COMPOSITION OF THE TEAM

#	Registration No	Name	Role
01	11008	Ashan Lakshitha	Programmer
02	10916	Chamara Chandrasekara	Supportive Programmer
03	11285	Anjana Naween	Tester
04	10908	Sandalu Soyza	Requirements Gatherer
05	11547	Akila Dhananjana	Requirements Gatherer
06	11007	Tharaka Jayasekara	Documenter
07	11325	Tharaki Dewmini	Documenter
08	11287	Iresha Udayangani	Documenter
09	11360	Imesha Sewwandi	Interface Designer
10	11396	Chapa Janandi	Interface Designer
11	11411	Miurangi Manodhya	Logical Diagram Designer

ACKNOWLEDGEMENT

We extend our gratitude to Mr. Geeth Weerasinghe, our beloved mentor for his sincere guidance and help for completing this project.

We also grateful to all the lectures of DiTEC programme for their support and guidance during this course period.

A debt of gratitude is also owed to all administration and library staff for their assistance in finishing this project on time.

Finally, we acknowledge the contribution of all those who have helped us directly or indirectly with their good wishes and constructive criticism which lead to successful completion of our project.

Contents

1. BACKGROUND OF THE STUDY

1.1 Introduction

In this 21'st century, all is about technology. The schools and education institutes nowadays have also already changed. For not only University and secondary school, small-scale education institute also have their own management system.

However, the current management systems those educational institutes have already defined as old generation management systems and it cannot satisfy the user needed as they expect.

As a result, a system called Student Management System has evolved as an upgrade version of the old system to replace the manual system in order to solve the problems that face when using the old manual system.

A Student management system is a system, which deals with all kind of student details. It uses for the betterment of the students and for the administrative purpose as well.

This student management system has three users. They are,

- Administrator
- Teachers
- Students

Administrator can register students, and also can view all the information about students, exam details and lecture detail s. Teachers can search student details, update exam details, add announcement, add new student, change password. Student can search exam details, change password and search own details.

Proposed system design can facilitate client to explore the activities happening in the institute effectively and efficiently. This SMS is an automated version of manual student management system, which almost all work is computerized.

1.2 Background

1.2.1 Goals

This project helps for maintain the student data base. We can easily access any student's details any time and can be kept safely for long period of time without any damage.

Student management system can be used by this institute to keep records of students easily. Achieving this goal is difficult by using a manual system. The information is scattered.

It can be redundant and collection relevant information may be very time consuming. All these problems are solved using this project.

This system must perform the following

- Register students
- Search Students details
- Update Students details
- Remove Students details
- Add exam result
- Search exam results
- Register teachers

Without a student management system managing and maintaining the details of the student is a tedious job for any organization. Student management system will store all the details of the students including their background information, personal details and all the information related to their resume.

1.3 Objectives

The main objective of this project is to establish an effective SMS, which enables to automate the dynamic administrative processes in the institute. In addition to that following sub objectives can be attained through this system:

I. Supporting the decision making process.

II. Improving the services provided to the students, and parents.

III. Improving the accuracy of the follow up and management of student data in the institute.

1.4 Challenge

- Here this outline the challenges faced during the implementation of the project.
- Identify the tools for the job.
- Mismanaged time looking for project documents or asset was wasting too much time on requirement gathering.
- Spend much time in meeting.

A successful project manager knows that repeatedly asking a team for the impossible can quickly result in declining morale and productivity. The odds of successfully completion project under unreasonable deadlines are generally not feasible exceptions.

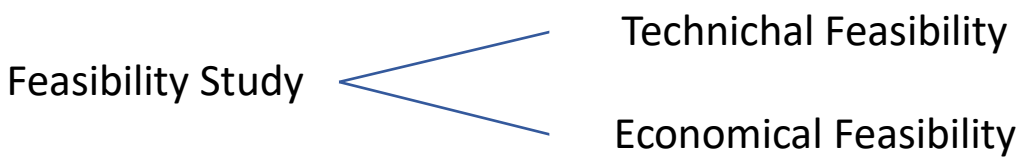
1.5 Feasibility of the Project

This is one of the most important sections of the feasibility study as it examines the marketability of the product or services and convinces readers that there is a potential market for the product or services.

If a significant market for the product or services cannot be established, then there is no project. Typically, market studies will assess the potential sales of the product, absorption and market capture rates and the project's timing.

Mainly the feasibility study outputs the feasibility study report, a report detailing the evaluation criteria, the study findings, and the recommendations.

According to the feasibility study two main sub areas were considered as follows,



1.5.1 Technical Feasibility

Feasibility say that is it technical feasibility, since there will not be much difficulty in getting required resources for the development and maintain the system as well.

All the resources needed for the development of the software as well as maintenance of the same is available in the organization.

Here we are utilizing the resources, which are available already.

1.5.2 Economic Feasibility

Development of this application is highly economical feasible. The organization not needed to spend much time on the development of the system because of it is already available.

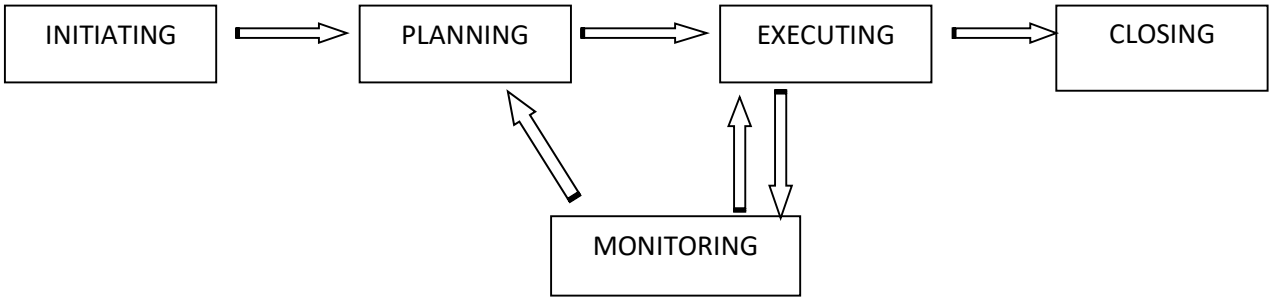
The only thing that want to be done is making an environment for the development with an effective super vision. If we are doing so, we can attain the maximum feasibility of the corresponding resources.

Even after the development, the organization will not be in a condition to invest more in the organization. Therefore, the system is automatically feasible.

2. PROJECT REVIEW

2.1 Project Planning

The proposed project of Student Management System were planned in such a way to take the maximum of the team members. Accordingly, each team member had his or her specified role in the overall process of this Student Management System. As a result, the process of initiating, planning, executing, closing, and overall monitoring were used as the key base phases for this project and all the other this with respect to this project were planned accordingly.



2.2 SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)

Software development is not a single phase. There are set of activities.

A software development life cycle model is a descriptive and diagrammatic representation of the software life cycle.

A life cycle model representation all the activities required to make a software product transit through is life cycle phase. It also captures the order in which these activities are to be undertaken.

When a software product is being developed by a team there must be a clear understanding among team members about when and what to do. Otherwise it would lead to chaos and project failure.

2.3 Software Development Model

The software process is a set of activities associated with productions of a software Product for the effective control of the software process is the essential to have a phase development state.

That is it consists of number of stages that produce one or more documents and programming codes.

The software development model is also called SDLC (software development life cycle) as the process is repeated when software need to be changed as long as the software is in use.

