

student's name appears underneath with a "Mark Present" Button, which is used to mark a student present. Figure 4.46 show this activity.

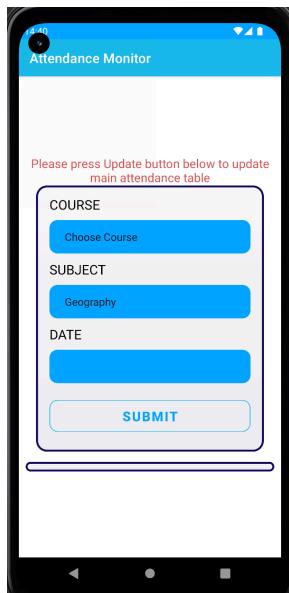


Figure 4.44

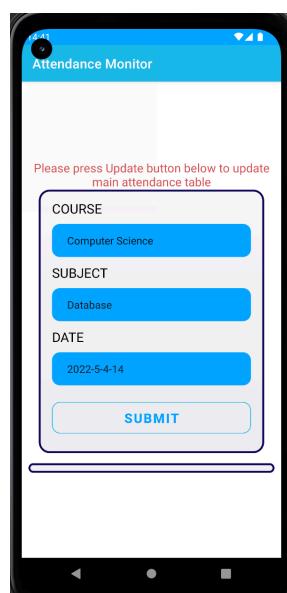


Figure 4.45

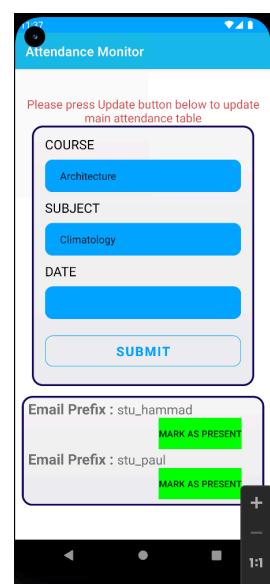


Figure 4.46

Download Attendance

Figure 4.47 shows the download Attendance activity. The lecturer selects the course and subject for which the attendance must be downloaded. After clicking the submit button, the attendance is downloaded in the external memory, in the code file and it is saved with the date as name as shown in Figure 4.48.

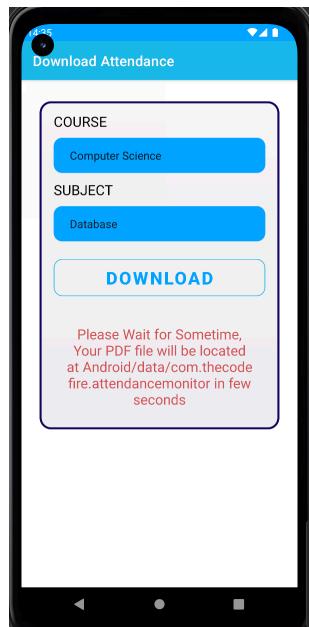


Figure 4.47

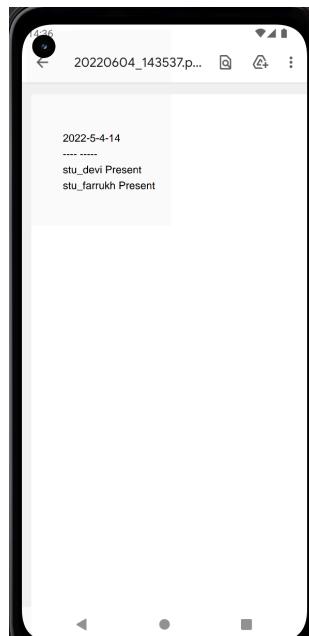


Figure 4.48

Chapter 5. Testing

This application has been tested using Unit Testing and Integration Testing. In Unit Testing all logic and design of the application was tested. All individual unit of code and at times set of more than one unit was tested to ensure it is working. For instance, testing the functionality of registering a lecturer. All types of data are entered to see if the application still runs. All the breaking points and failures are kept in mind while testing to ensure correct functionality. For integration testing, few features which are joined together were tested. For example, when a student is registered successfully, it should appear on the lecturer student list and on the student dashboard, the student's detail should appear.

