

**AN ANDROID BASED MOBILE APPLICATION FOR SMART COUNSELLING SYSTEM**

**SUBMITTED BY**

**Tashrif Rahman**

**ID: 182-15-11481**

**Humayra Rahman Silve**

**ID:182-15-11366**

This Report Presented in Partial Fulfillment of the Requirements for the Degree of  
Bachelor of Science in Computer Science and Engineering

**Supervised By**

**Ms. Nazmun Nessa Moon**

Associate Professor

Department of CSE

Daffodil International University

**Co-Supervised By**

**Mr. Md. Sadekur Rahman**

Assistant Professor

Department of CSE

Daffodil International University



**DAFFODIL INTERNATIONAL UNIVERSITY**

**DHAKA, BANGLADESH**

**January, 2023**

## **APPROVAL**

This Project “AN ANDROID BASED MOBILE APPLICATION FOR SMART COUNSELLING SYSTEM” submitted by Tashrif Rahman ID No: 182-15-11481; Humayra Rahman Silve ID No: 182-15-11366; to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 24 January.

### **BOARD OF EXAMINERS**

**Chairman**

\_\_\_\_\_

**Dr. Touhid Bhuiyan**

**Professor and Head**

Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University

\_\_\_\_\_



**Abdus Sattar**

**Assistant Professor**

Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University

**Internal Examiner**

\_\_\_\_\_



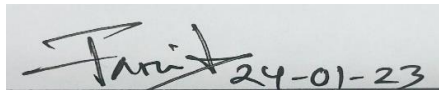
**Fatema Tuj Johra**

**Senior Lecturer**

Department of Computer Science and Engineering  
Faculty of Science & Information Technology  
Daffodil International University

**Internal Examiner**

\_\_\_\_\_



**Dr. Dewan Md Farid**

**Professor**

Department of Computer Science and Engineering  
United International University

**External Examiner**

## DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Ms. Nazmun Nessa Moon, Associate Professor, Department, of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

### Supervised by:



---

**Ms. Nazmun Nessa Moon**

Associate Professor

Department of CSE

Daffodil International University

### Co-Supervised by:

---

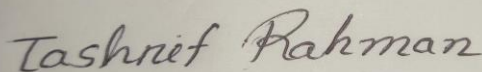
**Mr. Md. Sadekur Rahman**

Assistant Professor

Department of CSE

Daffodil International University

### Submitted by:



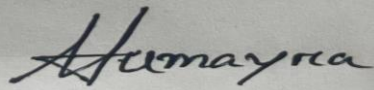
---

**Tashrif Rahman**

ID: 182-15-11481

Department of CSE

Daffodil International University



---

**Humayra Rahman Silve**

ID: 182-15-11366

Department of CSE

Daffodil International University

## ACKNOWLEDGEMENT

First, we express our heartiest thanks and gratefulness to almighty God for his divine blessing making us possible to complete the final year project/internship successfully.

We are grateful and wish our profound indebtedness to **Supervisor Ms. Nazmun Nessa Moon, Associate Professor, and Co-Supervisor Mr. Md. Sadekur Rahman, Assistant Professor,** Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of “Field name” to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

We would like to express our heartiest gratitude to ---- Professor **Dr. Touhid Bhuiyan Head, Department of CSE,** for his kind help to finish our project and also to other faculty members and the staff of the CSE department of Daffodil International University.

We would like to thank our entire course mates at Daffodil International University, who took part in this discussion while completing coursework.

Finally, we must acknowledge with due respect the constant support and patients of my parents.

## **ABSTRACT**

The title of our project is "**AN ANDROID BASED MOBILE APPLICATION FOR SMART COUNSELLING SYSTEM**". We named this in this program, our primary goal was to secure appointments from university or college faculty members. Students are the ones making this appointment. Faculty members and students share a single application. In this project, in order to communicate with one another, An institution's teachers and students should enroll in this system. Students can use their name to search for a certain professor. and can send an appointment with the appropriate justification. A student can find out more information about this lecturer. The lecturer has access to this appointment request's specifics. After receiving a notification that the request has been sent. The faculty member might then respond with a brief message accepting or rejecting this request. After accepting or rejecting this response, the student is notified. Students and faculty members can both view their appointment in the application's dashboard. The "Note Section" is where the faculty member can record his or her appointment to teach which will save in this system.

# TABLE OF CONTENTS

<b>CONTENTS</b>	<b>PAGE</b>
Approval Page	i
Declaration	ii
Acknowledgements	iii
Abstract	iv
<b>Chapter 1: Introduction</b>	<b>1- 3</b>
1.1 Introduction	1
1.2 Motivation	1
1.3 Objectives	2
1.4 Expected Outcomes	2
1.5 Project Management and Finance	2
1.6 Report Layout	3
<b>Chapter 2: Background</b>	<b>4 - 5</b>
2.1 Preliminaries/Terminologies	4
2.2 Related Works	4
2.3 Comparative Analysis	4
2.4 Scope of the Problem	5
2.5 Challenges	5
<b>Chapter 3: Requirement Specification</b>	<b>6 - 10</b>
3.1 Business Process Modeling	6
3.2 Collection and Analysis of Requirements	7
3.3 Use Case Modeling and Description	8

<b>Chapter 4: Design Specification</b>	<b>10 - 34</b>
4.1 Front-end Design	10
4.2 Back-end Design	28
4.3 Implementation Requirements	34
<b>Chapter 5: Implementation and Testing</b>	<b>35 - 41</b>
5.1 Implementation of Database	35
5.2 Front-end design implementation	40
5.3 Implementation, results, and reporting of testing	40
<b>Chapter 6: Ethical Aspect</b>	<b>42-45</b>
6.1 Ethical impacts on society	42
6.2 Ethical impacts on environment	43
6.3 Ethical aspects	43
6.4 Sustainability plans	44
<b>Chapter 7: Conclusion and Future Scope</b>	<b>46</b>
7.1 Discussion and Conclusion	46
7.2 Limitation	46
7.3 Scope for Further Developments	46
<b>References</b>	<b>47</b>
<b>Plagiarism Check Report</b>	<b>48</b>

## LIST OF FIGURES

<b>FIGURE</b>	<b>PAGE</b>
Figure 3.1: Data Flow Diagram	06
Figure 3.2: Use case illustration	08
Figure 4.1: Splash screen	10
Figure 4.2 Register screen	11
Figure 4.3 Registration verification mail	11
Figure 4.4 A user's login page	12
Figure 4.5 Select screen type for user	13
Figure 4.6 Screen to register students	13
Figure 4.7 Screen 1 of the instructor register	14
Figure 4.8 Screen 2 of the instructor register	14
Figure 4.9 A student's main page	15
Figure 4.10 Inquiry by instructor name on the main page	16
Figure 4.11 Consultation approval/rejection debate	17
Figure 4.12 Awaiting scheduling conversation	17
Figure 4.13 All of the instructors' displays listed	18
Figure 4.14 Splash display or instructor title inquiry	19
Figure 4.15 Instructor information as well as reservation scheduling display	19
Figure 4.16: Display for student profiles	20
Figure 4.17: Display for student edit profile	20
Figure 4.18: Landing page for teacher	21
Figure 4.19: Search on the instructor's main display	22
Figure 4.20: Receiving instructor note	23
Figure 4.21: Consultation approval display	23
Figure 4.22: Consultation display for requests	24
Figure 4.23: Note sheet for instructors	24
Figure 4.24: Window to add notes	25
Figure 4.25: View of an instructor's profile	25



Figure 4.26: Instructor upgrade consultation display	26
Figure 4.27 Display 1 of the instructor profile upgrade	26
Figure 4.28: Display 2 of the instructor profile upgrade	27
Figure 4.29: Back-end layout that revealed a partially separated Java file. Data source, adapter, and di all refer to the dependency injection bundle	28
Figure 4.30: Back-end layout that revealed a partially separated Java file. Model, notice, and source bundle are present	29
Figure 4.31: All this belongs to the java file. which display the UI, features, and models bundle	30
Figure 4.32: Includes layout template. All front-end design for an Android platform is contained in the layout file	31
Figure 4.33 All required libraries and third-party libraries were included by Gradle Script-1	32
Figure 4.34: All relevant libraries and third-party libraries were included in Gradle Script-2	33
Figure 5.1: Firebase authenticate users	35
Figure 5.2: Firebase Storage	36
Figure 5.3: All teacher information is kept in the fire store directory "all teacher."	37
Figure 5.4: All of the data for student profiles and appointment information is kept in the "student profile" fire store directory.	38
Figure 5.5: All user device tokens are stored in the fire store directory "tokens."	39
Figure 5.6: All user roles (teacher/student) are stored in the fire store directory "user role"	39

## LIST OF TABLES

<b>TABLES</b>	<b>PAGE</b>
TABLE 1.1: Firebase's pricing and free plans	02
TABLE 5.1: Testing result	41

# CHAPTER 1

## INTRODUCTION

### 1.1 Introduction

We frequently use appointment systems in our daily lives. However, in the present period, this appointment system is carried out by means of modern technology, such as phone calls, emailing, or any other technique. We can see that our nation uses a variety of appointment systems, such as physician and patient Appointments. They occasionally submit an application to be appointed. However, there is no application process that allows a student to request a faculty appointment for educational purposes. In general, the institution has many students. And there are many students in each department of a university. In this project we develop an android application so that a student can get an appointment directly from the teacher according to their schedule. In this one-to-one communication method, the student can connect with their teachers immediately. Students can access all faculty contact information, including phone numbers, email addresses, offices, and departments, through this application. However, before using this system, instructors and students must both register. It supports instructors and students in keeping appointments in an efficient and hassle-free manner.

### 1.2 Motivation

We maintain contact over the email, phone and messaging. However, this way you may not be able to reach the faculty. Students sometimes do not know when their teacher will come to the office Cause, counseling schedules are updated each semester or month. When students visit the office, they don't know their free time, so they don't meet with the faculty. Based on our observations of real-world situations, we believe that if there were a platform where students could access all of a faculty's information, including email, phone, counseling schedules, office locations, and other details, they could also schedule appointments with faculty for counseling. Faculty may also respond to this appointment. Additionally, this technology enables communication between instructors and students on one platform. This is how we intend to make this project using this style of thinking.

### 1.3 Objective

- To make the communication between student and faculty is easier.
- In one application student get every information of faculty.
- Receive faculty appointment schedule updates in real-time

### 1.4 Expected Outcomes

The major goal is to create a user-friendly application that allows students and instructors to connect easily.

### 1.5 Project Management and Finance

Two different cost kinds exist. in this project which is database cost and developer cost. There is some cost difference due to using Google Firebase database. The database is free to use, but as the number of users and the volume of data increases then a paid subscription is required. For this project, the developer should budget between 30,000k and 35,000kThe cost maintained based on developer. We have a graphic that shows the price differences between the free and paid Firebase plans based on the number of users and the volume of data. We make use of the Cloud Fire store service and Firebase authentication. When the free limit is reached, the cost is necessary [4].

TABLE 1.1: Firebase's pricing and free plans

Service Name	Free Plan	Cost Plan
Authentication with phone	11k/month	\$0.07/verification
Other Authentication services	No limit	No Cost
Stored data	1 GiB total	\$0.20/GiB
Document writes	21K/day	\$0.20/100K
Document reads	55K/day	\$0.08/100K
Document deletes	21K/day	\$0.04/100K

## **1.6 Report Layout**

### **Chapter1: Introduction**

Introduction of the project, its purpose, its goal, its anticipated results, its management and funding, and its report format which is covers in our project this chapter.

### **Chapter 2: Background**

Some projects similar to ours which is describe this chapter. We will compare our project with other projects and try to state why our project is better than others.

### **Chapter 3: Requirement Specification**

Diagrams showing the use case model, requirements gathering, and our project's business process model which is describe in this chapter.

### **Chapter 4: Design Specification**

Front end design and back-end design of our project which is describe in this chapter. We'll demonstrate the operation of the back-end file structure and the entire design screen. Additionally, talk about the implementation requirements.

### **Chapter 5: Implementation and Testing**

We will describe about front-end design implementation, database design and implementation, and various testing outcomes in this chapter.

### **Chapter 6: Conclusion and Future Scope**

We will find the outcome and limitations of our application in this chapter. Additionally updated on the project's future domain.

## **CHAPTER 2**

### **BACKGROUND**

#### **2.1 Introduction**

This section presents some of the works in our country or in other countries that recruit student-teachers and provide many details about student applications.

#### **2.2 Related Works**

The creators implement a web-based application developed in .NET languages to implement an appointment management system that allows students and teachers to arrange meetings in an efficient and convenient way at the university [1]. Tutors can schedule appointments through a web and mobile application using a GPS system. Tutors can track whether they have completed their work or not through this system [2]. A web-based system where both students and teachers can introduce themselves and receive appointments from teachers. Students and teachers can communicate by sending messages through this method [3].

