

**ADMISSION HELPER: A Mobile Application for Undergraduate Admission
Applicants**

BY

**SHAIFUL ISLAM
ID: 161-15-7644**

**MD. RAKIBUL ISLAM
ID: 161-15-7194**

**SAIKAT DAS
ID: 161-15-7649**

**SOHEL RIHAN OPU
ID: 161-15-7617**

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

Supervised by

Moushumi Zaman Bonny

Senior Lecturer
Department of CSE
Daffodil International University

Co-Supervised by

Mr. Ahmed Al Marouf

Lecturer
Department of CSE
Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

DECEMBER 2019

APPROVAL

This Project/internship titled “**ADMISSION HELPER: A Mobile Application for Undergraduate Admission Applicants**”, submitted by SHAIFUL ISLAM, ID No: 161-15-7644, MD. RAKIBUL ISLAM, ID No: 161-15-7194, SAIKAT DAS, ID No: 161-15-7649, SOHEL RIHAN OPU, ID No: 161-15-7617 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 7th December, 2019.

BOARD OF EXAMINERS



Dr. Syed Akhter Hossain
Professor and Head

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

Chairman



Md. Zahid Hasan
Assistant Professor

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

Internal Examiner



Sadekur Rahman
Assistant Professor

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

Internal Examiner



Dr. Dewan Md. Farid
Associate Professor

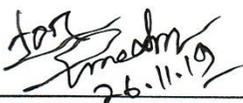
Department of Computer Science and Engineering
United International University

External Examiner

DECLARATION

I hereby declare that, this project has been done by me under the supervision of **MOUSHUMI ZAMAN BONNY**, **Department of CSE** at Daffodil International University. I also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

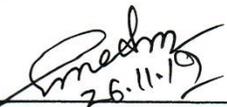
Supervised by:


26.11.19

Moushumi Zaman Bonny

Senior Lecturer

Department of CSE
Daffodil International University
Daffodil International University


26.11.19

Mr. Ahmed Al Marouf

Lecturer

Department of CSE
Daffodil International University
Daffodil International University

Submitted by:



Name: Rakibul Islam
Id Number: 161-15-7194
Department of CSE
Daffodil International University

Shaiful Islam

Name: Shaiful Islam
Id Number: 161-15-7644
Department of CSE
Daffodil International University

Saikat

Name: Saikat Das
Id Number: 161-15-7649
Department of CSE
Daffodil International University

Sohel Rihan Opu

Name: Sohel Rihan Opu
Id Number: 161-15-7617
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

I am grateful to omnipotent Allah, the most beneficial and merciful, for giving me the opportunity to complete my project work.

I would like to convey our earnest gratitude to our honorable thesis supervisor, **Moushumi Zaman Bonny**, for her persistent support, continuous steering, and invaluable advice during my project work. Her rigorous attitude towards technological research and her approach to solving unexpected problems have assisted me contriving my own way of performing project. Without her suggestion, encouragement and patience, I wouldn't be able to successfully complete this project. It has been a great pleasure and honor for me to have her as my supervisor.

I would like to thank the Department of Computer Science and Engineering, Daffodil International University, all the teachers.

Last but not least, I would like to express my deepest gratitude to Our family. Thanks for their generous and endless love to me. They are always with me when I go through tough situations and provide encouragement and support to me. Without their support, this project work would not happen. My project is dedicated to them.

ABSTRACT

Admission Test is the matter of concern for all the admitted candidates. So to minimize the sufferings of these aspirants, our admission test application made in small effort. At present, the use of smartphones can be noticed everywhere. A large part of our admissions candidates live in the village or do not have admissions coaching in the city, so they are often unaware of all university admission tests, question pattern, exact time of form filling and other important information. Therefore, if an admission examiner wants to install this application on his emperor's phone, he can benefit greatly through internet connection very easily, in a short time and with little effort. Our application has the advantage of almost all public and private information of Bangladesh, last year's question, university's question pattern and online test. Not only that through this application, an examiner can see the results of his or her examination from any university. Through this application you can take the online test and know the result as well. Through which a student can verify himself which will increase their confidence.

TABLE OF CONTENTS

CONTENTS	PAGE
APPROVAL	I
BOARD OF EXAMINERS	I
DECLARATION	II
ACKNOWLEDGEMENT	III
ABSTRACT	IV
 CHAPTER 1	 1-2
Introduction	1
1.1 Introduction	1
1.2 Motivation	1
1.3 Objectives	2
1.4 Expected Outcome	2
1.5 Report Layout	2
 CHAPTER 2	 3-8
Background	3
2.1 Introduction	3
2.2 Related Works	3
2.3 Comparative Studies	4
2.4 Scope of the Problem	4
2.5 Challenges	8

CHAPTER 3	9-13
Requirement Specification	9
3.1 Business Process Modeling	9
3.2 Requirement Collection and Analysis	10
3.3 Use Case Modeling and Description	10
3.4 Logical Data Model	12
3.5 Design Requirements	13
CHAPTER 4	13-36
Design Specification	14
4.1 Front-end Design	14
4.2 Back-end Design	30
4.3 Interaction Design and UX	36
4.4 Implementation Requirements	37
CHAPTER 5	38-41
Implementation and testing	38
5.1 Implementation of Database	38
5.2 Implementation of Front-end Design	38
5.3 Implementation of Interactions	39
5.4 Testing Implementation	40
5.5 Test Results and Reports	41
5.5.1 Sign In verification	41
5.5.2 Logical Table for Sign In	41

CHAPTER 6	42
Conclusion and future scope	42
6.1 Discussion and Conclusion	42
6.2 Scope for Further Developments	42
Appendices	43
Appendix A: Project Reflection	43
Appendix B: Related Issues	43
REFERENCE	44

List of Figures

3.1.1 Business Process Modeling	20
3.3.1 Use Case Diagram	22
3.4.1 Logical Data Model	23
4.1.1 Sign In page	26
4.1.2 Sign In page and user phone number	27
4.1.3 Sign In page with valid phone number	28
4.1.4 Sign In page with verified code	29
4.1.5 Application front page	30
4.1.6 Public university front page	31
4.1.7 University internal front page	32
4.1.8 University Information interface page	33
4.1.9 University Link page	36
4.1.10 Question set front page	39
4.1.16 Answer page	40
4.2.1 Back-end design	41
4.2.2 Back-end design	42
5.2 Implementation of front design	49
5.5.1 Sign In verification	37

List of Tables

2.4.1.1 Key capabilities	5
5.4.1 Testing Implementation	39
5.5.2 Logical table for sign	40

CHAPTER 1

INTRODUCTION

1.1 Introduction

It is an online Android application. At present it is unimaginable that a learner would go and collect information from various university admission exams. To help ease this problem in the learners “**Admission helper**” application, the script is created. Using this application students will find all the information provided in most public and private university admission tests very easily.

Apart from this, the application will get the question of the lost year by dividing the section of the universities and the benefit of the model test for the learners themselves.

There was a time when people believe a lot in manual work but now this time people believe in online. IN the student life we should be use time in proper way. For reduce waste of time we create this android application.

1.2 Motivation

There are many students in our country who cannot apply and attend the right admission test for the right information at the right time. On the other hand, many students do not collect last year’s question at the university as they choose to prepare, most of them are from rural area. Students will not have to face such problems later through the “Admission helper” Android Application.

Because all the information and question of all the universities here are available at the same time. If you want the learner to test himself according to the type of university question, the application can also be accessed through the model test, which will encourage learner lazer.

This application can be a great way to reduce the amount of the time you take for university admissions candidate. We have seen very closely to face this problem. So we decided to make an “**Admission helper**” Application as our final project

1.4 Objectives

1. Ensuring the benefit of the students and to reduce the sufferings related to admission test
2. Making MCQ exam facilities for individual university
3. Enhancing self-judgment and self-dependence of students
4. Saving money and time of every student

1.4 Expected Outcome

1. User can find different information of different university for admission test.
2. User can appear for model test and get the instant result.
3. User can evaluate himself/herself with the help of these results of model tests.
4. They can get university-wise result for admission test

1.5 Report Layout

First, in our reporter layout, we started with 4 content and we have described the chapters in section 4 and gave reference. The first chapters are packed with Introduction, Motivation, Objective and Expected Results. The second chapter has been integrated with background intro and relational works, and this application has been compared to some related applications and our application challenges have been highlighted.

Chapter 3 provides the business model, the use case diagram and its descriptions and logical data model and design requirements.

Chapter 3 is arranged with front end design and back end design

Then the Implementation has been done in Database and Front and Design and Interaction and Testing Implementation.

And last chapter is given in Discussion and Conclusion

CHAPTER 2

BACKGROUND

2.1 Introduction

Adobe XD is a vector based user experience design for web apps and mobile apps. Developed and published by Adobe inc. It is available for macOs and windows , although there are versions for ios and android to help preview the result of work directly on mobile devices. XD supports website wireframing and creating simple interactive click-through prototypes. We are using java as well. Java is very popular language for android apps . This option is effective for students who have passed the HSC exam . The user interface for Adobe XD, Java as well as the design of this application has been designed to do this. By using this app, a student will be able to benefit a lot more easily, in less time, as we will find important information from almost all the public and private universities of Bangladesh at the same time, which will greatly reduce the suffering of an admission test. Not only that, through this app, an examiner will know the result of his admission. By using this app, a student will be able to benefit a lot more easily, in less time, as we will find important information from almost all the public and private universities of Bangladesh at the same time, which will greatly reduce the suffering of an admission test. Not only that, through this app, an examiner will know the result of his admission.

2.2 Related Works

The admission helper application is a very effective Android application for university admissions candidates. When a student wants to get all the information of a university for a university admission test, our application can be a really helpful tool for that student. The next step a user enters in our application will get 2 options for obtaining information such as its convenience, that is, public universities and private universities. Usually a student who wants to get all the information from many universities will have to collect information from different places, but the next step in our application is to cure this problem. Where a user will get information about most university admission tests together. A student does not have to display different varsity websites repeatedly for different information, so the suffering is reduced in many respects as a student being a time safe. An examiner will be able to access all the

information, admission information, question type, past questions and model tests through this app, so he does not have to collect books in many markets, does not have to go to any coaching center to test himself.

2.3 Comparative Studies:

We found several websites and play stores, we have 2-5 similarity applications like this one. These websites and play store applications only have the advantage of giving model tests according to the question pattern of different public universities, but university information, admission information, last year's questions are not available in these applications simultaneously. But our application is all within one, where university information, admission information, last year's questions can be found in only one application for which the user does not have to visit different websites. Apart from this, this application has the advantage of providing university information, past questions and model tests to almost all private universities in Bangladesh. We have this application

Another great feature is that an examiner can see his / her merit position in any university admission test.

