

**ANDROID – BASED SMART GUARDIAN ASSISTANT**

**MD. MEHEDI HASAN SAGOR**

**162-15-8172**

**MD. AL-KAFI**

**162-15-7766**

**AND**

**BULBUL AHMED**

**162-15-8167**

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

Supervised By

**Fahad Faisal**

Assistant Professor

Department of CSE

Daffodil International University

Co-Supervised By

**Md. Azizul Hakim**

Lecturer

Department of CSE

Daffodil International University



**DAFFODIL INTERNATIONAL UNIVERSITY**

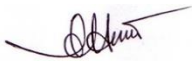
**DHAKA, BANGLADESH**

**SEPTEMBER-2020**

## APPROVAL

This Project titled “**Smart Guardian Assistant**”, submitted by **Bulbul Ahmed 162-15-8167**, **Md. Mehedi Hasan Sagor 162-15-8172** and **Md. Al-Kafi 162-15-7766** 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 October 07, 2020.

## 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. Sadekur Rahman**  
**Assistant Professor**

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

**Internal Examiner**

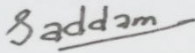


---

**Subhenur Latif**  
**Assistant Professor**

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

**Internal Examiner**



---

**Dr. Md. Saddam Hossain**  
**Assistant Professor**

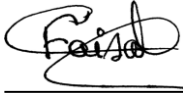
Department of Computer Science and Engineering  
United International University

**External Examiner**

## DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Fahad Faisal, Assistant 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:**



---

**Fahad Faisal**  
Assistant Professor  
Department of CSE  
Daffodil International University

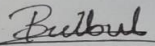
**Co-Supervised by:**



---

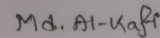
**Md. Azizul Hakim**  
Lecturer  
Department of CSE  
Daffodil International University

**Submitted by:**



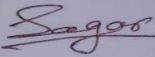
---

**Bulbul Ahmed**  
ID: 162-15-8167  
Department of CSE  
Daffodil International University



---

**Md. Al-Kafi**  
ID: 162-15-7766  
Department of CSE  
Daffodil International University



---

**Md. Mehedi Hasan Sagor**  
ID: 162-15-8172  
Department of CSE  
Daffodil International University

## ACKNOWLEDGEMENT

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

We really grateful and wish our profound our indebtedness to **Fahad Faisal, Assistant Professor of CSE** Daffodil International University, Dhaka. Deep Knowledge & keen interested of our supervisor in the field of “*Mobile Application Development*” 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 stage have made it possible to complete this project.

We would like to express our heartiest gratitude Prof. Dr. Syed Akhter Hossain, Head of and Head, Department of CSE for his kind help to finish our project and to other faculty member and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

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

## **ABSTRACT**

This project name is Android and IOS – based smart guardian assistant. We know the present era of internet with the passage of time the digital platform has become so popular at almost every phase of our life. Mainly we want to make our educational life digital also that was the driving force for us to develop a system that can be used. We try to arrange a way of communication between teacher and guardian with to inform guardian about the student's activities and results. This project makes more user-friendly fee transaction. Guardians can check balance of ID with students can pay the bills of canteen without any hassle. Our project saves enough time for students, guardians also teachers because of this project create more appropriate attendance system. One part is for user application And Second part is for admin panel can control the transactions. This project will be very useful because it will make good communication between Teacher and guardian with reduce time.

# TABLE OF CONTENTS

## CONTENTS

APPROVAL.....	i
DECLARATION .....	ii
ACKNOWLEDGEMENT .....	iii
ABSTRACT.....	iv
<b>CHAPTER 1.....</b>	<b>1</b>
1.1: Introduction.....	1
1.2: Motivation .....	1
1.3: Objective.....	1
1.4: Expected Outcome .....	2
1.5: Report Layout .....	2
<b>CHAPTER 2.....</b>	<b>3</b>
2.1: Introduction.....	3
2.2: Related Work .....	3
2.3: Comparative Studies.....	4
2.4: Scope of the Problem .....	4
2.5: Challenges.....	5
<b>CHAPTER 3.....</b>	<b>6</b>
3.1: Introduction.....	6
3.2: For Desktop Application (Admin) .....	6
3.3: For Android Application (User).....	7
3.3.1: Guardian .....	7
3.3.2 Teacher .....	9
<b>CHAPTER 4.....</b>	<b>10</b>
4.1: Business Process Modeling.....	10
4.2: Requirement Collection and Analysis.....	10
4.3: Use Case Modeling and Description.....	11
4.4: Design Requirements.....	12
4.5: Flowchart.....	12
<b>CHAPTER 5.....</b>	<b>13</b>
5.1: DESKTOP APPLICATION.....	13

Step1: Admin Log in Option .....	13
Step2: Add Student .....	14
Step3: Add Teacher.....	15
Step4: Add Notice .....	16
Step5: Delete & Update Student Data.....	17
Step6: Delete & Update Teacher Data .....	18
Step7: Student Activity .....	19
Step8: Account.....	20
Step9: Canteen.....	21
Step10: Tuition Fee .....	22
5.2: ANDROID APPLICATION .....	23
Front End Design.....	23
<b>CHAPTER 6 .....</b>	<b>36</b>
6.1: Implementation of Database.....	36
6.2: Implementation of Front-end Design .....	36
6.3: Implementation of Interactions.....	36
6.4: Testing Implementation .....	36
6.5: Test Results and Repots.....	37
<b>CHAPTER 7 .....</b>	<b>38</b>
7.1: Discussion and Conclusion.....	38
7.1.1: Limitations of Our application .....	39
7.2: Scope for Future Developments .....	39
<b>Appendices.....</b>	<b>40</b>
<b>References .....</b>	<b>43</b>

## LIST OF FIGURES

Figure 4.1: Business Process Modeling .....	10
Figure 4.2: Use Case Modeling and Description .....	11
Figure 4.3: Flowchart.....	12
Figure 5.1:Screenshot of Log In with User Name .....	13
Figure 5.2:Screenshot of Add Student Page .....	14
Figure 5.3: Screenshot of Add Teacher Page.....	15
Figure 5.4: Screenshot of Add Notice Page.....	16
Figure 5.5: Screenshot of Delete & Update Student Data Page.....	17
Figure 5.6: Screenshot of Delete & Update Student Data Page.....	18
Figure 5.7: Screenshot of Student Activity Page .....	19
Figure 5.8: Screenshot of Account Page .....	20
Figure 5.9: Screenshot of Canteen Page .....	21
Figure 5.10: Screenshot of Tuition Fee Page.....	22
Figure 5.11: Screenshot of Our Application icon. ....	23
Figure 5.12: Screenshot of User Log in page.....	24
Figure 5.13: Screenshot of Student User Profile.....	25
Figure 5.14: Screenshot of Teacher User Profile Menu Bar .....	26
Figure 5.15: Screenshot of Teacher User Profile Menu Bar .....	27
Figure 5.16: Screenshot of Student Activity .....	28
Figure 5.17: Screenshot of Student Result.....	29
Figure 5.18: Screenshot of Notice.....	30
Figure 5.19: Screenshot of Messenger .....	31
Figure 5.20: Screenshot of Account Report.....	32
Figure 5.21: Screenshot of Expense Report.....	33
Figure 5.22: Screenshot of Upload Mark.....	34
Figure 5.23: Screenshot of Take Attendance .....	35

## LIST OF TABLES

Table 6-1: Test Case for Project on Secure Messaging .....	37
--	----



# CHAPTER 1

## Introduction

### 1.1: Introduction

The Project is an android and IOS based project. This is a mobile application that is used to easy for busy life. First of all, when we design our project, we have made some plan. Because, this application will use guardians and teachers of different phones. So, the UI design must be user friendly and the activity must easily understandable. Basically, this project is divided into two parts. One part is user part that will be used by all student guardians, teachers and students. And other part is that will be used by admin officers. User can login, easy to see payments reports, notice, attendance history, result, message by this application. Admin officer can control everything. If all is ok with system, then teachers and guardians are use our application very easily. It will be very useful to reduce time, good effective for students and good communication with teachers and guardians.

### 1.2: Motivation

Our system provided all of digitalized system. When our teachers are taken attendance by analog. That's why we are thinking about digital attendance for everyone. Our system has regular attendance system for reduce time. Then fees payment is the big problem for student and guardians. There needs a serial line for payment fees. This is another problem for everyone. That's why we provided online fees payment by using our application [3]. Then students are taken foods in canteen and bill pay by this system. Which payments are given guardian can know about by our system? So, guardians can easily maintain our provided service without any suffering.

### 1.3: Objective

The main goals of us under the android applications are:

- i. We want to reduce the suffering of hand cash in fees payment.
- ii. We want to reduce time for both of teachers and guardians.
- iii. We want to reduce the suffering of retail cash canteen bill.

- iv. It is fully digital platform for paying money in online payment gate.
- v. A user-friendly interface where guardians easily login for needed service.
- vi. It will make safe for students and comfortable for guardians.
- vii. Users can see the all payments history.

#### **1.4: Expected Outcome**

We want to run this project from help in ICT division. We export the result of our project to be completely correct and precise. The major infrastructure accessible for good communication and guardian gets proper update his/her own child by our application. User should feel more comfortable to paying canteen bill and fees payment using this application. In this time, we want to work with government level of all government school in full Bangladesh. We want to produce our service all most 100 schools in one year. If it works properly then our education system developed day by day and guardians are leading very comfortable life for his/her child without any suffering.

#### **1.5: Report Layout**

A complete description of our process and related activities and terminology will be slowly provided in this mobile application. In chapter 1, we discuss about introduction, motivation, objectives, expect outcome of our project. In chapter 2, we discuss introduction, related work, comparative studies, scope of the problem and challenges. This chapter we talk about focus and backdrop of the project. In chapter 3, we design and poised business process modeling, Requirement collection and analysis, Use case modeling and description, Logical data model, design requirement in our project. In chapter 4, we give front end design, back end design and implementation requirement in our project. In chapter 5, we described implementation of firebase database and MySQL, interaction, testing result and report of our project. In chapter 6, we described discussion and conclusion and limitation of this application. This chapter we talk about conclusion, future work and limitation of our project.

## **CHAPTER 2**

### **Background**

#### **2.1: Introduction**

This is an android and IOS application which is developed good communication with teachers and guardians. Our application can divergence between guardian and teacher by messaging [11]. This service can reduce gap for both. Then this application which is online fees payment that is reducing hand cash problem, paying canteen bill [3]. Guardians can also see their children are present or not today in the school by our attendance system.

Teachers are sending all information or notice to guardian that's why they are known about everything. In this project we worked some month or days in field level. We talk about our project in different schoolteachers and guardians. They were very much interested to using this method application. We described our project with some general people. They were appreciated in this project and wishes good luck. We started to collect some information and found new services which are important and include this project. After completing our work in field level then we started our main flow of work. First, we design front end and then started working in back end. Then we started admin panel work in desktop platform using MySQL database. We were so much excited for this project because we were facing this problem in every moment in our regular day life.