#### **2.3 Comparative Analysis**

In this case, the initial piece of work is a web-based application that allows teachers and students to schedule meetings, but in our project, we'll construct a mobile-based application that is more streamlined than web applications.

GPS monitoring system is a second that allows tutors to schedule appointments and keep track of their whereabouts in order to ensure that they have completed their work to standard. Upon registering, the teacher enters their office address and room number to our system. That's why no additional GPS system is required.

Has some similarities to our project but also some differences are the third. This system, teachers have the ability to record student's appointments. Teachers receive real-time appointment notifications, and students also receive real-time notifications when teachers respond to notifications.

## **2.4 Scope of the problem**

Much of what we have narrated in the preceding section is web based. However, a mobile-based application is easier to use than a web-based one. Our program uses a Firebase data base, which provides real-time updates and notifies users teachers and students accordingly. While some of the other projects are difficult, ours is incredibly user-friendly and simple to utilize. Additionally, the user is informed when the application is closed.

## **2.5 Challenges**

While creating this project, we ran into a few issues. This application has two types of users. We had trouble keeping track of user type. After the project was completed, we made an effort to fix all of the bugs. We usage advantage of certain cutting-edge technologies for Android programming, including the MVVM coding pattern, Dagger for dependency injection, Android X, and other libraries. Which for us was more challenging.

# CHAPTER 3

## REQUIREMENT SPECIFICATION

### 3.1 Business Process Modeling

This is data flow diagram. The data stream of an information is addressed using a data stream chart. In the financial information framework, DFD streamlines the information stream process from contribution to record accumulating. The Flow Diagram of Data (DFD) consists of two parts. They are physical and sensible. The practical DFD uses a framework to explain the flow of all information and determine the value to company. The use of a genuine information stream is demonstrated by the physical DFD.

Our combined DFD and business process model is shown in figure 3.1 below.

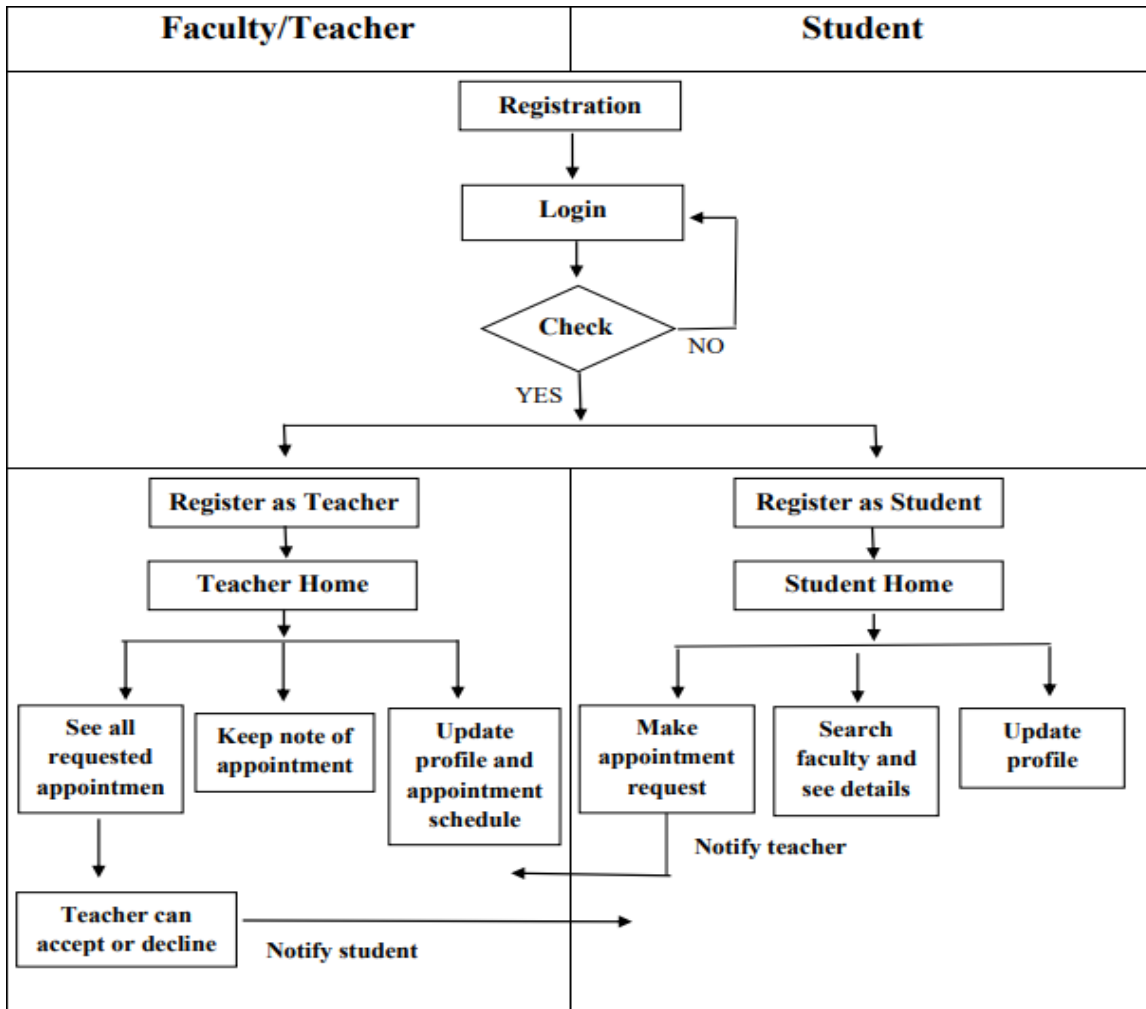


Figure 3.1: Data Flow Diagram



## 3.2 Collection and Analysis of Requirements

This application has some requirements because it is an android-based application. As follows:

- **Hardware set up:**
  - An Android smartphone for each student and teacher.
  - A computer was used to create this software.
  
- **Configuration of the software:**
  - System of operation: Android (API level 21 or higher)
  - Tools: Android Studio IDE (Version: 4.0.1)
  - Database: Firebase Fire store and Storage
  - Server: Firebase server
  
- **Features:**
  - Email & password registration
  - User login
  - Separate instructor and student registration with detailed information;
  - See the full list of teachers;
  - Student requests an appointment with a reason;
  - Teacher accepts or rejects with a response;
  - Teacher makes or note of appointment;
  - Both the Teacher and the Student can view the list of appointments on the Home Page.
  - Update profile
  - Search
  - search, receive notifications for both teachers and students, and
  - Logout

### 3.3 Use Case Modeling and Description

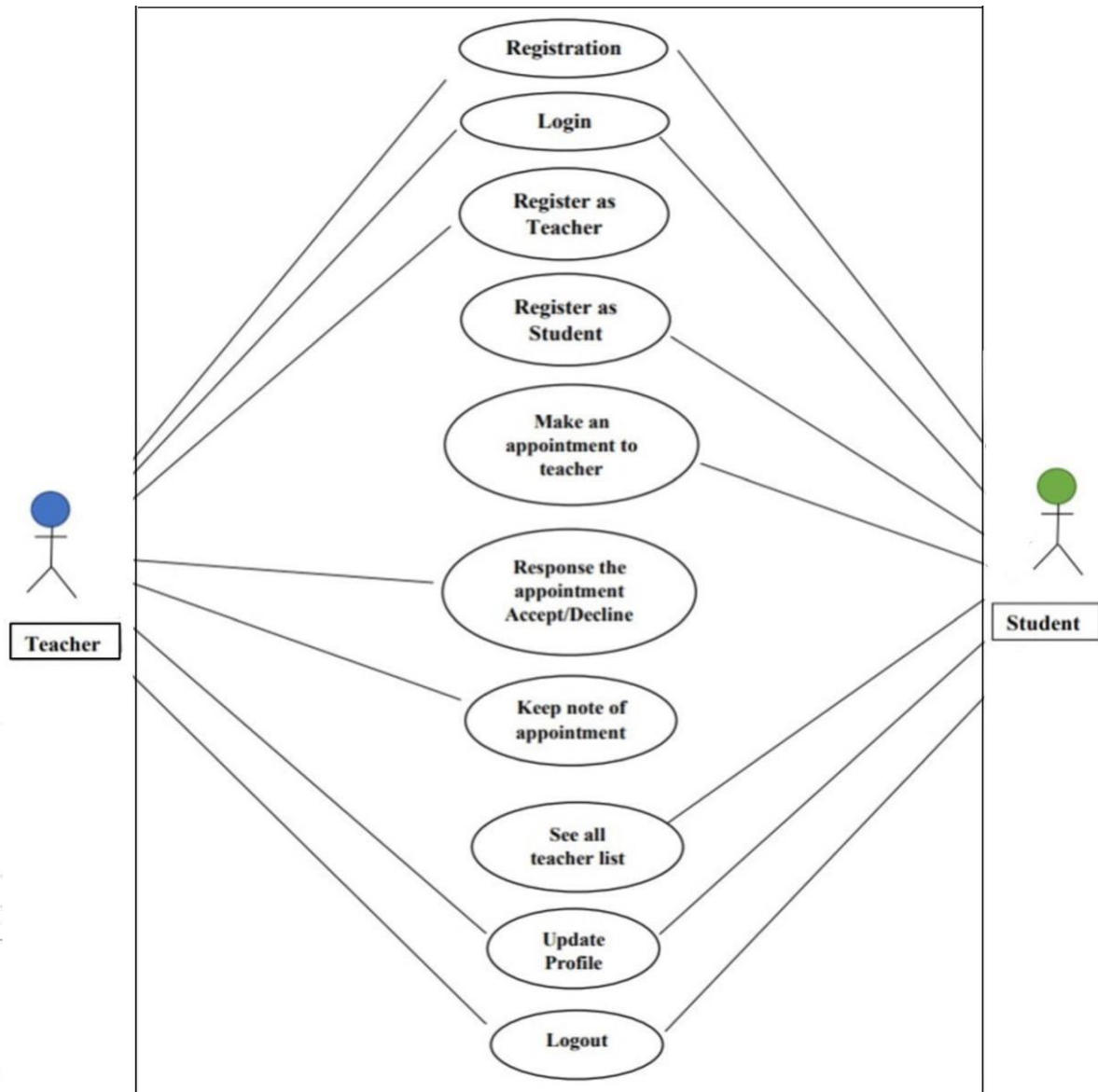


Figure 3.2: Use case illustration

There are two actors represented in this use case model diagram. Which is figure 3.2. One is a teacher, the other a pupil. Students and teachers must register with an email address and password. They get a confirmation email in the mail after registering. After confirmation, they must log in. For the first time, both the teacher and the student must register separately as a teacher and a student. Student can view the list of all teachers with

people are already enrolled in this system in detail. Then the student can schedule an appointment with a reason. Teacher gets notification after making an appointment. The student's home page will display all of the appointments they have set, along with their status (pending, accept, decline). Similar to how a teacher can get a list of all appointment requests with student information. Teachers will notify when they respond to an appointment, for as by accepting it or rejecting it with a reply message. The note section is where the teacher can record appointments. The technology allows both teachers and students to log out and update their profiles.

## CHAPTER 4 DESIGN SPECIFICATION

### 4.1 Front-end Design

➤ **Splash Screen:**

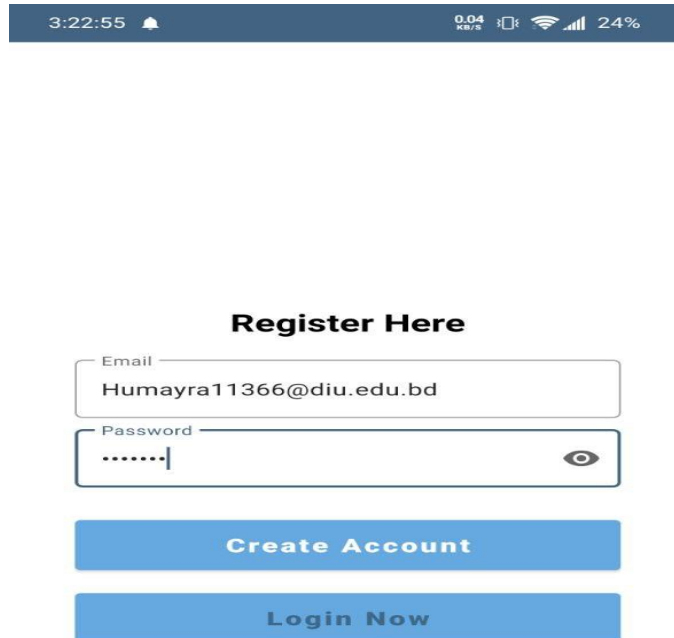
This is our application's splash screen which is figure 4.1. It appears when our application launches.