2.4 Scope of the Problem:

- Database Design
- Testing
- Data collection
- Work flow (model)

2.4.1 Database Design:

Store and sync data with our NoSQL cloud database. Data is synced across all clients in real-time, and remains available when your app goes offline. The Firebase Real-time Database is a cloud-hosted database. Data is stored as JSON and synchronized with each connected client in realtime. When you build cross-platform applications with our iOS, Android and JavaScript SDKs, all your clients share a realtime database instance and automatically receive updates with new data.

2.4.1.1 Key capabilities

Realtime	Firestore uses realtime database data synchronization instead of normal HTTP requests - each connected device receives that update within milliseconds every time the data is changed. Provide collaborative and immersive experiences without having to worry about networking code.
Offline	Firestore applications are responsive even when offline because the Firestore Realtime Database SDK keeps your data on disk. After the connection is reopened, it synchronizes with any current server state that the client device has missed.
Accessible from Client Devices	The Firestore Realtime Database can be accessed directly from a mobile device or web browser; No application server required. Security and data validation is available through the Firestore Realtime Database Protection Rules, expression-based rules that execute when reading or writing data.

Scale across multiple databases	You can support your application's data needs by scaling your data by splitting your data into multiple databases for the same Firebase project, including the Firebase Realtime Database in the Blaze pricing plan. Streamline authentication with Firebase Authentication in your project, and authenticate users in your database examples. Control data access to each database, including custom Firebase realtime database rules for each database instance.
---------------------------------	--

Table: 2.4.1.1 Key capabilities

The precise equation of access to database troops directly from the FireWells realtime data client-side load should create team applications. As the article maintains and replays of uploading programs are about to happen, the last observer reviews. Get the device back in connection, Realtime Database Client Offline has been contacted with Cakalin's remote update, Long Island has been contacted .

RealTime Database Firebase provides a flexible, expression-based rule language called how to structure your data and when to read and write data. When integrated with Firebase authentication, developers can determine who has access to the data and how they can access it.

The Realtime Database is a NoSQL database and has different optimizations and functionality compared to one such related database. The Realtime Database API is designed to allow activities that can only be executed. It enables you to create a great realtime experience that can serve millions of users without compromising responsiveness. Because of this, it's important to think about how users need to access your data and then structure it accordingly.

2.4.1.2 Next Steps:

- Set data and listen for changes using the Android, SDKs, or the REST API.
- Add Firebase Real-time Database to your iOS, Android, or Web app.
- Learn about how to secure your files using Firebase Real-time Database Rules.

2.4.2 Data collection

No other student-generated category is required to create this application. We first need to add two separate part of the public and private university to the application to add data. Then the name of all the Universities has been added serially according to the Government and Private. This is where all the information of each university whose profile has been added to it. All the information entered in the university is individually added to the application for the benefit of the students.

Adequate information is added here to ensure that any student in any department can get admission information as per his requirement. All the question papers from last year were added to the application by both the public and private university departments.

Next step is the model test facility based on the admission test type of universities. All this information in the application will serve as a package for a university student.

As information about university admissions changes every year, we need to update this information regularly. So Google's help in collecting some data for the application was indispensable.

2.4.3 Testing

In testing part have being facing so many problem. Matching date and time is big challenge for me. In software testing we have to need to check is all of the part is working or not. If working then we have to check is working in perfect way or not that means what I want is it working this way. And also check by the wrong information.

In many time I face so much error. Also check is this application Is first or not. Then check all the contents are right or not. All the edit, view, delete option is working perfectly or not.

2.4.4 Work Flow

Workflow is the definition, execution and automation of business processes where tasks, information or documents are transmitted from one participant to another partner for action according to a set of rules of operation.

Consists of an orchestrated and repeatable pattern of a workflow activity, entities transform processes into resources that enable materials to be transformed, provide services or process information .It is the sequence of operations, the work of an individual or group, the work of employees' organizations. Or can be portrayed as one or more simple or complex processes Ray.

From a more abstract or higher-level perspective, workflow can be considered as a philosophy or representation of real work .The flow described can refer to a document, service, or product that is moving from one step to another.

2.5 Challenges

In the beginning, this project was very difficult for us, as a good Android application requires Java and database skills to be very good. Apart from this, we have a lot of features in our application tee, increasing the Android skill was a big challenge for us.

The first challenge of launching the application was to raise last year's questions in the thick of the section, along with most public and private university information. Then, implementing this information in applications was a difficult task.

The biggest challenge of this project was to prepare the model test according to the question pattern of each university and to implant it in the application, because the student will submit the Model Test by submitting the test and will show the correct result as well.

CHAPTER 3

REQUIREMENT SPECIFICATION

3.1 Business Process Modeling

Business process modeling is a graphical representation of an expected business process or work flow, as a means of identifying potential improvements. This is usually done through different graphing methods, such as the flowchart, data-flow diagram, etc.

Our application of the business process model is given in the diagram bellows:

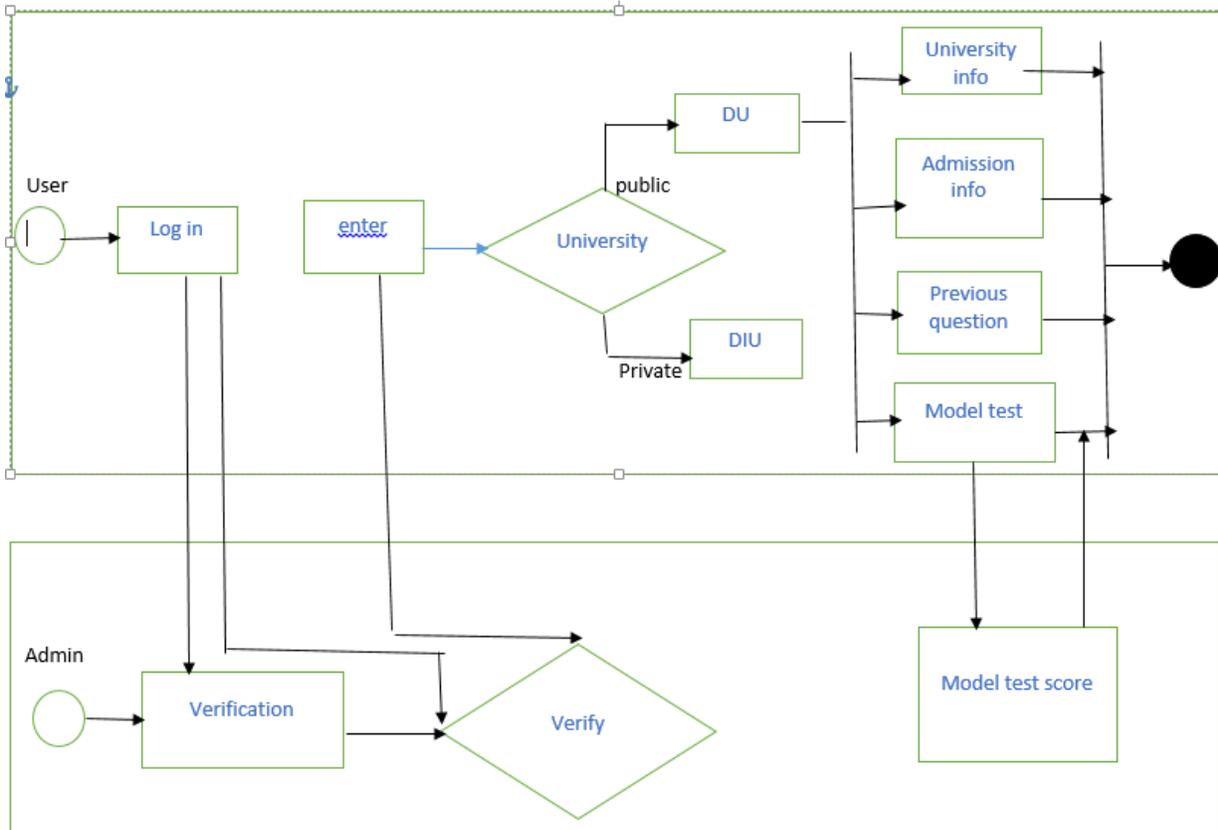


Figure 3.1.1: Business Process Modeling

3.2 Requirement Collection and Analysis

Every development system or app should have a Requirement Collection as well as some of our System Requirements collection that we have listed below:

Our application is for admission to the University so in this case we should have information of almost all the university and this information should be accurate, at the bottom every model model test should be given according to the type of that university question so that the students do not have to buy books and the result of that model test. So that shows up right away we need some functional requirement in our application.

For example, updating the university thickness questions every year, as well as increasing the number of universities as much as possible so that students can attend the universities of their choice.

3.3 Use Case Modeling and Description

The type of actuator that a system has, the way it relates to the system, is shown through the case diagram.

There are two types of actors, the first one is the admin and the other is the user. To access this user's system, they must first log in and get verified through Verification, after which they will be able to view the information at any university they wish.

Users will enter different universities to learn about that particular university's information and admissions information and previous questions and model tests.

The admin will first see the verification function, if the validation is correct, then the admin will allow the user to enter the system.



Figure 3.3.1: Use case Diagram

3.4 Logical Data Model

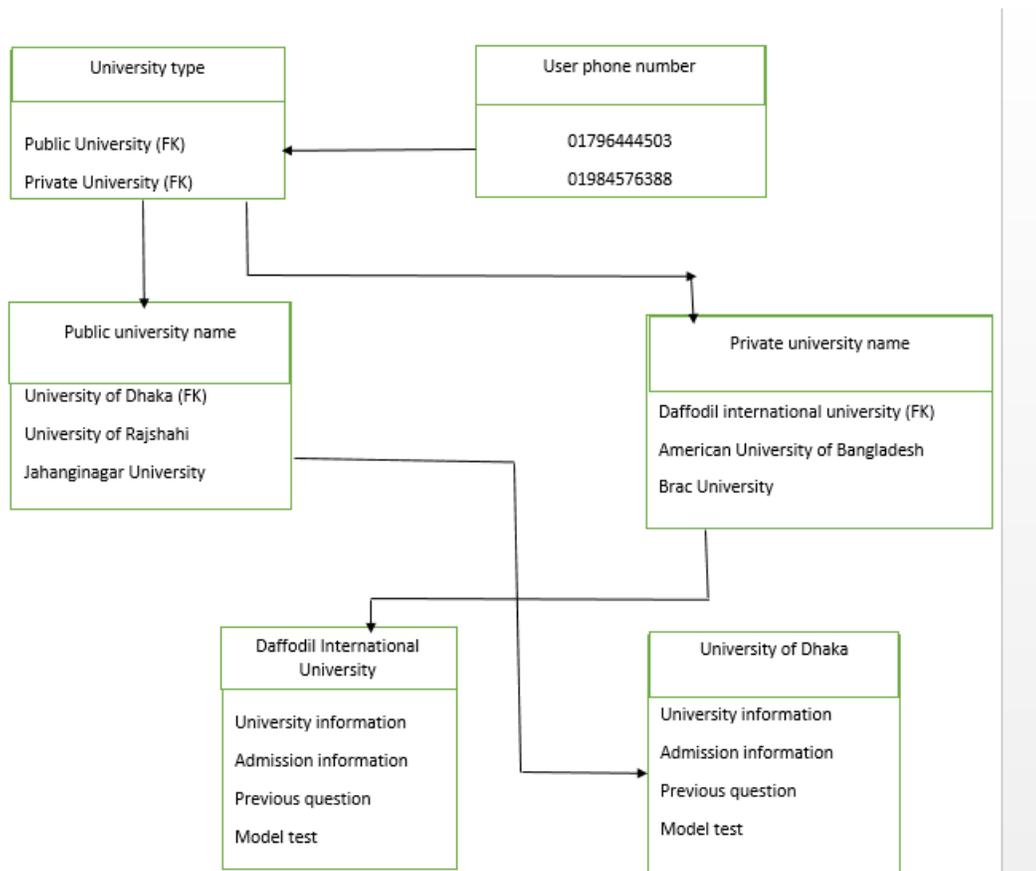


Figure 3.4.1: Logical Data Model

3.5 Design Requirements

Two design are required for building a web application one is front-end design and another is back-end design. Depending on an application design, the application will run or shut down. The application must be user friendly.