Different software development models

- ✓ Agile method
- ✓ The waterfall model
- ✓ "V" model
- ✓ Prototyping model
- ✓ Evolutionary model
- ✓ Spiral method
- ✓ Rapid application development

2.4 Waterfall Model

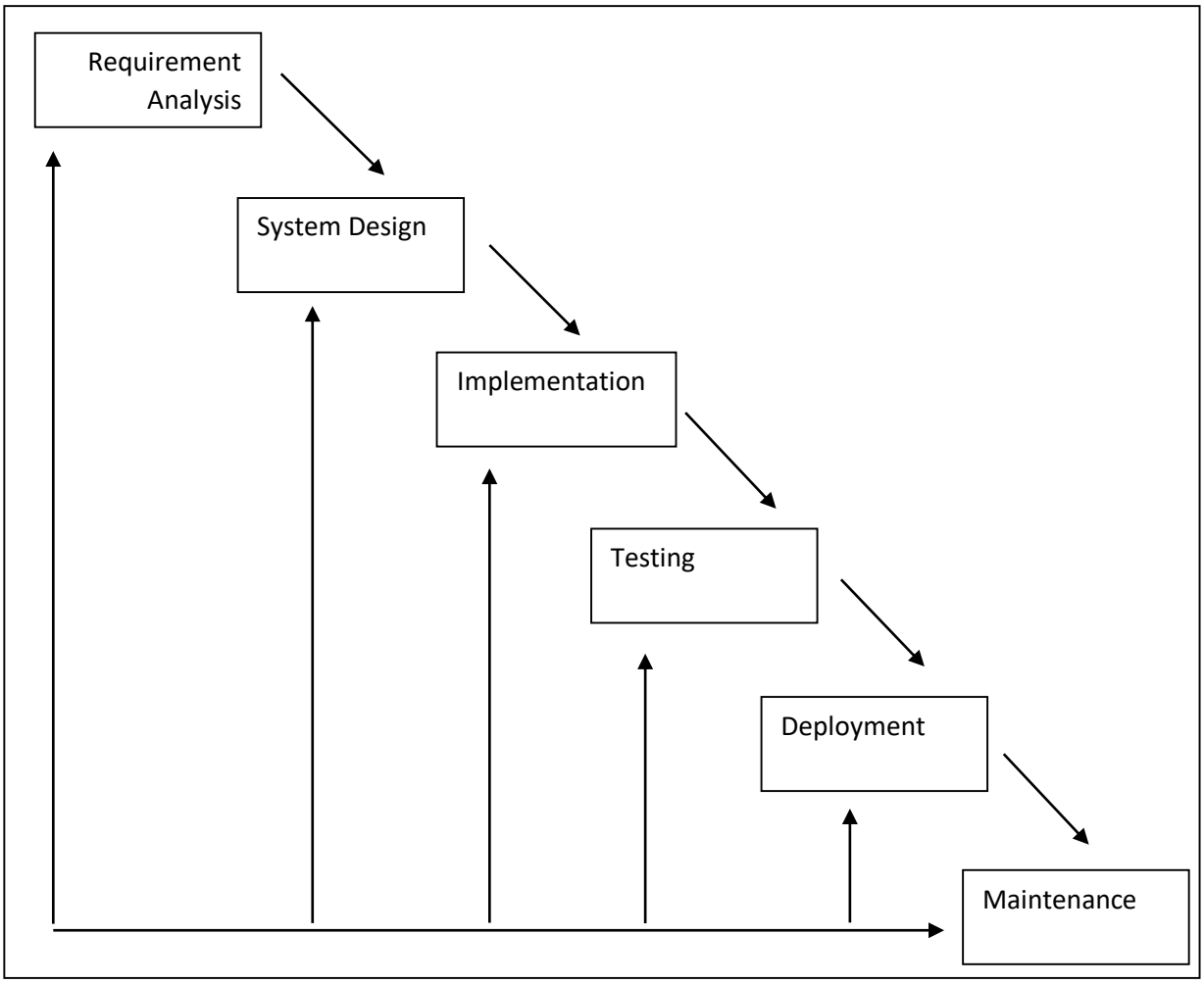
Referring to the Basic Project Planning Process team prioritize the software engineering model that can use for the SMS.

Accordingly, Waterfall Model was selected and overall SMS was created based on its phases.

Waterfall Model is a software development approach that used in earliest Software Development Life Cycle (SDLC).

It illustrates the software development in sequential manner without any overlapping in the phases.

That means the outcome of one-phase acts as the input for the next phase sequentially.



1. **Requirements Analysis and definitions**

The system services, limitations (constraints) and goals are established by consolation with system users. They are then defined in a manner, which is understandable by both users and development.

2. **System and software design**

The system design process partitions the requirements the requirements to either hardware or software systems. It established overall system architecture.

Software involves representing the software system function in a form that may be transformed into one or more executable programs.

3. **Implementation and unit testing**

During this stage the software design is realized as a set of programs or program units. Unit testing involves that each meets its specification.

4. **Integration and system testing**

Individual program units or programs are integrated and tested as a complete system to ensure that software requirements have been met. After testing, software system is delivered to the consumer.

5. **Operations and maintenance**

Normally (although not necessarily) this is the longest life cycle phase. The system is installed and put into practical use.

Maintenance involves correcting errors which are on to discover in earlier stages of the life cycle, improving the implementation of system units and adding the systems requirements as a new requirement are discovered.

2.4.1. Suitability of Waterfall Model

As the focus client of the SMS is a small-scale private educational services provider, the system should be very simple and user friendly. Therefore, waterfall model is the comparatively most suitable software engineering model compared to other models.

Because of the positive points of the waterfall model such as easiness in system designing and documentation, clarity in model phases, and its ability to identify the user requirements quickly.

2.5 Requirement Phase

Requirements phase focuses on defining and capturing the needs and problems that a software application is to address and solve.

The requirements phase is the first phase in waterfall model, setting the stage for the rest of the phases of the software application development.

As per the Student Management System, there were two dedicated team members to collect facts and figures of our client. Accordingly, this was carried out as one of the initial task as per our basic project planning process.

2.5.1 Sources of Data

As per the initial discussions of our team, we decided to have a brainstorming with our client. Accordingly, planned an initial discussion and to do observations thereafter.

2.5.1.1 Initial Discussion/Interview

Under this, more priority was given to the business and functional requirements of the client. Accordingly, we gathered some information in this regard and obtained kind of an understanding of on the client's business.

2.5.1.2 Observations

Entire process investigation, documents and papers investigation were subjected to this stage as it facilitates to see what is going on exactly.

2.5.2 Design Description

This is the second stage of the waterfall model. Under this stage, gathered information and requirement specifications are studied thoroughly and start to plan accordingly.

In line with Student management system, the system design was done in two phases as logical design and physical design.

Under the logical design, more priority was given to interface and database designs respectively and system design was taken into consideration in physical design phase.

2.5.3 Logical Design Phase

Logical Design means the way of designing relationships and other mapping subject to a logical schema. This only shows the physical framework of the student management system.

With respect to the student management system, this was done based on Unified Modeling Language diagrams such as use case diagrams, class diagrams and entity relationship diagrams.