Figure 4.1: Splash screen

### ➤ Authentication Screen


Both types of users must first register in this system by providing an email address and a password. After registering, a mail-based confirmation email is delivered. Then, you must click this mail link to confirm the user's registration. User cannot log in without confirmation. Which is shown in figure 4.2 & figure 4.3 & figure 4.4.



3:22:55 0.04 KB/s 24%

### Register Here

Email  
Humayra11366@diu.edu.bd

Password  
.....| 

**Create Account**

**Login Now**

Figure 4.2: Register screen

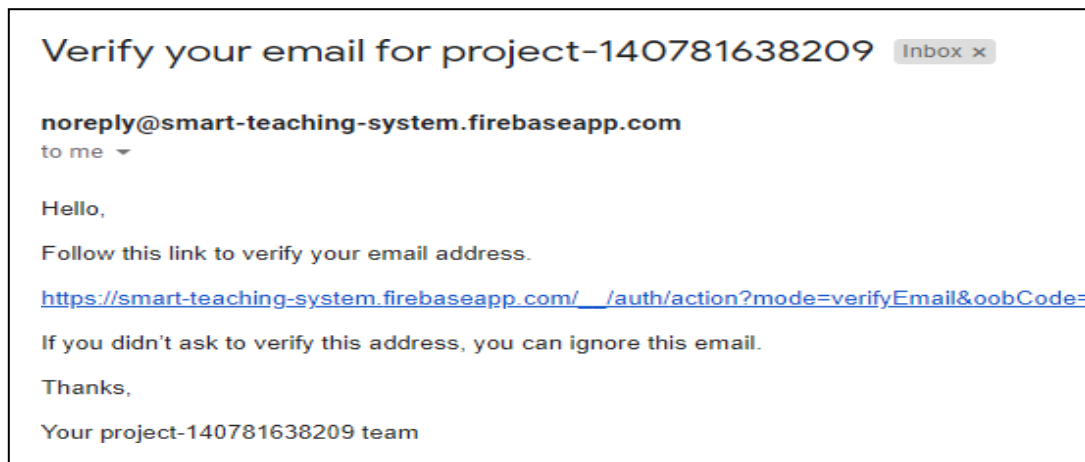


Figure 4.3: Registration verification mail

### Login Here

Figure 4.4: A user's login page

#### ➤ Registration Screen

User must re-register after logging in and providing specific information. Become a teacher or a student by registering. Both the teacher and the student must input their information during registration. We also keep track of the user role through this registration. Which is shown in figure 4.5& figure 4.6 & figure 4.7 & figure 4.8.

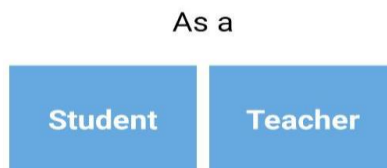



Figure 4.5: Select screen type for user

Student Register

\* Pick a image



\* Enter your name  
Humayra Rahman Silve

\* Enter student ID  
humayra15-11366@diu.edu.bd

\* Enter university email  
humayra15-11366@diu.edu.bd

\* Enter mobile number  
01627200545

\* Dept  
CSE

CSE

\* Enter your section  
A

Register

Figure 4.6: Screen to register students

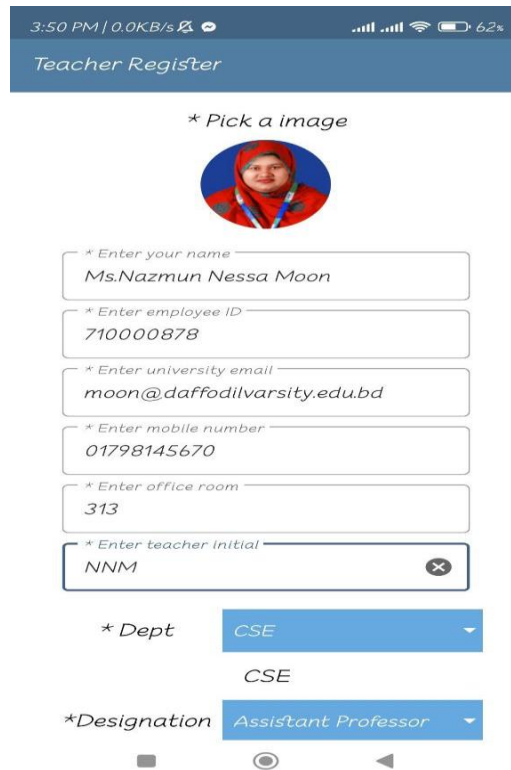


Figure 4.7: Screen 1 of the instructor register

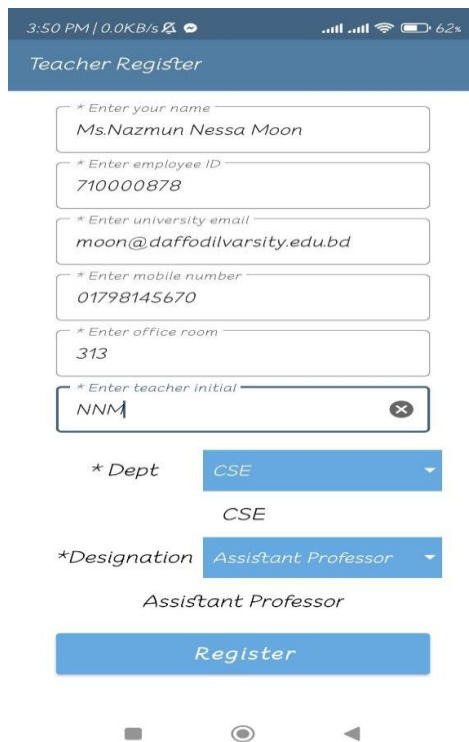


Figure 4.8: Screen 2 of the instructor register



## ➤ Student activities

The student can view a list of all the appointments they have already requested, along with the status of each one, on their home page. Students can search for any appointment by the teacher's name or first initial. When an appointment is accepted or declined, the student may also delete the appointment. But you cannot erase a pending appoint. Which is shown in figure 4.9 & figure 4.10.

The screenshot shows a user profile for Humayra Rahman Silve (ID: 182-15-11366) at the top. Below it is a 'Counseling List' section with a search bar labeled 'Search by name or initial'. The list contains six entries, each with a profile picture, name, title, department, and status:

Name	Title	Department	Initial	Status
Krishno Dey	Lecturer	Dept. of CSE	KD	Approve
Ms. Nazmun Nessa Moon	Assistant Professor	Dept. of EEE	NNM	Pending
Khairul Islam	Lecturer	Dept. of CSE	KI	Pending
md. sadekur Rahman	Professor	Dept. of CSE	msr	Pending
Ms. Nazmun Nessa Moon	Assistant Professor	Dept. of EEE	NNM	Approve
Dr. professor Tohuid Bhuiyan	Head and Professor	Dept. of CSE	TB	Approve

A blue circular button with a white plus sign is located at the bottom right of the list.

Figure 4.9: A student's main page

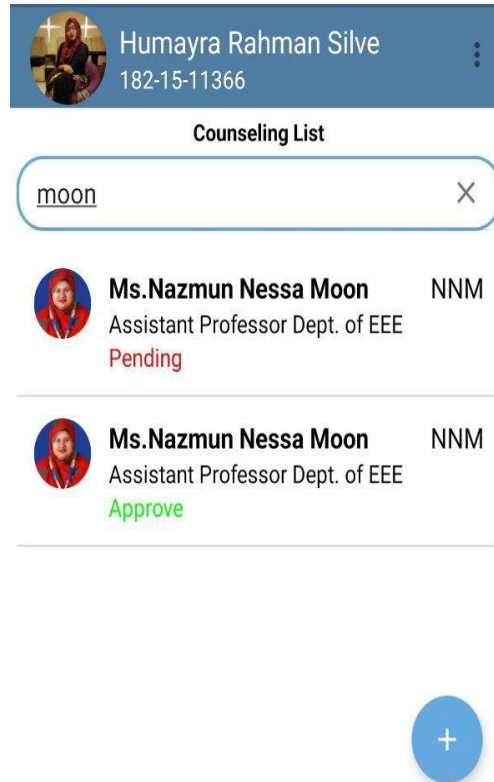


Figure 4.10: Inquiry by instructor name on the main page

The appointment's information and status are also visible to the student. The student's message will appear in a dialogue box with the status "Pending" when an appointment is still scheduled. However, whether an appointment is accepted or rejected, the student will get the teacher's message with the appropriate status, as seen below in figure 4.11 & figure 4.12.

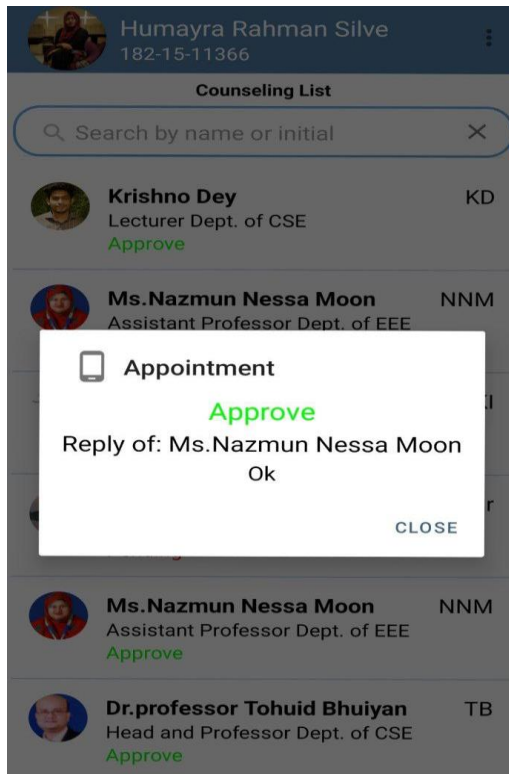


Figure 4.11: Consultation approval/rejection debate

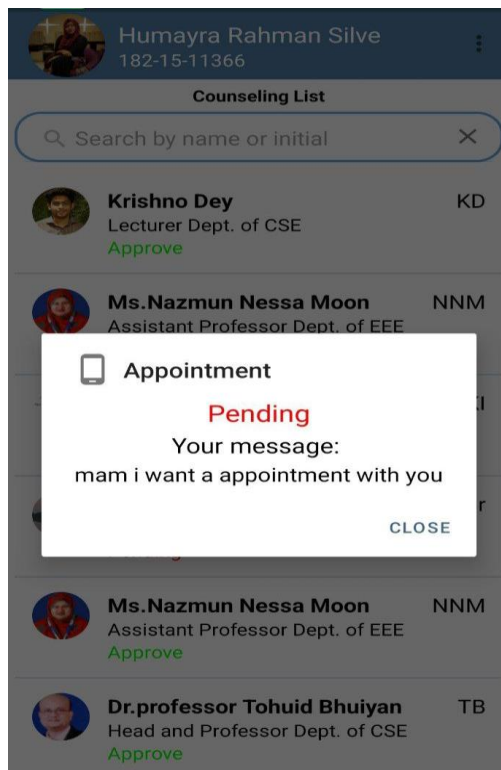


Figure 4.12: Awaiting scheduling conversation

Students have access to a list of all teachers who have signed up for this system. They may look for the instructor by their name or initial and view information about them such as name, email, phone, office address, and counseling hours. Additionally, students may submit an application for an appointment with justification. After that, the instructor is informed. As seen below in figure 4.13, figure 4.14 & figure 4.15.

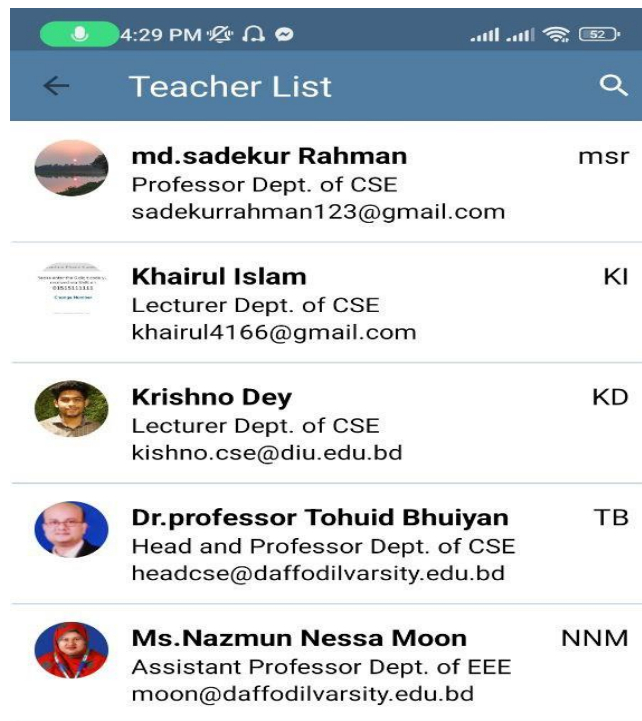


Figure 4.13: All of the instructors' displays listed

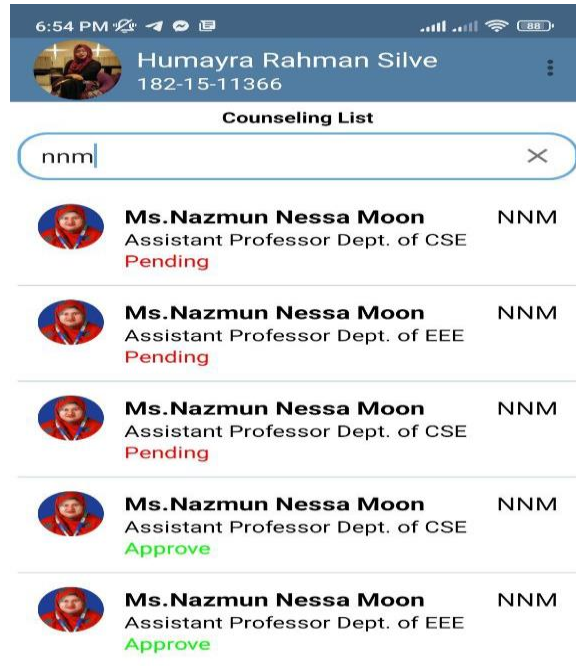


Figure 4.14 Splash display or instructor title inquiry



Figure 4.15: Instructor information as well as reservation scheduling display

Students have access to their profiles and can edit them. As seen below in figure 4.16 & figure 4.17.