#### **2.2: Related Work**

We see one or two year ago some university almost use some service [1]. But if a guardian could be link up with teachers then they are getting update everything, and they can talk to each other by messaging in every day [11]. This is our unique idea in Bangladesh scenarios. We noticed that, nexus pay Dutch Bangla bank have already generated a method for paying bill. They have many types of payment gateway application in our country like bKash, nexus-pay [3]. When we worked in field level and talk about many different guardians and teachers. They were not interested on Dutch Bangla system. Because, they

must open a bank account. But in our applications, they can easily pay fees payment and canteen bill. So, it is very easy for users. We know already continue so much place online attendance [7][9] system by using fingerprint [8], CC camera and canteen bill payment system already running some university. But Our main target in this process is, which food are taking their child that's information is see guardians. This is another new idea in our country. Most probably some students are not appreciating this process, but this is good for everyone. So, this system is very helpful for users.

### **2.3: Comparative Studies**

We have developed our project to use it as fully Android and IOS based application benefits to satisfy the fantastic aspect of the problem. We will have a proper digitalized educational environment containing features like smart communication with guardians and teachers, smart transaction, guardian assistant, digital attendance and smart payment system. So, our system makes educational life more comfortable and will help us to lead better life.

### **2.4: Scope of the Problem**

Yet accessing all the relevant document at a period is a very hard and its resource-consuming to do all those issues as well. So, from this perspective, we want to build up a mobile application is very helpful that payment gateway for user on mobile in a single platform. Important and tough work is field work. Users were interesting about our project. But, some of user or guardians and teachers are not use smart phone. Then we described manual option will be ok. When we build our venture, we need to so many fields work. Then we will try how to communicate with teachers and guardians. But in this moment, we

want to do for only smart phone users. When we talk to guardians, then we understand that so much guardians are not updated the smart phone. They are not using smart phone properly. So, we will try understood to people and teach them how to use our application. Finally, we published our project properly.

## **2.5: Challenges**

When we must do some task, certain difficulties must be faced. If we choose to build up our project, we take our option into consideration. Neither task can be smooth at all. Likewise, we will face certain obstacle in order develop of our project. Expenditure in the first phase is a big problem for us. Another problem we face in the suitable compilation of information. In this venture, when we work on admin panel, we faced some problem. The problem is user part admin desktop-based and we connected MySQL database. But our user part is android-based admin panel. We want to connect it MySQL database. We want our user part was connected MySQL and firebase database. This was very tough for us. But we could not connect MySQL database easily. Quality and portrayal application depend on user interface design and payment method. Payment gateway is tough for us. We based on user friendly service development with payment method in our project.

## CHAPTER 3

### Our Services

#### 3.1: Introduction

Our project is Android and IOS based application and that is Smart Guardian Assistant. We want to make our educational life digital also that was the driving force for us to develop a system that can be used. There has mainly two parts. One is for desktop application and another for mobile application. Our main service is conversation with teachers and guardians. Especially in the case of working parents the rate is huge. These two parts has different type of services. We described here about our target services.

#### 3.2: For Desktop Application (Admin)

Only Admin can control this service. There no needs to android application for admin. All type of works completes by desktop. The main Service of us under the desktop applications are:

- i. **Add Student:** Mainly this is the registration process for students. Admin can create student guardians' profile and can change anything anytime. Admin can add student here also.
- ii. **Add Teacher:** Mainly this is the registration process for teachers. Admin can create student guardians' profile and can change anything anytime.
- iii. **Add Notice:** Admin can post any notice you need when it comes to this notice page. Admin can add notice any time and can change or delete any time. Through this service, admin can always connect with teachers and parents.
- iv. **Delete Students Data:** If admin thinks, there has unnecessary information or data, and then admin can delete the student's data anytime.

- v. **Delete Teachers Data:** If admin thinks, there has unnecessary information or data, and then admin can delete the teacher's data anytime.
- vi. **Student Activity:** Admin will send ideas to parents about how students are attending regular classes. Admin can continue to inform parents of where the students are performing or can take any kind of notice. If students do not come to class for any reason, admin will be able to send news to parents when they see this attendance. If admin want, admin can get good ideas or any initiative by seeing the attendance of the students.
- vii. **Account:** Students all types of transactions admin can tell students guardian how much money has been deposited or how much money is owed. If something goes wrong here, then admin can change or edit anytime if needed changes.
- viii. **Canteen:** Admin can stay up to date on what's on the canteen. Admin can keep the list of what students are eating. How much money is spent on food, they will be able to collect the transaction and keep the admin. And admin will be able to inform parents of the list and transaction.
- ix. **Logout:** Admin can anytime login and can anytime logout if needed.

### **3.3: For Android Application (User)**

There have two types of android application user. One is guardian and other is teacher. Their services are different. But our all services are very easy to use. We described here of all android application services.

#### **3.3.1: Guardian**

Things that only parents can do by using this application, all of these are described here. The main Service of us under the android application for guardians are:

- i. **View Notice:** Parents will be able to all kind of notices immediately. By looking at these notices they can easily keep up with all kinds of news. Seeing the notices allows their children to make the right decisions at the right time. So, this service is very important for both of guardians and students.
- ii. **Student Activity:** Parents can see their child activities here, where their children are performing. By watching these activities here, parents will be able to better care for their children. By watching student activity guardian can make good decisions at any time. This service is very helpful for users.
- iii. **Message:** This is our main service in our application. If a parent can talk to the teachers on a regular basis, the parents can be sure if this suggestion is given. We know, there have lot of divergence between guardian and teacher. So, using this service, guardian can message with teachers.
- iv. **Result:** When all kind of academic result is published then guardian can see it. That's why guardians can more take care about their children. If parents are aware of this, students will be able to study responsibly.
- v. **Account:** The guardian can see how much money has been deposited, how much money is owed. Parents can find out about the date of deposit and types of transaction.
- vi. **Payments Report:** Here guardians can find the report where and when how much payments. If they see the news about the payment, they will be alert in advance about the transaction. Because of this, there will never be any pressure on the parents.
- vii. **Logout:** Guardians can anytime login and anytime logout from this application.



### 3.3.2 Teacher

Things that only teachers can do these services by using this android application, all of these are described here. The main service of us under the android application for teachers are:

- i. **View Notice:** Teachers will be able to all kind of notices immediately. By looking at these notices they can easily keep up with all kinds of news. Seeing the notices allows their students to make the right decisions at the right time.
- ii. **Upload Mark:** Teachers can upload student's exam mark. If there needed to change then teachers can any time edit or delete the mark.
- iii. **Take Attendance:** Teachers can take attendance regular by using this application. Only teacher can edit it.
- iv. **Message:** Sometimes teachers need to talk to their parents about their students. That's why teachers can text with guardians. So, our message system is the very important for this application. This service can reduce divergence between teacher and guardian.
- v. **Logout:** Teachers can anytime login and anytime logout from this application. It is very easy to user.

## CHAPTER 4 Requirement Specification

The specific need state what a framework must be completed. The specific requirement state what should be done together to overcomes the association incentive. Specific requirement is specific many types of model like business process modeling, Use case diagram, Flowchart.

### 4.1: Business Process Modeling

The following figure 4.1, the activity begins by logging in. If the user already login then go home activity. When the user will complete login then he will go Menu activity. If user never login then he must be registration and login.

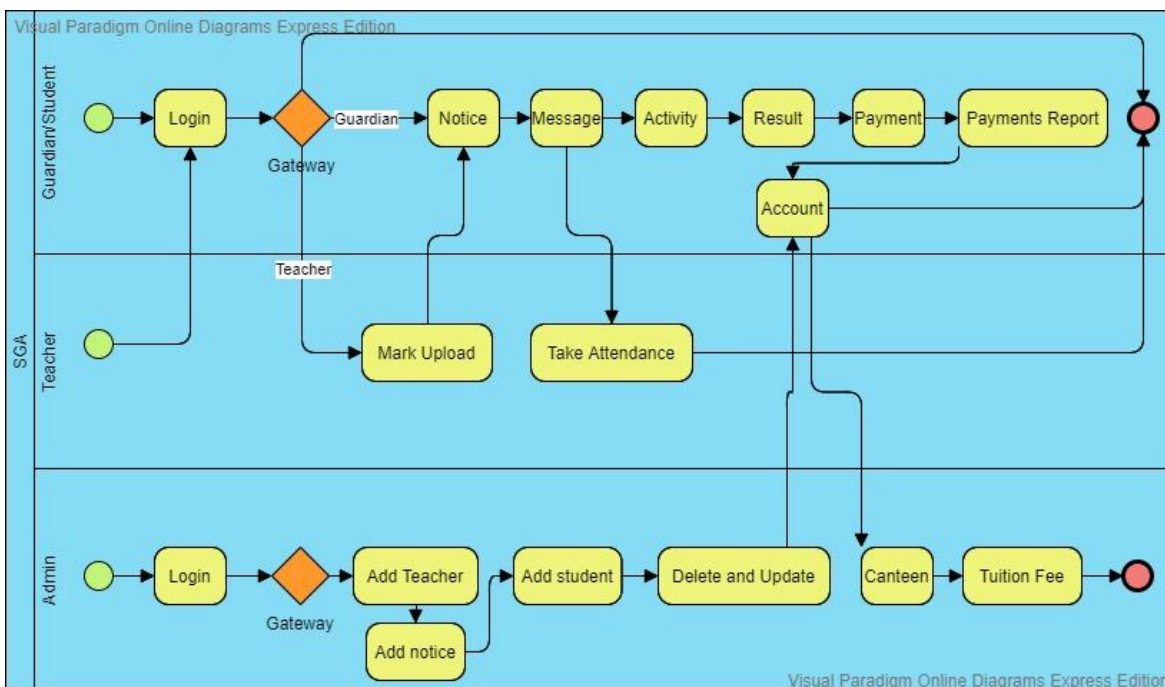


Figure 0.1: Business Process Modeling[16]

### 4.2: Requirement Collection and Analysis

For completing this application, we used Android Studio Software. We use Dart language to build this application. And for data store we use MySQL and Firebase as it is secure and reliable. We also use many Rest API with raw PHP for getting data [4][12]. After doing all of that we tested to some user. We changed some of UX design part according to user choice.

### 4.3: Use Case Modeling and Description

The use case modeling is a useful tool for securing preconditions. This provides visual overviews of the brand system. Use case model is very useful for our smart guardian assistant android base mobile application. The following figure 4.2, this has three actors Guardian/Student, Teacher and Admin and nine use case.