Some features of the application may not be given to students so that they do not make sense. You must pay attention to the system running fast or otherwise it will be difficult to stand up.

Every part of this application should be accessible to students. Design our application very well so that students do not have any problems.

We have tried to make the application better for all students and make it run faster. The kind of benefits students are getting bellow:

1. Students will be able to take university information of their choice
2. Students can take admission information from the university
3. They can get previous questions from the university through our application
4. And they will be able to take a model test to test their talents and get results immediately

CHAPTER 4

DESIGN SPECIFICATION

For building an application firstly we select a niche that means I will work on this. After selection niche then we have work core functionality of this niche. Then we have to a plan to design and working methodology of this application. After that we will make a perfect design for this application. Design specification discuss about this matter that how a design will be developed for a system. Now this time I will show the font-end and back end design of my admission helper android application and admin panel. And in this section we will discuss about many tool and platforms.

4.1 Front-end Design

Adobe XD is a vector based user experience design for web apps and mobile apps. developed and published by Adobe inc. It is available for macOs and windows , although there are versions for ios and android to help preview the result of work directly on mobile devices. XD supports website wire framing and creating simple interactive click-through prototypes interactive click-through prototypes. We are using java as well . java is very popular language for android apps . This option is effective for students who have passed the HSC exam . The user interface for Adobe XD, Java as well as the design of this application has been designed to do this. By using this app, a student will be able to benefit a lot more easily, in less time, as we will find important information from almost all the public and private universities of Bangladesh at the same time, which will greatly reduce the suffering of an admission test. Not only that, through this app, an examiner will know the result of his admission. By using this app, a student will be able to benefit a lot more easily,

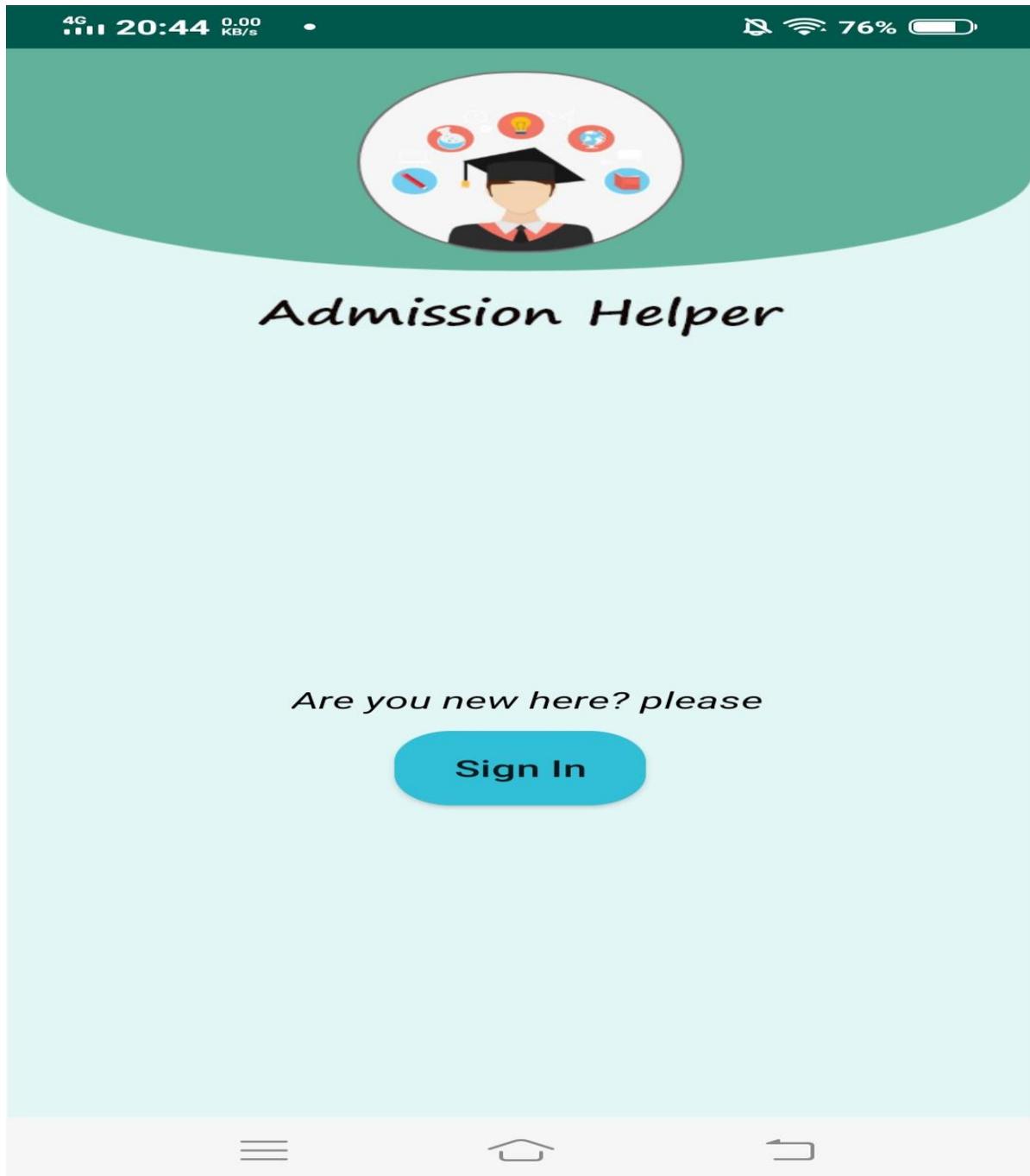


Figure: 4.1.1 Sign In page

When an examiner has installed these apps, he / she will be able to see this front page sign in and proceed to the next step by selecting sign in.

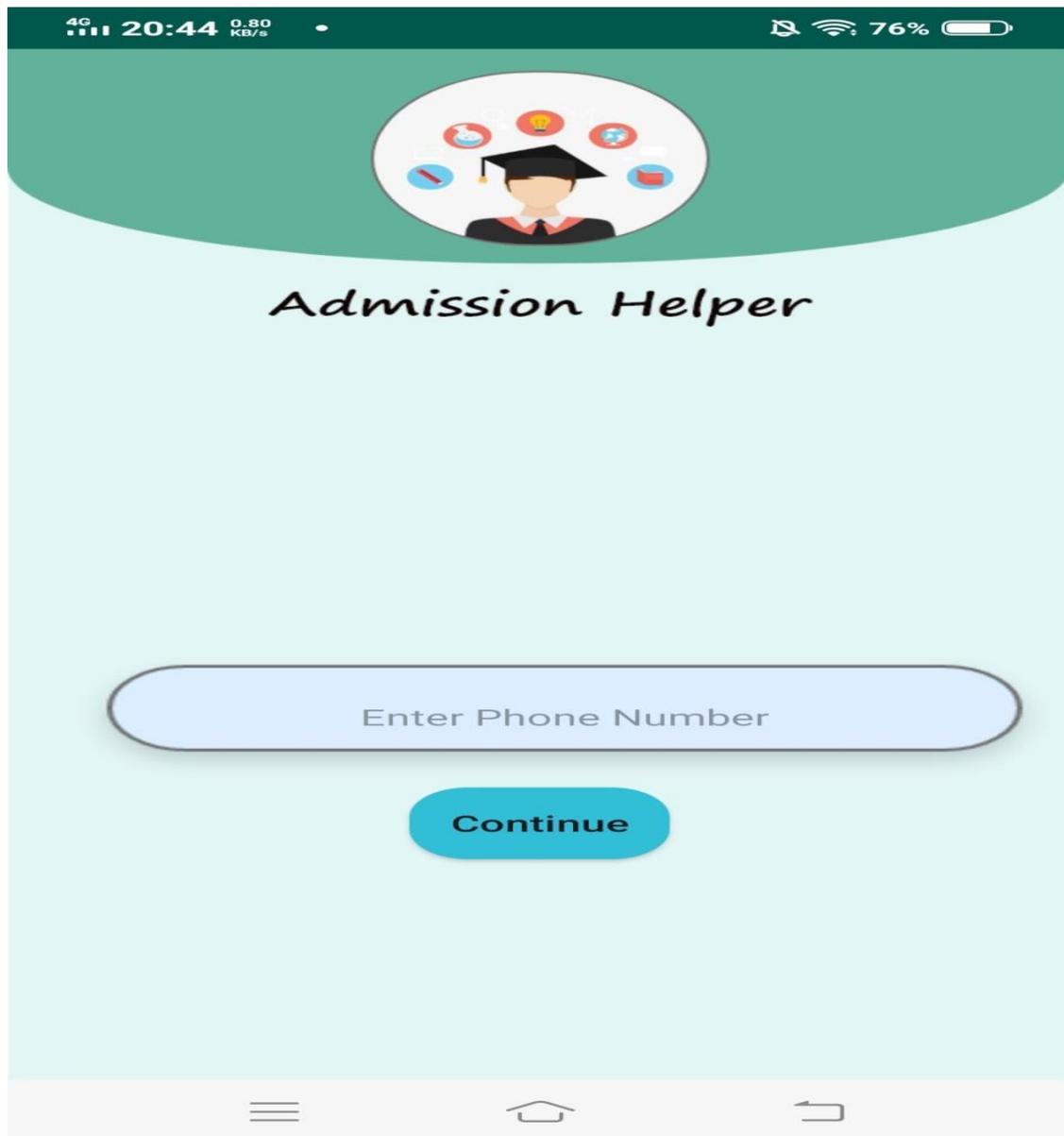


Figure: 4.1.2 Sign In page and user phone number

A student will be able to view this interfacing after signing in, where the student must provide a valid or a usable mobile number. Students will be able to view the Next Interface by pressing the Container Button by providing a mobile number.

