Republic of the Philippines
RIZAL TECHNOLOGICAL UNIVERSITY
Cities of Mandaluyong and Pasig
COLLEGE OF ENGINEERING, ARCHITECHTURE AND TECHNOLOGY

COMPUTER PROGRAMMING 1
(ITC111L)

LearnLog

Classroom Management System

GRADE

Submitted by:

Manalo
Eisenpol D.G
BSIT
CEIT-37-202A

Malota
Fiona Louella A.
BSIT
CEIT-37-202A

Mesa
Sebastien V.
BSIT
CEIT-37-202A

Pabiona
Michael John O.
BSIT
CEIT-37-202A

Talinao
Nicole J.
BSIT
CEIT-37-202A

Sarmiento
Alessandra Hajiah F.
BSIT
CEIT-37-202A

Zamora
Kimverly A.
BSIT
CEIT-37-202A

Submitted to:

MS. MAY BACERLONA FIGUEROA

Date of Defense

I. INTRODUCTION

The modern world is now ruled by technology, which has the capability to make human labour simple and efficient. Systems were developed for variety of goals and uses. Setting up a system for a business is one step closer to innovation and another way of creating a productive workplace for all employees and consumer. A traditional way of listing or gathering information from a student is to write it on a class record. But, because we live in a modern world and want to help teachers with less paperwork, we came up with this system.

The developers created **LearnLog** classroom management system that makes information more accessible at all times. A system that will allow the faculty and students to easily access information for a certain person or for personal use and to have a greater potential for higher student engagement.

II. DESCRIPTION OF THE PROJECT

A **LearnLog** classroom management system is a digital method to display every student's information. It is also to help lessen the teacher's paperwork by storing and keeping track of the students' information. Their project features a log-in and log out system where everyone can create an account to register themselves, as well as forgot password, a student library where students can encode their information (name, age, gender, address, email account, contact number, user's image), all of which is stored in the interior file of the system, and lastly, it has search feature that allows the user to search a certain student number and it shows the student's information.

III. Objectives

The main objective of the system is to develop a classroom management software using the C++ programming language that can serve as a digital way of efficiently sorting and displaying information about different students.

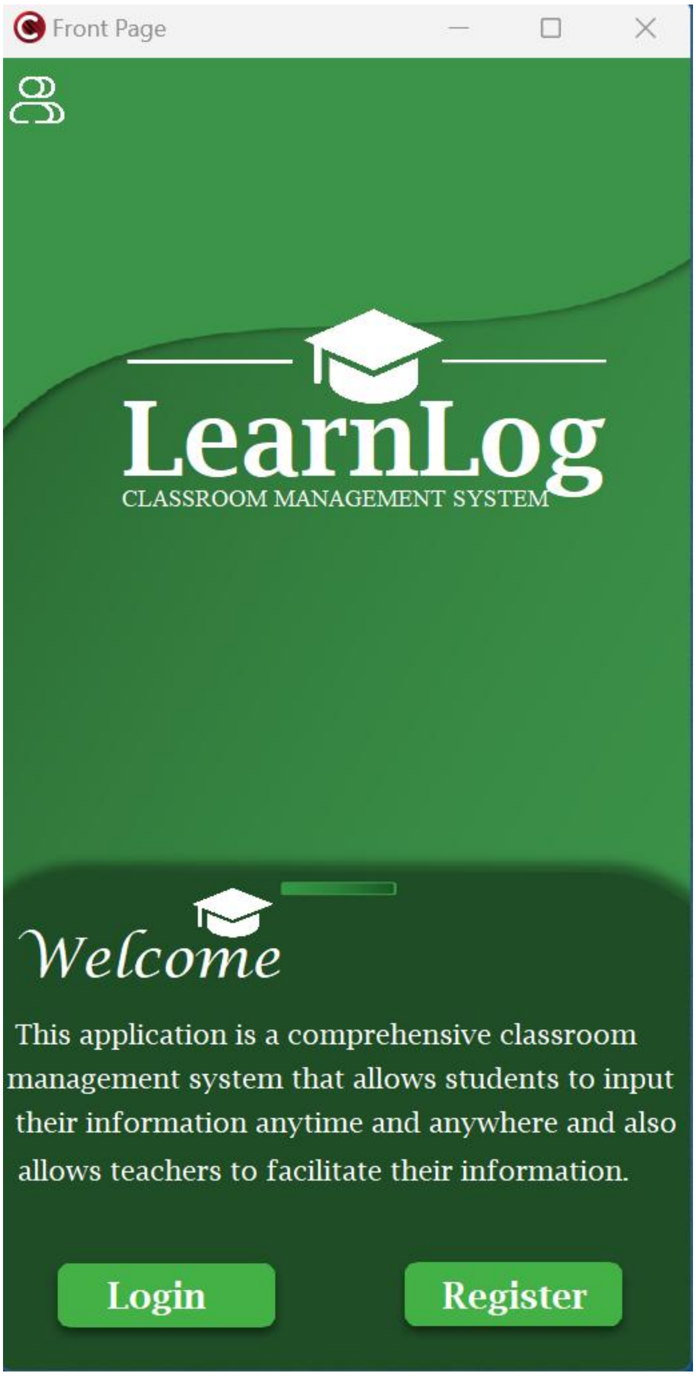
IV. Significance of the Study

The developers wanted to have a more convenient way to keep information on track and secure. This study gave importance to those three reasons. The system will appeal to the teachers, faculty, and students' benefit by reducing paperwork and allowing them to easily access the data they require at any time and from any location. Furthermore, the system is secure because it is ensured that an individual can only access the data with the account he or she registers in the system. It is relevant at this time because we are surrounded by technology and as a result, many people are looking for other ways to be more convenient about keeping their information, but in a safe way.