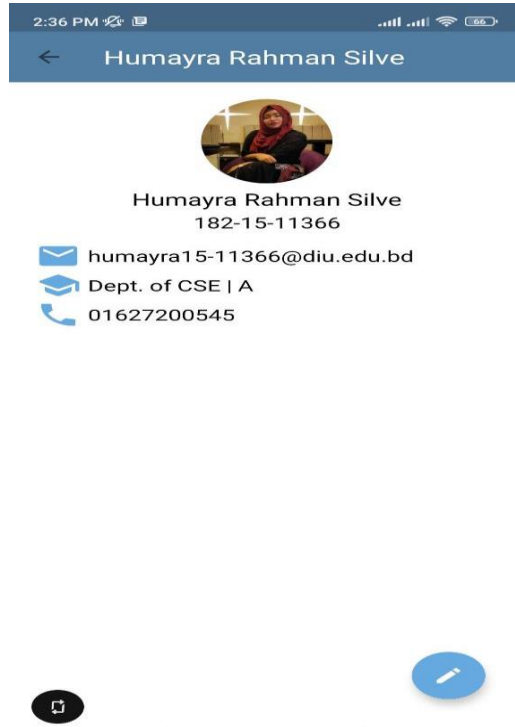


Figure 4.16: Display for student profiles

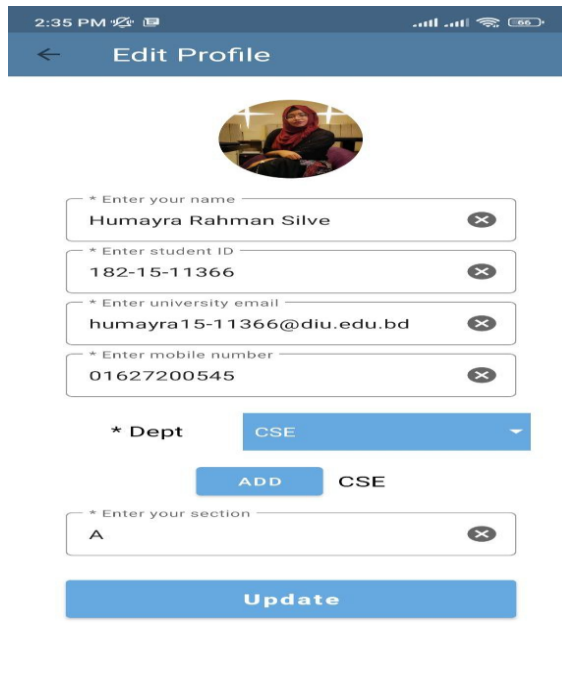


Figure 4.17: Display for student edit profile

➤ **Teacher activities**

On the teacher's home page, all of the students' appointments are listed. Additionally, you can search by student name to find any individual name. Only once an appointment has been accepted or declined by the teacher may the appointment be deleted. The student home page is comparable. As seen below in figure 4.18 & figure 4.19.

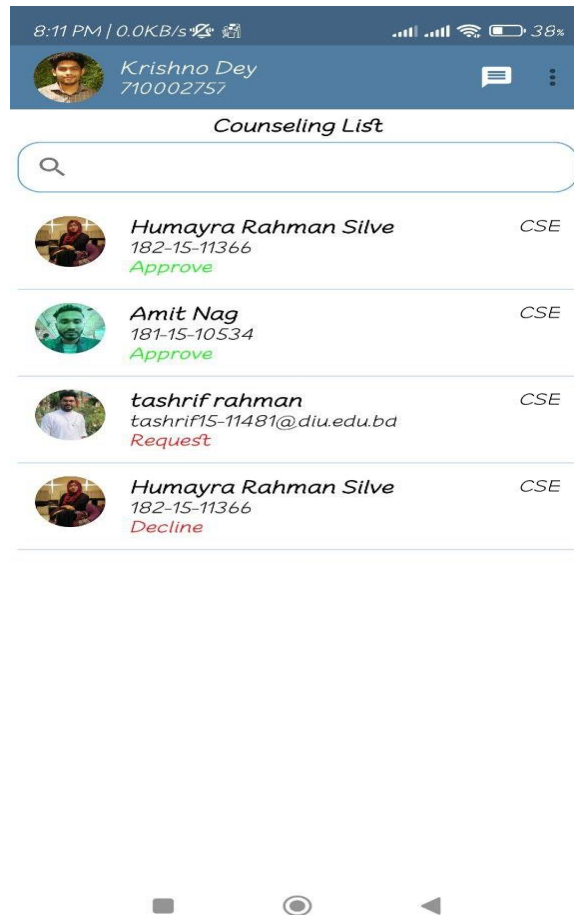


Figure 4.18: Landing page for teacher

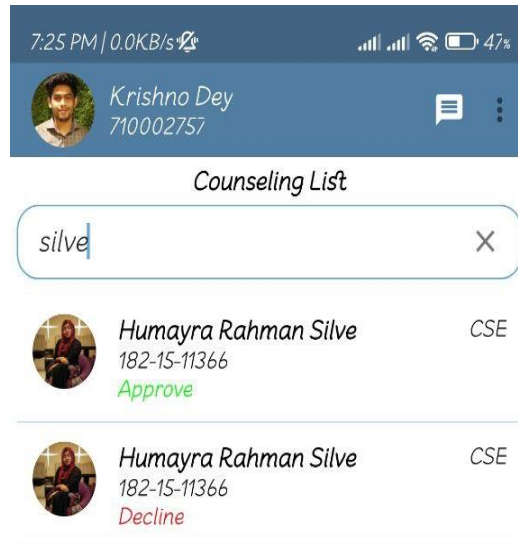


Figure 4.19: Search on the instructor's main display

A notice was sent to the teacher when a student made an appointment. It is the same for students. Student receives notified when teacher responds to a request. Teacher accesses appointment information together with student data. and give a message in return. The approve/decline option is disabled once an appointment has been accepted or rejected. As seen below in figure 4.20, figure 4.21 & figure 4.22.



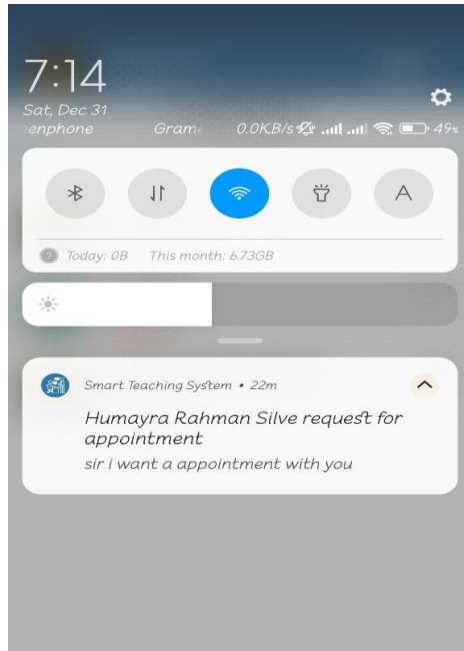


Figure 4.20: Receiving instructor note

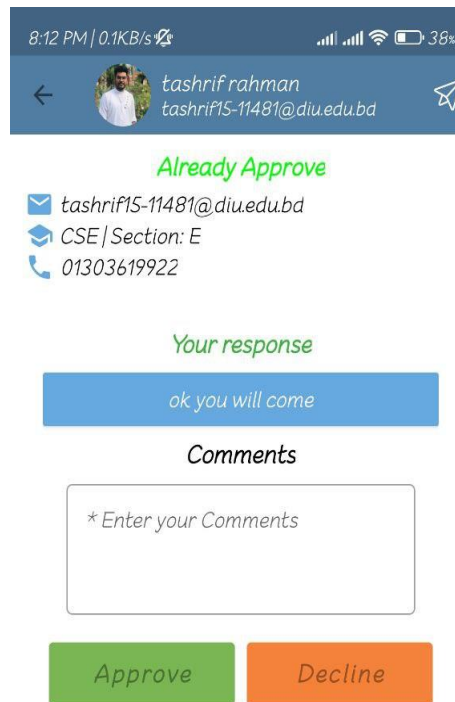


Figure 4.21: Consultation approval display

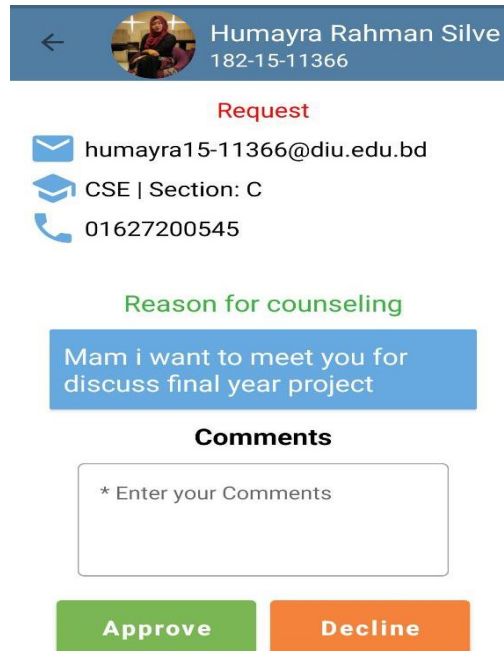


Figure 4.22: Consultation display for requests

The appointment can be noted by the teacher in the Note item page. As seen below in figure 4.23 & figure 4.24.

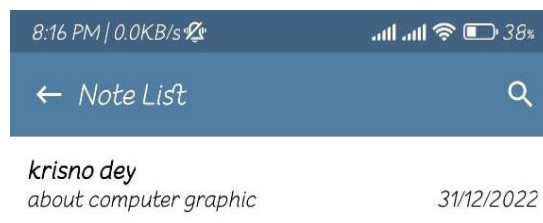


Figure 4.23: Note sheet for instructors

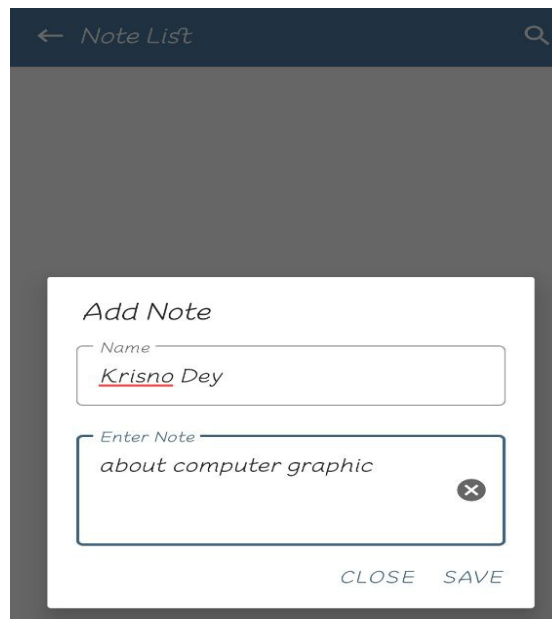


Figure 4.24: Window to add notes

The profile and profile screen are visible to the teacher. Teachers can edit their counseling availability in their profiles. A teacher's profile can also be updated. Shown in figure 4.25, figure 4.26, figure 4.27, figure 4.28.