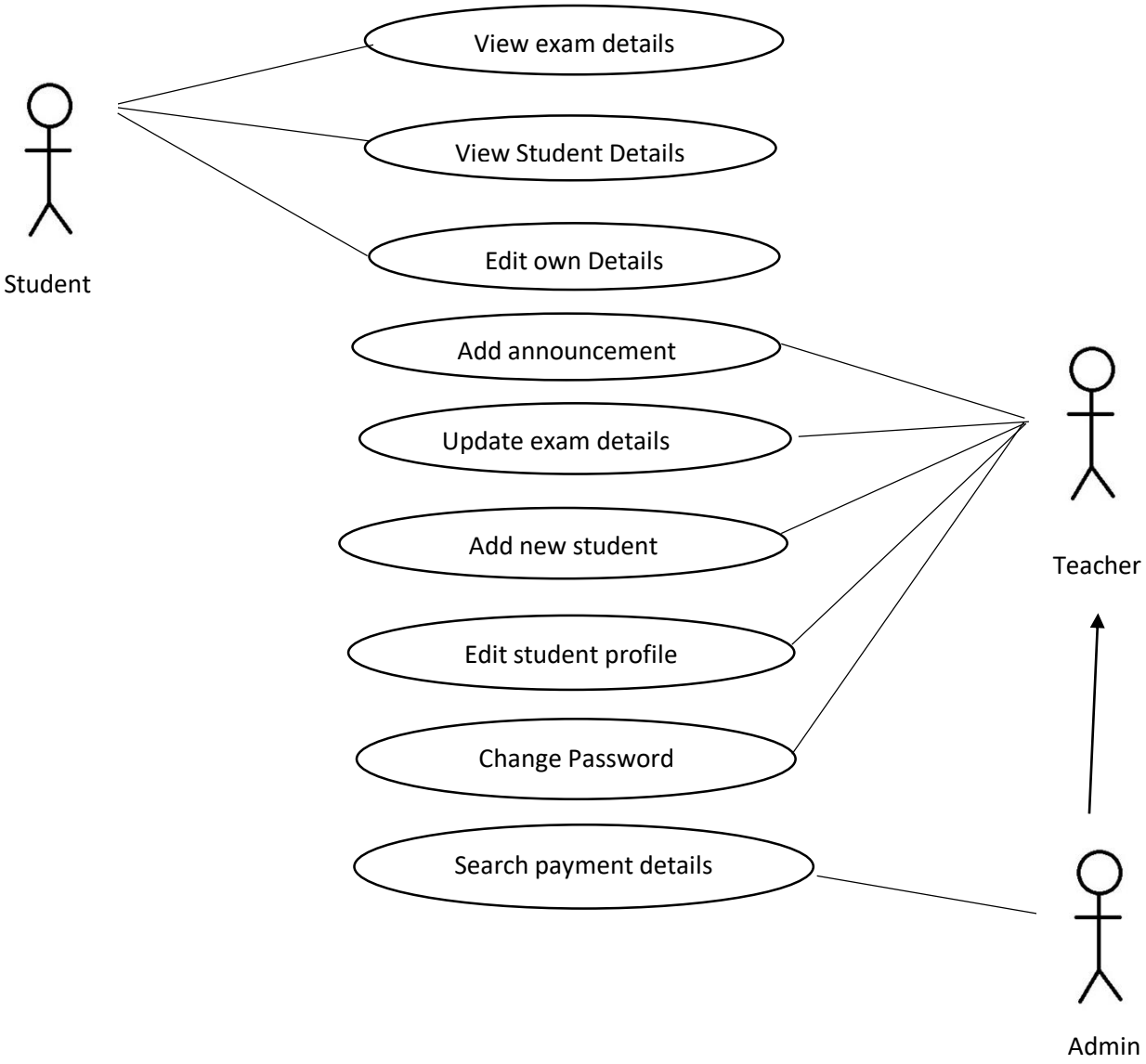
2.5.3.1 Use Case Diagram

Use Case Diagram is a set of tasks that a particular system can or should perform with the related parties to it.

The main purpose of is to show the related persons to the system and main goals they achieve with it.

Use case diagram comprises with four main components as Actor, Use Case, Subject, and Communication Line. With respect to the Student management system, there are two main Actors as Student and Teacher while student management system playing its role as Subject.

Further, the tasks that they perform. such as registration, course details updating and etc can denote as Use Cases and the way they see these tasks as Communication Lines. Accordingly, the use case diagram with respect to student management system can be depicted as follows,



As depicted in the above use case diagram, the main tasks of student registration, payment collections, paper marking, and class schedule management do in four ways as insert, update, search, and delete respectively.

From student's perspective, the Student management system allows only searching of his or her respective exam marks and class schedule within the system.

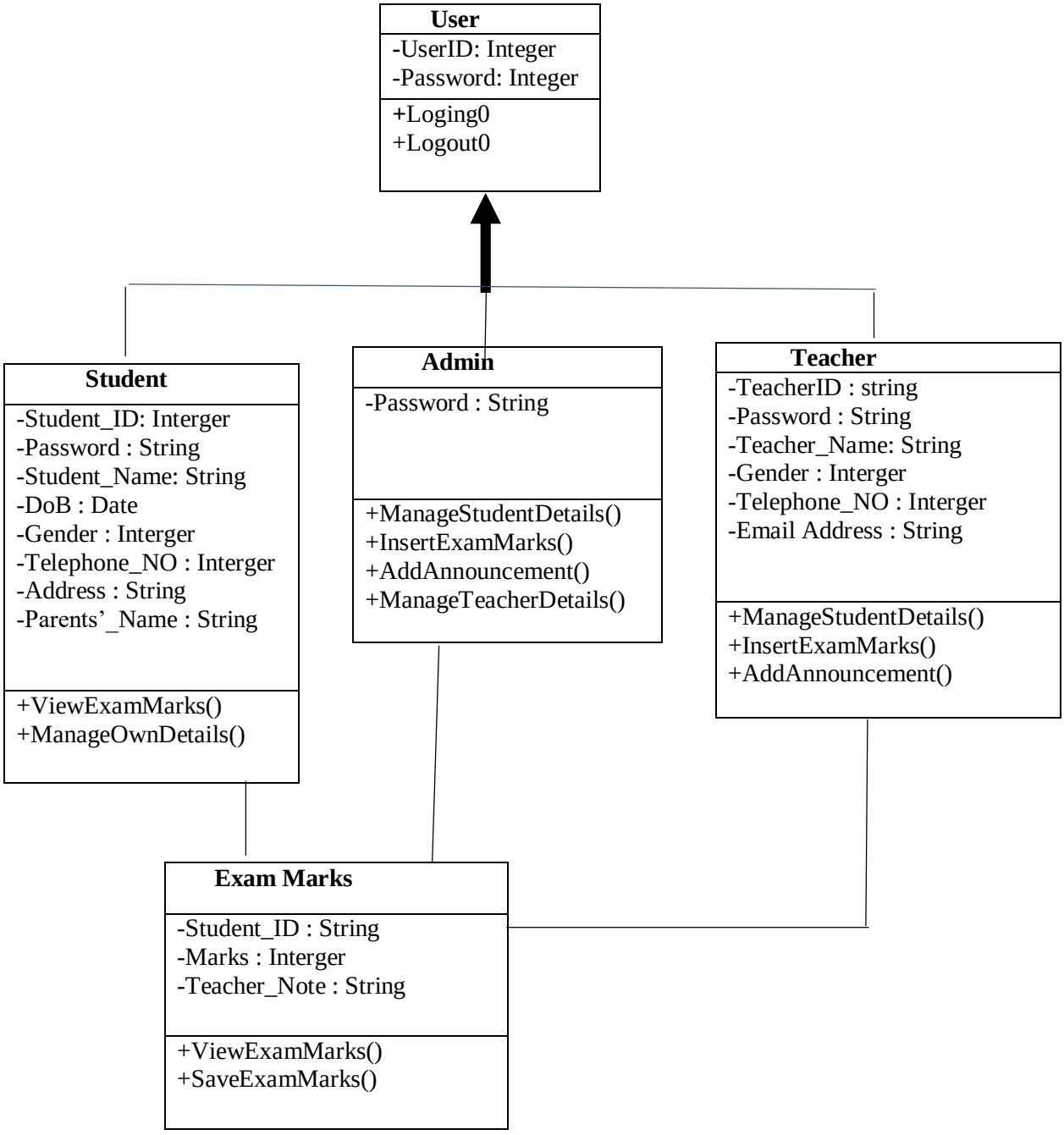
Whereas, it allows teacher to do all the functions within her scope.

2.5.3.2 Class Diagram (CD)

Class Diagram (CD) is a static structural diagram that shows the structure of the system by way of system's classes, their attributes, operations (or methods), and the relationships among objects.

The main aim of the class diagram is to give a structural picture for the future development of the system. As mentioned above class diagram comprises of classes and combined relationships to them.

A class has three parts as class name, class attributes, and class operations.



As depicted in the above class diagram, it has five classes namely user, student, teacher, exam marks, attendance, course details, and payments.

Each class has its respective attributes and operations along with the visibility of minus and plus marks respectively.

The minus mark denotes the privately available attributes while plus mark standing for operations that can be performed in a public manner.