V. Scope and Delimitations

The system focuses on providing teachers and students a more efficient, easy, and secure way to keep data on every student in a classroom. It was designed with features in mind that will allow the user to sign up for an account, input their data, retrieve and display the data they have provided. The program itself is bound to such limitations and therefore doesn't have a feature that will let the user to display additional information such as subjects, grades, and edit information.

VI. Screen Output



This application is a comprehensive classroom management system that allows students to input their information anytime and anywhere and also allows teachers to facilitate their information.

Figure 1. Home Page

Figure 1 shows the option you may choose that include the following:

1. **LOGIN** - This option will let you login in your registered account.

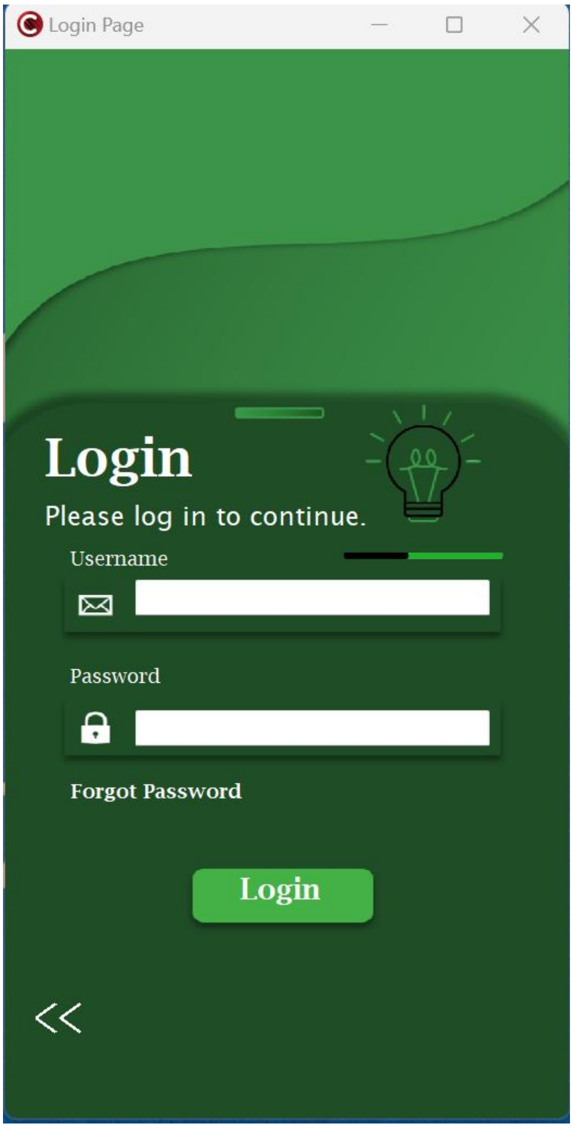
1.1 **FORGOT PASSWORD** - This option will let you reset your password if happens that you forgot it.

2. **REGISTER** - This option will let you register or create your account.

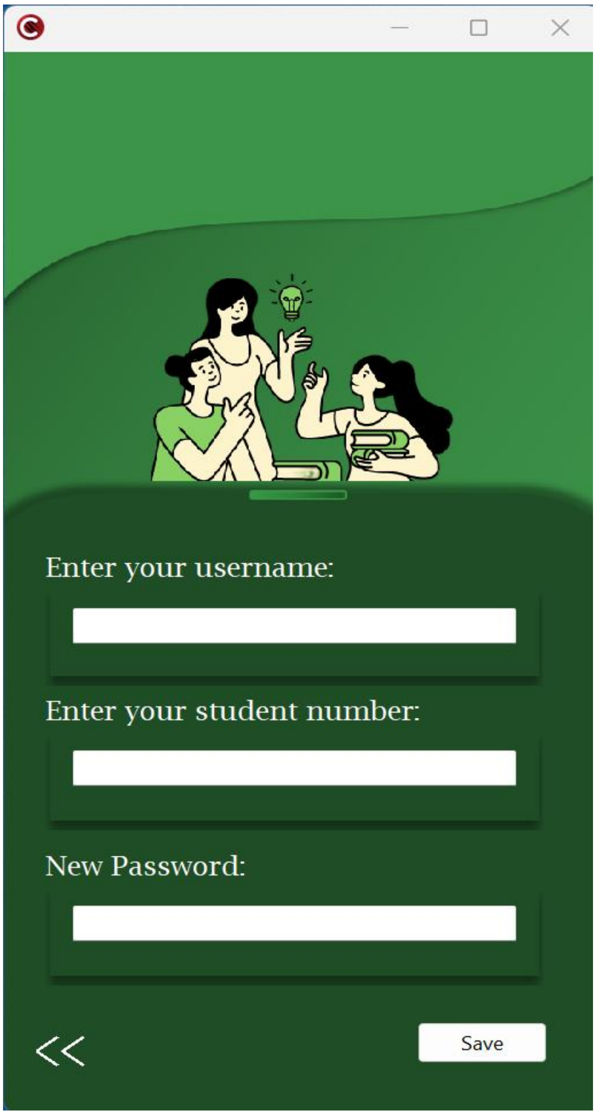
2.1 **DATA PRIVACY CONSENT -** This page allows the user to be aware of the terms and condition of creating an account inside the application.

3. **ABOUT US** - This button will let you view the developers and their respective contribution in developing the system.