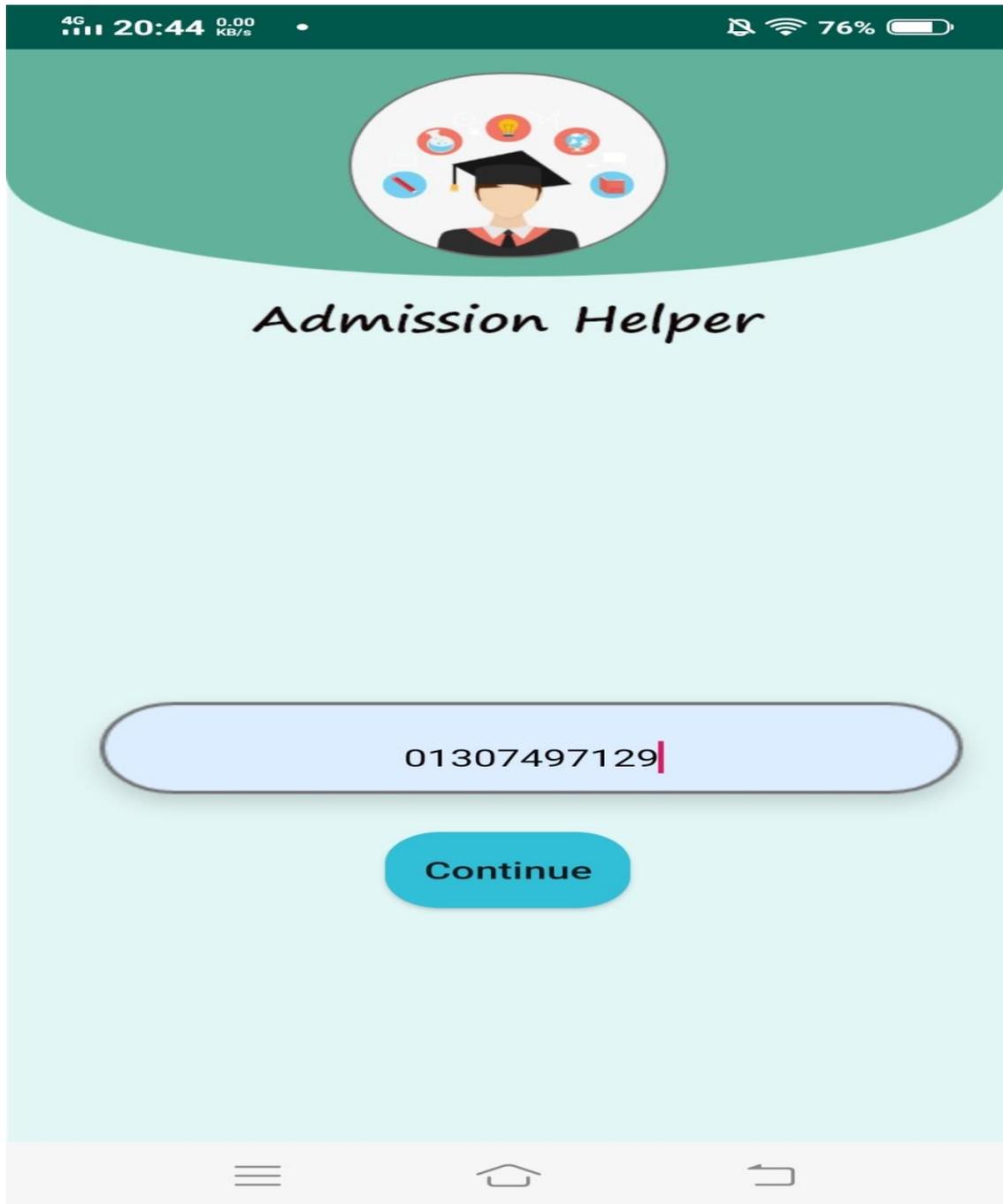


Figure: 4.1.4 Sign In page with valid phone number

Here the candidate will input the valid phone number and press Continue. Through which he will move on to the next step.

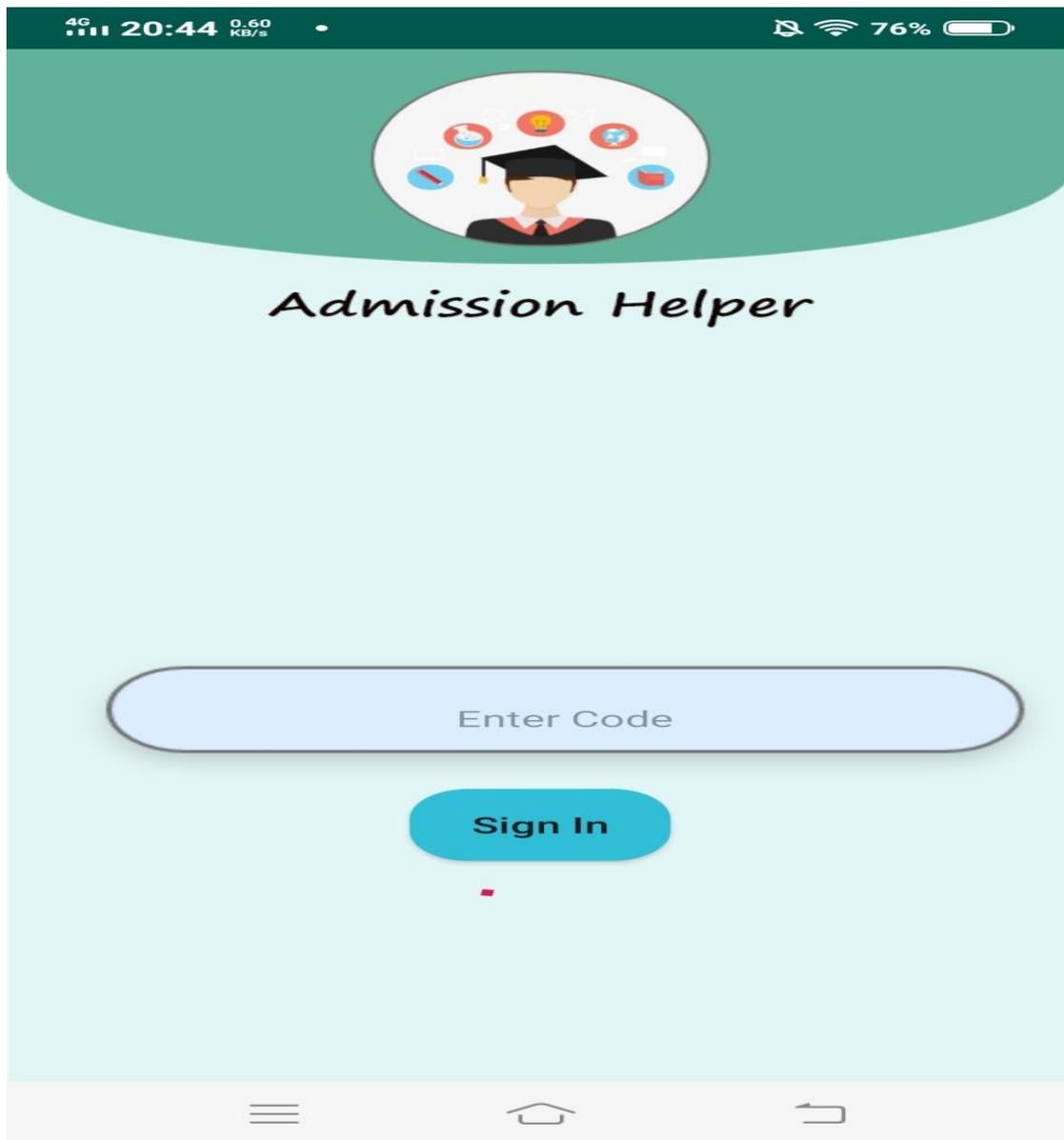


Figure : 4.1.5 Sign In page with verified code.

A message of a verified code of Six Digits will be sent to the phone number that the examiner had inputted on the previous interface. The user has to input Six Digit's verified code here. Here, if the examiner makes a mistake in inputting the code, they will not be able to access the next interface. The software will show him the code error.

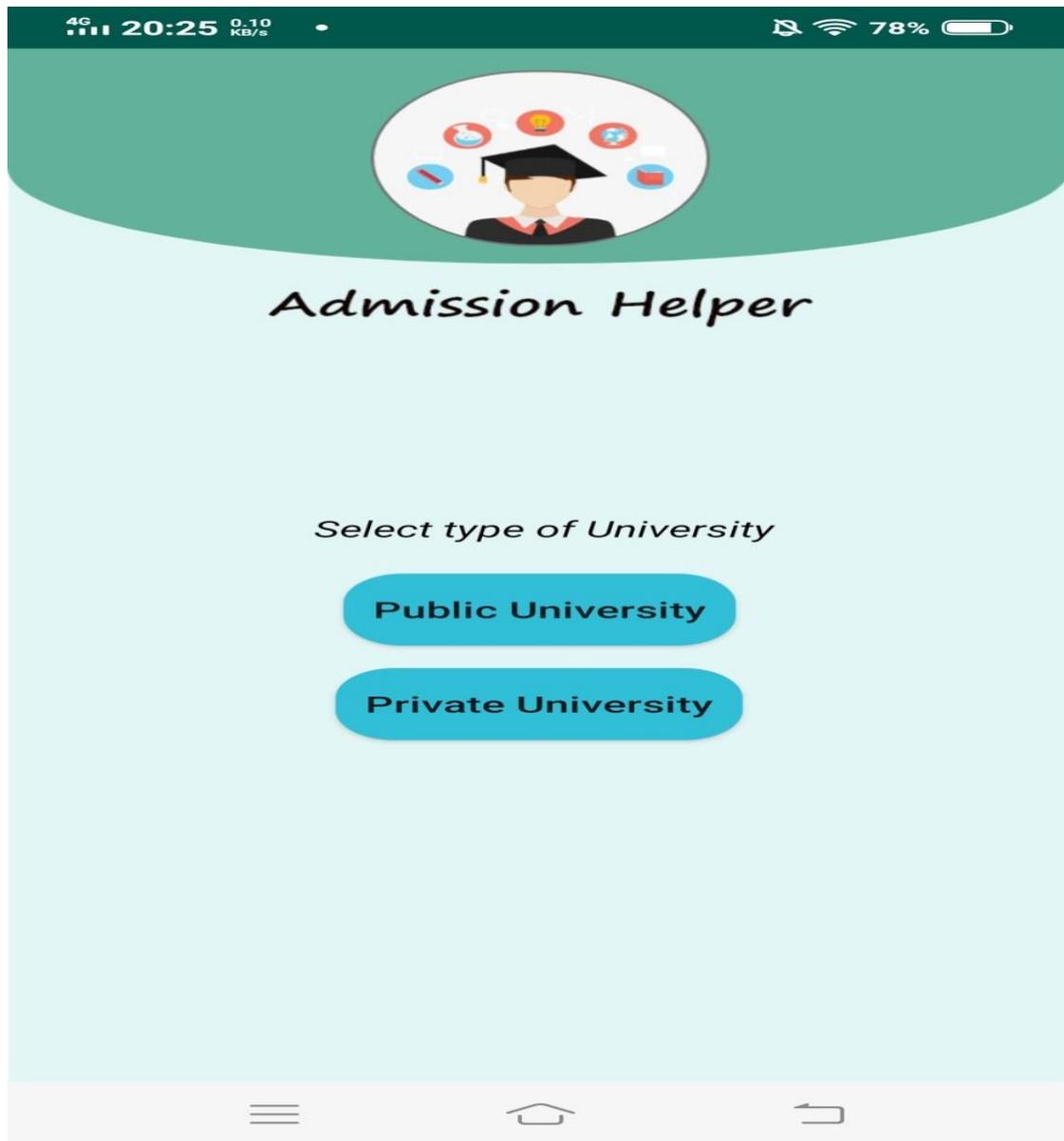


Figure: 4.1.6 Application front page.