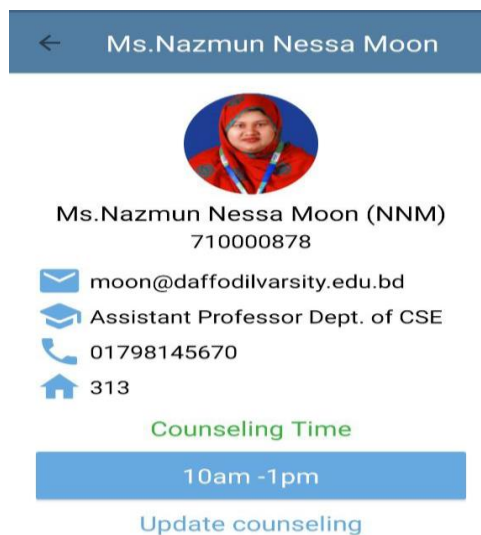


Figure 4.25: View of an instructor's profile

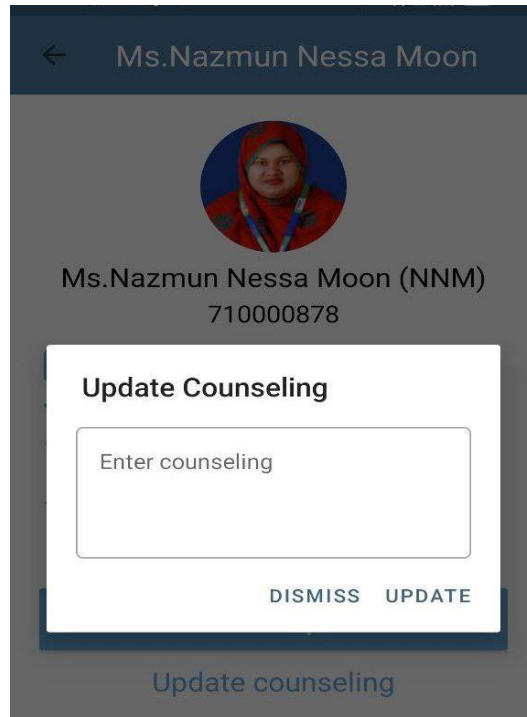


Figure 4.26: Instructor upgrade consultation display

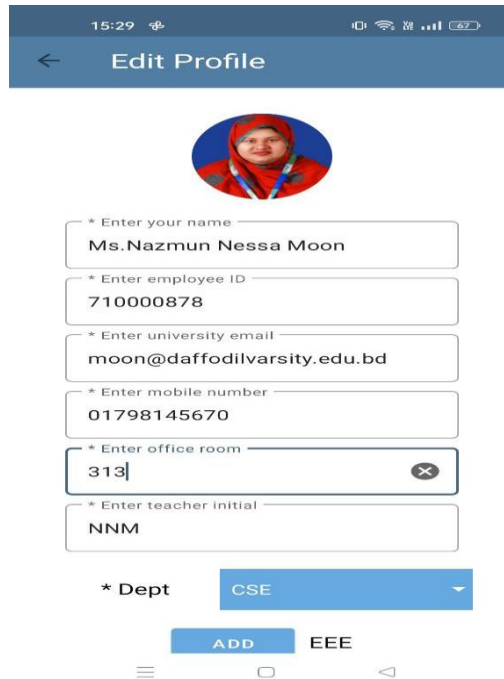


Figure 4.27: Display 1 of the instructor profile upgrade

15:29 67%

← Edit Profile

\* Enter your name  
Ms. Nazmun Nessa Moon

\* Enter employee ID  
710000878

\* Enter university email  
moon@daffodilvarsity.edu.bd

\* Enter mobile number  
01798145670

\* Enter office room  
313

\* Enter teacher initial  
NNM

\* Dept CSE

ADD EEE

\* Designation Assistant Professor

ADD Assistant Professor

Figure 4.28: Display 2 of the instructor profile upgrade

## 4.2 Back-end Design

In the project's back end, there are certain file structures. They are Gradle Script, Resource, Manifest, Java/Kotlin, and Resource. Layout, drawable, mipmap, and values are included in the resource file. Below is the added file structure for the backend. As seen below in figure 4.29, figure 4.30, figure 4.31, figure 4.32, figure 4.33, figure 4.34.

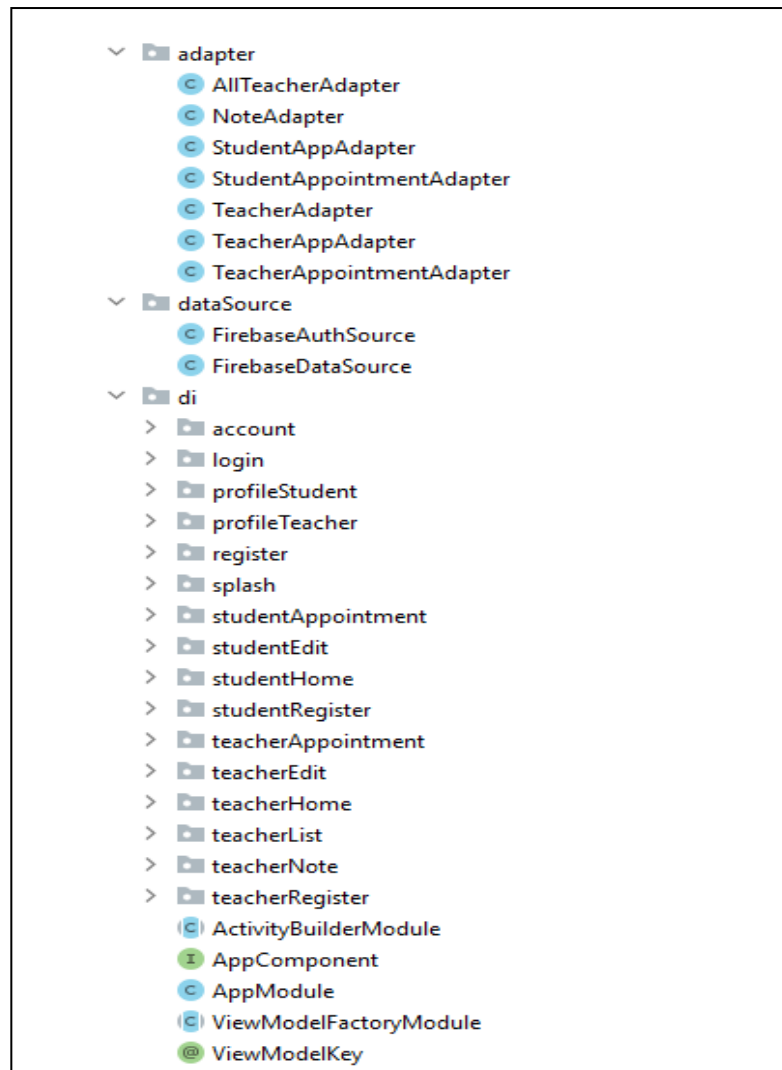


Figure 4.29: Back-end layout that revealed a partially separated Java file. Data source, adapter, and di all refer to the dependency injection bundle

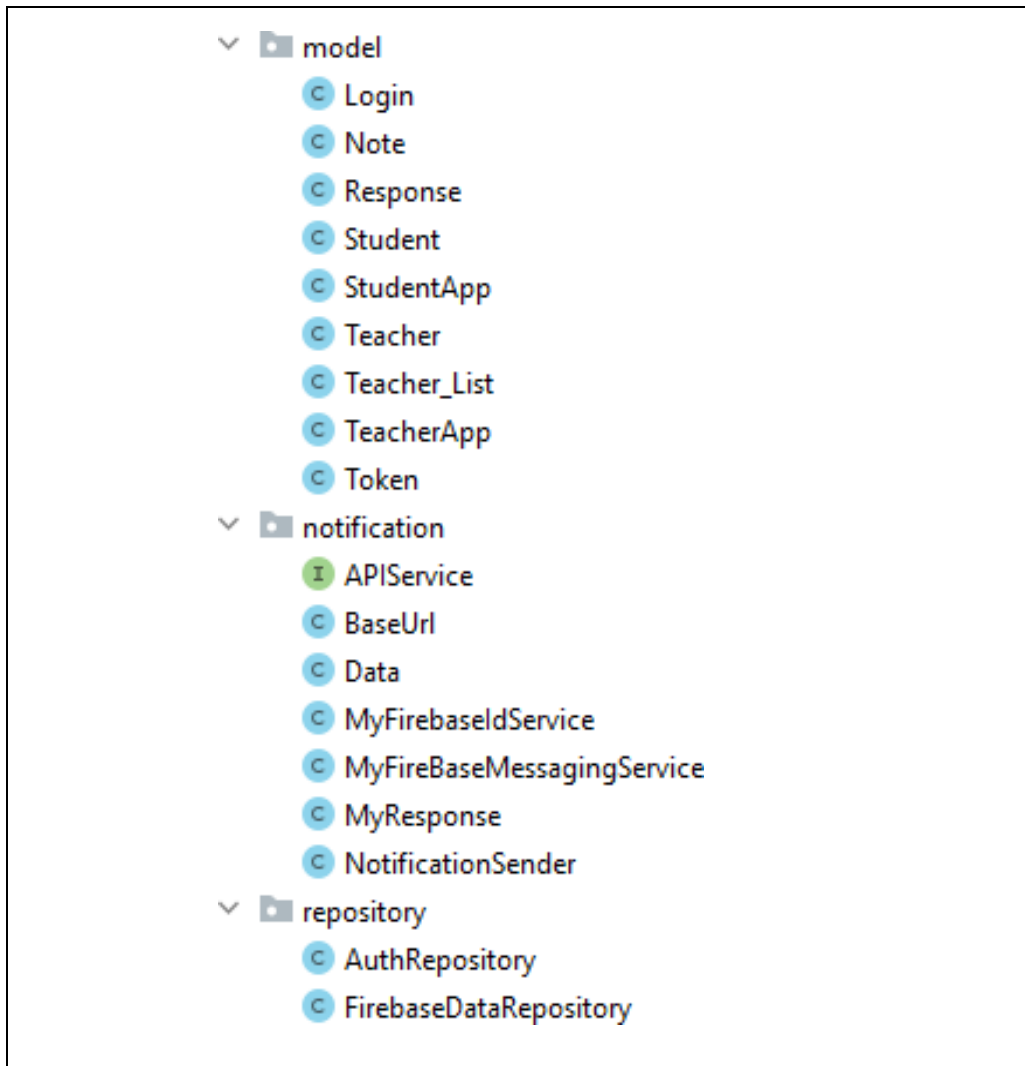


Figure 4.30: Back-end layout that revealed a partially separated Java file. Model, notice, and source bundle are present

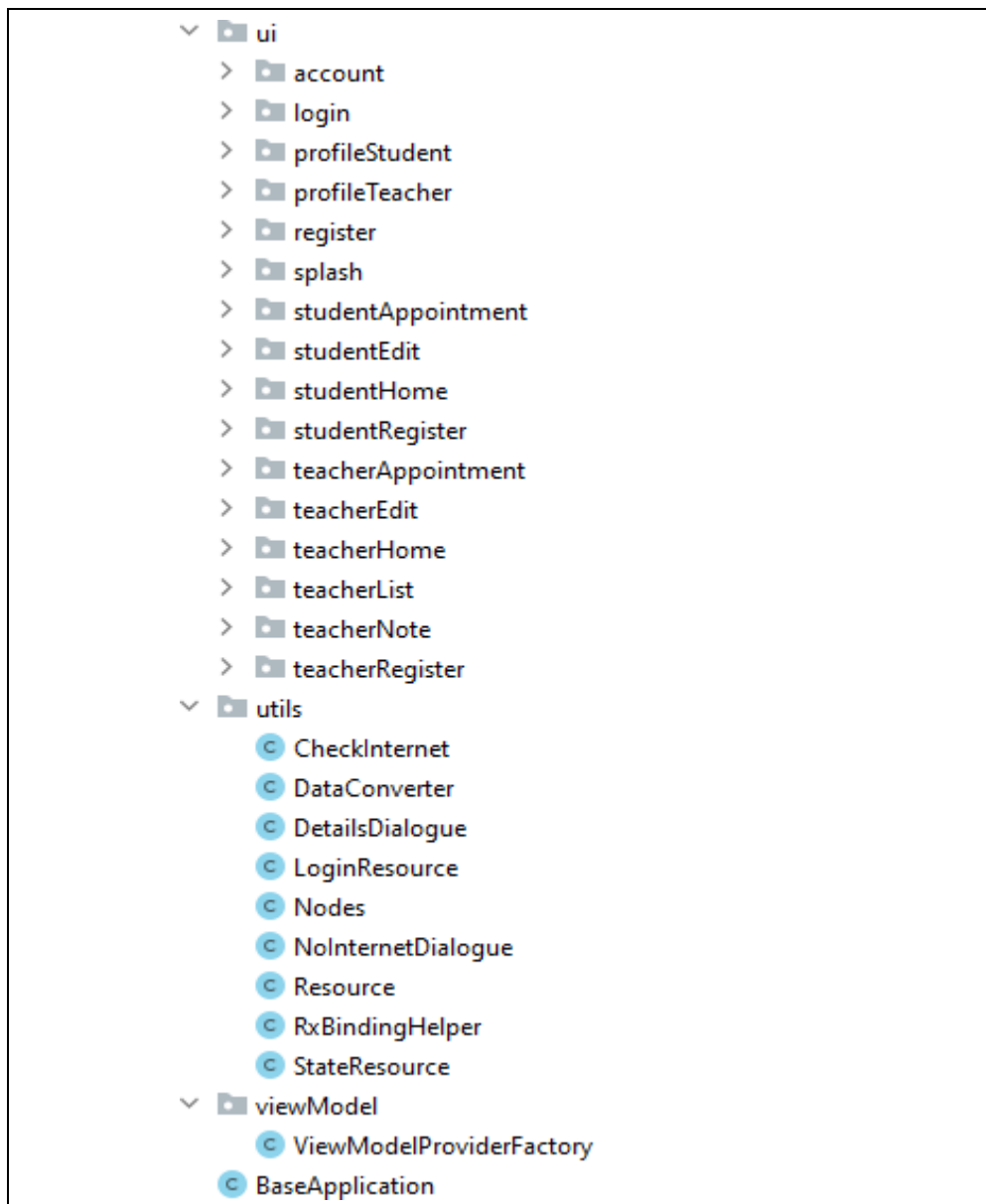


Figure 4.31: All this belongs to the java file. which display the UI, features, and models bundle