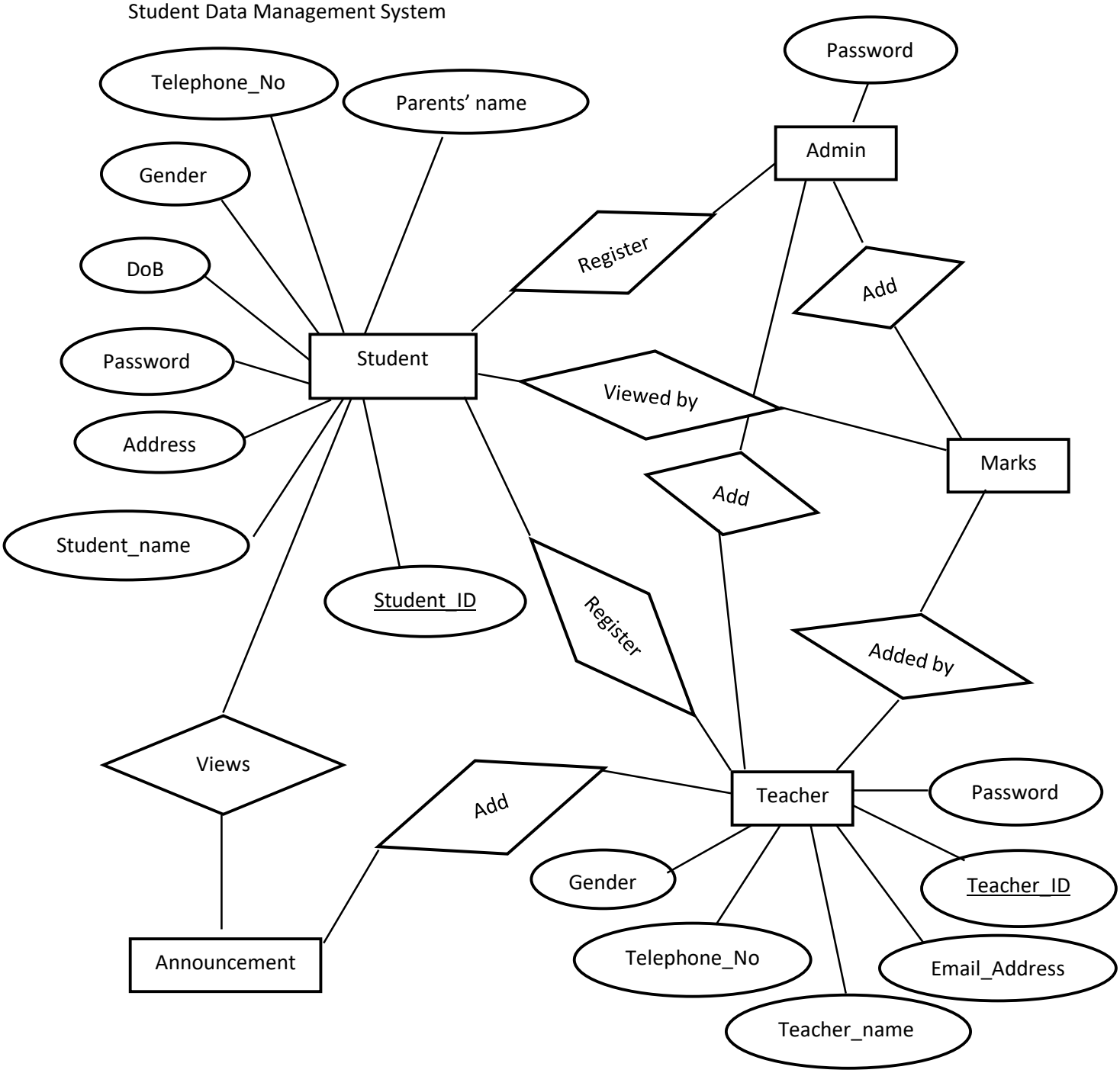
2.5.3.3 Entity Relationship Diagrams

It is a graphical representation of how entities (concepts or things) relate to one another. The aim of the Entity Relationship Diagram is to analysis the existing systems to find and resolve issues in rational and development.

In an ER diagram, there are model components as entities, attributes, and relationships. Entities means a thing or concept in a system, which is represented using a rectangular.

The attributes mean a detail description about the entity. It depicts using an ellipsis. Further, the relationship is a logical interaction between entities.

Accordingly, a separate ER Diagram was developed with respect to the student management system,



As per the above diagram, student and teacher entities have one to many relationship and the underscored ellipses denotes the unique attribute with respect to that particular entity.

2.5.4 Physical Design Phase

It is the actual input and output processes of the system. That means, how data is given as an input into the system, how it is verified, how it is processed and how it is represented as outputs.

With respect to the Student management system, the physical design was done in order to create more user-friendly system.

Accordingly, user-friendly fonts, colors, themes, and data designs were used in the entire system design while giving attention to the process designs as well.

2.5.5 Database Design

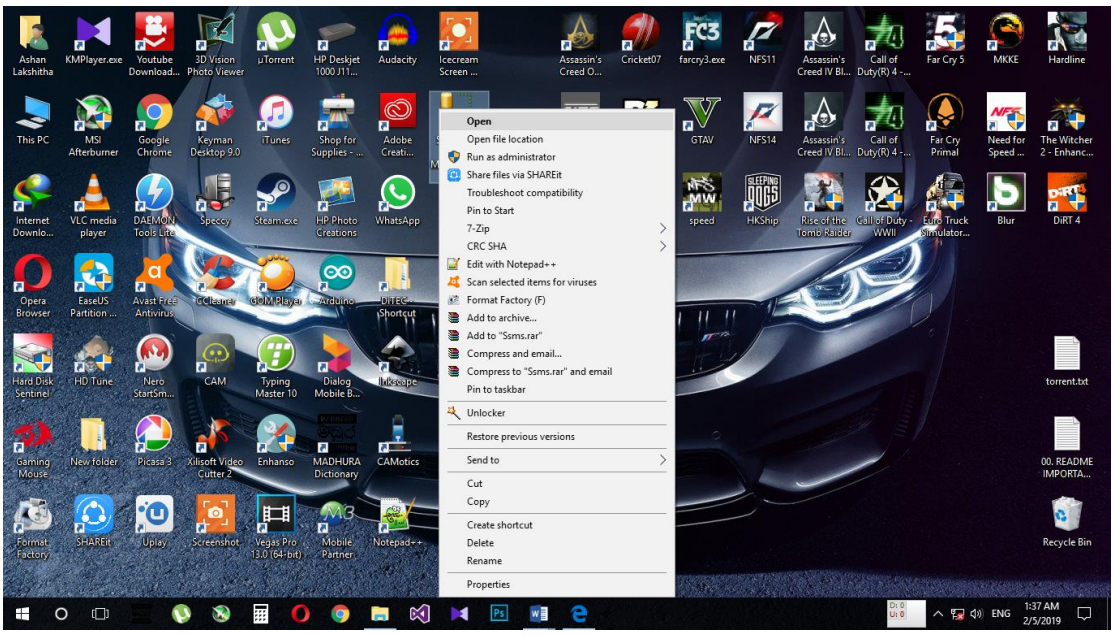
The database design involves the creating of tables where each table deals a particular set of information and each table contains columns where each column is a field name the data is inserted into the column.

The database used in the making of this Student Management System is SQL Server management 2014. There are five three in the database. There are primary key fields that uniquely identify a record in a table.