Some of the testing performed are:

- ❖ Adding Duplicate Username
- ❖ Date of Birth
- ❖ Registering Student
- ❖ Registering Lecturer
- ❖ View student
- ❖ View Lecturer
- ❖ Active Attendance
- ❖ Take Attendance
- ❖ Mark Attendance
- ❖ Download Attendance
- ❖ View percentage attendance
- ❖ Approval from Admin
- ❖ Deleting student
- ❖ Deleting Lecturer

Adding username

If a username is already being used and a new user is being registered with same email address, an error is thrown saying “User Already exist”. Figure 5.0 shows an example of this testing.

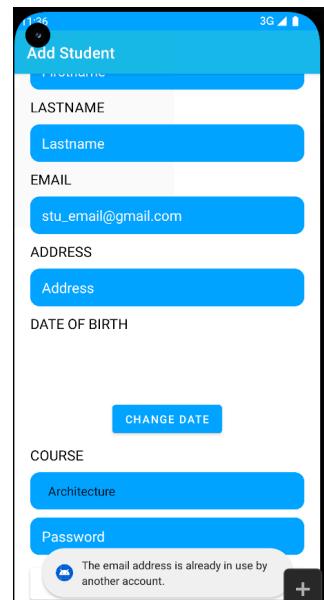


Figure 5.0

Date of birth

A date of birth can be a date in the past, if a date from the future is entered, the registration form is not saved, and a Toast message is output. Student Age should be greater than 16 years old. Figure 5.2 shows a testing screen shot of this.

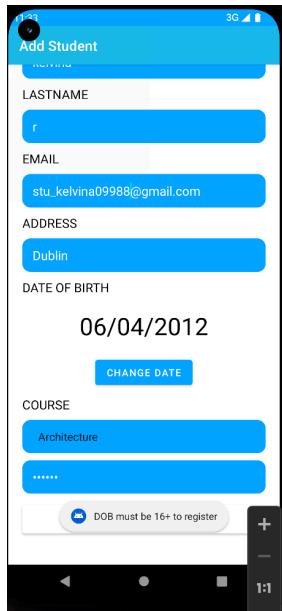


Figure 5.2

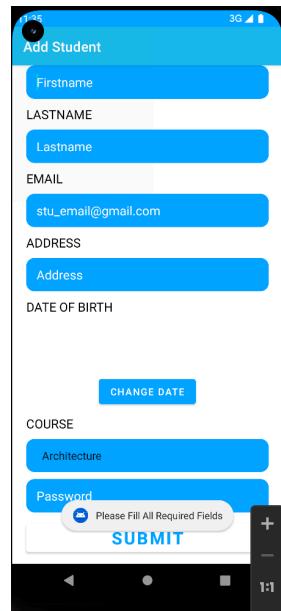


Figure 5.3

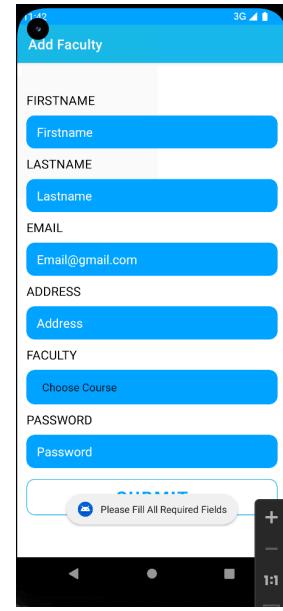


Figure 5.4

Register Student

This Activity was tested by not filling all the field of information and trying to save or register the student. As seen in Figure 5.3 it throws an error message.