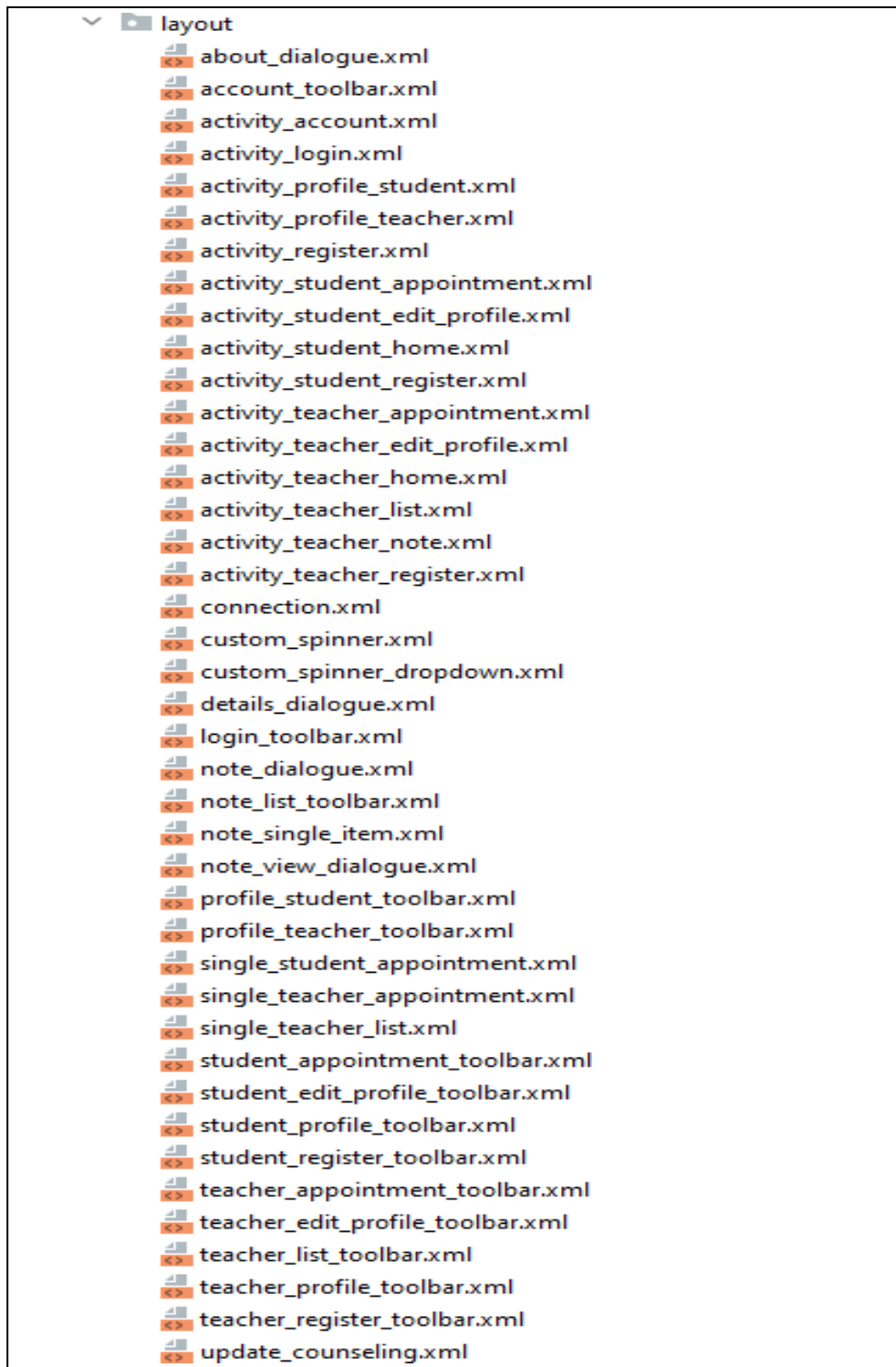


Figure 4.32: Includes layout template. All front-end design for an Android platform is contained in the layout file

```

dependencies {
    implementation 'androidx.appcompat:appcompat:1.2.0'
    implementation 'com.google.android.material:material:1.3.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
    implementation 'com.google.firebase:firebase-auth:20.0.1'
    implementation 'com.google.firebase:firebase-firestore:22.0.1'
    implementation 'com.google.firebase:firebase-storage:19.2.1'
    implementation 'com.google.android.gms:play-services-auth:19.0.0'
    implementation 'com.google.firebase:firebase-messaging:21.0.1'
    testImplementation 'junit:junit:4.+'
    androidTestImplementation 'androidx.test.ext:junit:1.1.2'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'
    // FirebaseUI for Cloud Firestore
    implementation 'com.firebaseui:firebase-ui-firestore:7.1.1'
    //circle image View
    implementation 'de.hdodenhof:circleimageview:3.1.0'
    // Reactive Streams (convert Observable to LiveData)
    def reactivestreams_version = "1.1.1"
    implementation "android.arch.lifecycle:reactivestreams:$reactivestreams_version"
    // Rx Java & Rx Android
    implementation 'io.reactivex.rxjava3:rxandroid:3.0.0'
    implementation 'io.reactivex.rxjava3:rxjava:3.0.0'
    //gilde
    implementation 'com.github.bumptech.glide:glide:4.11.0'
    annotationProcessor 'com.github.bumptech.glide:compiler:4.11.0'

    def lifecycle_version = "2.2.0"
    // ViewModel
    implementation "androidx.lifecycle:lifecycle-viewmodel:$lifecycle_version"
    // LiveData
    implementation "androidx.lifecycle:lifecycle-livedata:$lifecycle_version"
    // Lifecycles only (without ViewModel or LiveData)
    implementation "androidx.lifecycle:lifecycle-runtime:$lifecycle_version"
    // Annotation processor
    annotationProcessor "androidx.lifecycle:lifecycle-compiler:$lifecycle_version"
}

```

Figure 4.33: All required libraries and third-party libraries were included by Gradle Script-1

```

// Dagger
implementation 'com.google.dagger:dagger:2.33'
annotationProcessor 'com.google.dagger:dagger-compiler:2.33'
// Dagger Android
implementation 'com.google.dagger:dagger-android:2.33'
implementation 'com.google.dagger:dagger-android-support:2.33' // if you use the support libraries
annotationProcessor 'com.google.dagger:dagger-android-processor:2.33'

//Rx Binding
implementation 'com.jakewharton.rxbinding4:rxbinding:4.0.0'
implementation 'com.jakewharton.rxbinding4:rxbinding-material:4.0.0'
//Compressor
def compressor = "2.1.0"
implementation "id.zelory:compressor:$compressor"
//crop image...
implementation 'com.theartofdev.edmodo:android-image-cropper:2.8.+'

// Room database
def room_version = "2.2.5"

implementation "androidx.room:room-runtime:$room_version"
annotationProcessor "androidx.room:room-compiler:$room_version"
//retrofit
implementation 'com.squareup.retrofit2:retrofit:2.9.0'
// Gson
implementation 'com.google.code.gson:gson:2.8.6'
implementation 'com.squareup.retrofit2:converter-gson:2.9.0'

// swif refresh
implementation "androidx.swiperefreshlayout:swiperefreshlayout:1.1.0"
}

```

Figure 4.34: All relevant libraries and third-party libraries were included in Gradle Script-2

### **4.3 Implementation Requirements**

- Database: Firebase Cloud Fire store, Firebase Storage
- Dependency Injection: Dagger (2,3)
- Retrofit: for notification response
- Rx Java
- MVVM Architecture Pattern
- View Model & Live Data
- Android X
- Language: Java

## CHAPTER 5

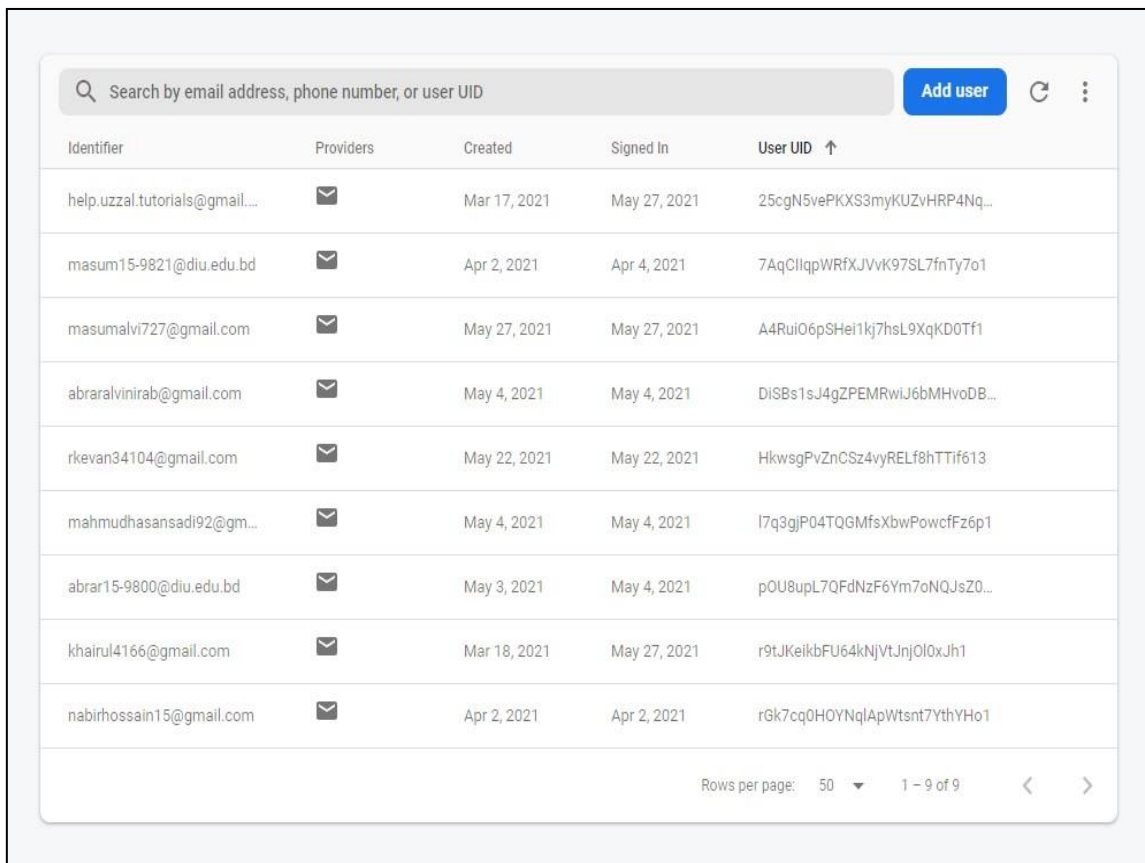
# IMPLEMENTATION AND TESTING

### 5.1 Implementation of Database

We used Firebase database for our project. The user may be managed and controlled with ease. There are several authentication methods available in Firebase, including Facebook, Google Account, Phone, Email Password, and Yahoo. We employ a mechanism for email password authentication. As seen below in figure 5.1.

➤ **User authentication**

A distinct user id is produced by this database when a user registers in this system. Because of this, it is simple to use and allows for user identification.