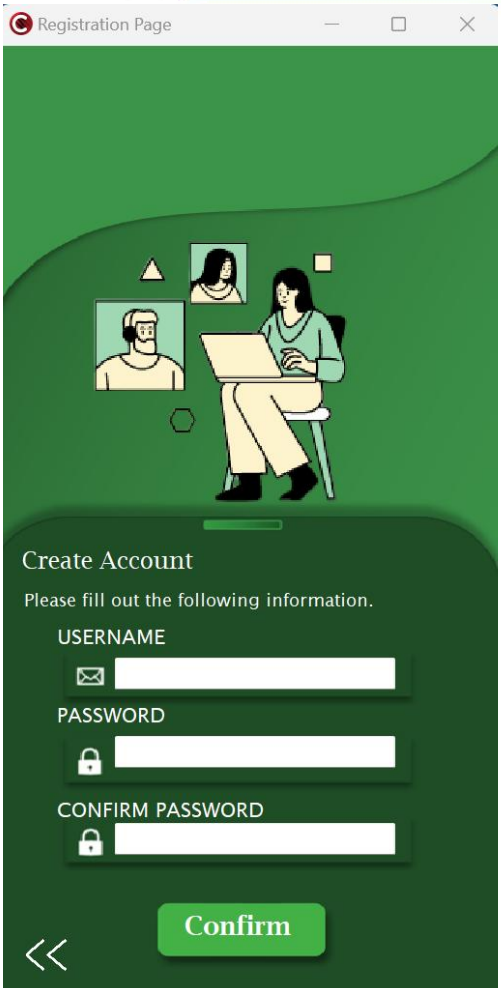
1



1.1



2



Create Account

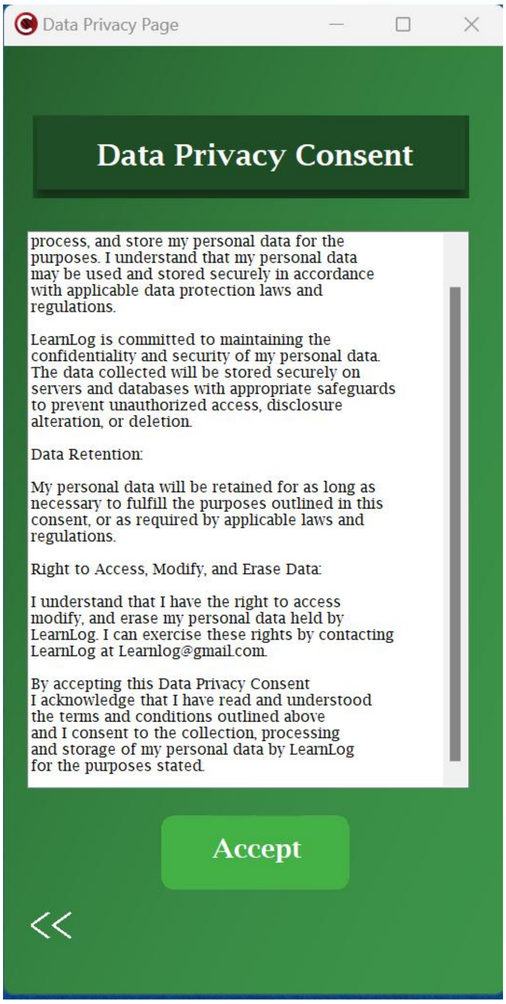
Please fill out the following information.

USERNAME

PASSWORD

CONFIRM PASSWORD

2.1



Data Privacy Consent

process, and store my personal data for the purposes. I understand that my personal data may be used and stored securely in accordance with applicable data protection laws and regulations.

LearnLog is committed to maintaining the confidentiality and security of my personal data. The data collected will be stored securely on servers and databases with appropriate safeguards to prevent unauthorized access, disclosure alteration, or deletion.

Data Retention:

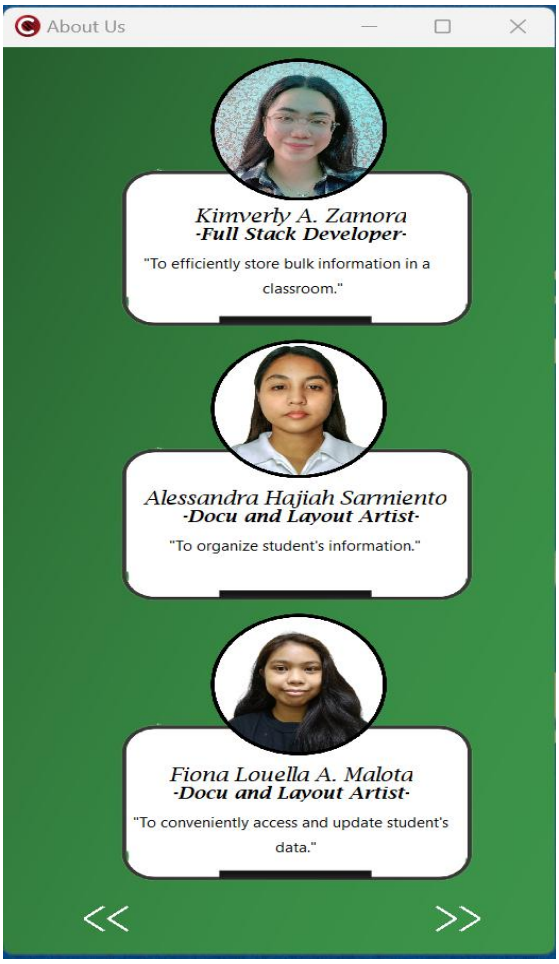
My personal data will be retained for as long as necessary to fulfill the purposes outlined in this consent, or as required by applicable laws and regulations.

Right to Access, Modify, and Erase Data:

I understand that I have the right to access modify, and erase my personal data held by LearnLog. I can exercise these rights by contacting LearnLog at Learnlog@gmail.com.