Register Lecturer

This Activity was tested by not filling all the field of information and trying to save or register the lecturer. As seen in Figure 5.4 an error message appears on screen.

View Student

View student testing was done to ensure only student from the selected course appears on the screen. Figure 5.5 and Figure 5.6 shows this testing activity.

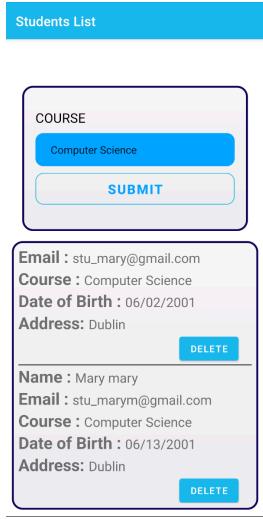


Figure 5.5

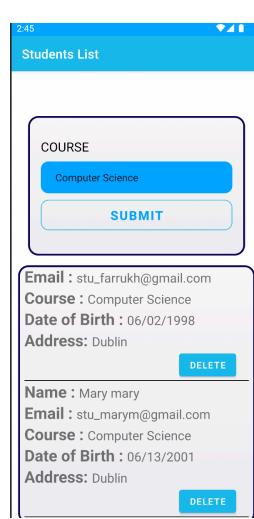


Figure 5.6

View Lecturer

View lecturer testing was done to ensure only lecturer from the selected course appears on the screen. Figure 5.7 shows this testing activity.

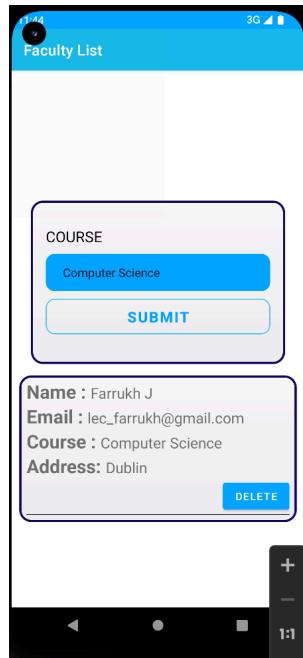


Figure 5.7

Active Attendance

This test was done to ensure that when a lecturer start a timer, it becomes active for the student of that course and subject only. Figure – shows the page where lecturer enable the attendance. Figure 5.8 shows the screen of a student from the course when the lecturer has enabled attendance.

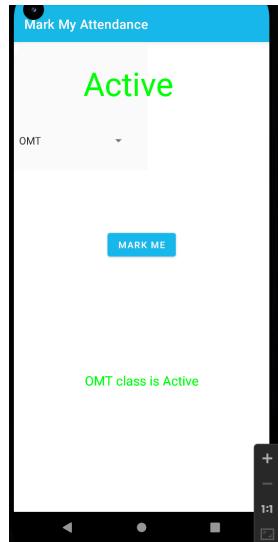


Figure 5.8

Mark Attendance

The student can mark its attendance when attendance is enabled. When attendance is enabled, students can login in the application and mark themselves as present. Figure - -- show a student that has its attendance active. Figure 4.42 shows the page when the student has marked himself as present.

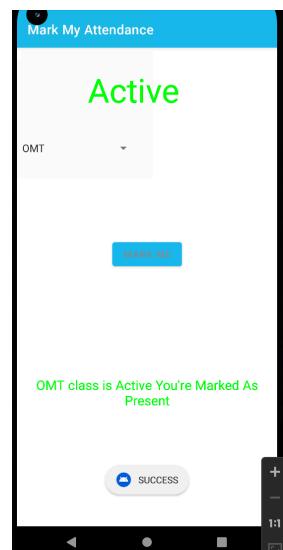


Figure 5.9

Download attendance

During this test, it was confirmed that the attendance list is being created and downloaded and contain all the dates. Figure 5.11 shows a downloaded PDF that is saved in the Android folder in the package of the application after selection as shown in Figure 5.10.