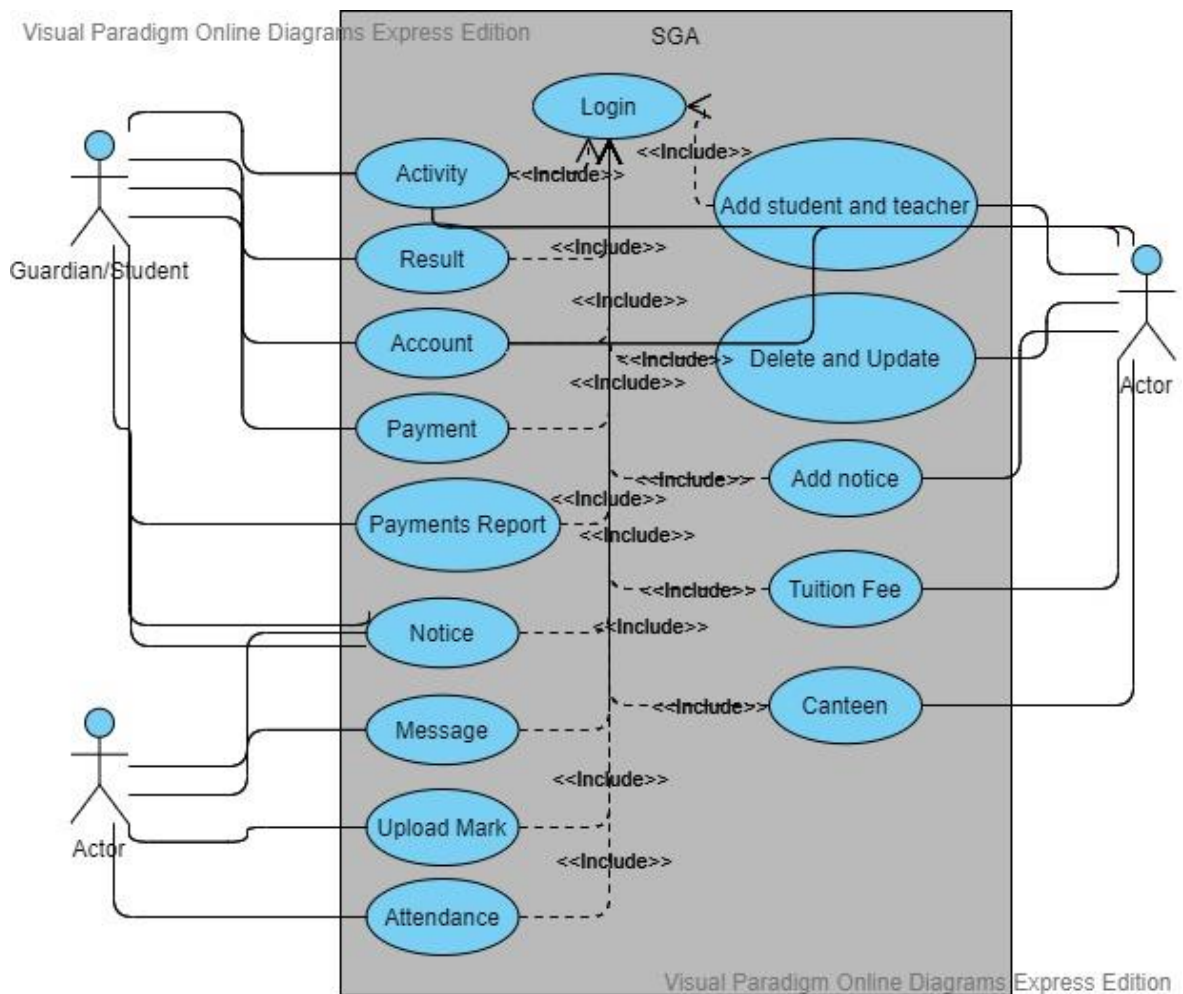


Figure 0.2: Use Case Modeling and Description

#### 4.4: Design Requirements

- We have an encryption in our process for our user. We have design and model user login. We have a model the user login his/her children ID number authentication
- We design a model admin panel. He can login with admin ID and password.
- We make architecture for check result, attendance, payments report for user in our project.
- We have a layout for payment method by payment gateway [3].
- We add to design permission declare by admin.
- We also design a messenger list chart and show data using firebase database [2][11].

The user interface is designed using Flutter Framework. Adobe PS used to design logo and icon photo design. We used draw.io online drawing tool for drawing business process modeling, use case diagram that are very helpful for our venture.

#### 4.5: Flowchart

Flowchart is very useful for our smart guardian assistant Android and IOS base mobile application. The following figure 4.3.

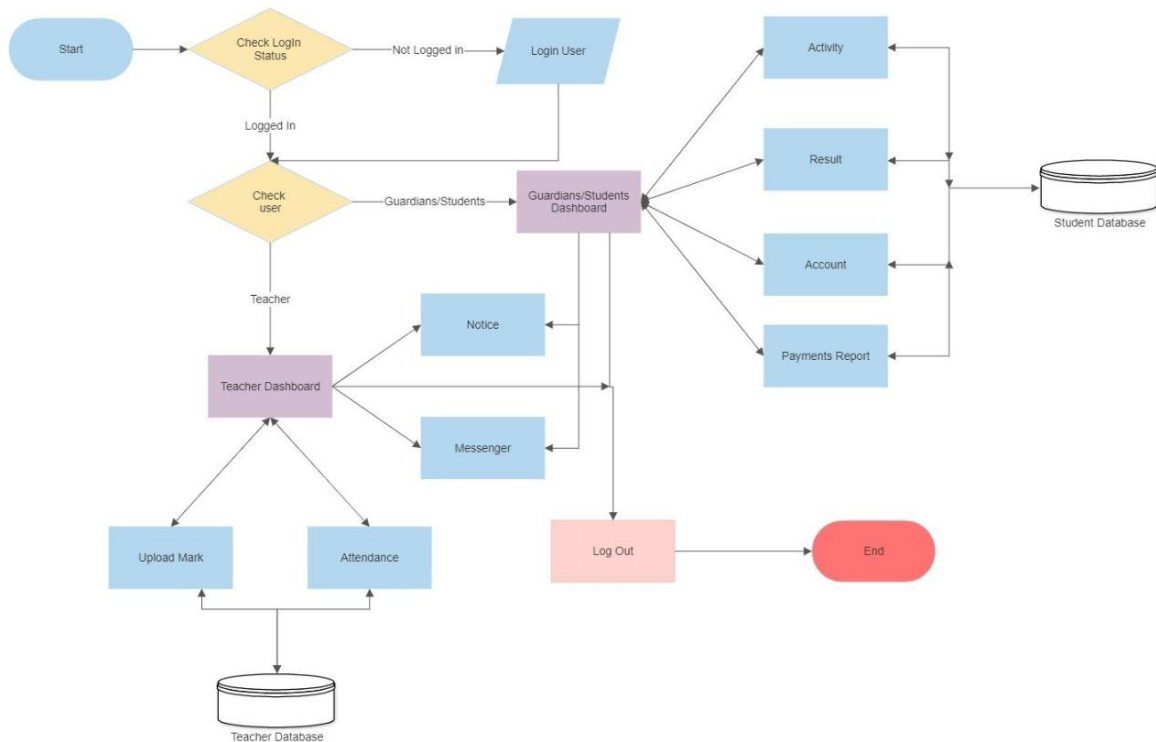


Figure 0.3: Flowchart [15]

## CHAPTER 5 Design Specification

### 5.1: DESKTOP APPLICATION

#### Step1: Admin Log in Option

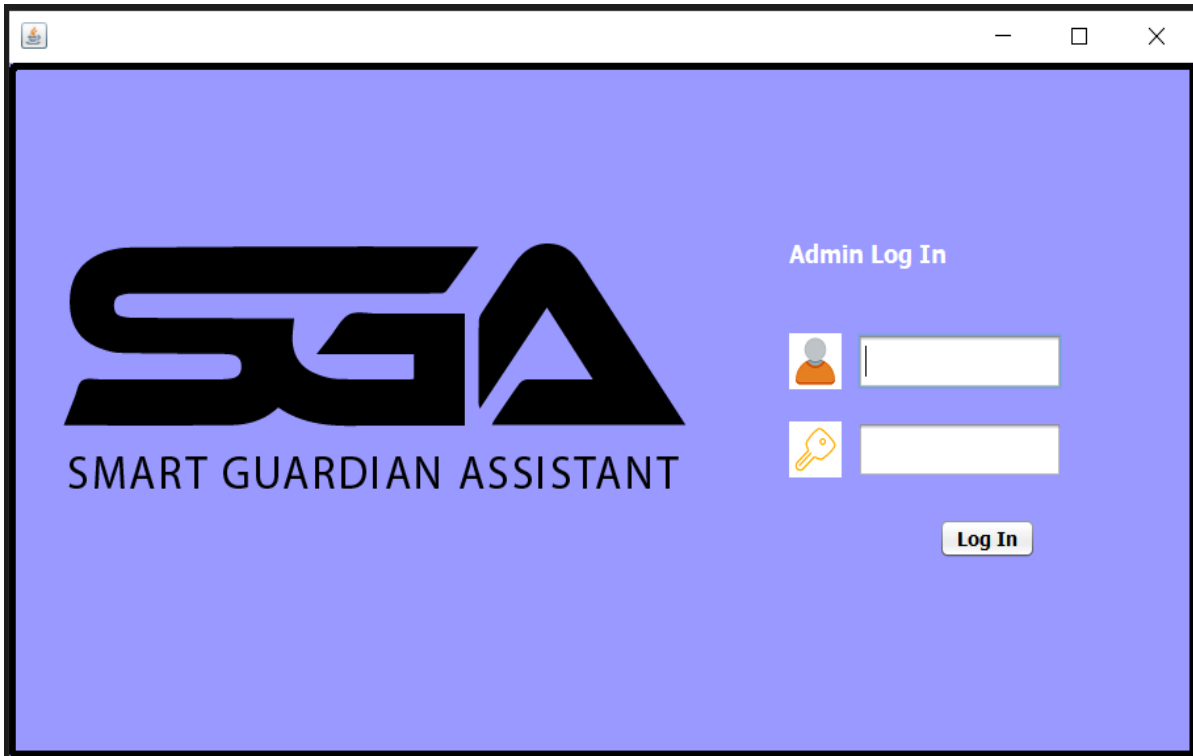


Figure 0.1: Screenshot of Log In with User Name

The following figure 5.1, log in layout, admin can easily login via username to the digital smart guardian assistant application. We give here only one types of log in schemes.

## Step2: Add Student

The screenshot shows a web application window titled "Add Student". On the left is a dark blue sidebar with a list of menu items: "Add Student", "Add Teacher", "Add Notice", "Delete & Update Student Data", "Delete & Update Teacher Data", "Student Activity", "Account", "Canteen", "Tuition Fee", and "Log Out". The "Add Student" menu item is highlighted in light blue. The main content area has a dark blue header with the "SSA SMART GUARDIAN ASSISTANT" logo and the title "Add Student". Below the header is a form with the following fields: "Id" (text input), "Name" (text input), "Father's Name" (text input), "Mother's Name" (text input), "Class" (dropdown menu with "06" selected), "Roll No" (text input), "Group" (dropdown menu with "Null" selected), "Sex" (dropdown menu with "Male" selected), "Blood Group" (dropdown menu with "A+" selected), "Mobile No" (text input), "Father's Mobile" (text input), "Mother's Mobile" (text input), and "Address" (text area). A "Submit" button is located at the bottom right of the form.