Seeing this interface of the application means that the user has completed his sign-in process correctly, and this interface will show the fast front-phase of our application, which is how many times the user enters this application. There are 2 buttons on this interface. The first is a public university and the second is a private university. This time the user will randomly enter the interface that the button presses.

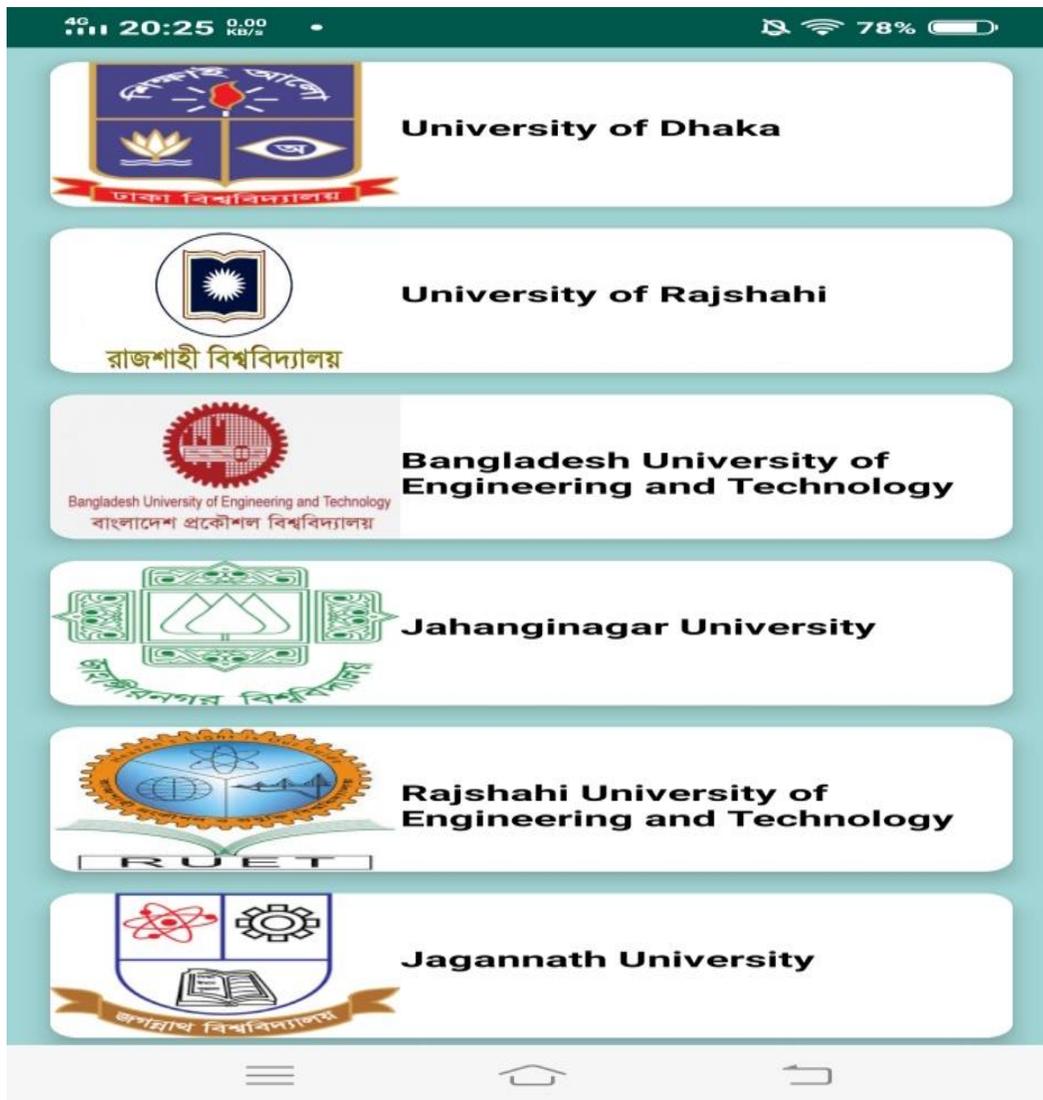


Figure: 4.1.7 public university front page.

Suppose one of the candidates randomly selected the public university, then he would see the name of the top listed universities of Bangladesh. This time, when an examiner clicks on a university of his choice, the next step will be to see all the activities of that university. The University will be aware of important information such as important information during the university's establishment, the time and date of admission test, the university's previous quiz repository and the model test.

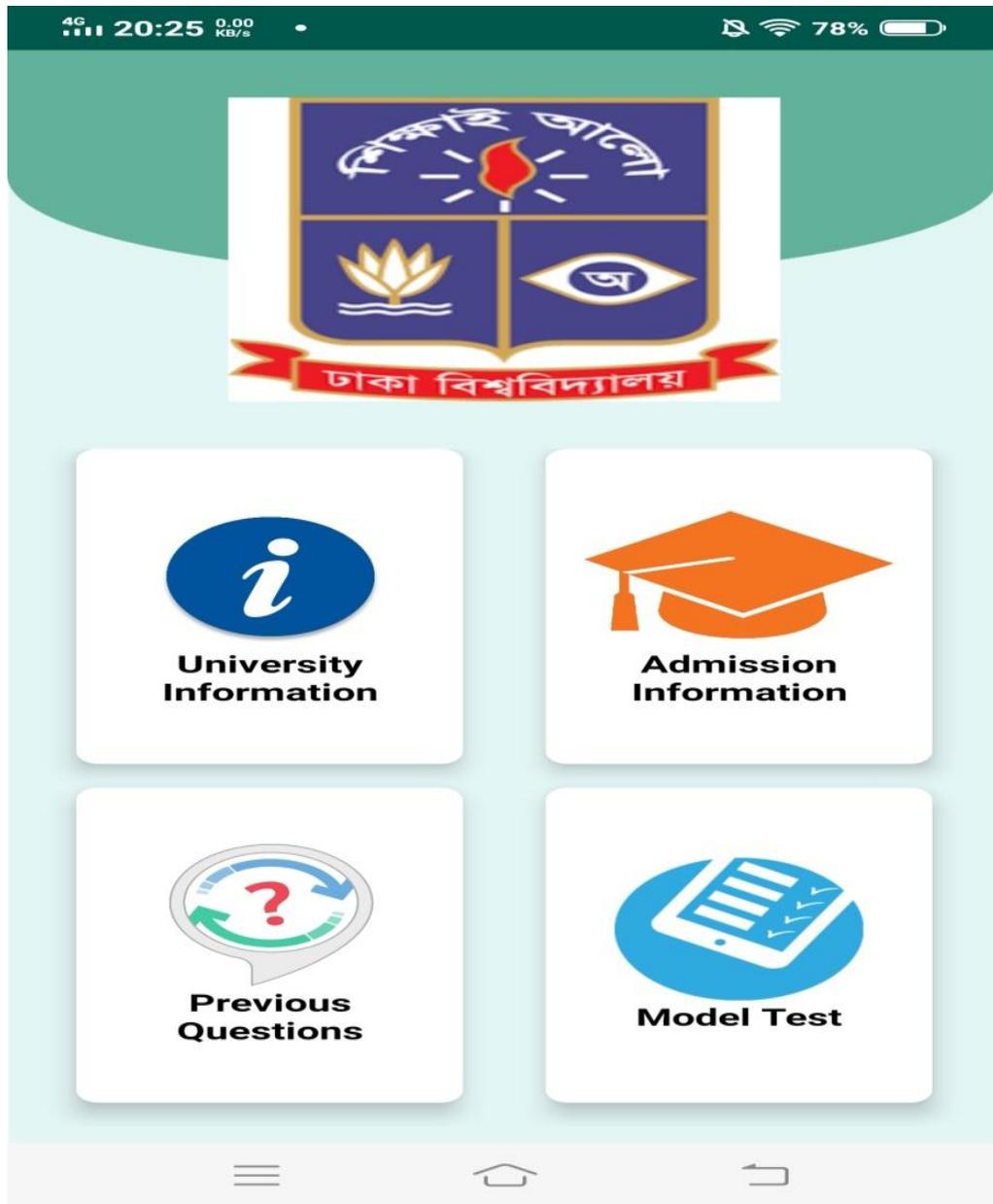


Figure: 4.1.8 university internal front page.

When an examiner presses a button on a particular university, it will be able to see the category of the church type belonging to that university. They are University Information, Admission Information, Review Questions and Model Tests. This time the examiner will see the next step of the button that he chooses.



Figure: 4.1.9 University information inter-page page.

When an examiner clicks on the University Information button, he or she will see the interface in the next step. How many years has Varsity T been settled there? What is the number of students present? What is the meaning of this varsity? All the important information will be visible.



Figure: 4.1.10 university link page.

Now if an examiner clicks on the Admissions Information button, he / she will see this interface. Here are some links, clicking on this link, the examiner will see all the information about the admission test. Such as the time and date of the admission test.



Figure: 4.1.11 university unit page.

Here you can see the previous question according to the unit and if he / she can click on any unit of his choice. For example, there are four units.