By accepting this Data Privacy Consent I acknowledge that I have read and understood the terms and conditions outlined above and I consent to the collection, processing and storage of my personal data by LearnLog for the purposes stated.

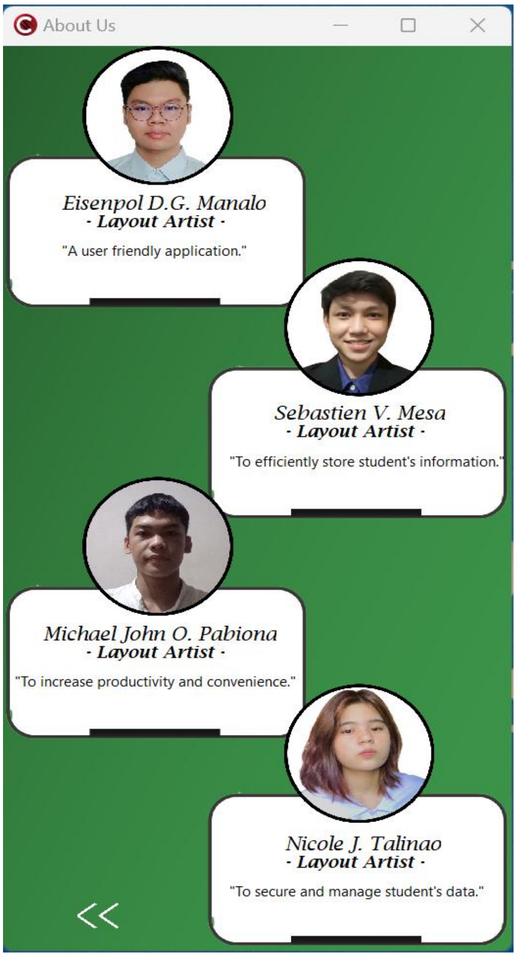
3



Kimverly A. Zamora
-Full Stack Developer-
"To efficiently store bulk information in a classroom."

Alessandra Hajiah Sarmiento
-Docu and Layout Artist-
"To organize student's information."

Fiona Louella A. Malota
-Docu and Layout Artist-
"To conveniently access and update student's data."



Eisenpol D.G. Manalo
- Layout Artist -
"A user friendly application."

Sebastien V. Mesa
- Layout Artist -
"To efficiently store student's information."

Michael John O. Pabiona
- Layout Artist -
"To increase productivity and convenience."

Nicole J. Talinao
- Layout Artist -
"To secure and manage student's data."

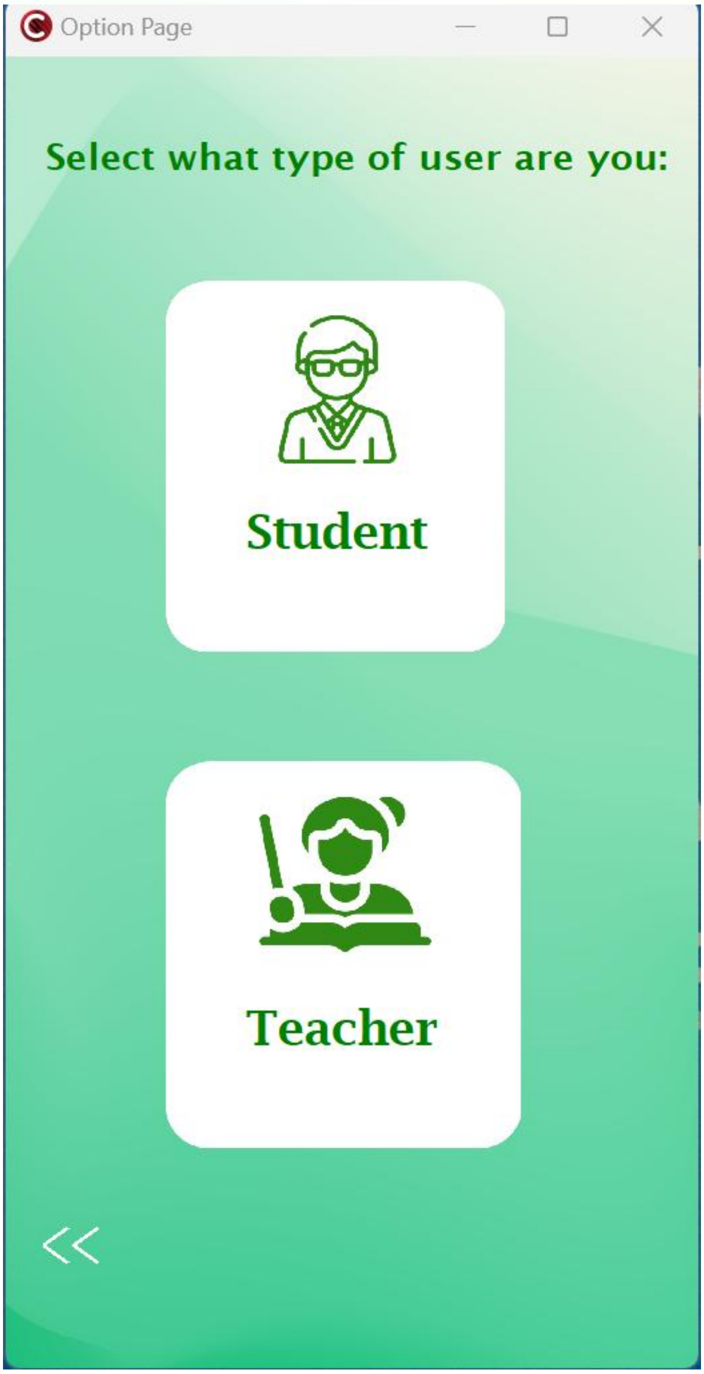


Figure 2. Option Page

Figure 2 shows the main page after you register and login in your account, the following option you can choose are:

1. STUDENT **-** this option is for student use.

2. TEACHER - this option is for teacher use.

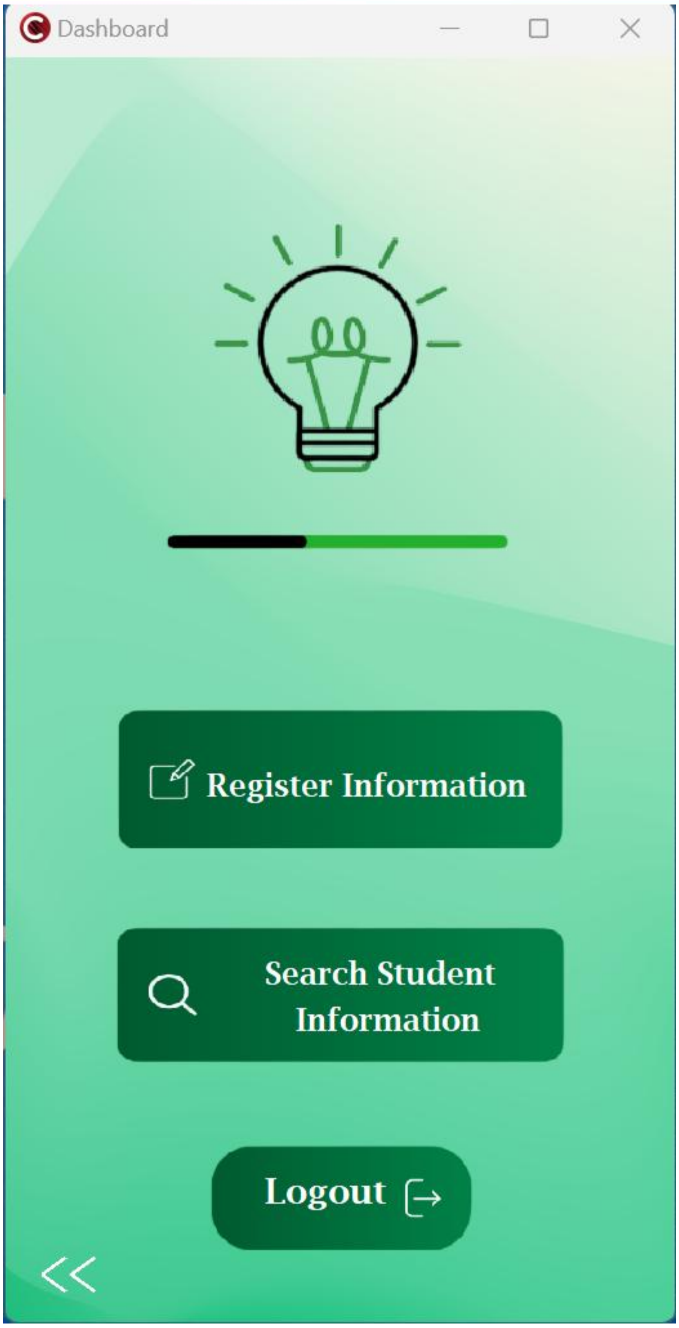


Figure 3. Dashboard

Figure 3 shows the dashboard after you choose what kind of user you are, the following features you can choose are:

1. REGISTER INFORMATION - this option will let you register the following information needed.

1.1 **UPLOAD PHOTO -** this option allows the user to upload a photo of her/himself.

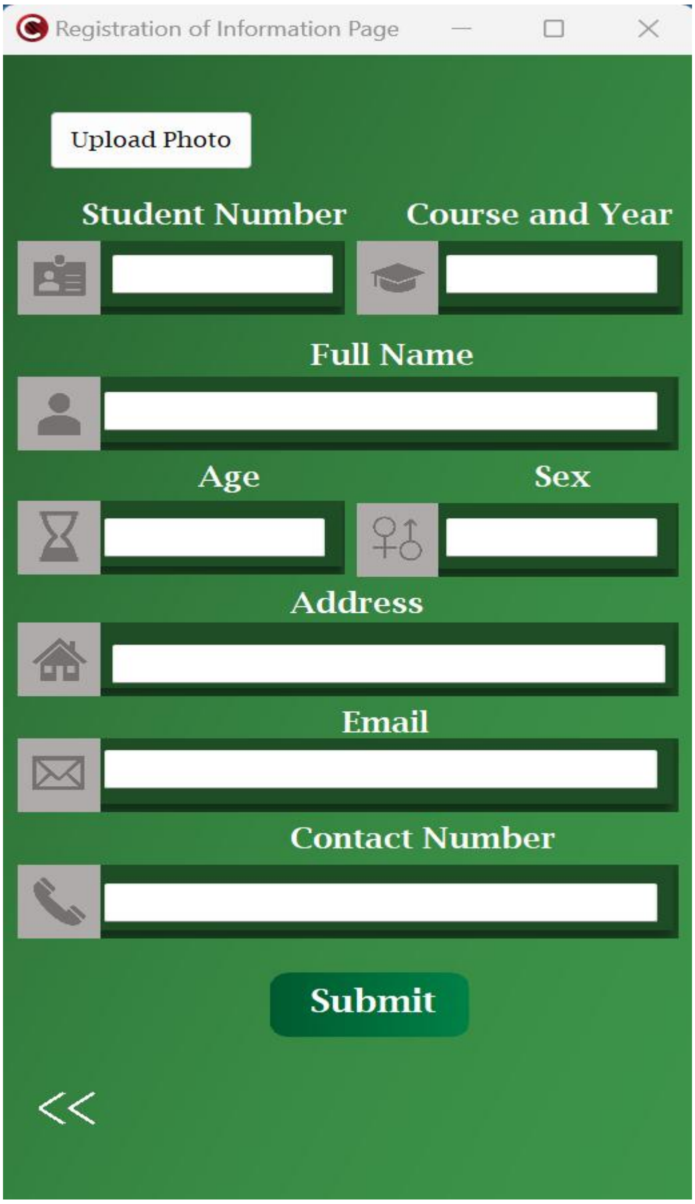
1.2 **SUBMIT** - this option allows the user to submit all the information that has been input by the user.