Figure 0.2: Screenshot of Add Student Page

The following figure 5.2, add student page. Make this option available to add new students. This process is also student's registration process. Students all information needed to complete it. We collect a lot of information from students for adds student process. First of all we create a student permanent id. Next time students needed this id for log in. That's why this id is very important for students. Then needs to student's name, Fathers name, Mothers name, Class, Roll No, Group, Sex, Blood group, Students mobile number, Fathers mobile number, Mothers mobile number and full address. If all information is input here, then a student can add here. Then our data base takes all information.

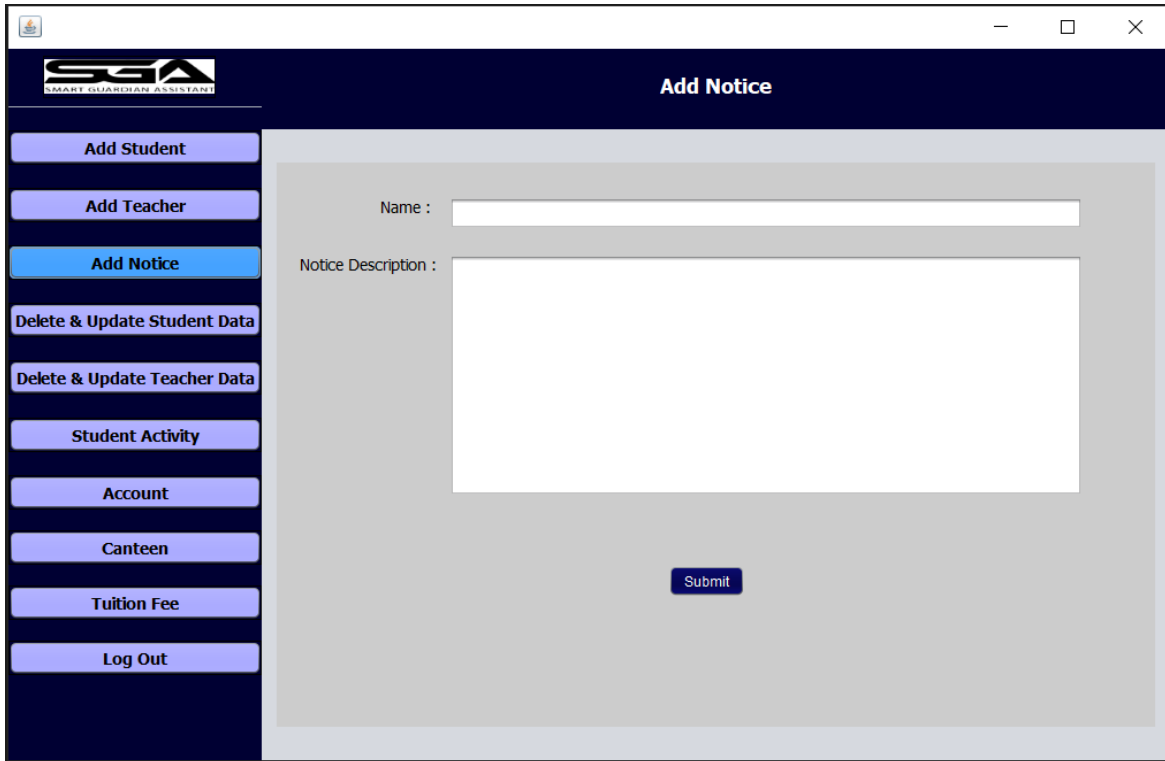
### Step3: Add Teacher

The screenshot shows a web application interface for adding a teacher. The page title is "Add Teacher". On the left, there is a sidebar with navigation options: "Add Student", "Add Teacher" (which is highlighted in blue), "Add Notice", "Delete & Update Student Data", "Delete & Update Teacher Data", "Student Activity", "Account", "Canteen", "Tuition Fee", and "Log Out". The main content area contains a form with the following fields: "Id" (text input), "Name" (text input), "Father's Name" (text input), "Mother's Name" (text input), "Blood Group" (dropdown menu with "A+" selected), "Sex" (dropdown menu with "Male" selected), "Mobile No" (text input), "Email" (text input), and "Address" (text area). A "Submit" button is located at the bottom right of the form.

Figure 0.3: Screenshot of Add Teacher Page

The following figure 5.3, add teacher page. Make this option available to add new teacher. If a new teacher needed, then we add here. This process is also teacher's registration process. Teachers all information needed to complete it. We collect a lot of information from teachers for adds teacher process. First, we create a teacher's permanent id. Next time teachers needed this id for log in. That's why this id is very important for teachers. Then needs to teacher's name, Fathers name, Mothers name, Sex, Blood group, Teachers mobile number and full address. If all information is input here, then a teacher can add here. Then our data base takes all information.

## Step4: Add Notice



The screenshot shows a web application window titled "Add Notice". The window has a dark blue header with the SGA logo and the text "SMART GUARDIAN ASSISTANT" and "Add Notice". On the left side, there is a vertical sidebar with several buttons: "Add Student", "Add Teacher", "Add Notice" (which is highlighted in blue), "Delete & Update Student Data", "Delete & Update Teacher Data", "Student Activity", "Account", "Canteen", "Tuition Fee", and "Log Out". The main content area is light gray and contains a form with two input fields: "Name :" followed by a text input box, and "Notice Description :" followed by a larger text area. Below the text area is a "Submit" button.

Figure 0.4: Screenshot of Add Notice Page

The following figure 5.4, add notice layout. This layout creates for added some notice from admin or teachers. First option is the name or title for notice. Then notice description box here. If all information input here, then there has submit button to goes everywhere. Admin, Students, Teachers both of can see this notice. Admin can change notice privacy secure.



## Step5: Delete & Update Student Data

The screenshot shows a web application window titled "Delete And Update Student Data". The sidebar on the left contains the following buttons: Add Student, Add Teacher, Add Notice, Delete & Update Student Data (highlighted), Delete & Update Teacher Data, Student Activity, Account, Canteen, Tuition Fee, and Log Out. The main content area contains a form with the following fields and controls:

- Id :**
- Name :**
- Father's Name :**
- Mother' Name :**
- Class :**
- Roll No :**
- Group :**
- Sex :**
- Blood Group :**
- Mobile No :**
- Father's Mobile :**
- Mother's Mobile :**
- Address :**

At the bottom of the form, there are two buttons:  and .

Figure 0.5: Screenshot of Delete & Update Student Data Page

The following figure 5.5, delete and updated student data layout. Already we are talking about add student data process. There we collect lot of information from students. So, we know which information is collecting from students. If there needed any information delete or updated, then we complete it there very easily. There has search option. Which student's data delete, or update needed only his/her id input here, then we found that student. Then there has update and delete option, just choose and complete the target process.

## Step6: Delete & Update Teacher Data

The screenshot shows a web application window titled "Delete And Update Teacher Data". The interface includes a sidebar on the left with the following menu items: Add Student, Add Teacher, Add Notice, Delete & Update Student Data, Delete & Update Teacher Data (highlighted), Student Activity, Account, Canteen, Tuition Fee, and Log Out. The main content area contains a form with the following fields and controls:

- Id :** Text input field with a "Search" button to its right.
- Name :** Text input field.
- Father's Name :** Text input field.
- Mother' Name :** Text input field.
- Blood Group :** Dropdown menu with "A+" selected.
- Sex :** Dropdown menu with "Male" selected.
- Mobile No :** Text input field.
- Email :** Text input field.
- Address :** Large text input field.
- Update** and **Delete** buttons at the bottom of the form.

Figure 0.6: Screenshot of Delete & Update Student Data Page

The following figure 5.6, delete and updated teacher data layout. Already we are talking about add teacher data process. There we collect lot of information from teachers. So, we know which information is collecting from students. If there needed any information delete or updated, then we complete it there very easily. There has search option. Which teacher's data delete, or update needed only his/her id input here, then we found that teacher. Then there has update and delete option, just choose and complete the target process.

## Step7: Student Activity

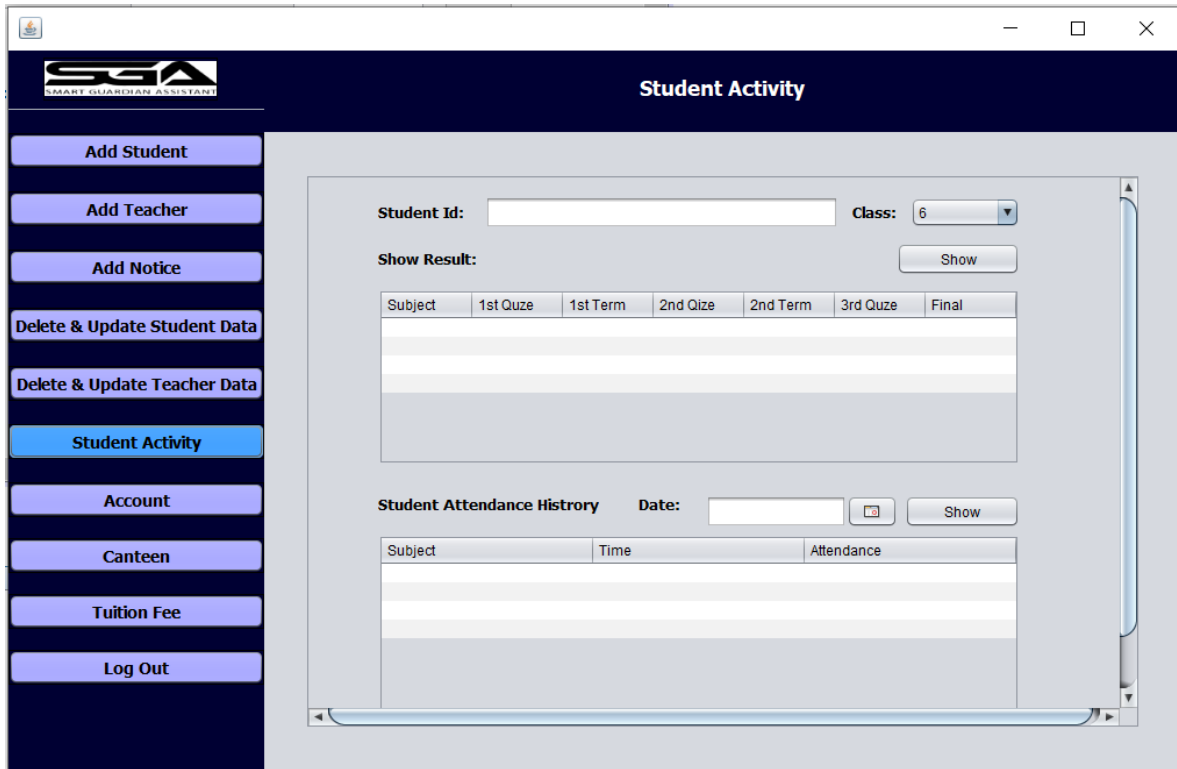


Figure 0.7: Screenshot of Student Activity Page

The following figure 5.7, student activity layout. This option is very important for our project. Students result show here. Subject wise quiz, term all result show here. Then there has student's attendance history. Subject and date wise attendance are show here. That's why guardians are take care about his/her child very easily.