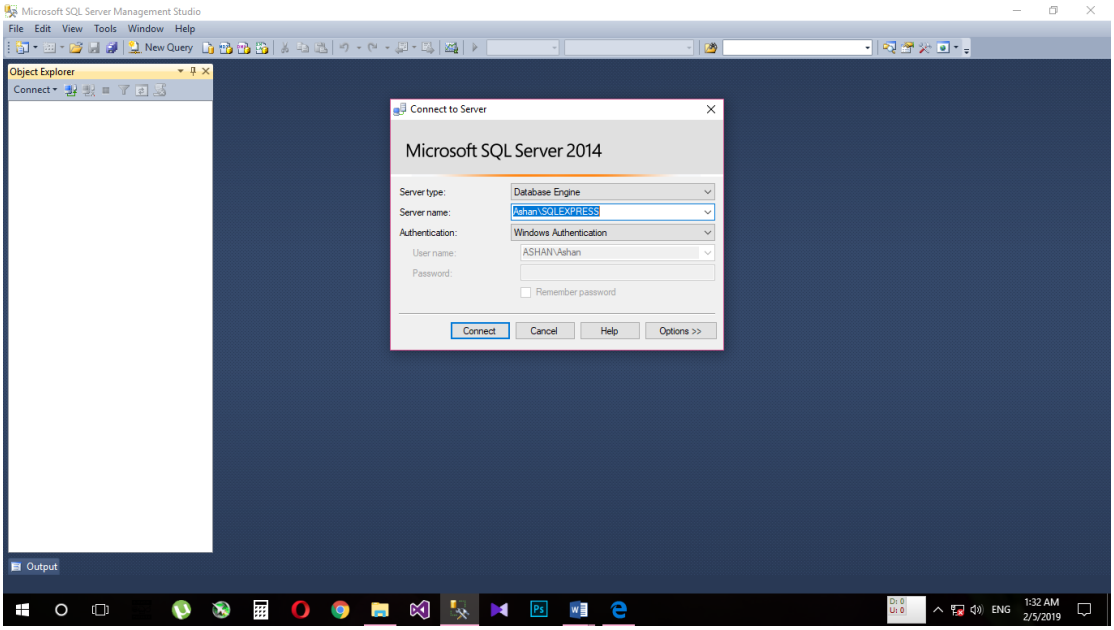
There are also fields that contain primary key from another table called foreign keys.

2.3.5.1 Database Pathway

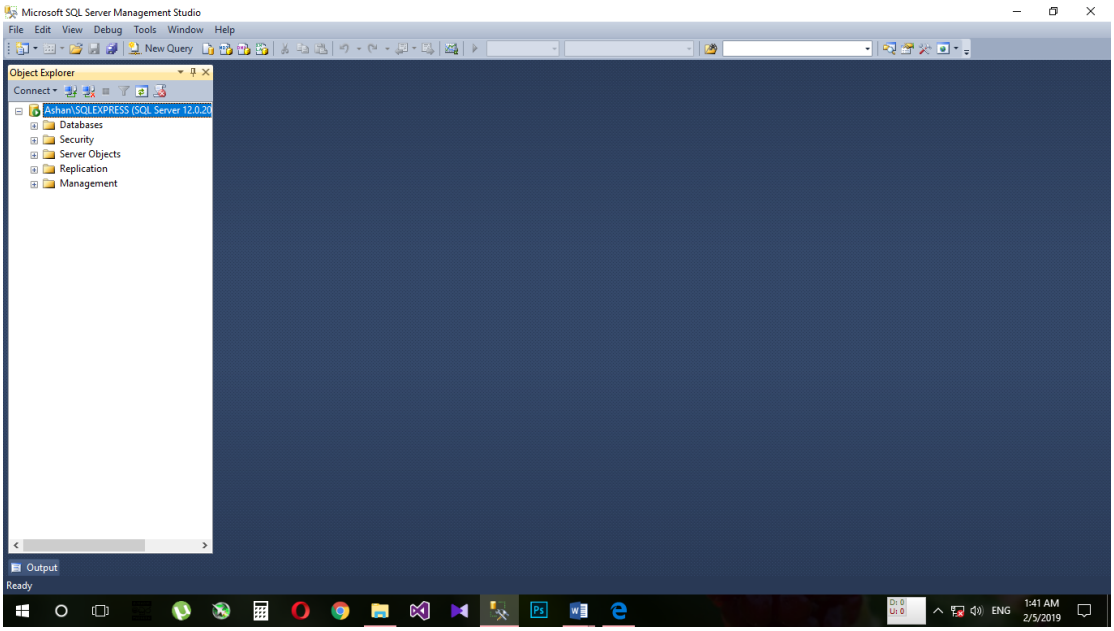
Go to SQL Server 2014 Management Studio and Open it



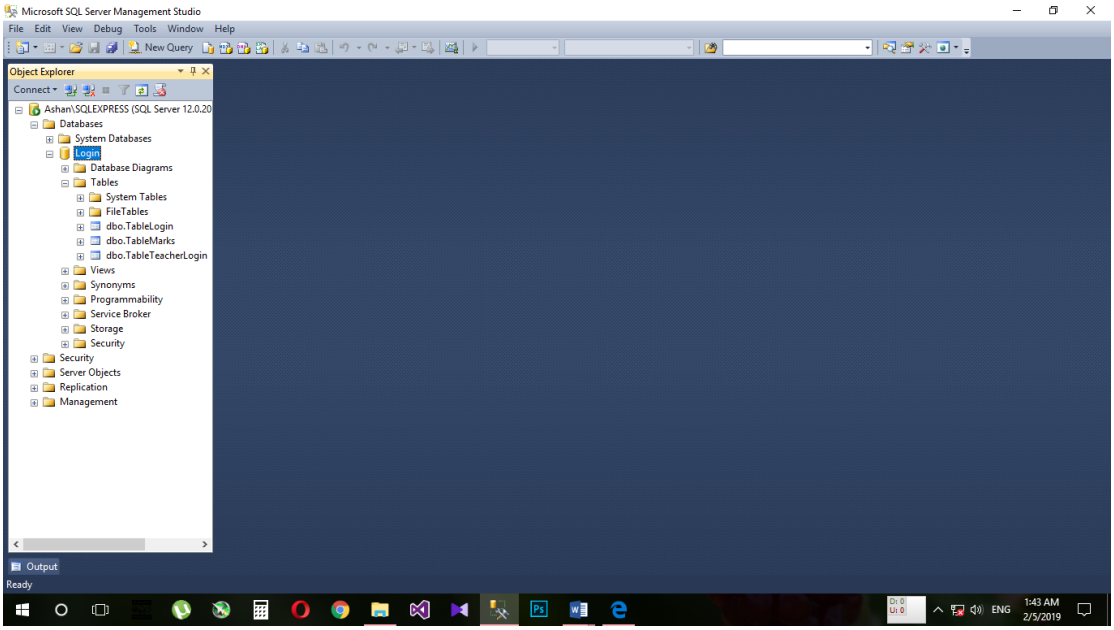
Then, you can see the SQL Server home screen and type server name. Then click on “Connect” button.



Once you click on the Connect button, you can see a window like below.



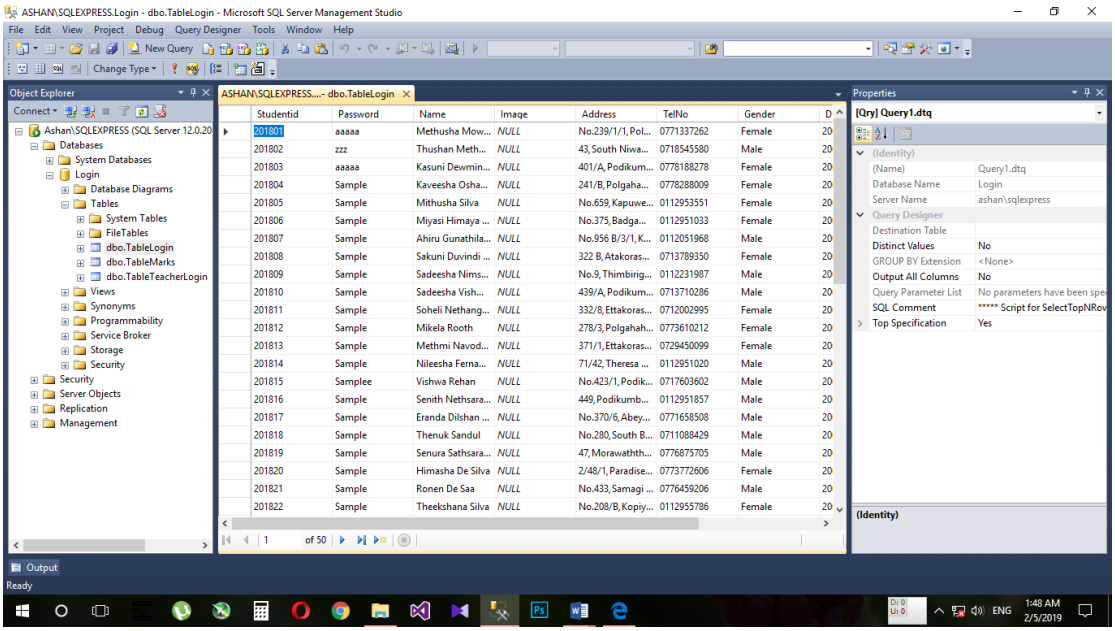
Then click on “Databases” to see databases and ten click on database name and then click on tables. Now you can see all the data tables in your database.