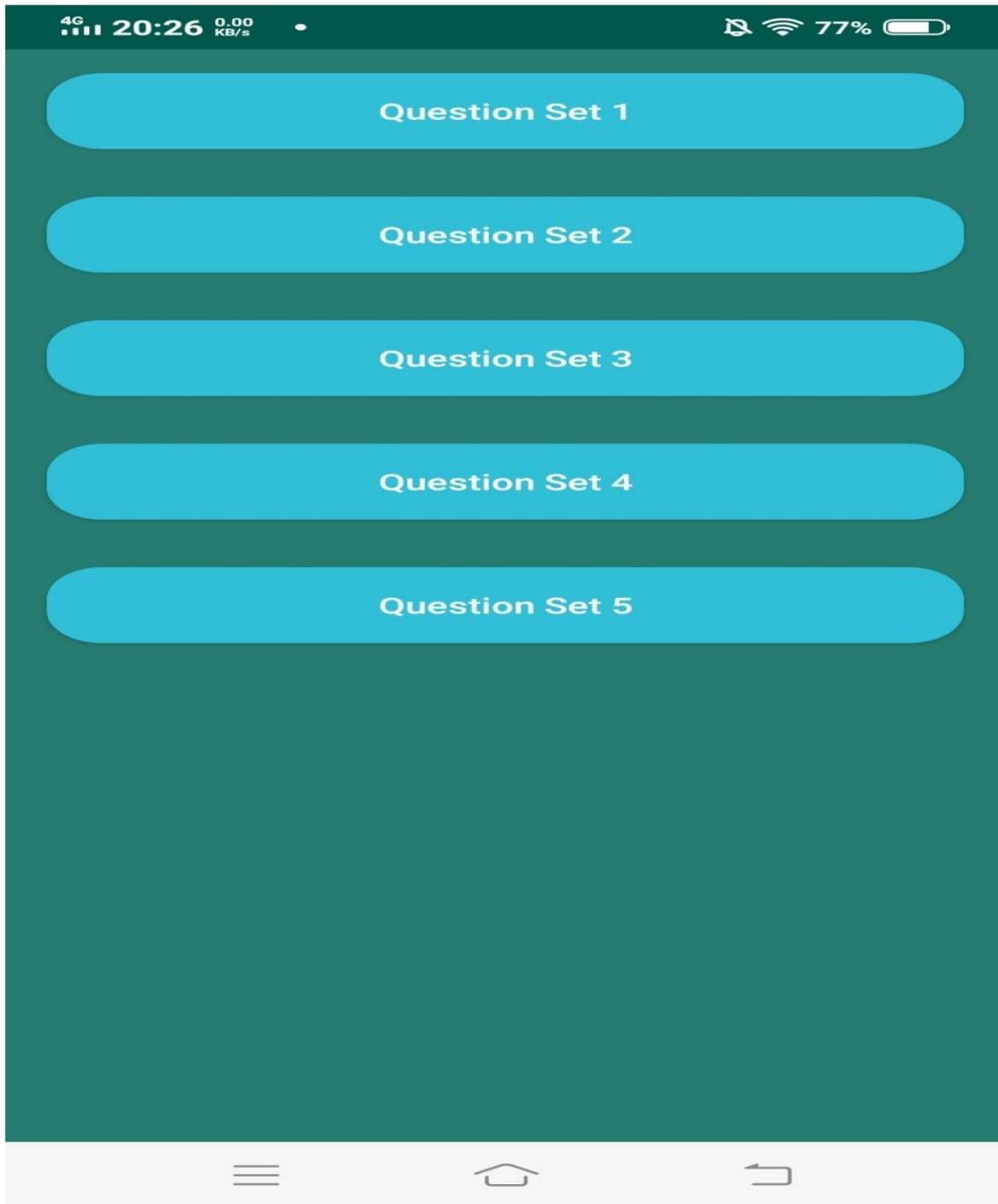


Figure: 4.1.12 Question set front page.

If an examiner clicks on the Model Test button, he will see 5 options there; option 3 is the number of units of that school, A unit, B unit, C and unit, D unit. Now the user will click on the unit button, the unit will be able to test the question pattern model.

4G 20:26 2.00 KB/s 77%

00:59:49

গুটিকুলাম হয় কোনটিতে?

লেবু

জুই

আখ

পেয়াজ

বৃক্ষের একক?

নেফোসাইট

গ্লোমেরুলাস

কর্টেক্স

নেফ্রন

মানুষের সর্জীব রক্তের তাপমাত্রা?

৩২-৩৪

৩৪-৩৬

৩৬-৩৮

SUBMIT

Figure: 4.1.13 Question pattern page.

The user will see the question here and can take the test here to test himself. There will be questions according to the unit.

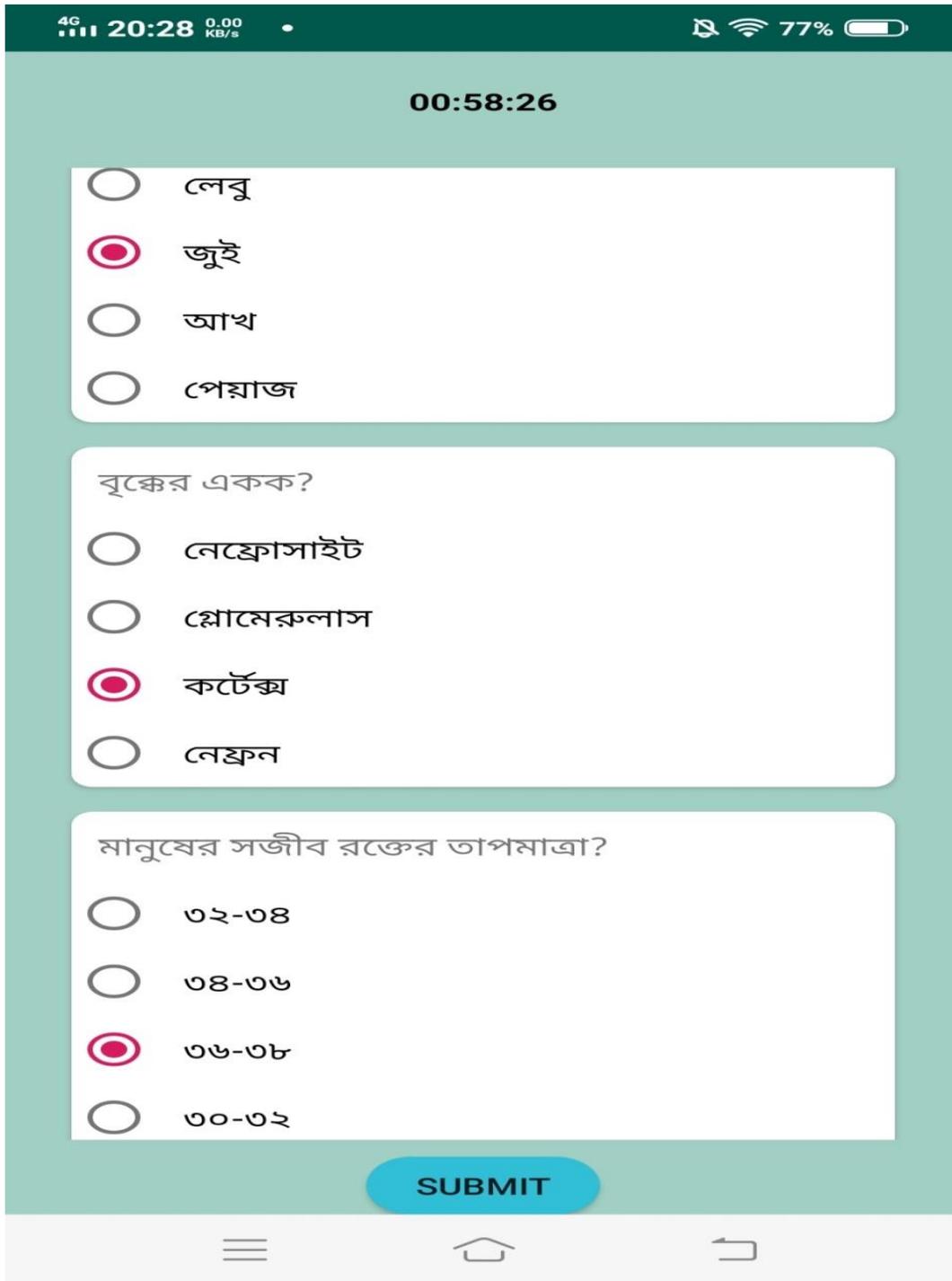


Figure: 4.1.14 Question pattern page.

Now, if the user selects a question set, a question will come up accordingly and there will be one question and 3 options. And after seeing each question, he will select the correct answer.

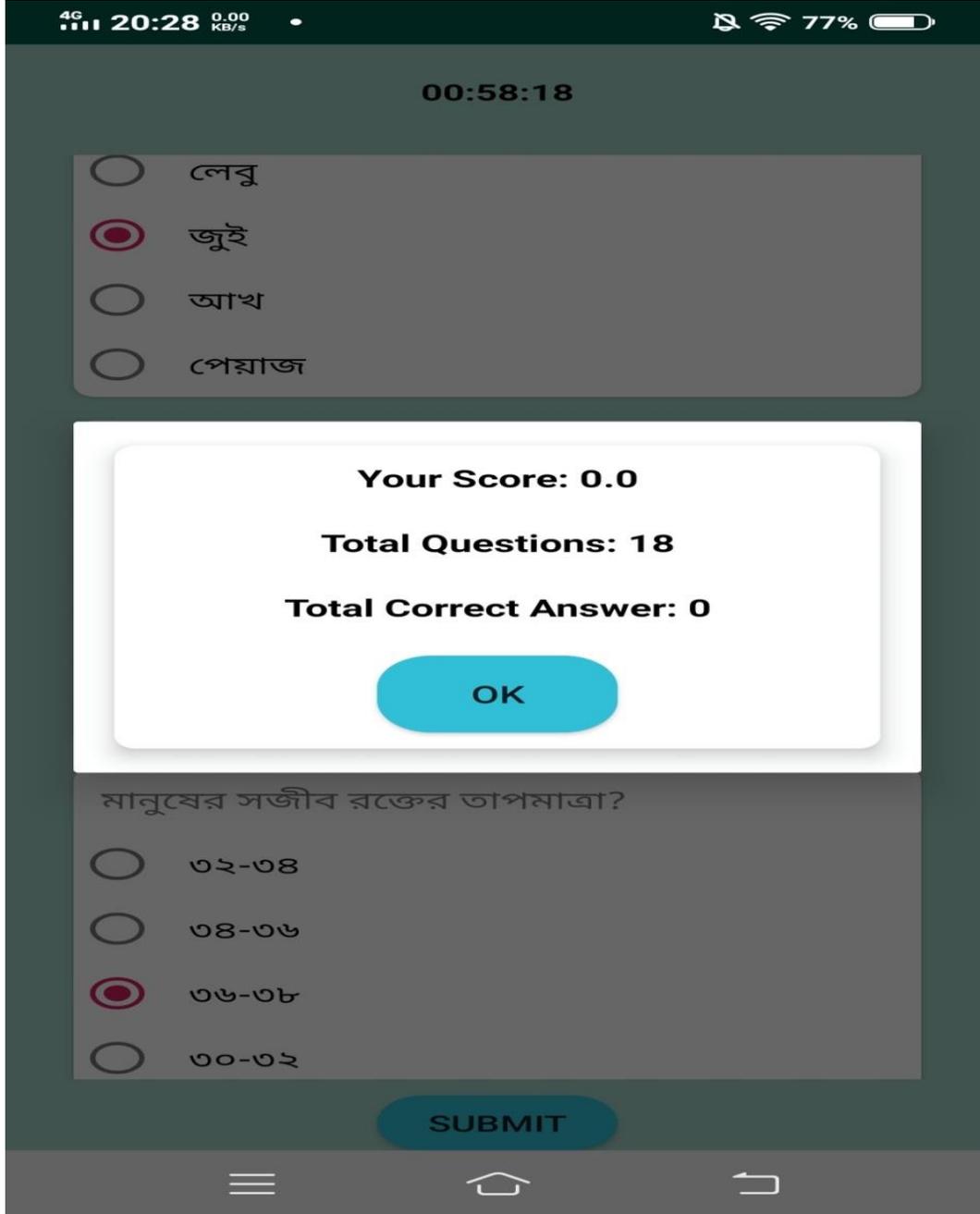


Figure: 4.1.15 Score page.