## Step8: Account

Amount	In	Out

Figure 0.8: Screenshot of Account Page

The following figure 5.8, account layout. When needs load money in student's id card, that's moment this process will be work. Just input the money amount and card number, then money adds to student's ID card. Students all transaction are show here. Admin can search student ID very easily and that's student transaction will be show here.

## Step9: Canteen

The screenshot shows the 'Canteen' page of the SSA application. On the left is a sidebar with navigation options. The main area contains a form for managing food items and orders. The form has sections for adding new food, selecting an item (currently 'Birany'), specifying quantity and price, viewing a list of items, and calculating the total amount. Below this is a payment section with fields for card number and ID number, and buttons for scanning or online payment.

Figure 0.9: Screenshot of Canteen Page

The following figure 5.9, canteen layout. Admin can add all kind of food list here. Just input food name and submit here. There has select food option, food quantity, amount per item, list of food and show here total amount. Then a student can order here and payment by card very easily. Then needed student's card number and id number. So, this way to students takes food very easily and guardians are known about their child's food list.

## Step10: Tuition Fee

The screenshot shows a web application interface for managing tuition fees. On the left is a vertical sidebar with navigation options. The main area is titled 'Tuition Fee' and contains a form for entering student information and payment details. The form includes fields for class selection, total amounts, student identification, and payment method options.

Figure 0.10: Screenshot of Tuition Fee Page

The following figure 5.10, tuition fee layout. This way to s student can payment their tuition fee very easily. That's for needed students some information how much amount payable and there has payment of way. And there needed students card number and id number.

## 5.2: ANDROID APPLICATION

### Front End Design

User interface development is the process of creating interfaces in software and electronic product with an emphasis on aesthetics and fashion. We are building a front-end design that is easier for user to manage. We designed the front-end architecture for all type of user who will able to be controlling this application easily. We used Flutter framework front end architecture [14][6].

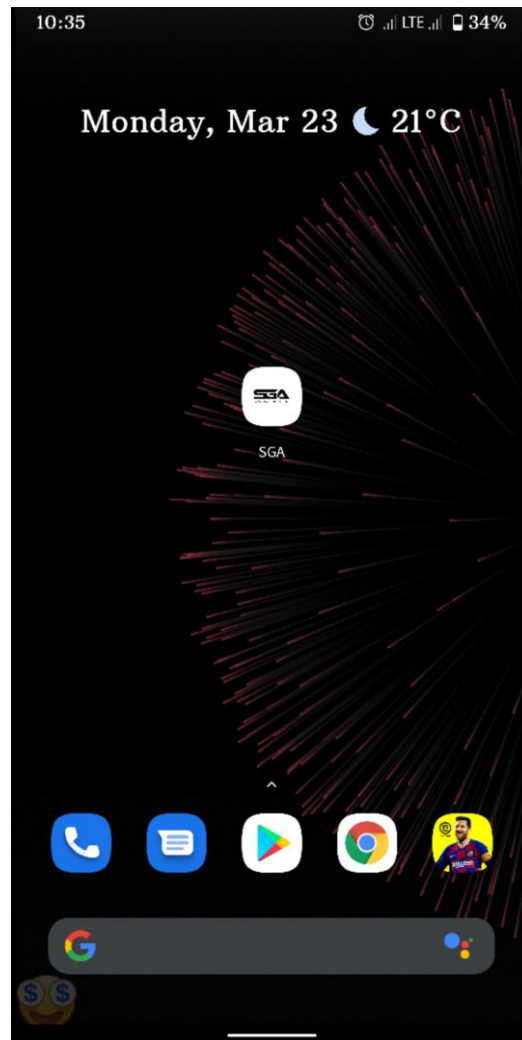


Figure 0.11: Screenshot of Our Application icon.

This is our application icon. When we design our application icon, we think about our user. We designed our application icon by using adobe Photoshop and some online tool. These are very helpful for our icon designing. Our icon is very simple.

## Step1: User Log in Option

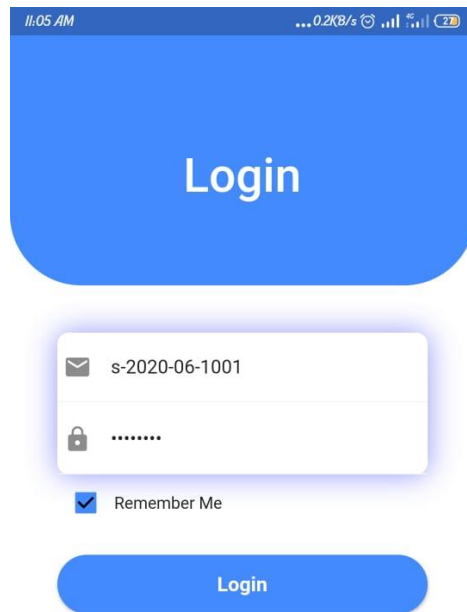


Figure 0.12: Screenshot of User Log in page

The following figure 5.12, log in layout. User can easily login via ID number to the smart guardian assistant application. We give here only two type of log in schemes, one Gaudian/Student and another login scheme is teacher. If user doesn't have any account of smart guardian assistant application, then he can register in via admin panel. There have a remember me option. If there needed save password, then it will be works. Our data base collects this password and save it. Our applications log in service is very easier for user.



## Step2: Student User Profile

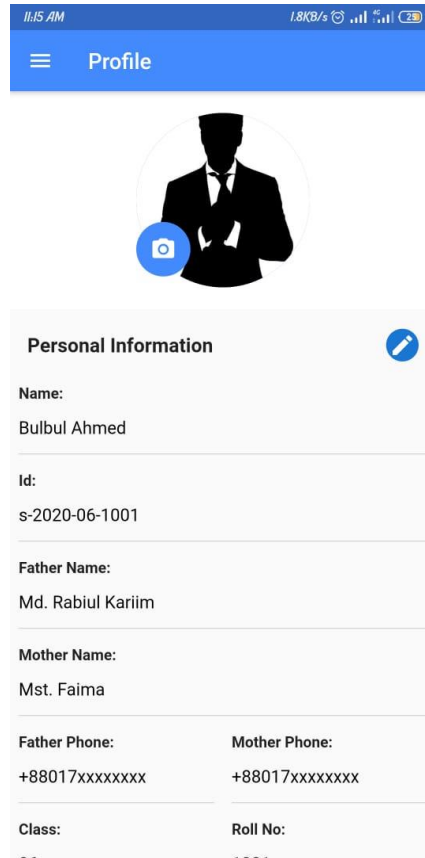


Figure 0.13: Screenshot of Student User Profile

The following figure 5.13, student user profile layout. This is the student's personal information page. There will be show users name, ID, Fathers name, Mothers name, Fathers mobile number, Mothers mobile number, Class, Roll No. There has edit option. Users can update his/her profile anytime if users want.

## Step3: Student User Profile Menu Bar

The following figure 5.14, student user profile menu bar layout. This is the student profile menu bar. There has some service for student user. This user can see Notice, Activity, Result, Account, Payments report and messages option. This messages part is the very important for our application. Guardians can talk with teachers by this messages option. Finally, there has logout option. Users can logout this application by using logout option.

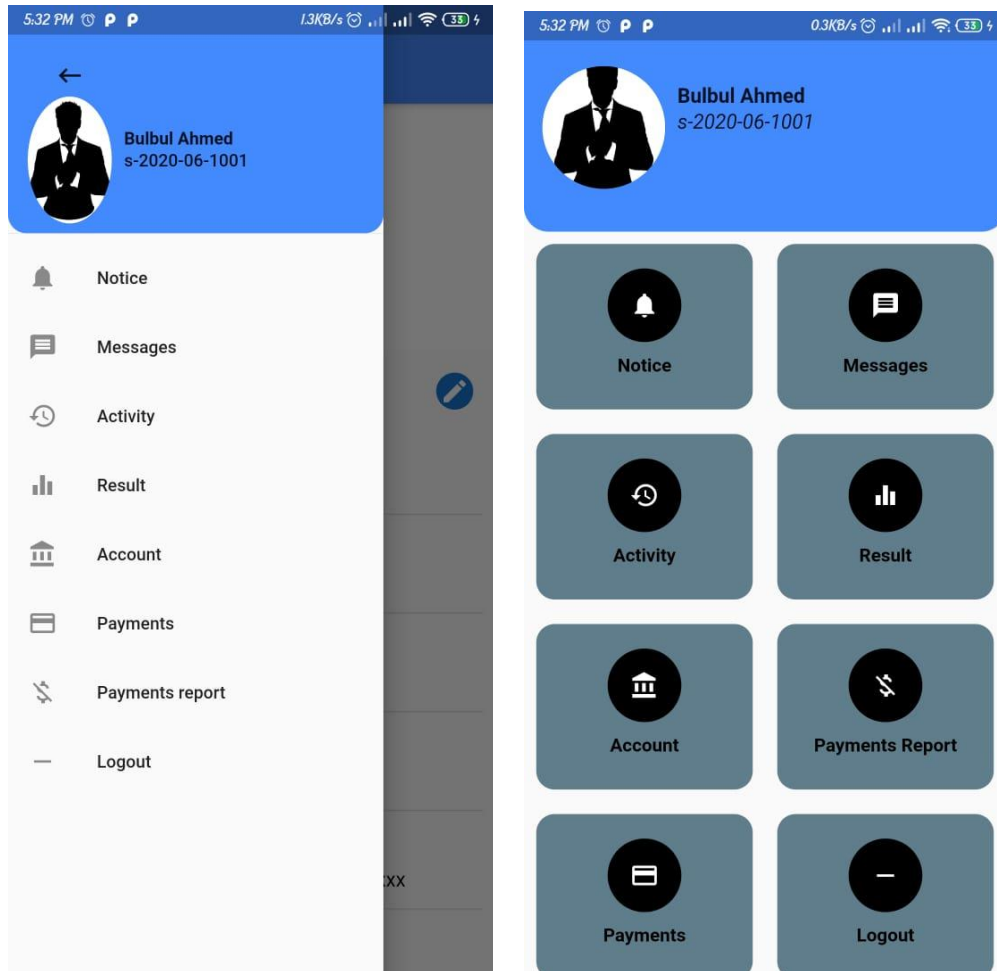


Figure 0.14: Screenshot of Teacher User Profile Menu Bar

#### Step4: Teacher User Profile

The following figure 5.15, teacher user profile layout. This is the teacher's personal profile menu bar. There has some service Notice, Attendance, Mark upload, Messages and logout. Teachers can see notice, updated attendance. Teachers and admin edit this option. We design this menu bar for users because of users can understand very easily. Our application tools design looks very good. That's why user fell comfort by using our application.