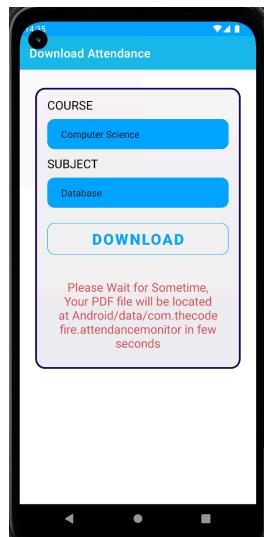


Figure 5.10

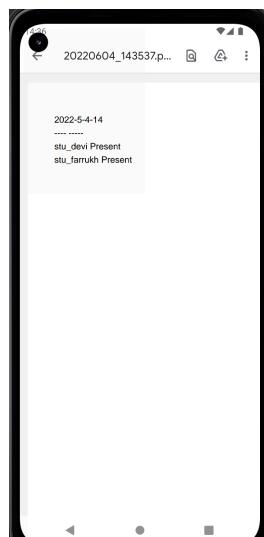


Figure 5.11

View Attendance Percentage

The student has a functionality to view his / her percentage attendance. This testing was done to ensure the calculation is showing up right and the percentage attendance is displayed. Figure shows the activity with the percentage of attendance.

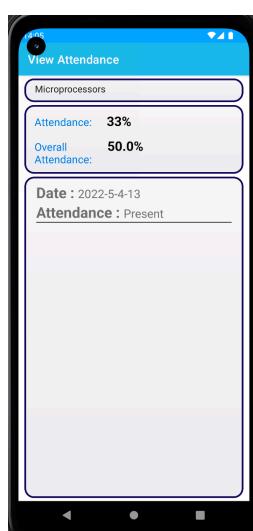


Figure 5.12

Approval from Admin

This testing was done to confirm that a student cannot login before approval of a admin if the student has registered by itself as shown in Figure 5.13.



Figure 5.13

Deleting Lecturer

Admin can delete a lecturer if the lecturer is longer teaching in that institution. Figure 5.17 shows the lectures before deleting and Figure 5.18 shows the page when a lecturer has been deleted.

Name : Barry Denby
Email : lec_barry@gmail.com
Course : Computer Science
Address: Dublin
DELETE

Name : Farrukh J
Email : lec_farrukh@gmail.com
Course : Computer Science
Address: Dublin
DELETE

Figure 5.17

COURSE
Computer Science
SUBMIT

Name : Farrukh J
Email : lec_farrukh@gmail.com
Course : Computer Science
Address: Dublin
DELETE

DELETED

Figure 5.18

Values used for testing

Student

The Table 1 below shows the data used to test for adding user through admin registration and student registration followed by Approval.

Email	Password	Faculty
Stu_junaid@gmail.com	123456	Art
stu_keshika@gmail.com	123456	Computer Business
Stu_hammad@gmail.com	123456	Architecture
Stu_kashif@gmail.com	123456	Architecture
Stu_farrukh@gmail.com	123456	Computer Science
Stu_marym@gmail.com	123456	Art
Stu_devi@gmail.com	123456	Computer Science

Table 1

Lecturer

The tables below show the username and password used during testing of the application.

Email	Password	Faculty
Lec_farrukh@gmail.com	1111111	Computer science
Lec_paddy@gmail.com	123456	Computing Business
Lec_sarah@gmail.com	123456	Architecture
Lec_gemma@gmail.com	123456	Art