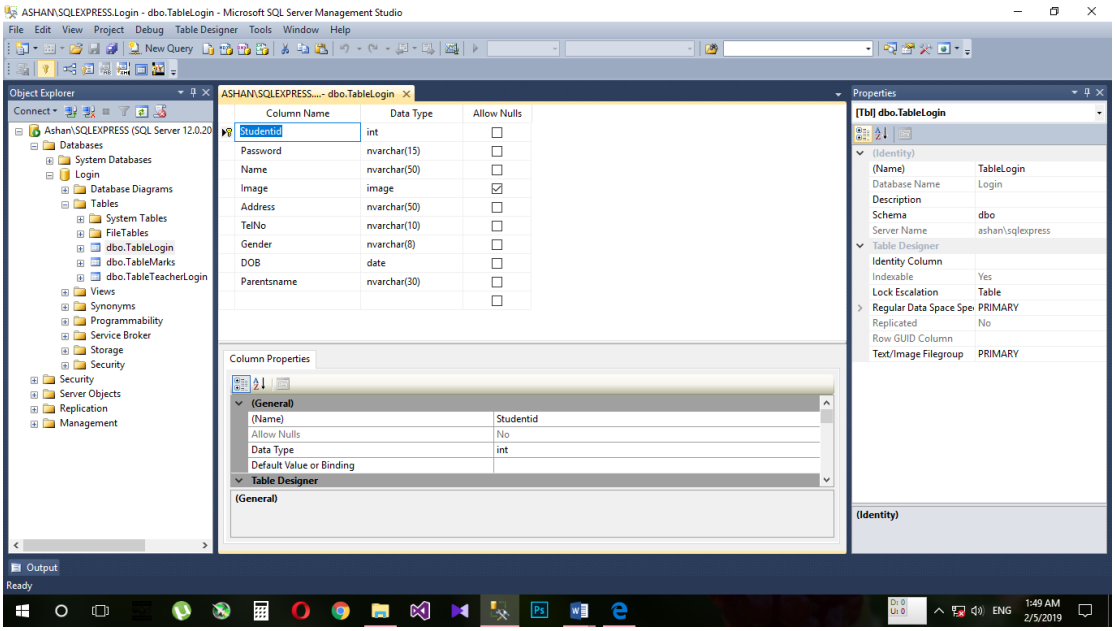
2.3.5.2 Database Composition

Under this all the database tables and its respective table structure has depicted under its respective entity name.