Identifier	Providers	Created	Signed In	User UID ↑
help.uzzal.tutorials@gmail...	📧	Mar 17, 2021	May 27, 2021	25cgN5vePKXS3myKUZvHRP4Nq...
masum15-9821@diu.edu.bd	📧	Apr 2, 2021	Apr 4, 2021	7AqCllqpWRfXJVvk97SL7fnTy7o1
masumalvi727@gmail.com	📧	May 27, 2021	May 27, 2021	A4RuiO6pSHei1kj7hsL9XqKD0Tf1
abraralvinirab@gmail.com	📧	May 4, 2021	May 4, 2021	DISBs1sJ4gZPEMRwiJ6bMHvoDB...
rkevan34104@gmail.com	📧	May 22, 2021	May 22, 2021	HkwsgPvZnCSz4vyRELf8hTTif613
mahmudhasansadi92@gm...	📧	May 4, 2021	May 4, 2021	l7q3gjP04TQGMfsXbwPowcfFz6p1
abrar15-9800@diu.edu.bd	📧	May 3, 2021	May 4, 2021	pOU8upL7QFdNzF6Ym7oNQJsZ0...
khairul4166@gmail.com	📧	Mar 18, 2021	May 27, 2021	r9tJKeikbFU64kNjVtJnjOl0xJh1
nabirhossain15@gmail.com	📧	Apr 2, 2021	Apr 2, 2021	rGk7cq0HOYNqlApWtsnt7YthYHo1

Figure 5.1: Firebase authenticate users

## ➤ **Firestore Storage**

Images for user profiles are saved in Firestore storage. Each user's unique ID is saved alongside this picture. As seen below in figure 5.2.

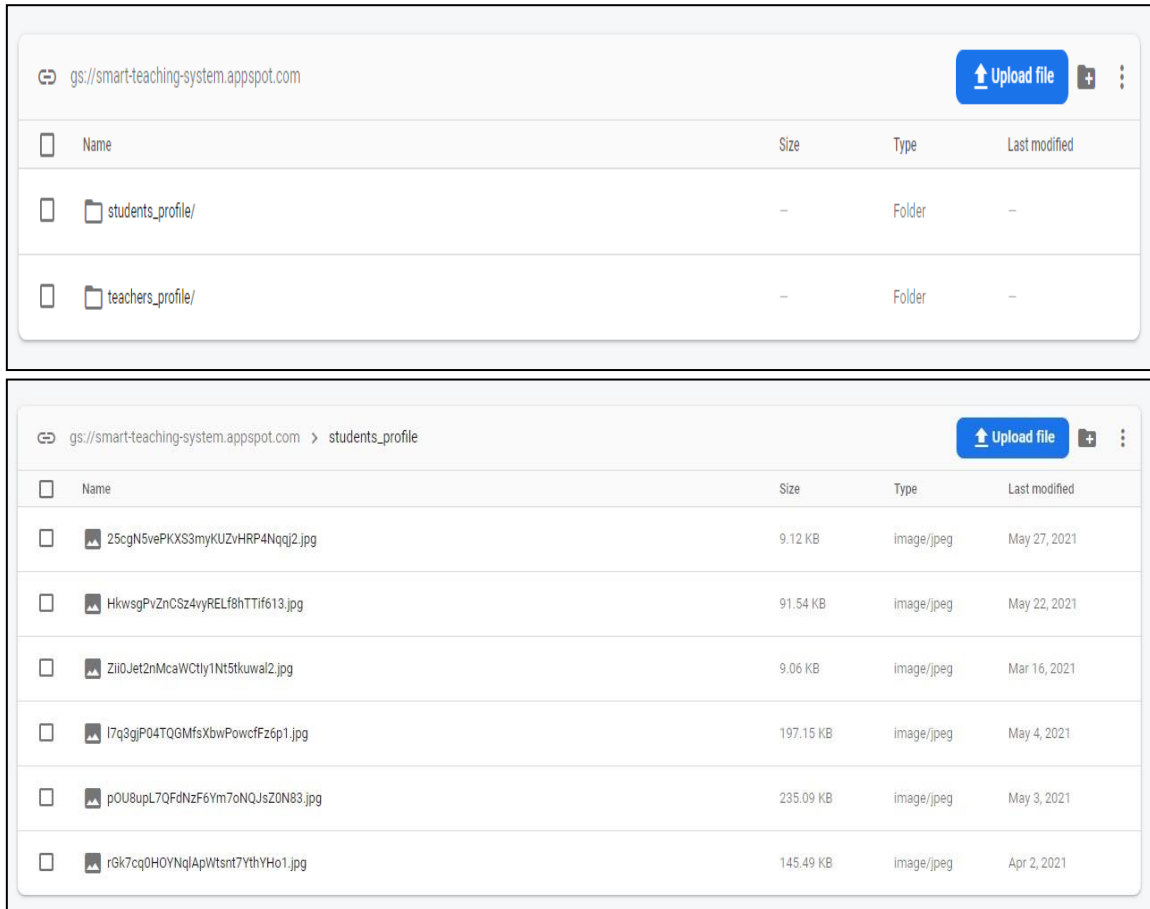


Figure 5.2: Firestore Storage

## ➤ Firebase Cloud Fire store

As an update to Firebase Real Time Database, we used Firebase Cloud Fire store. Complex queries may be executed with ease. We set up a separate directory to manage all of the data for this application. As seen below in figure 5.3.

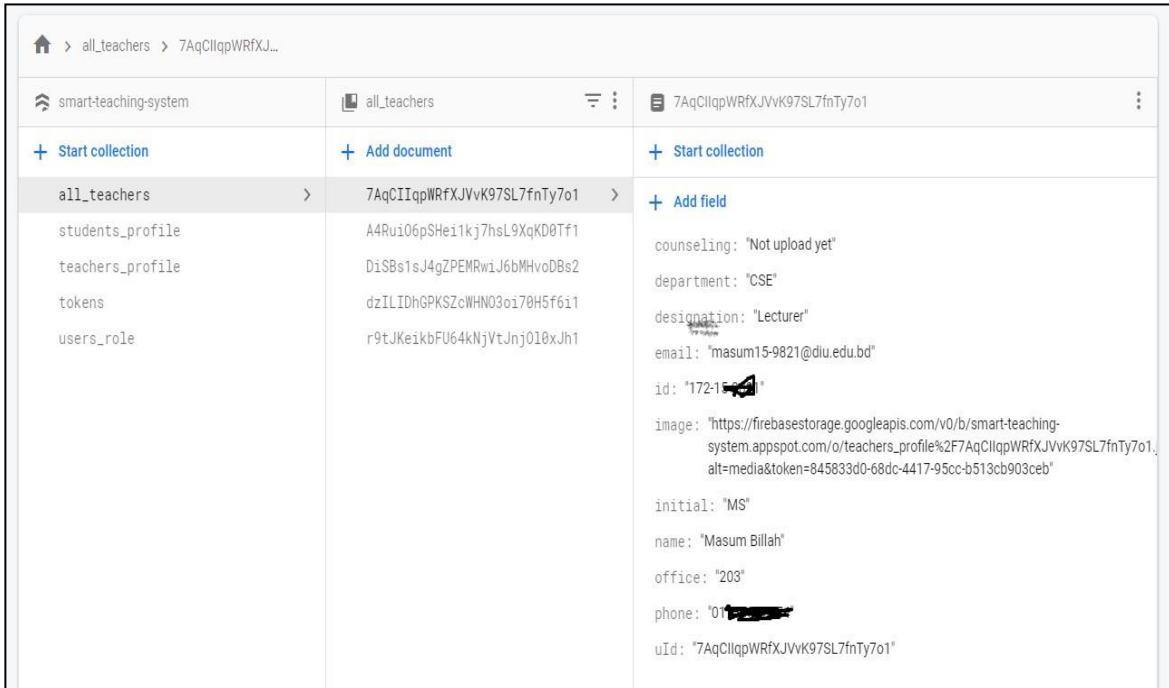


Figure 5.3: All teacher information is kept in the fire store directory "all teacher."

The technique for storing data in the folders "student profile" and "teacher profile" is same here. As seen below in figure 5.2.

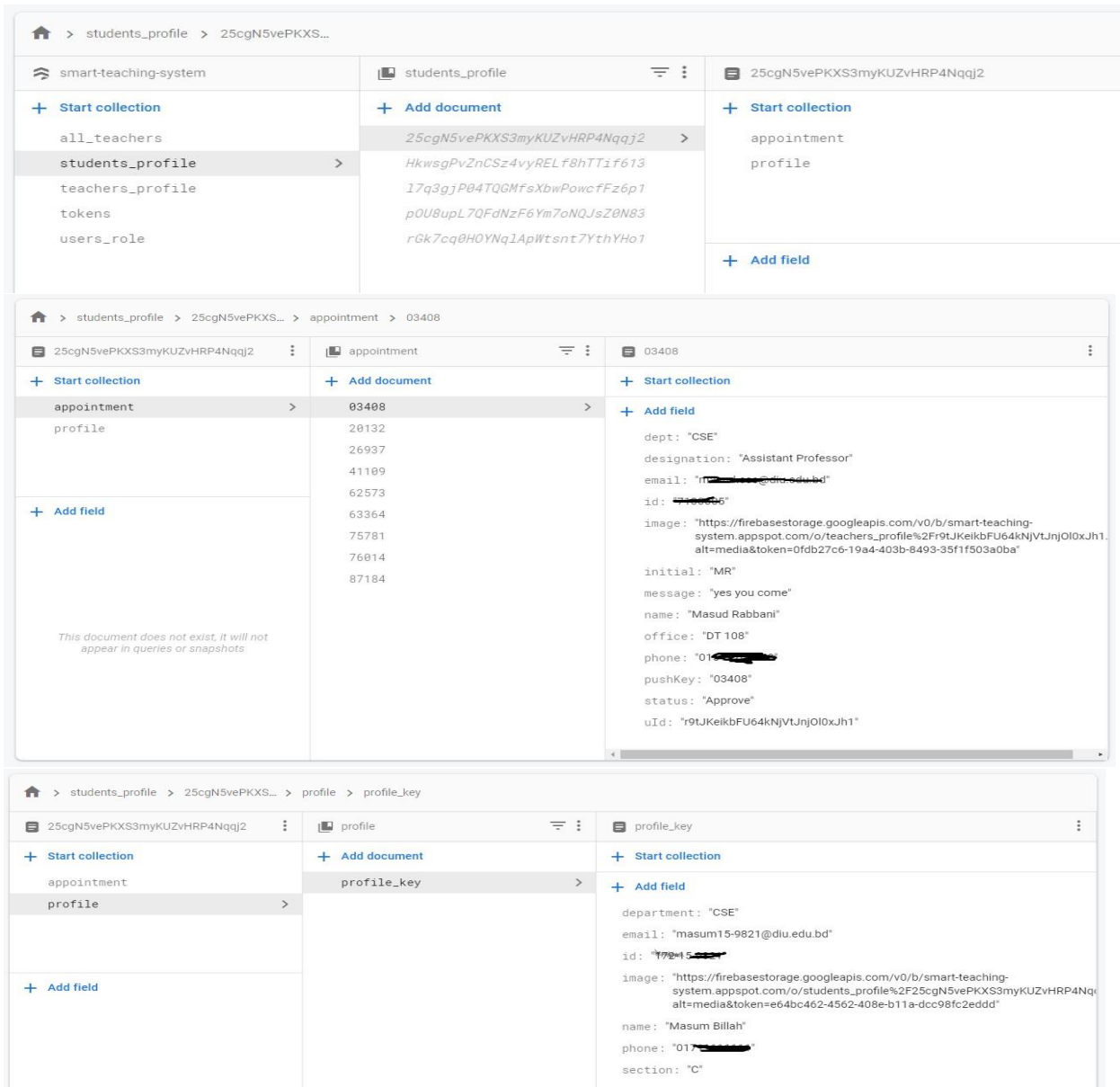


Figure 5.4: All of the data for student profiles and appointment information is kept in the "student profile" fire store directory.



All user device access tokens should be stored in the "tokens" directory. It constantly updates based on various devices. It facilitates sending notifications to users who are teachers and students. A user may be a teacher or a student, and the "user role" directory stores all user types. As seen below in figure 5.5 & figure 5.6.