Table 2

Admin

Email	Password	Faculty
admin_me@gmail.com	000000	admin

Table 3

Chapter 6. Evaluation

List of Proposed Tasks

User	Proposed Task	Task Completed	Not Completed
Lecturer:	Login	✓	
	Enable attendance to mark attendance	✓	
	Attendance is enabled for Five minutes only	✓	
	Mark student's attendance Manually	✓	
	View all modules	✓	
	View student list	✓	
	Download Attendance list	✓	
	Reset Password	✓	
Student	Login	✓	
	Mark present when attendance is enabled	✓	
	View percentage attendance	✓	
	View Module	✓	
	View profile details	✓	
Admin	Reset Password	✓	
	Login	✓	
	Register student	✓	
	Approve registration of student	✓	
	Register a member of faculty	✓	
	View Student	✓	
	View Lecturer list	✓	
	Reset Password	✓	

Table 4

The proposed features of the system are all implemented except for Reset password for each of the users. Since the username / email being used is not real / valid, the reset password will not be possible, but in a real time system using this application with valid email id the reset password will be working and password can be reset.

In the proposed form of storage for the application MySQL and Knuth Server was included. In the final solution, these are not present because the Knuth server was not working. Therefore, I had to review my technologies and choose another suitable one. After careful consideration, I decided to use Firebase and NoSQL.

Chapter 7. Conclusions and Future Work

Conclusion

In this fast-developing world of technology, mobile phone is one of the devices that have been integrated in our daily life. Most people cannot stay for more than an hour without their mobile phone. Mobile phone has software and applications which can be accessible from anywhere and anytime. Therefore, there is a big trend in mobile phone and mobile applications.

One of the applications that is being used nowadays is the mobile attendance system. This application has been taken up nearly everywhere, in schools and offices. It is great to monitor attendance of student and employee. The mobile attendance system has various advantages for example it is user-friendly, it can save time, it reduces amount of paperwork and administration work.

This project has resulted in helping instructors and students to quickly record attendance. Moreover it helps to save faculty time by exporting the attendance to a well-known format (pdf) digitally and download it to the lecturer's device. This application mainly focuses on educational institutes use cases and can be further improved to support any type of institution where attendance is still manual.

This application helps the admin to register Student, View Student, Register Lecturer, View Lecturer, Delete Lecturer and Authenticate New Student.

The student can mark themselves as present, view attendance for each subject, view overall attendance and mark attendance when attendance is open.

The lecturer can view student, manage attendance by taking attendance, manually mark attendance and download attendance.

Future Work

For future works:

- ❖ The application can be developed to suit other OS and Desktop.
- ❖ The application can be improved to use biometric or sensor to capture attendance rather than trying to mark by themselves.
- ❖ For More Convenience for login purposes the system can use biometric instead of passwords as passwords are hard to remember.
- ❖ Prepare an attendance system for people with physical disabilities as they have a right to everything as normal people.
- ❖ Lecturer should see only assigned modules for attendance
- ❖ Student can only login to one device. to register another device, he must need to unregister first device and then he can use his unique ID number to login to new device example AIB mobile app

References

Anon., n.d. *Andriod set up*. [Online]

Available at: <https://firebase.google.com/docs/android/setup>

Anon., n.d. *Build attendance Application*. [Online]

Available at: <https://medium.com/geekculture/build-simple-attendance-app-in-kotlin-part-1-dashboard-layout-c111b84ea9cf>

Anon., n.d. *Button*. [Online]

Available at: <https://abhiandroid.com/ui/button>

Anon., n.d. *Future wprk*. [Online]

Available at: <https://hrone.cloud/future-of-work-reinvented-attendance-management/>

Anon., n.d. *How to download*. [Online]

Available at:

https://www.reddit.com/r/androiddev/comments/u7ayii/how_to_download_file_to_external_storage_for/

Anon., n.d. *Kotlin*. [Online]

Available at: <https://www.tutorialkart.com/kotlin-android-tutorial/>

Anon., n.d. *Mobile Attendance System Project*. [Online]

Available at: <https://nevonprojects.com/mobile-attendance-system-project/>

Anon., n.d. *Read and write*. [Online]

Available at: <https://firebase.google.com/docs/database/android/read-and-write>

Denby, B., n.d. *Android Examples*. s.l.:s.n.