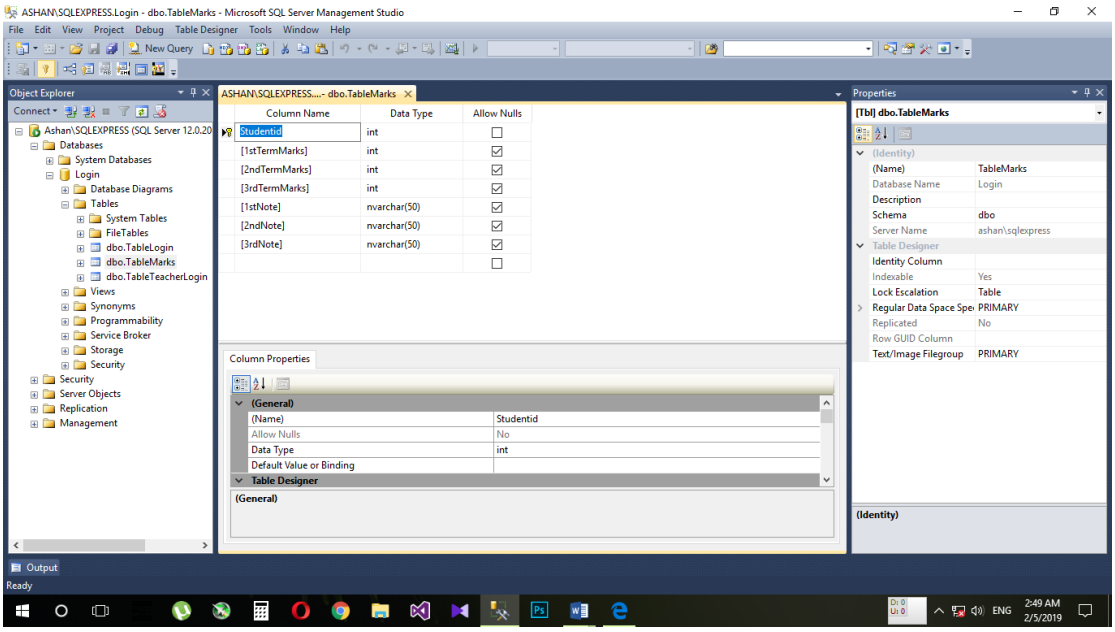
- Student Details Database



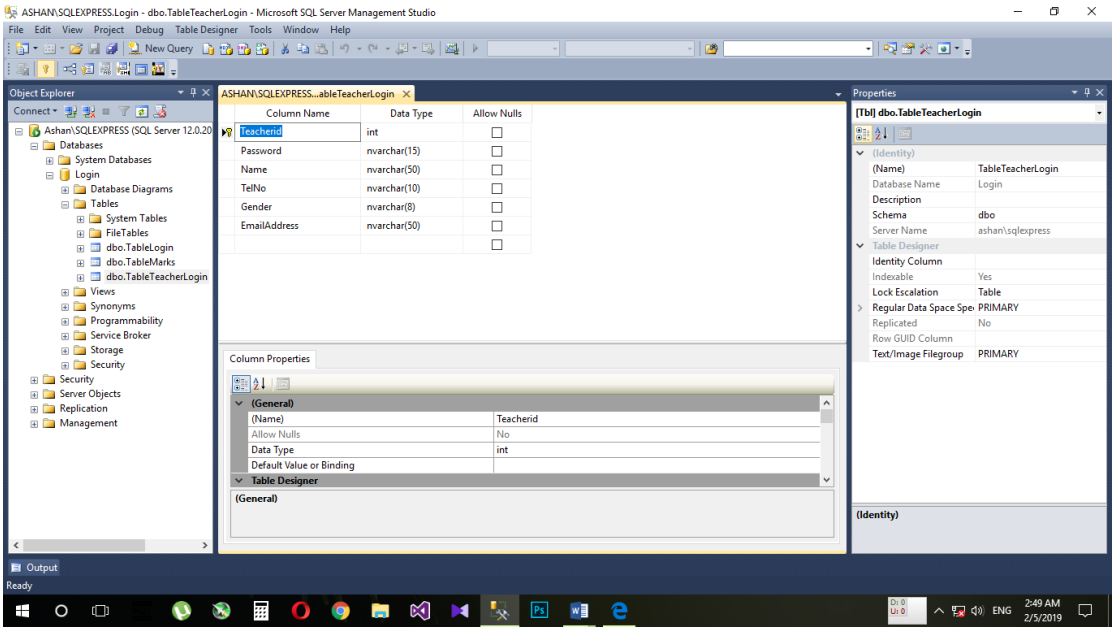
- Student Details Database Structure



- Exam Marks Database Structure



- Teacher Login Database Structure



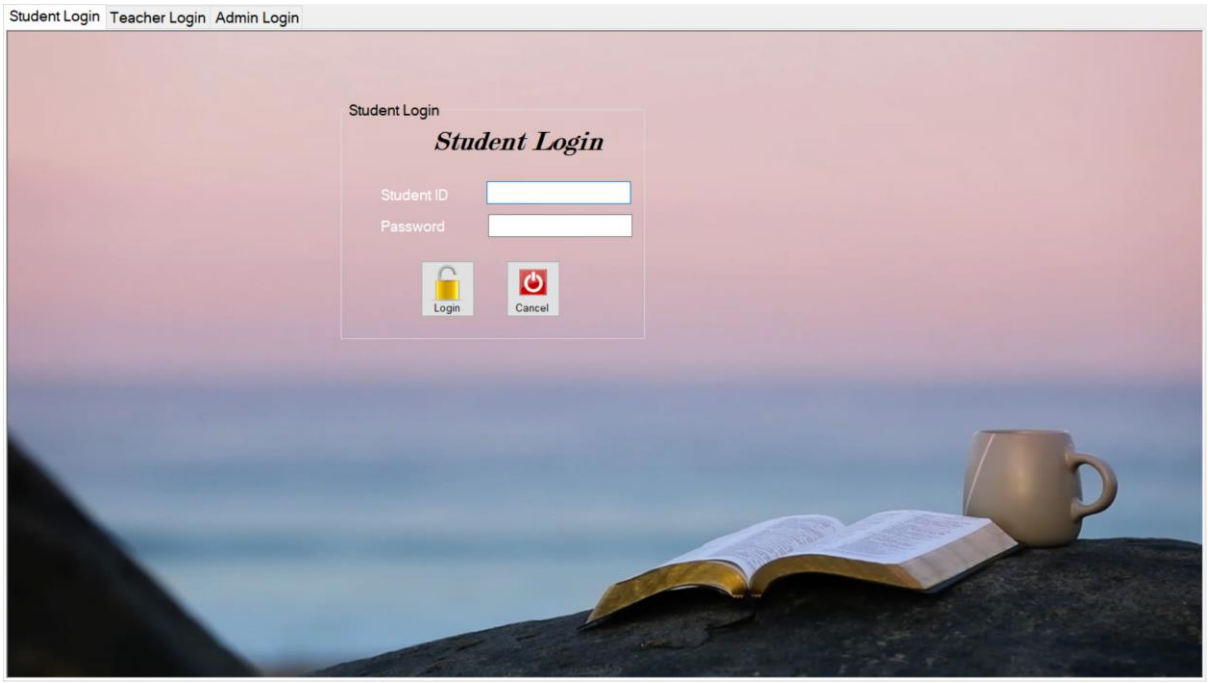
3. 3 CODING/CONSTRUCTION

This is the phase where requirements implementation is done as per the predetermined design scope. That means, under this phase a visible system can be seen after this phase.

As per the SMS, there is programmer with a backup programmer to bring all the design specification into reality. Withal, the visual C # was used as the source code configuration tool in creating this SMS. Accordingly, systematic coding flow can be depicted with its respective form as follows;

3.1 System Coding Flow

3.1.1 Login Form



3.1.1.1 Login Button

```
private void bttnLogin_Click(object sender, EventArgs e)
      {
         //Set the static variable
         useridpassing = txtUsername.Text;
         //Check Student id text field
         if (txtUsername.Text == string.Empty)
         {
            MessageBox.Show("Student ID is required!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            txtUsername.Focus();
         }
         else
         {
            //Check Password text field
```

```
            if (txtPassword.Text == string.Empty)
            {
                MessageBox.Show("Password is required!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                txtPassword.Focus();
            }
            else
            {
                if (txtUsername.TextLength<6)
                {
                    MessageBox.Show("Enter valid Student ID!", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                    txtUsername.Select();
                    txtUsername.SelectAll();
                }
                else
                {
                //Using error handler tool
                try
                {
                    //set the SQL statement
                    sql = "select * from TableLogin where Studentid='" + txtUsername.Text + "'";

                    //open the connection
                    sqlconn.Open();
                    //set the statement in command control
                    sqlcomm = new SqlCommand(sql, sqlconn);
                    //read the command
                    sqlread = sqlcomm.ExecuteReader();
                    //Using error handler tool
                    try
                    {
                        //Load the records
                        sqlread.Read();
                        {
                            //Creating memory variable
                            string login = sqlread[1].ToString();
                            usernamepassing = sqlread[2].ToString();


                            //Checking password
                            if (login == txtPassword.Text)

                            {
                                //Student login
                                MessageBox.Show("Welcome " + sqlread[2] + "!!", "Welcome",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                                this.Hide();
                                frmStudent df = new frmStudent();
                                df.Show();
                            }


                            else
                            {
```

```
                        //Clearing password text field if password is wrong
                        txtPassword.Clear();
                        MessageBox.Show("Student ID or Password is wrong!", "login Faild",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                    }

                }
            }
            catch (Exception ex)
            {
                //Disply error
                txtPassword.Clear();
                MessageBox.Show("Error!" + ex.Message, "Student Details",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
        catch (Exception ex)
        {
            //display error
            txtPassword.Clear();
            MessageBox.Show("Error!" + ex.Message, "Student Details",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
        finally
        {
            //close the connection
            sqlconn.Close();
        }
    }
  }
}
```

3.1.1.2 Cancel Button

```
private void bttnCancel_Click(object sender, EventArgs e)
    {
      //Display message with yes no buttons
      DialogResult dirRes;
      dirRes = MessageBox.Show("Do you want to exit?", "Login Form",
MessageBoxButtons.YesNo, MessageBoxIcon.Question);
      if (dirRes == DialogResult.Yes)
      {
        //Exit the program
        Application.Exit();
      }
    }
```

3.1.1.2 Student ID Validation

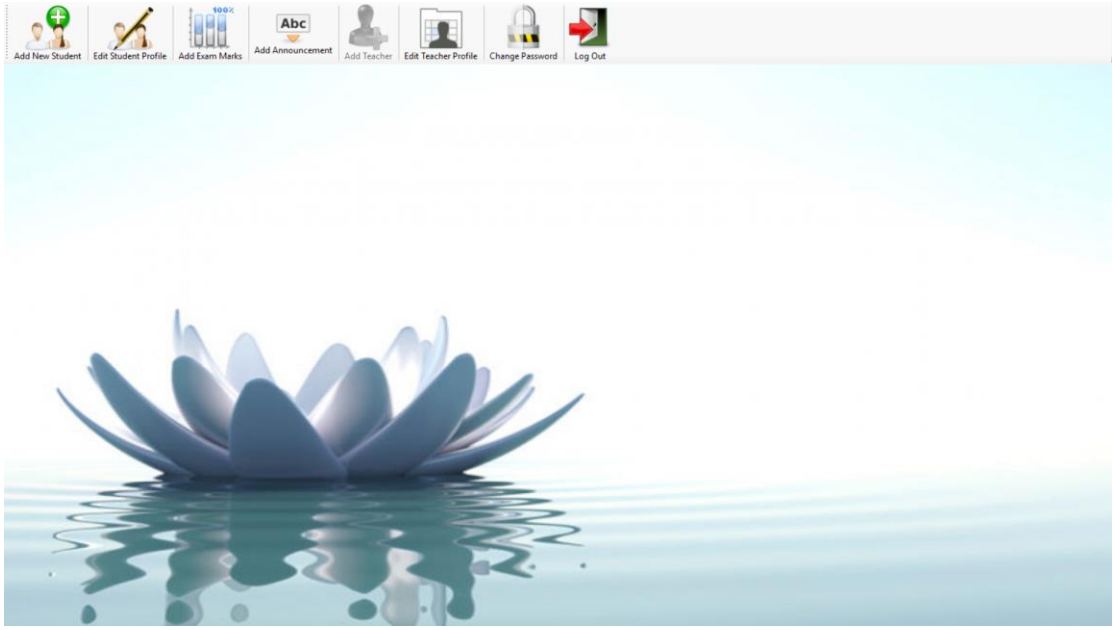
```
private void txtUsername_TextChanged(object sender, EventArgs e)
      {
         //Validate Username
         if (txtUsername.Text.Length > 0)
         {
            if (!System.Text.RegularExpressions.Regex.IsMatch(txtUsername.Text, "^[0-
9]+$"))
            {
               MessageBox.Show("Please enter a valid User Id", "Add Marks",
MessageBoxButtons.OK, MessageBoxIcon.Information);
               txtUsername.Text = txtUsername.Text.Substring(0, txtUsername.Text.Length -
1);
               txtUsername.Select(5, 0);
            }
         }
      }
```