2. SEARCH STUDENT INFORMATION - this option will let you search for a certain student information using your student number/ their student number.

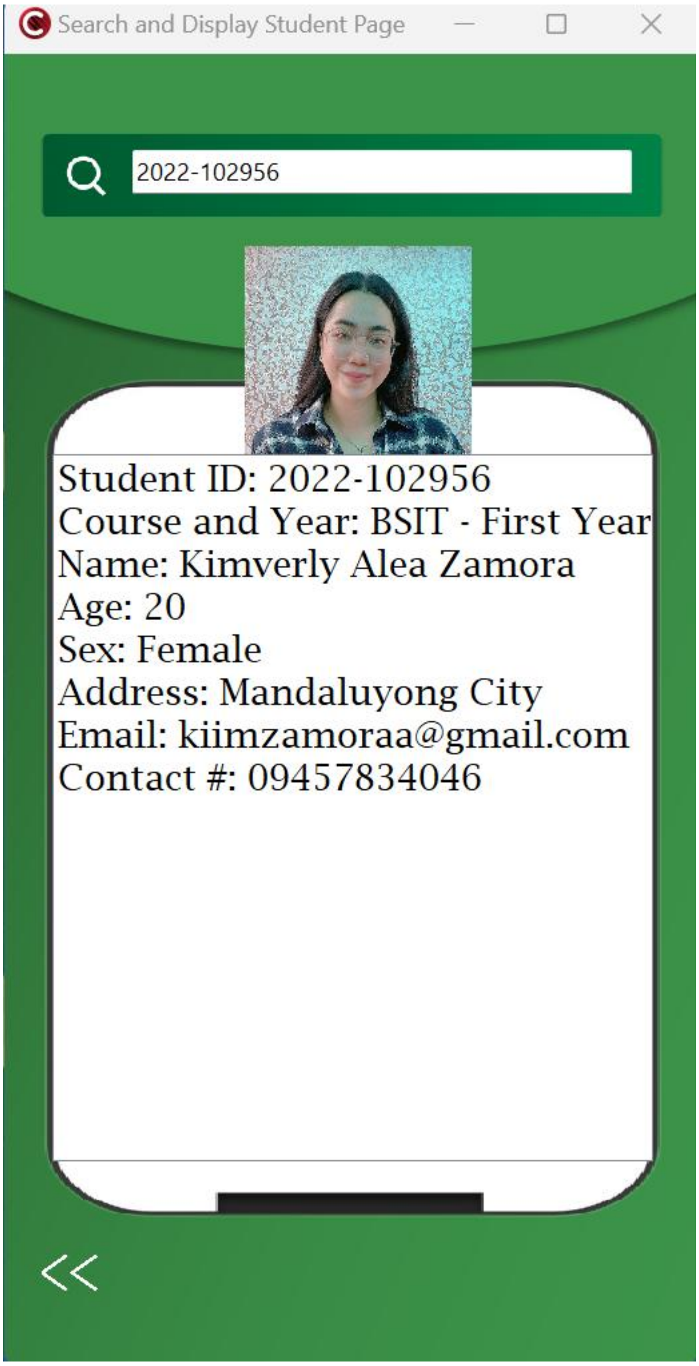
2.1 **SEARCH BAR -** this is where the user will enter the student number to search for a certain student number.

3. **LOGOUT -** this option will let you logout your account.

1

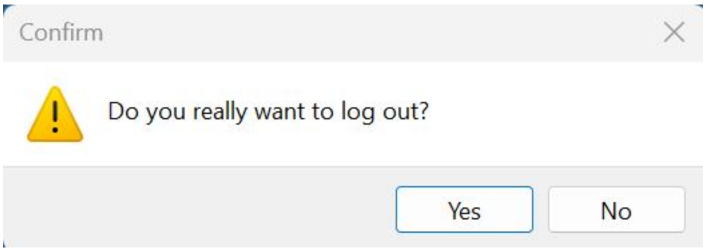


2



Student ID: 2022-102956
Course and Year: BSIT - First Year
Name: Kimverly Alea Zamora
Age: 20
Sex: Female
Address: Mandaluyong City
Email: kiimzamoraa@gmail.com
Contact #: 09457834046

3



Do you really want to log out?