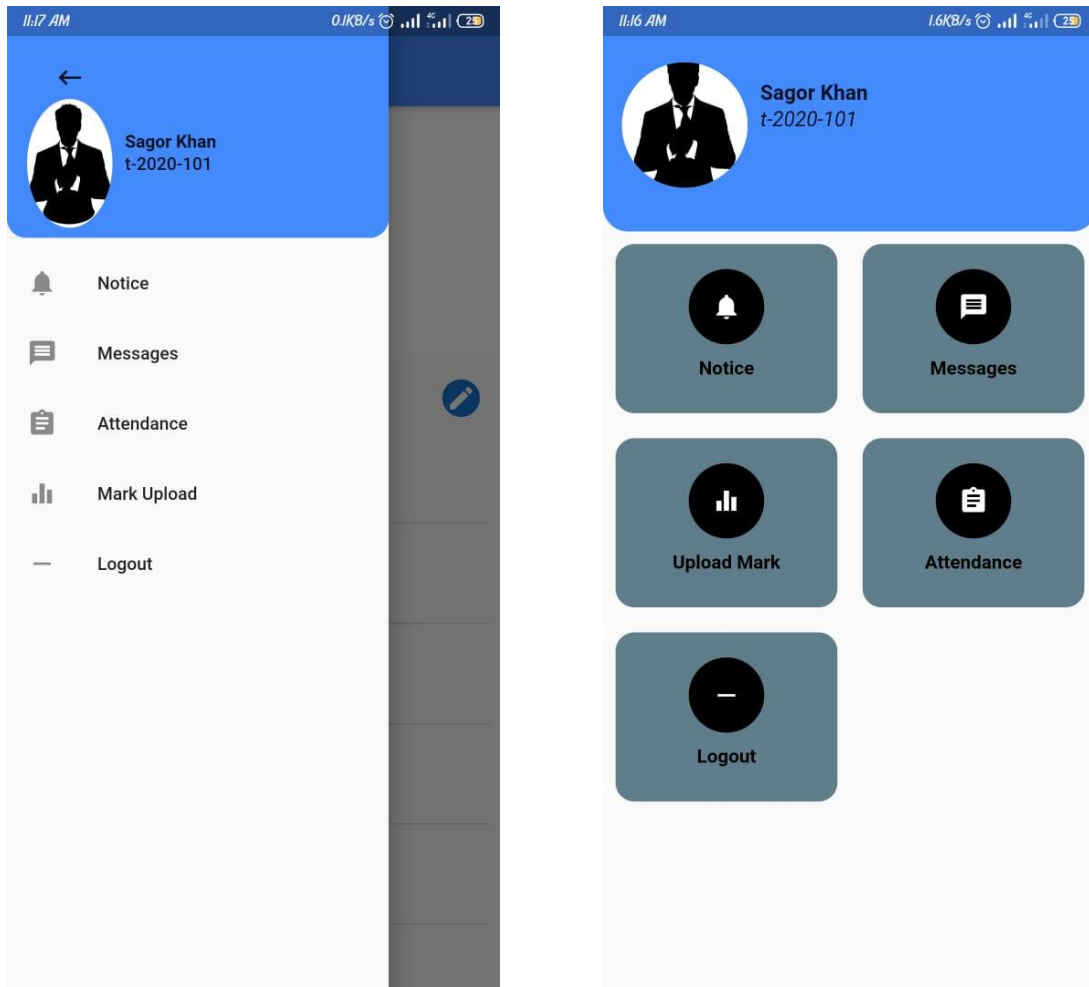


Figure 0.15: Screenshot of Teacher User Profile Menu Bar

### Step5: Student Activity

The following figure 5.16, student activity layout. Guardians can see student activity. First of all, there will be show date and attendance. If student attend in the class, then attendance result will be show 'Yes' otherwise result will be show 'No'. When this activity known guardians then they are satisfied their child's. If student is not attending in the class one day, then guardians can see it and act hurry up.

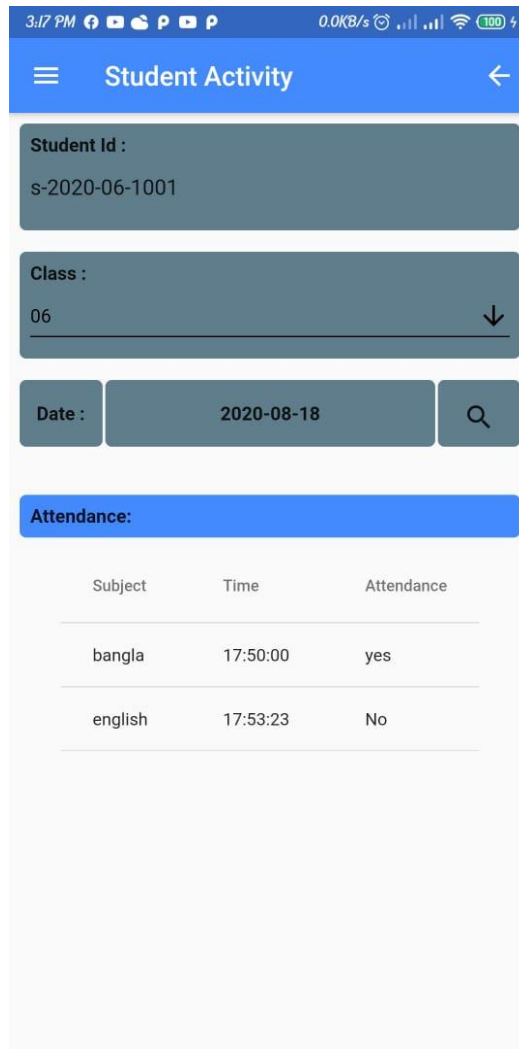


Figure 0.16: Screenshot of Student Activity

### Step6: Result

The following figure 5.17, student result layout. Then guardians can see students result. They can see each subject and all divided marks. That's why guardians are take care about child's result. When admin published this result, then guardians can see it.

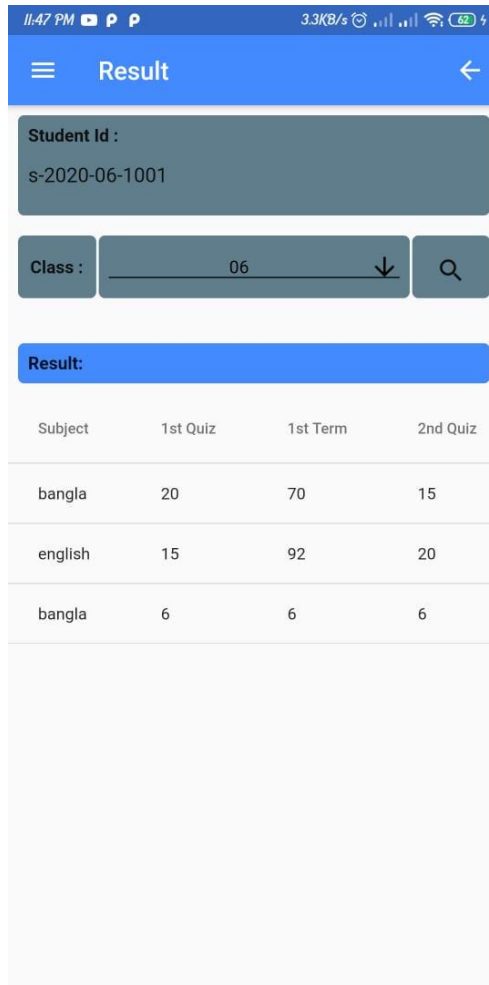


Figure 0.17: Screenshot of Student Result

### Step7: Notice

The following figure 5.18, notice layout. This service works for both users. There has notice title and notice upload date. In this moment there has no delete option for notice.

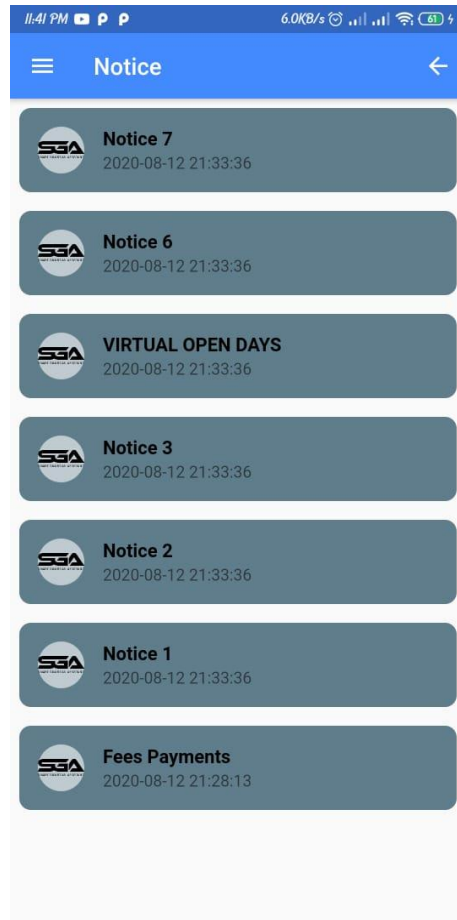


Figure 0.18: Screenshot of Notice

### Step8: Message

The following figure 5.19, message layout. We create this option for conversation between teacher and guardians. This option as like messenger, that's why we called it messenger. There has chat list option and teacher and students different list. That's why we found our target member for chatting. We provided this service mainly for guardians [11].

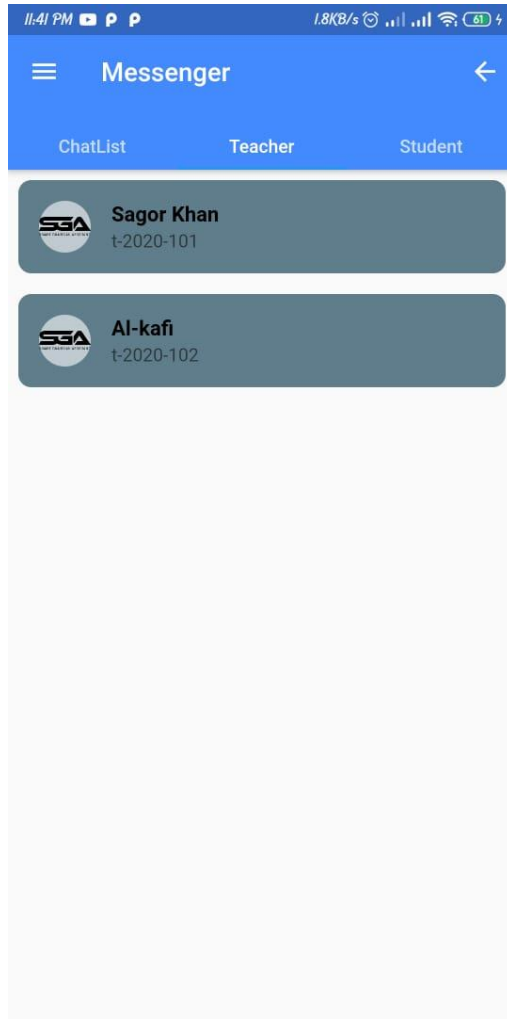


Figure 0.19: Screenshot of Messenger

### **Step9: Account Report**

The following figure 5.20, account report layout. Admin maintain this service. Student or guardians can only see this account report. There will be show student id, account No, and amount. And bottom of page show history of transaction. History of transaction has amount, date and 'IN/OUT' option. We design it very simply for users. That's why users can understand her /his transaction very easily.