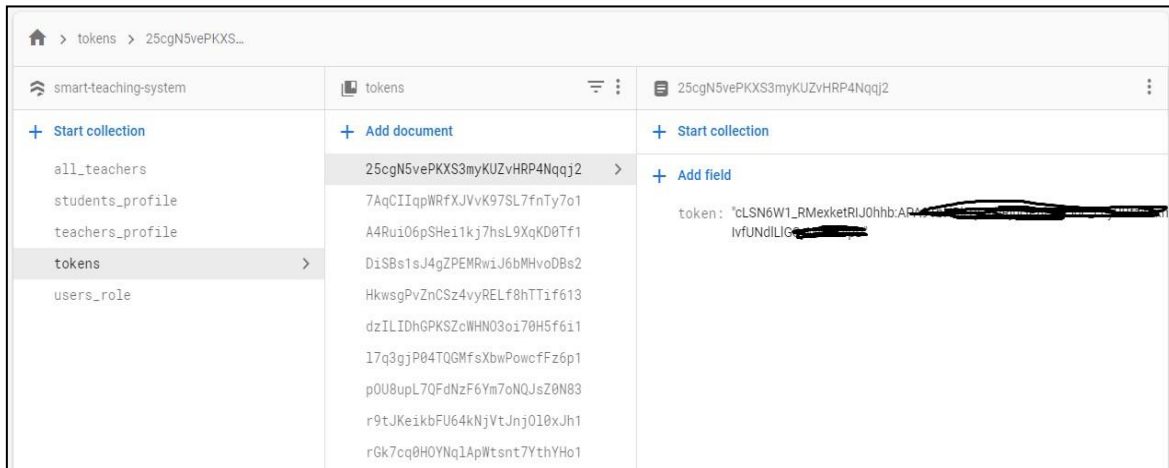


Figure 5.5: All user device tokens are stored in the fire store directory "tokens."

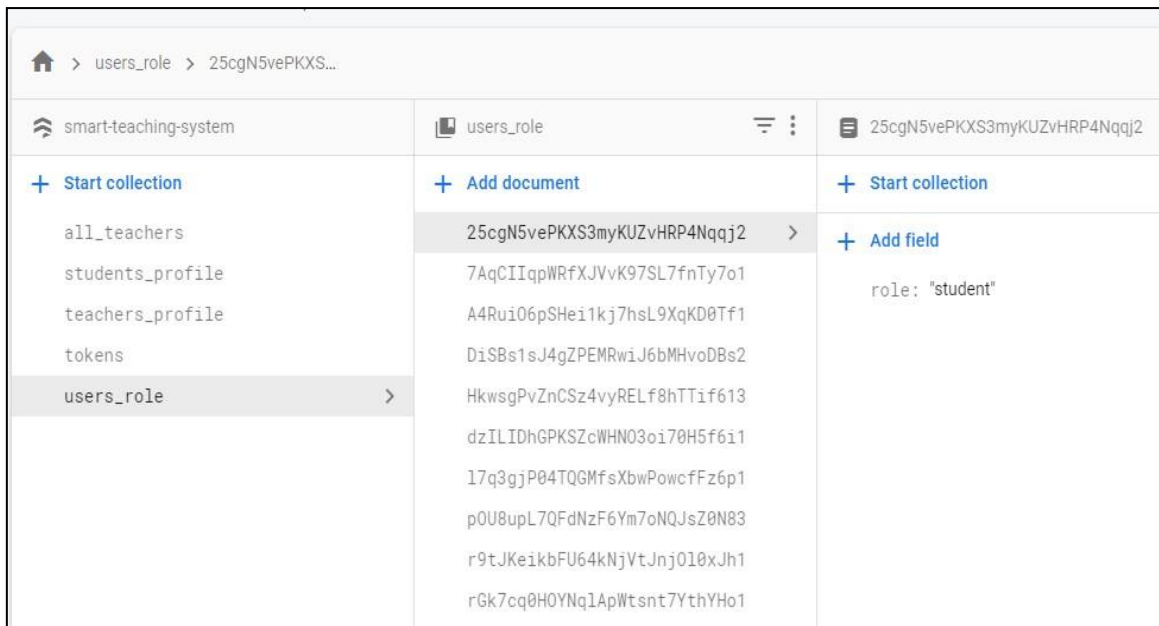


Figure 5.6: All user roles (teacher/student) are stored in the fire store directory "user role."

## 5.2 Front-end design implementation

An application's user interface should be visually appealing. First, we paid attention to the UI design. We maintain our design in a suitable manner so that it will remain consistent across all device sizes. We manage devices with various screen sizes.

## 5.3 Implementation, results, and reporting of testing

We test our app using various input devices and input scenarios. The majority of our applications have produced successful results.

TABLE 5.1: Testing result

Test case	Test input	Expected outcome	Obtained outcome	Passed/Failed	Testing Period
Install	Test in different version of android phone	Installed	Installed	Passed	25/06/22
Sign in	Not signing in without signing up	not sign in	not sign in	Passed	25/06/22
Register	Enter invalid email	NOT Summarize	NOT Summarize	Passed	25/06/22
Sign in	Sign in without mail verification	not sign in	not sign in	Passed	25/06/22
Password	incorrect password	Invalid password	Invalid password	Passed	25/06/22
Sign in	After all registration	Go to home page	Go to home page	Passed	25/06/22
Registration	Register without upload image	Forced to upload image	Say to upload image	Passed	25/06/22
Submit appointment	Submit appointment to teacher without reason	Forced to enter reason	Say to enter reason	Passed	25/06/22

Delete	Get remove of the pending appointment for student	not remove	not remove	Passed	25/06/22
Delete	Delete the request appointment without accept/decline	Not delete	Not delete	Passed	20/06/22
Authentication	If user already logged in not need to login when open this application again	Go to home page without login	Go to home page without login	Passed	20/06/22
signing out	after signing out	Log in now page	Log in now page	Passed	25/06/22
Internet Connection	the absence of the internet connection	Null internet connection message	Null internet connection message	Passed	25/06/22

## **CHAPTER 6**

### **ETHICAL ASPECT**

#### **6.1 Ethical impacts on society**

The question of appointment systems comes up often in daily life. However, in the present period, this appointment procedure is carried out by modern technology, such as phone calls, emailing, or any other system. We can observe that our nation uses a variety of appointment systems, such as doctor-patient appointments. They occasionally submit an application to be appointed. However, there is no application process that allows a student to request a faculty appointment for educational purposes. In general, a school has a large number of students. And there are many students in each department of a university. In this project, we create an Android application that allows a student to directly arrange appointments with teachers. In this one-to-one communication method, the student can speak with the teacher immediately. Students may access all of the faculty's contact information, including phone numbers, email addresses, offices, and departments, through this app. However, before using this system, instructors and students must both register. It supports instructors and students in keeping appointments in an efficient and hassle-free manner. **SMART COUNSELLING SYSTEM** is a mobile application for Android that allows users to schedule and communicate with teachers for counseling sessions. In this endeavor, our primary goal was to secure appointments from university or college faculty members. Students are the ones making this appointment. Faculty members and students share a single application. In this project, in order to communicate with one another, both students and Faculty at an institution are required to sign up for this system. Students can use a name search to look for a certain professor. and can send an appointment with the appropriate justification.

## **6.2 Ethical impacts on environment**

The major goal is to create a user-friendly application that allows students and instructors to connect easily.

It will serve as a link between the teacher and the students. A communication link that keeps track of the teacher's schedule will enable the student to efficiently contact the teacher without disrupting the faculty room. Cardinality-based one-to-one contact between the instructor and the student

- Teachers are accessible to students and may be reached easily.
- The teacher can easily keep up with every pupil.
- The teacher will be informed of the student's arrival, any issues they may have with the course, and their anticipated arrival time.
- How many pupils had time and what issues they solved may be readily recorded in a database.
- The teacher will find it simple to inform the class of his holidays and activities.
- To facilitate contact between teachers and students.
- Receive faculty appointment schedule updates in real-time.

## **6.3 Ethical aspects**

It's actually not a terrible idea to require a guidance and counseling course of a student enrolled in teacher education. Having some exposure to both guidance and counseling course work would be extremely beneficial for a prospective teacher going for their certification. In your role as a teacher, you will frequently be questioned by parents and students regarding the courses that their children should enroll in, as well as in circumstances where a student will seek your assistance with a personal issue. Knowing when to recommend a kid to school counselors or psychologists and when your assistance as an instructor will be sufficient is crucial. Keep in mind that every member of the school staff is there to support students in realizing their full academic,

professional, and psychological potential. Teachers can assist with these chores, but it's crucial to know when to offer advice and when to refer.

College is crucial for a student to communicate with professors in a setting like a university. We continue to communicate by phone, SMS, and email. However, there are times when this approach cannot be used to get to the faculty. Sometimes a student is unaware of the availability of a faculty member in their office. due to the monthly or semester updates to their counseling schedule. When a student visits a faculty member's office, they do not schedule a meeting since they are unaware of their availability. We believe that if there is a platform where students can find all of a faculty member's information, such as email, telephone, counseling routine, workplace, and other details, they can schedule appointments with faculty members for counseling. This is based on our observations of real-life situations. Faculty may also comment on this appointment. Additionally, this technology enables communication between instructors and students on one platform. This is how we want to make this project using this style of thinking.

## **6.4 Sustainability plans**

In this section, we outline several applications of appointment or teacher-student contact in our own nation or in another country, or we represent some work in this topic. The authors use a web-based appointment management system created in the.NET programming language to help students and professors schedule meetings in a productive and suitable manner at a university or institution. Tutors may schedule appointments using a GPS system through a web and mobile application-based approach. This approach allows them to monitor whether the tutor is doing their job correctly. a web-based system where students and teachers may identify themselves and set up appointments. Students and teachers can communicate by sending messages using this method. In this case, the initial piece of work is a web-based application that allows teachers and students to schedule meetings, but in our project, we'll construct a mobile-based application that is more streamlined than web applications.

The second is a GPS monitoring system that allows tutors to schedule appointments and keep track of their whereabouts in order to ensure that their work is being done

correctly. But upon registering, the instructor enters their office address and room number to our system in order to avoid adding a GPS system. The third has some similarities to our project but also some differences. In our system, teachers have the ability to record student appointments. Teachers get real-time appointment notifications, and students also receive real-time notifications when teachers respond to notifications.

## **CHAPTER 7**

### **CONCLUSION AND FUTURE SCOPE**

#### **7.1 Discussion and Conclusion**

Our application is now finished. We anticipate that continuing their counseling communication will be extremely beneficial to the student and instructor. Today, applications for mobile devices are becoming more and more popular than those for other platforms. We thus assumed that all students and teachers would be provided access to this application. We anticipate that using this application will assist students learn more about the teachers they will be working with and simple to get in touch with the teacher they choose.

#### **7.2 Limitation**

Our application has some restrictions. As follows:

- If a student changes their information after submitting an appointment request, the instructor dashboard will not be updated.
- Similar to how if a teacher changes their information after making an appointment request, The list of teachers will be updated, but not the list of student home appointments.
- Sometimes there are issues with device tokens that prevent notifications from being sent.

#### **7.3 Scope for Further Developments**

- Add a real-time chat feature for students and teachers.
- Include a file-sharing option for students and teachers.
- To create a web portal and admin panel system to make this program more interactive.



## References

- [1] M.H.N.M Nasir, S.H. Hamid and H. Hasan, “WebMD and .NET Architecture for Developing Students Appointment Management System”, *Journal of Applied Science* 9(8): 1432- 1440, pp.87,2009 ISSN 1812-5654 © 2009 Asian Network for Scientific Information.
- [2] Al-Monte Vince M Calo, Jocelyn B Barbosa, Amparo V Dinagsao, “GrabTutor: A web and mobile application for tutor appointment system with GPS security feature”, Calo et al. / *Indian Journal of Science and Technology* 2020;13(19):1955–1964.
- [3] BELLO Ridwan Oluwaseun, OLUGBEBI Muyiwa, BABATUNDE Abdulrauph Olanrewaju, BELLO Bashir Omolaran, BELLO Shakirat Iyabo, “Student-Teacher Online Booking Appointment System in Academic Institutions”, *Journal of Computer Science and Control Systems* Volume 9, Number 2, pp.15-18, November 2016.
- [4] Firebase database, available at<< <http://firebase.google.com/pricing/>>>, last accessed on 26 March 2022 at 12:10 PM.

# PLAGIARISM CHECK REPORT

## An Android Based Mobile Application For Smart Counseling System

### ORIGINALITY REPORT

<b>19%</b>	<b>19%</b>	<b>3%</b>	<b>%</b>
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

### PRIMARY SOURCES

<b>1</b>	<b>dspace.daffodilvarsity.edu.bd:8080</b> Internet Source	<b>14%</b>
<b>2</b>	<b>bellota.ele.uva.es</b> Internet Source	<b>1%</b>
<b>3</b>	<b>dspace.mit.edu</b> Internet Source	<b>&lt;1%</b>
<b>4</b>	<b>Yonghee Shin, Robert M Bell, Thomas J Ostrand, Elaine J Weyuker. "On the use of calling structure information to improve fault prediction", Empirical Software Engineering, 2011</b> Publication	<b>&lt;1%</b>
<b>5</b>	<b>repository.mines.edu</b> Internet Source	<b>&lt;1%</b>
<b>6</b>	<b>businessdocbox.com</b> Internet Source	<b>&lt;1%</b>
<b>7</b>	<b>cco-sj-2.cisco.com</b> Internet Source	<b>&lt;1%</b>