VII. Source Code

LOGIN BUTTON

```
//---------------------------------------------------------------------------
#include <vcl.h>
#pragma hdrstop
#include "FPage.h"
#include "LPage.h"
#include "DPage.h"
#include "AboutP.h"
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TFrontP *FrontP;
//---------------------------------------------------------------------------
__fastcall TFrontP::TFrontP(TComponent* Owner)
        : TForm(Owner)
{
}
//---------------------------------------------------------------------------
void __fastcall TFrontP::Label5Click(TObject *Sender)
{
        Hide();
        LoginP->Show();
}
```

```
//-
---
---------------------------------------------------------------------------
void __fastcall TFrontP::Label6Click(TObject *Sender)
{
        Hide();
        DataP->Show();
}
//---------------------------------------------------------------------------

void __fastcall TFrontP::Image2Click(TObject *Sender)
{
        Hide();
        aboutUsP->Show();
}
//---------------------------------------------------------------------------
```

REGISTER BUTTON

```
 //---------------------------------------------------------------------------

#include <vcl.h>
#pragma hdrstop

#include "FPage.h"
#include "LPage.h"
#include "DPage.h"
#include "AboutP.h"
```

```
//-
---
---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TFrontP *FrontP;
//---------------------------------------------------------------------------
__fastcall TFrontP::TFrontP(TComponent* Owner)
        : TForm(Owner)
{
}
//---------------------------------------------------------------------------
void __fastcall TFrontP::Label5Click(TObject *Sender)
{
        Hide();
        LoginP->Show();
}
//---------------------------------------------------------------------------
void __fastcall TFrontP::Label6Click(TObject *Sender)
{
        Hide();
        DataP->Show();
}
//---------------------------------------------------------------------------


void __fastcall TFrontP::Image2Click(TObject *Sender)
{
        Hide();

        aboutUsP->Show();

}

//---------------------------------------------------------------------------
```

LOGIN FEATURE

```
#include <vcl.h>

#include <fstream>

#include <sstream>

#include <string>

#pragma hdrstop


#include "LPage.h"

#include "FPage.h"

#include "MPage.h"

#include "ForgotPage.h"

#include "OptPage.h"


using namespace std;

//---------------------------------------------------------------------------

#pragma package(smart_init)

#pragma resource "*.dfm"

TLoginP *LoginP;

//---------------------------------------------------------------------------


__fastcall TLoginP::TLoginP(TComponent* Owner)
```

```
TForm(Owner)
{
}

void __fastcall TLoginP::backbtnClick(TObject *Sender)
{
        Hide();
        FrontP->Show();
}

void __fastcall TLoginP::Label4Click(TObject *Sender)
{
        AnsiString username = this->username->Text;
        AnsiString password = this->password->Text;

        if (username.IsEmpty() || password.IsEmpty()) {
                ShowMessage("Please input the following information needed.");
                return;
        }


        fstream file;
        file.open("users.txt", ios::in);

        if (file.is_open()) {
```

```
bool found = false;
    string line;
    while (getline(file, line)) {
        size_t comma = line.find(",");
        string db_user = line.substr(0, comma);
        string db_pass = line.substr(comma+1);

        if (username == db_user.c_str() && password == db_pass.c_str()) {
            found = true;
            break;
        }
    }

    file.close();

    if (found) {
        ShowMessage("Login successful.");

        // Create an instance of the next form (FPage)
        OptionP->Show();

        // Hide the current login form
        Hide();
    } else {
        ShowMessage("Invalid username or password.");
    } else {
        ShowMessage("Error opening file.");
    }
}




void __fastcall TLoginP::usernameClick(TObject *Sender)
{
    username->Clear();
}
//---------------------------------------------------------------------------


void __fastcall TLoginP::passwordClick(TObject *Sender)
{
  password->Clear();
}
//---------------------------------------------------------------------------

void __fastcall TLoginP::Label5Click(TObject *Sender)
{
    Hide();
```

```
        ForgotP->Show();

}

//---------------------------------------------------------------------------
```

FORGOT PASSWORD FEATURE

```
void __fastcall TForgotP::Button1Click(TObject *Sender)

{



        AnsiString targetUsername = username->Text;

   AnsiString newPassword = reset->Text;

        AnsiString studentNumber = student->Text;

   AnsiString snFile = student->Text + ".txt";


        bool unVerified = false;

        bool snVerified = false;



        ifstream verifyUN("users.txt");

        if(verifyUN.is_open()){


                string line;

                while(getline(verifyUN, line)){


                size_t commaPos = line.find(",");

                string un = line.substr(0, commaPos);
```

```
     if (username->Text == un.c_str()) {

              unVerified = true;

              break;

         }

     }

     verifyUN.close();

 }else{

         ShowMessage("Username does not exist in database!");

         return;

         }

 fstream verifySN (snFile.c_str(), ios::in);

 if(verifySN.is_open()){

  string line;

  while(getline(verifySN, line)){

     size_t commaPos = line.find(",");

string dbSN = line.substr(0,commaPos);

     if(student->Text == dbSN.c_str()){
```

```
   snVerified = true;

            break;


            }

     }

   }

   verifySN.close();


   if (unVerified == true && snVerified == true ) {


   ifstream searchUser("users.txt");

   if (searchUser.is_open()){


ofstream tempFile("temp.txt");


            if (tempFile.is_open()){

   string line;

   bool usernameFound = false;


                    while (getline(searchUser, line)){


                              stringstream ss(line);

      string username, password;

      getline(ss, username, ',');

                              getline(ss, password);
            if (username == targetUsername.c_str() && reset->Text.Trim() != ""){


    password = newPassword.c_str();

    usernameFound = true;

                  }else{




                  }


  tempFile << username << "," << password << endl;

}


            searchUser.close();

tempFile.close();


            remove("users.txt");

            rename("temp.txt", "users.txt");


            if (usernameFound){

                  ShowMessage("Password has been successfully reset.");

            }




      }
```

```
}
```