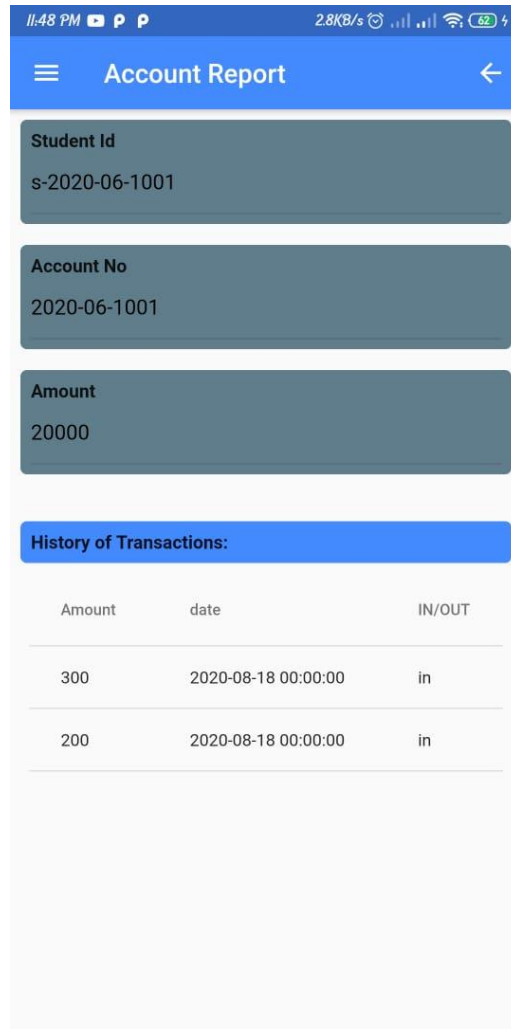


Figure 0.20: Screenshot of Account Report

### Step10: Payments Report

The following figure 5.21, Payments report layout. If student buy anything in canteen, then that's elements are adding here and there will be show how much money expense here. That's why guardians are known everything about their child.



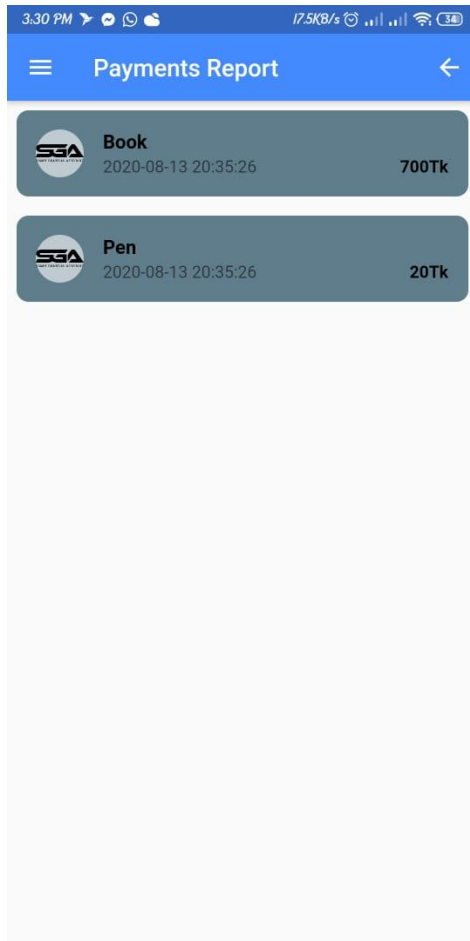


Figure 0.21: Screenshot of Expense Report

### **Step10: Upload Mark**

The following figure 5.22, upload mark layout. Teachers can upload mark here. Guardians are only seeing this mark. There has subject input option. Which subject needed, teachers input here and select the class bellow. There has student id, 1<sup>st</sup> quiz, 1<sup>st</sup> term, 2<sup>nd</sup> quiz, and 2nd term option. When data input here, and then needed to click save button for save upload mark.

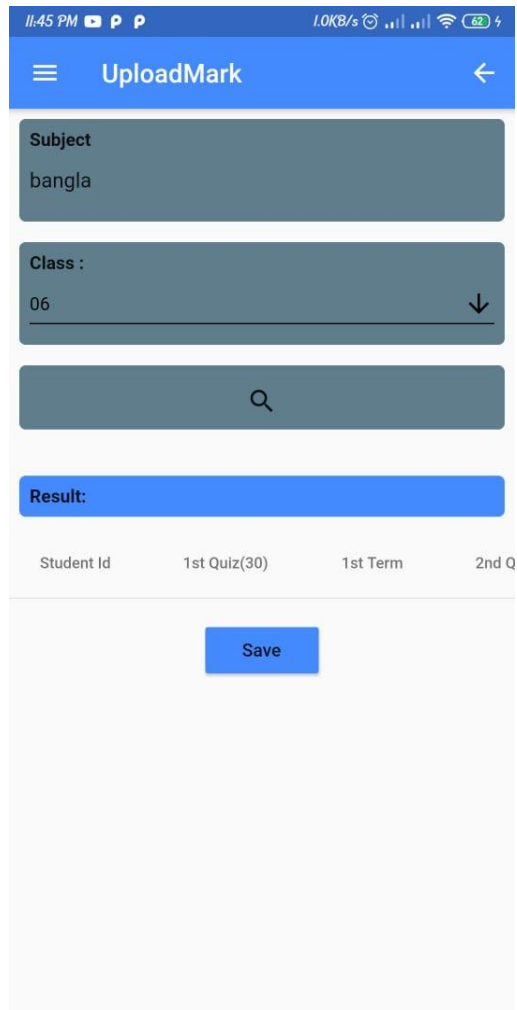


Figure 0.22: Screenshot of Upload Mark

### Step11: Take Attendance

The following figure 5.23, take attendance layout. Teachers can take attendance. There will be show subject wise all students. Teachers select class, subject, and date then take attendance very easily. When teacher click save button then all data goes to our database and save it.

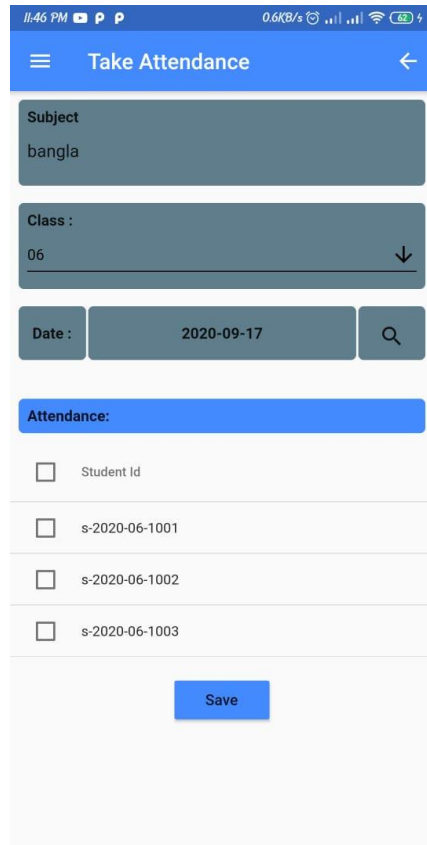


Figure 0.23: Screenshot of Take Attendance

## **CHAPTER 6**

### **Implementation and Testing**

#### **6.1: Implementation of Database**

We have made every effort to ensure the security and management of application in the database. We create our database with the quick response to Firebase and MySQL. Hopefully we will create a secured and stable database. We will progressively make the database safer. We were implemented real time database with very fast response [2][12][4].

#### **6.2: Implementation of Front-end Design**

The forehead start was created with Flutter design. We focused mainly UI architecture in the front-end. We implement very simple design for user interaction and user friendly. In this, we use tool Material Design for UI design [6][13].

#### **6.3: Implementation of Interactions**

An element and communication between those components are defined by a computer system design. All data from the database is named in our plan and transferred to application. That's why our mobile application is lighter. Interaction can be found nearly everywhere. User can get service from app on the Smart Guardian Assistant system with the admin panels mainly regulated. We successfully execute our application and the communication between our applications and user is very remarkable.

#### **6.4: Testing Implementation**

After the task, we have tested and successfully completed our project. Transfer data adequately. When we acted a user and put a data in our application then the data is transfer to database properly. Our test results are appropriate, but we work hard to make them more precise.

## 6.5: Test Results and Repots

The reports are necessary to formally represents test results and to quickly calculate the results. We present the test case, test data, predicted outcomes, current output, and eventually find out what our application is looking for. The test result 96 percent that we are pleased with. Hope, we are going to be more successful. Main testing results are simple to use application, improve contact user interaction, reduces hand cash and retail money, save times.

Table 0-1: Test Case for Project on Secure Messaging

Test Case	Test Input	Expected Outcome	Actual Outcome	Result	Tested On
1.Install Application	Tested on Various – 1. Jellybean (4,1-4.3.1) 2. KitKat (4.4-4.4.4) 3. Lollipop (5.5-5.0.2) 4. Marshmallow (6.0) 5. Nougat (7.0-7.1) 6. Oreo (8.0-8.1) 7. Pie (9.0) 8. Q (10.0)	Successfully install all those versions.	Install successful	Passed	28.10.2019

## **CHAPTER 7**

### **Conclusion and Future Scope**

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

This project would make our life comfortable and easier on the regular life. We have to face a lot of problems. We always try a good communication with guardians and teachers. We try to give good service for guardians. We hope our project is very helpful for our busy life. Guardians can feel relaxed to their family. When guardians know everything about their child by using our application, then they aren't take tension for Childs. We try to design of this app is very simple and user friendly so, there was no problem running this application. Our applications solve the hand to hand cash problem. Users pay money by online payment gateway. It reduces hand-cash problem and retail money. This project was designed to explain all the following elements and activities during the application development. We try to reduce time for both of teachers and guardians. We try to reduce suffering of retail cash canteen bill. Our all service will be helped to guardians for easier life. When guardians are known details about our application then they will be very much interested to using this method or application. We described our project with some general people. They were appreciated in this project. We stated to collect some information and we started our main flow of work. First of all we design front end and then started design in back end. Then we started admin panel work in web-based platform using firebase database. We were so much excited for this project because we were facing this problem in real life. The application will improve and give better performance by more users or guardians. More use and more feedback allow to give more accurate information. It will help to reach to user effortlessly and effectively. Many challenges have to be confronted when we do many functions. We consider our choice into account when we choose to construct our design. Any task can at all be seamless. In addition, to build our endeavor we must face other obstacles. Our project is based on user-friendly design of products with payment methods. We hope our project application good effective for students and good communication with teachers and guardians.

### **7.1.1: Limitations of Our application**

As with all applications, our application also has some limitation. Those obstacles will be solved in the future. Some of the most significant drawbacks are:

- The application must be connected to the internet. So, without a strong internet connection, the user could not coordinate our proposal.
- Developed by two platform mobile application and web application.
- Some data are very sensitive one of them is ID number. We log in our application using this ID number. So, we need an extra security for our application.
- Several function or characteristic must be updated.
- We only dropped English in our application and should keep a Bangla version.
- All of users are teachers and guardians. Most of guardians are illiterate.