After selecting the user's answer, you will see how many questions went wrong and right. After watching this page you will understand how much you got in the test.

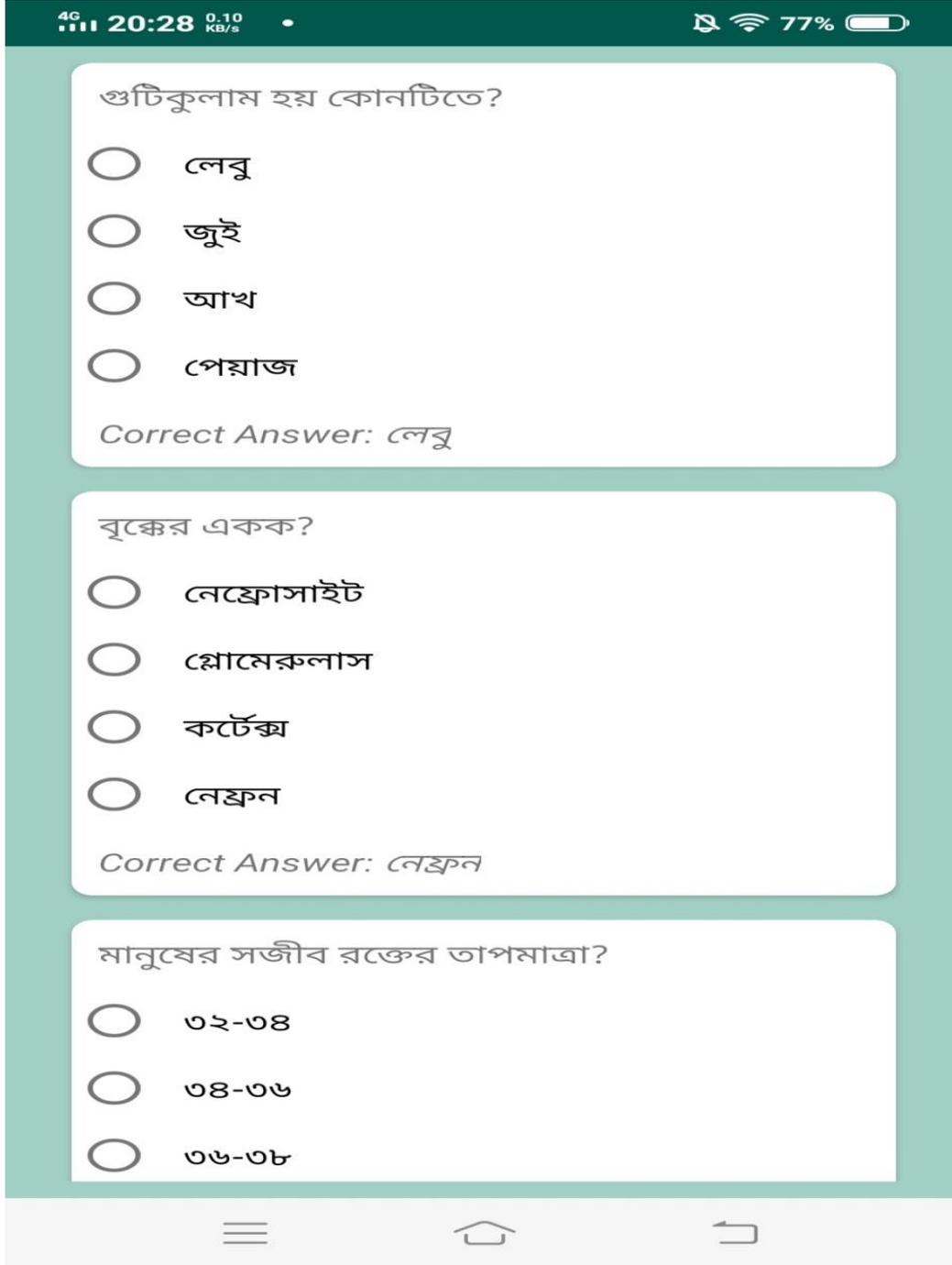
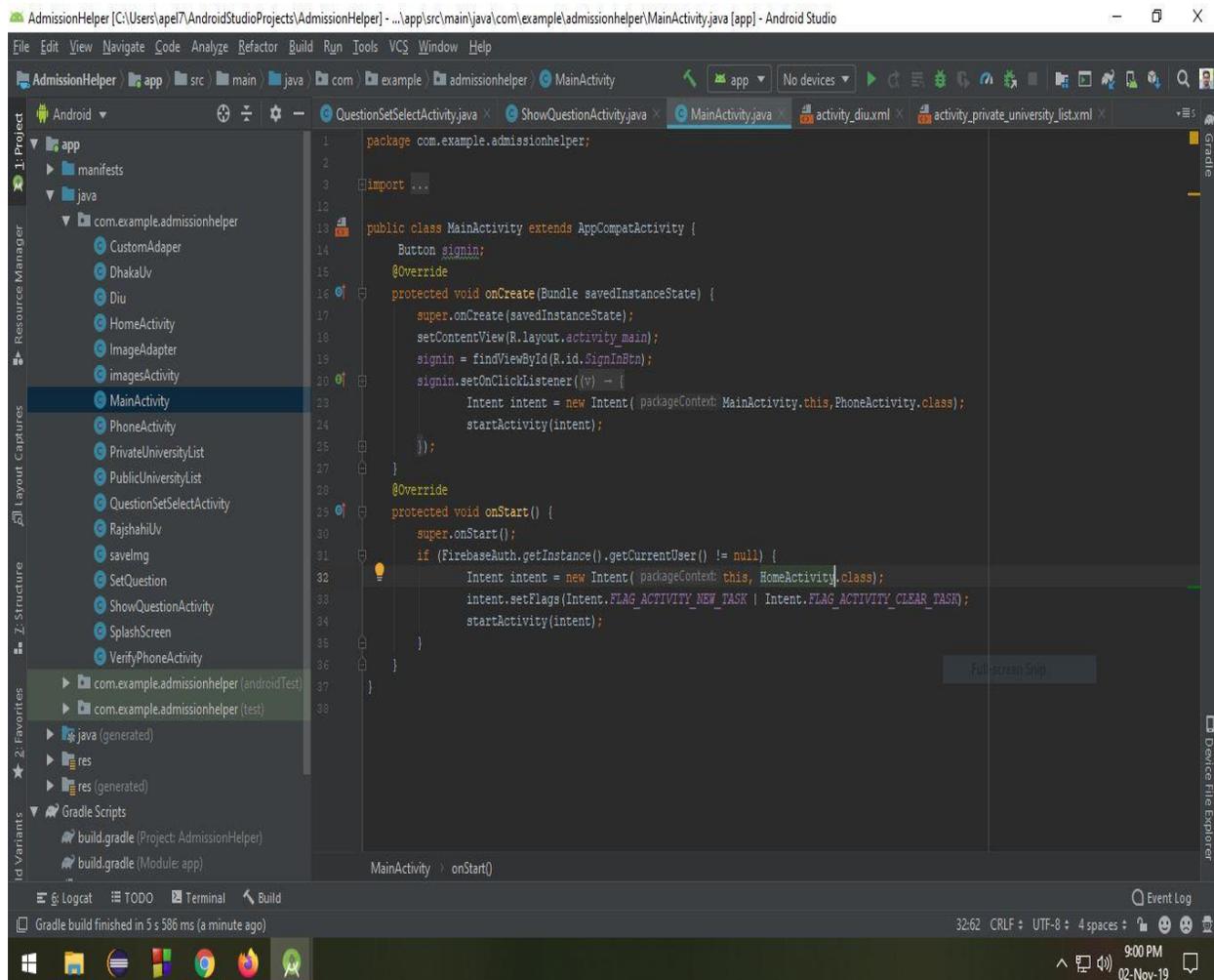


Figure: 4.1.16 Answer page.

If you click the OK button, he can see the answer to every question. By which he can know the answer to his mistake and can read from now on. He will benefit greatly.

4.2 Back-end Design

Back-end design part is very important for this application because all the logical part are including here. Another part also join in back-end part like server side language, database management security authentication, data backup. In my back-end part I used Android studio.



```
1 package com.example.admissionhelper;
2
3 import ...
4
12
13 public class MainActivity extends AppCompatActivity {
14     Button signin;
15     @Override
16     protected void onCreate(Bundle savedInstanceState) {
17         super.onCreate(savedInstanceState);
18         setContentView(R.layout.activity_main);
19         signin = findViewById(R.id.SignInBtn);
20         signin.setOnClickListener(v - {
21             Intent intent = new Intent( packageContext, MainActivity.this, PhoneActivity.class);
22             startActivity(intent);
23         });
24     }
25
26
27
28     @Override
29     protected void onStart() {
30         super.onStart();
31         if (FirebaseAuth.getInstance().getCurrentUser() != null) {
32             Intent intent = new Intent( packageContext, this, HomeActivity.class);
33             intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);
34             startActivity(intent);
35         }
36     }
37 }
38
```

Figure: 4.2.1 Back-end Design

```

36 progressBar = findViewById(R.id.progressBarId);
37 editText = findViewById(R.id.verifyPhoneEt);
38 button = findViewById(R.id.signIn);
39 String phoneNumber = getIntent().getStringExtra("phoneNumber");
40 sendVerificationCode(phoneNumber);
41 button.setOnClickListener((view) - {
42
43     String code = editText.getText().toString().trim();
44     if (code.isEmpty()) {
45         editText.setError("Invalid code");
46         editText.requestFocus();
47         return;
48     }
49     progressBar.setVisibility(View.VISIBLE);
50     verifyCode(code);
51 });
52
53 private void verifyCode(String code) {
54     PhoneAuthCredential credential = PhoneAuthProvider.getCredential(verificationId, code);
55     signUpCredential(credential);
56 }
57
58 private void signUpCredential(PhoneAuthCredential credential) {
59     mAuth.signInWithCredential(credential).addOnCompleteListener((task) - {
60         if (task.isSuccessful()) {
61             Intent intent = new Intent(packageContext, VerifyPhoneActivity.this, HomeActivity.class);
62             intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);
63             startActivity(intent);
64         } else {
65
66         }
67     });
68 }
69
70
71
72
73

```

```

1 package com.example.admissionhelper;
2
3 import androidx.appcompat.app.AppCompatActivity;
4
5 public class HomeActivity extends AppCompatActivity {
6     Button pri, pub;
7
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10        super.onCreate(savedInstanceState);
11        setContentView(R.layout.activity_home);
12        pri = findViewById(R.id.privateBtn);
13        pub = findViewById(R.id.publicBtn);
14        pri.setOnClickListener((v) - {
15            Intent intent = new Intent(packageContext, PrivateUniversityList.class);
16            startActivity(intent);
17        });
18        pub.setOnClickListener((v) - {
19            Intent intent = new Intent(packageContext, PublicUniversityList.class);
20            startActivity(intent);
21        });
22    }
23
24 }
25
26
27
28
29
30
31
32
33
34

```

Figure: 4.2.2 Back-end Design

4.3 Interaction Design and UX

A interaction design is a way that discuss between system and user. To understand the user problem or for knowing user requirement we use interaction design. Not only finding problem but also I find this problem solution. It also focus that user experience to use this application.