SEARCH STUDENT INFORMATION AND DISPLAY PHOTO FEATURE

```
void __fastcall TDisP::searchBtnClick(TObject *Sender)
{
        AnsiString studNum = search_bar->Text;
        AnsiString filename = studNum + ".txt";

        if (search_bar->Text == " ") {
                ShowMessage("Please input student number!");
                return;
        }

        TStringList* lines = new TStringList();
        string line;

        fstream file;
        file.open(filename.c_str(), ios::in);

        if (!file.is_open()) {
                ShowMessage("Student number wasn't found! Please check if input is correct.");
        } else {
                while (getline(file, line)) {
                        lines->Add(line.c_str());
```

```
            file.close();

    }


    for (int i = 0; i < lines->Count; i++) {

            string line = AnsiString(lines->Strings[i]).c_str();

            char* token = strtok(const_cast<char*>(line.c_str()), ",");


            infoDisplay->Items->Add("Student ID: " + AnsiString(token));

            token = strtok(NULL, ",");

            infoDisplay->Items->Add("Course and Year: " + AnsiString(token));

            token = strtok(NULL, ",");

            infoDisplay->Items->Add("Name: " + AnsiString(token));

            token = strtok(NULL, ",");

            infoDisplay->Items->Add("Age: " + AnsiString(token));

token = strtok(NULL, ",");

            infoDisplay->Items->Add("Sex: " + AnsiString(token));

            token = strtok(NULL, ",");

            infoDisplay->Items->Add("Address: " + AnsiString(token));

            token = strtok(NULL, ",");

            infoDisplay->Items->Add("Email: " + AnsiString(token));

            token = strtok(NULL, ",");

            infoDisplay->Items->Add("Contact #: " + AnsiString(token));

    }


     String username = search_bar->Text.c_str();
    String dir =
"/Users/kiimz/OneDrive/Documents/Embarcadero/Studio/Projects/LearnLogB/userdata/photos/";
    String photoPath;

    // Iterate over common image file extensions to find the existing photo file
    const String imageExtensions[] = { ".jpg", ".jpeg", ".png", ".gif", ".bmp" };
    for (const String& extension : imageExtensions)
    {
        String photoFileName = username + extension;
        String fullPath = dir + photoFileName;

   // Check if the photo file exists
   if (TFile::Exists(fullPath))
   {
    photoPath = fullPath;
    break;
   }
  }

  if (!photoPath.IsEmpty())
  {
   userImg->Picture->LoadFromFile(photoPath);
    }
}
```

REGISTER INFORMATION AND UPLOAD PHOTO FEATURE

```
void __fastcall TRegisP::Label8Click(TObject *Sender)
{
         AnsiString student = this->student->Text;
   AnsiString course = this->course->Text;
   AnsiString name = this->name->Text;
   AnsiString age = this->age->Text;
   AnsiString sex = this->sex->Text;
   AnsiString address = this->add->Text;
   AnsiString email = this->email->Text;
   AnsiString contact = this->contact->Text;

   AnsiString filename = student + ".txt";

   // Check if any of the fields are empty
   if (student.IsEmpty() || course.IsEmpty() || name.IsEmpty() || age.IsEmpty() ||
      sex.IsEmpty() || address.IsEmpty() || email.IsEmpty() || contact.IsEmpty()) {
      ShowMessage("Please input the required information.");
      return;
         }
                  fstream file;
         file.open(filename.c_str(), ios::app);

         if (file.is_open()) {
            file << student.c_str() << "," << course.c_str() << "," << name.c_str() << "," << age.c_str() <<
"," << sex.c_str() << "," << address.c_str() << "," << email.c_str() << "," << contact.c_str() << endl;
        file.close();


        ShowMessage("Student details added.");
            } else {
        ShowMessage("Error opening file.");
            }
}

//---------------------------------------------------------------------------

void __fastcall TRegisP::Image9Click(TObject *Sender)
{
            Hide();
            MainP->Show();
}
//---------------------------------------------------------------------------



void __fastcall TRegisP::Button1Click(TObject *Sender)
{
            if(pictureDialog->Execute()){

         String path = pictureDialog->FileName;
String dir =
"/Users/kiimz/OneDrive/Documents/Embarcadero/Studio/Projects/LearnLogB/userdata/photos/";


if (!DirectoryExists(dir))
            ForceDirectories(dir);


        String fileName =  student->Text.c_str();
        String destPath = dir + fileName;


   if (TFile::Exists(destPath + ".jpg"))
            TFile::Delete(destPath + ".jpg");
        if (TFile::Exists(destPath + ".jpeg"))
            TFile::Delete(destPath + ".jpeg");
        if (TFile::Exists(destPath + ".png"))
            TFile::Delete(destPath + ".png");


        String fileExt = ExtractFileExt(path);
        fileExt = LowerCase(fileExt);


        if((fileExt == ".jpg") || (fileExt == ".jpeg") || (fileExt == ".png")){


        try{


        destPath += fileExt;
```

```
    ifstream srcFile (path.c_str(), ios::binary);

        ofstream destFile (destPath.c_str(), ios::binary);


        destFile << srcFile.rdbuf();


        srcFile.close();

        destFile.close();

  ShowMessage("Photo has been set.");

        }catch(EInOutError &e){


            ShowMessage(e.Message);


        }


            }else{


                ShowMessage("Unsupported file format. Please select a .jpg, .jpeg or
a .png file");


            }
```

VIII. Conclusion

The developers, created this classroom management system they gave importance of applying more advance programming concepts therefore understanding and using advanced programming principles in the creation of systems using the C++ programming language is important since it can make certain activities simple and convenient. As developers, while using more advance programming techniques they faced some instances on a certain feature, edit information due to lack of time to execute such code for that. Overall, they are aware that it can help the users of the system to lessen the paperwork and files that they should pass at school or university. Instead, with the help of the classroom management system, it is easy to encode and display various information for a certain user or student. And it also promotes a higher level of student engagement.