### **7.2: Scope for Future Developments**

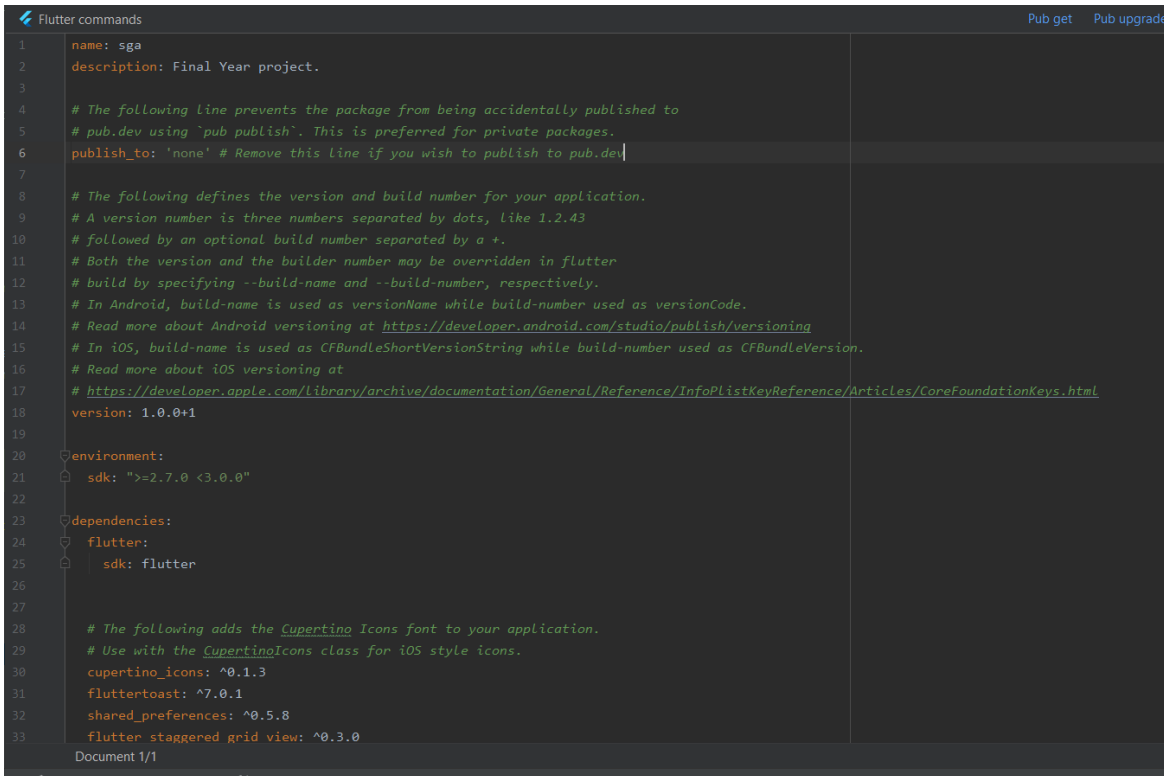
Our plan can be built in the future in many ways. Scopes for future development write bellow:

1. We are going to work on all educational institutions in Bangladesh.
2. We are going to offer our own program of protections.
3. In future we will be doing our project on IOT.
4. In future we will connect with Arduino in our venture.
5. We want to work with government level.
6. We are going to build our own user database system.
7. In future we will find our application is English language as well as Bengali language.
8. We will try our best for how to be more users friendly in future.
9. For addition our application requires more capacity that we attach this to our larger database system.
10. We want to work with ICT division by this venture.
11. We want to a large IT company in basic level.

# Appendices

## Appendix A:

Below those picture are Flutter framework pubspace.yaml and android build.gradle.



```
Flutter commands Pub get Pub upgrade
1 name: sga
2 description: Final Year project.
3
4 # The following line prevents the package from being accidentally published to
5 # pub.dev using `pub publish`. This is preferred for private packages.
6 publish_to: 'none' # Remove this line if you wish to publish to pub.dev
7
8 # The following defines the version and build number for your application.
9 # A version number is three numbers separated by dots, like 1.2.43
10 # followed by an optional build number separated by a +.
11 # Both the version and the builder number may be overridden in flutter
12 # build by specifying --build-name and --build-number, respectively.
13 # In Android, build-name is used as versionName while build-number used as versionCode.
14 # Read more about Android versioning at https://developer.android.com/studio/publish/versioning
15 # In iOS, build-name is used as CFBundleShortVersionString while build-number used as CFBundleVersion.
16 # Read more about iOS versioning at
17 # https://developer.apple.com/library/archive/documentation/General/Reference/InfoPlistKeyReference/Articles/CoreFoundationKeys.html
18 version: 1.0.0+1
19
20 environment:
21   sdk: ">=2.7.0 <3.0.0"
22
23 dependencies:
24   flutter:
25     sdk: flutter
26
27
28 # The following adds the Cupertino Icons font to your application.
29 # Use with the CupertinoIcons class for iOS style icons.
30 cupertino_icons: ^0.1.3
31 fluttertoast: ^7.0.1
32 shared_preferences: ^0.5.8
33 flutter_staggered_grid_view: ^0.3.0
Document 1/1
```



```

Flutter commands Pub get Pub upgrade
28 # The following adds the Cupertino Icons font to your application.
29 # Use with the CupertinoIcons class for iOS style icons.
30 cupertino_icons: ^0.1.3
31 fluttertoast: ^7.0.1
32 shared_preferences: ^0.5.8
33 flutter_staggered_grid_view: ^0.3.0
34 http: ^0.12.2
35 provider: ^4.3.1
36 progress_dialog: ^1.2.4
37
38 dev_dependencies:
39   flutter_test:
40     sdk: flutter
41   simple_animations: ^1.3.3
42
43 # For information on the generic Dart part of this file, see the
44 # following page: https://dart.dev/tools/pub/pubspec
45
46 # The following section is specific to Flutter.
47 flutter:
48
49 # The following line ensures that the Material Icons font is
50 # included with your application, so that you can use the icons in
51 # the material Icons class.
52 uses-material-design: true
53
54 # To add assets to your application, add an assets section, like this:
55 assets:
56   - assets/images/
57
58
59
60

```

Document 1/1

```

Flutter commands Open for Editing in Android Studio
1 def localProperties = new Properties()
2 def localPropertiesFile = rootProject.file('local.properties')
3 if (localPropertiesFile.exists()) {
4   localPropertiesFile.withReader('UTF-8') { reader ->
5     localProperties.load(reader)
6   }
7 }
8
9 def flutterRoot = localProperties.getProperty('flutter.sdk')
10 if (flutterRoot == null) {
11   throw new GradleException("Flutter SDK not found. Define location with flutter.sdk in the local.properties file.")
12 }
13
14 def flutterVersionCode = localProperties.getProperty('flutter.versionCode')
15 if (flutterVersionCode == null) {
16   flutterVersionCode = '1'
17 }
18
19 def flutterVersionName = localProperties.getProperty('flutter.versionName')
20 if (flutterVersionName == null) {
21   flutterVersionName = '1.0'
22 }
23
24 apply plugin: 'com.android.application'
25 apply plugin: 'kotlin-android'
26 apply from: "$flutterRoot/packages/flutter_tools/gradle/flutter.gradle"
27
28 android {
29   compileSdkVersion 28
30
31   sourceSets {
32     main.java.srcDirs += 'src/main/kotlin'
33   }

```

Dart Analysis Logcat Version Control Event Log Layout

```
Flutter commands Open for Edit
33 }
34
35 lintOptions {
36   disable 'InvalidPackage'
37 }
38
39 defaultConfig {
40   // TODO: Specify your own unique Application ID (https://developer.android.com/studio/build/application-id.html).
41   applicationId "com.example.sga"
42   minSdkVersion 16
43   targetSdkVersion 28
44   versionCode flutterVersionCode.toInteger()
45   versionName flutterVersionName
46 }
47
48 buildTypes {
49   release {
50     // TODO: Add your own signing config for the release build.
51     // Signing with the debug keys for now, so `flutter run --release` works.
52     signingConfig signingConfigs.debug
53   }
54 }
55 }
56
57 flutter {
58   source '../..'
59 }
60
61 dependencies {
62   implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"
63 }
64
```

Dart Analysis Logcat Version Control Ev

## References

- [1] DIU Student Portal: <http://studentportal.diu.edu.bd/> [Last Access 23-04-20] at 3:32am.
- [2] Android Firebase Database Read & Write: <https://firebase.google.com/docs/database/android/read-and-write> [Last Access 18-08-20] at 2:05am.
- [3] SSLCommerz Payment: <https://pub.dev/packages/sslcommerz> [Last Access 24-09-20] at 1:16am.
- [4] http: <https://pub.dev/packages/http> [Last Access 20-02-20] at 2:05am.
- [5] Shared Preferences: [https://pub.dev/packages/shared\\_preferences](https://pub.dev/packages/shared_preferences) [Last Access 09-04-20] at 3:05am.
- [6] Cupertino icons: [https://pub.dev/packages/cupertino\\_icons](https://pub.dev/packages/cupertino_icons) [Last Access 09-04-20] at 3:35am.
- [7] Attendance: <https://play.google.com/store/apps/details?id=in.codeshuffle.amftpro&hl=en> [Last Access 18-05-19] at 3:05am.
- [8] AttendanceByfingerprint: <https://play.google.com/store/apps/details?id=com.aandt.attendacewithfingerprint&hl=en> [Last Access 18-05-19] at 12:05pm.
- [9] GPSmartAttendance: <https://play.google.com/store/apps/details?id=com.inovacetech.inovacecentral&hl=en> [Last Access 01-05-20] at 01:25am.
- [10] Firebase Cloud Messaging: <https://firebase.google.com/docs/cloud-messaging> [Last Access 18-02-20] at 4:20pm.
- [11] WhatsApp Messenger: <https://play.google.com/store/apps/details?id=com.whatsapp&hl=en> [Last Access 18-02-20] at 5:15pm.
- [12] SQL and PHP: <https://www.w3schools.com/sql/> [Last Access 17-09-20] at 4:45pm.
- [13] Flutter framework: <https://flutter.dev/docs> [Last Access 18-09-20] at 5:15pm.
- [14] Material Components widgets: <https://flutter.dev/docs/development/ui/widgets/material> [Last Access 16-09-20] at 7:15am.
- [15] Smart Draw: <https://www.smartdraw.com/> [Last Access 08-08-20] at 12:15am.
- [16] Visual paradigm: <https://online.visual-paradigm.com/> [Last Access 08-08-20] at 12:15am.

# Turnitin Originality Report

Processed on: 20-Sep-2020 12:14 +06  
ID: 1391583149  
Word Count: 5726  
Submitted: 1

Similarity Index

3%

## Similarity by Source

Internet Sources: N/A  
Publications: N/A  
Student Papers: N/A

Android Based Smart  
Guardian Apps By Bulbul  
Ahmed