For UX we have tested it by twelve peoples. In those people ten are said that this application is very helpful and user friendly. When I share all think about my project somebody said that it's user friendly but admin can use some problem when routine update some day later.

So we can say that this application are user friendly.

4.4 Implementation Requirements

We had some basic requests for the project. The things that we are trying to accomplish in this project. Our Basic Requirements are an Examination Varsity Information, Admission Information, Prediction Question and Model Test through these apps.

That we have filled in with these apps. First, a learner will get an accurate idea of Varsity Information. And the date of the admission test will get an idea of these. The model will test itself through the test and get the correct answer about the previous question.

CHAPTER 5

IMPLEMENTATION AND TESTING

After all the planning one thing that comes up is the implementation and then the testing. This is the part we will discuss in detail about the implementation and testing of our project. Implementation in a project is just as important as testing and equally important. Because through testing, it is known that all the functions of the project are the correct course or not the course.

5.1 Implementation of Database

In the Android application, we have stored all the data in the database. For this, the database was created after installing hardware and DBMS first. Lists the database tree size for the data store. Where the data is stored in the tree nodes. Then the nodes are looped and the data is collected. We can control our data with code. In our database we can delete, edit and insert data.

5.2 Implementation of Front-end Design

For our project's front-end design, we used Adobe xd and basic Android xml design. Adobe xd enables a user to see the full range of designs. This makes it easier for a user to use. On the other hand, the advantage of using basic Android xml design for our Android application's layout designer is that it is a lightweight language. For this it will not load our layout.

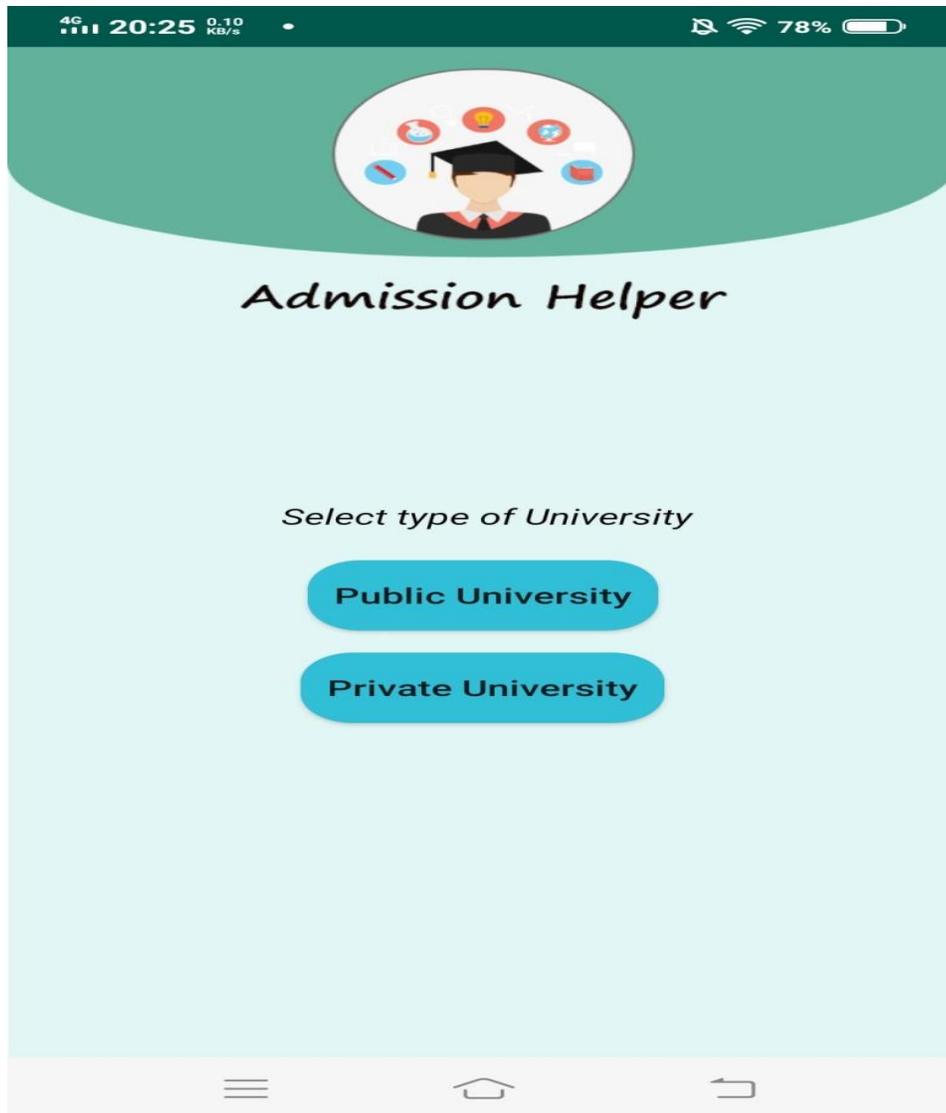


Figure: 5.2 Implementation of Front-end Design

5.3 Implementation of Interactions

To create the application, first we create a use case diagram according to the project request. Since our application is an Android based application, we make firebase combination with Android Studio and then we code it and implement it.

5.4 Testing Implementation

Test case	Test input	Expected outcome	Obtained outcome
1. Sign In	Input Mobile Number	Get 6 digit code from Google	Successful Sign In
2. Public university	Show all University Name	Show all University Name with logo	Shown successful
3. Dhaka university	Show four card view	Show four card view successfully	Successfully
4. Dhaka university information	Show all information of Dhaka university	Show successfully	Successfully
5. Dhaka university Admission information	Show some link for admission information	Student show successfully	Successfully
6. Dhaka university previous question	Show four unit	Show previous question for any unit successfully	Successfully
7. Model Test	Show four unit	Show all question Set successfully	Successfully Done the model test

Table: 5.4 Testing Implementation

5.5 Test Results and Reports

5.5.1 Sign In verification

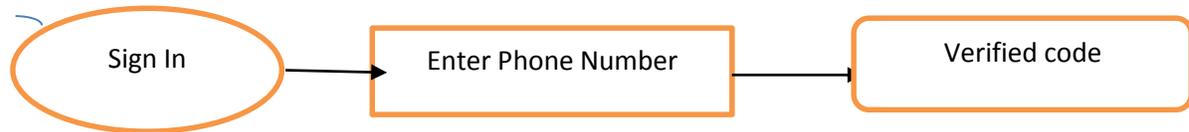


Figure: 5.5.1 Sign In verification

User Enter the Sign In button, Then need to phone number, Enter the verified code. Successfully done the Sign In.

5.5.2 Logical Table for Sign In

Test Case	Test Input	Expected outcome	Obtained outcome
Sign In	01*****89	028***	Successfully Sign In

Table: 5.5.2 Logical Table for Sign In

Similarly implementing the every part of this project we can successfully done our job. So we can say that our project is successfully done.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 Discussion and Conclusion

We think that through this project, students will benefit greatly from taking admission in university. Students will get all the information of the admission through an application. Besides, they will be able to test themselves through model tests.

We have learned a lot from doing this project. This project has greatly enhanced our thinking. We benefit greatly from the students as well

6.2 Scope for Further Developments

Future development

1. We will try to add technology and engineer universities later
2. We will try to ensure that the students can fill out the admission form through this application
3. We will increase our model test question number later
4. And I'll try to work out where the bugs are

Appendices

Appendix A: Project Reflection

We started our project from February to implement this application. Our main motive to this project is to find a way of proper guide line for every student to help the admission test. After compiling this project apse we think it will play a much more effective role for an examiner. After a long a Journey and hard work, we reach our goal.

Appendix B: Related Issues

Before working on this project we have to collect many question for each individual university. It was very tough and very challenging work for us . It was a challenge for us to set questions and make a marking system at the data base. We implement our project on android studio and fire-base database.

REFERENCE

[1] Get raw fundamental idea of Firebase, Available at:

<<https://firebase.google.com/?gclid=Cj0KCQjw9fntBRCGARIsAGjFq5GBAAOoWxw5vR9nVVrn3KnsRj279w1enp_uPiVEYVpCcC_-iHDoIiEaAgp5EALw_wcB>>

[2] Android xml design Available at: <<<https://code.tutsplus.com/tutorials/getting-started-with-xml-layouts--mobile-12749>>>

[3] Database Reference Available at:

<<<https://firebase.google.com/docs/reference/android/com/google/firebase/database/DatabaseReference>>>

[4] Android studio SDK at: <<<https://developer.android.com/studio>>>

Admission Helper Report

ORIGINALITY REPORT

15%

SIMILARITY INDEX

12%

INTERNET SOURCES

1%

PUBLICATIONS

13%

STUDENT PAPERS

Rectangular Snip

PRIMARY SOURCES

1

firebase.google.com

Internet Source

5%

2

en.wikipedia.org

Internet Source

3%

3

Submitted to Daffodil International University

Student Paper

2%

4

Submitted to Laureate Education Inc.

Student Paper

1%

5

www.basicknowledge101.com

Internet Source

1%

5 www.basicknowledge101.com 1%
Internet Source

6 blog.coepd.com 1%
Internet Source

7 www.cslabels.co.uk 1%
Internet Source

8 Submitted to Bridgepoint Education <1%
Student Paper

9 Submitted to Universiti Teknologi MARA <1%
Student Paper

10 Submitted to Bahrain Polytechnic <1%
Student Paper

11 Submitted to Kingston University <1%
Student Paper

12 Submitted to Vaal University of Technology <1%
Student Paper

13 Submitted to Limerick Institute of Technology <1%
Student Paper

14 Submitted to National University of Ireland,
Galway <1%
Student Paper

Exclude quotes Off

Exclude matches Off

Exclude bibliography Off