3.1.2 Student Dashboard

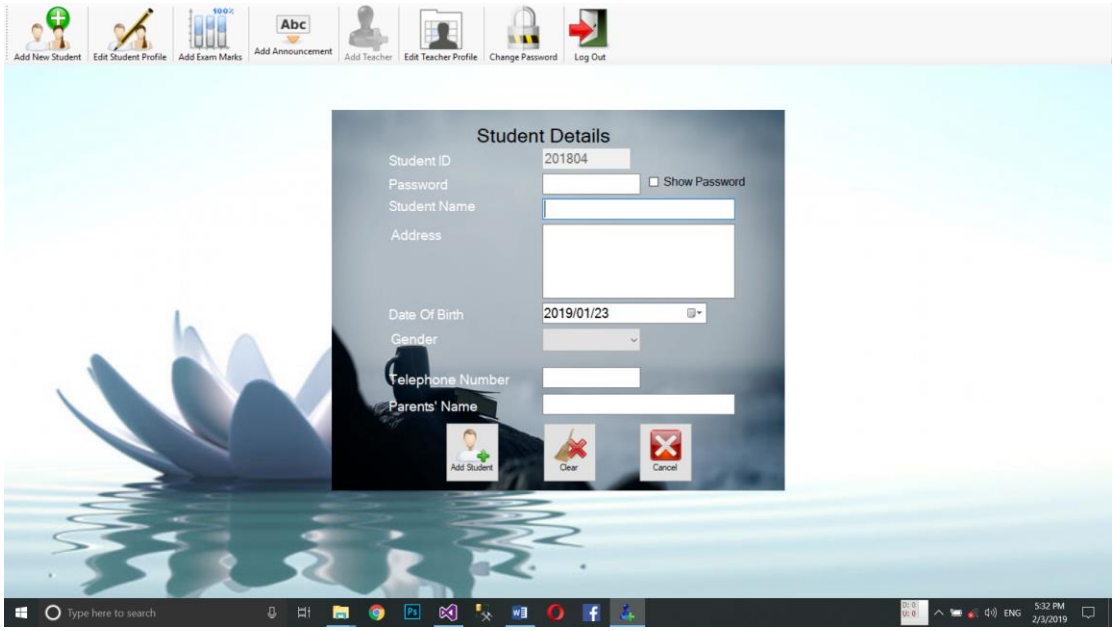
3.1.2.1 My details button

```
  private void toolStripButton1_Click(object sender, EventArgs e)
      {
         frmStudentDetails ab = new frmStudentDetails();
         ab.ShowDialog(this);

      }
```

3.1.3 Teacher Dashboard



3.1.4 Add Student Form



3.1.4.1 Automatic Student ID assigning

```
private void frmAddNewStudent_Load(object sender, EventArgs e)
        {
            txtPassword.UseSystemPasswordChar = true;
            txtSdudentname.Select();
            //Set the connection statement
            sql = "Data Source=Ashan\\SQLEXPRESS;Initial Catalog=Login;Integrated
Security=True";
            //Assigning the connection
            sqlconn = new SqlConnection(sql);
```

```
        //Set the SQL statement
        sql = "select MAX(Studentid) from TableLogin";

        //open the connection
        sqlconn.Open();
        //set the statement in command control
        sqlcomm = new SqlCommand(sql, sqlconn);
        //read the command
        sqlread = sqlcomm.ExecuteReader();
        {
           //Load the records
           sqlread.Read();
           {
              //Assigning the Values
              maxstudentid = sqlread[0].ToString();
           }
        }
        //close the connection
        sqlconn.Close();

        txtStudentid.Text = (Convert.ToInt32(maxstudentid) +
Convert.ToInt32(1)).ToString();

    }
```

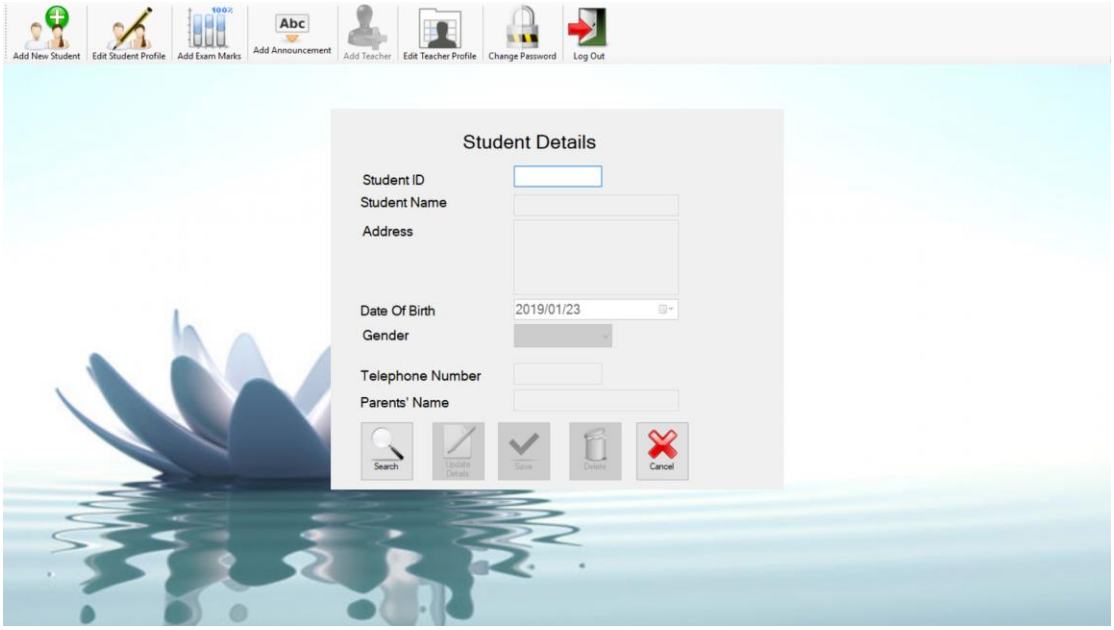
3.1.4.2 Add student button

```
    private void btnUpdate_Click(object sender, EventArgs e)
    {
        if ((txtSdudentname.Text==string.Empty) || (txtPassword.Text== string.Empty) ||
(txtAddress.Text== string.Empty) || (cmbGender.Text== string.Empty) || (txtTelno.Text==
string.Empty) || (txtParentsname.Text== string.Empty))
        {
            MessageBox.Show("Please fill all details", "Add Student",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        else
        {
            if (!System.Text.RegularExpressions.Regex.IsMatch(txtSdudentname.Text, "^[a-
zA-Z ]+$"))
            {
                MessageBox.Show("Please enter a valid Student Name", "Add Student",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                txtSdudentname.Focus();
                txtSdudentname.SelectAll();
            }
            else
            {
                if (txtPassword.Text.Length<5)
                {
                    MessageBox.Show("Your password must be at least 5 characters long. Please
try another","Add Student",MessageBoxButtons.OK,MessageBoxIcon.Information);
                    txtPassword.Focus();
                    txtPassword.SelectAll();
```

```
            }
            else
            {
                if ((!System.Text.RegularExpressions.Regex.IsMatch(txtTelno.Text, "^[0-
9]+$")) || (txtTelno.Text.Length<=9))
                {
                    MessageBox.Show("Please enter a valid Telephone Number", "Add
Student", MessageBoxButtons.OK, MessageBoxIcon.Information);
                    txtTelno.Focus();
                    txtTelno.SelectAll();
                }
                else
                {
                    if (!System.Text.RegularExpressions.Regex.IsMatch(txtParentsname.Text,
"^[a-zA-Z ]+$"))
                    {
                        MessageBox.Show("Please enter a valid Parents' Name", "Add Student",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                        txtParentsname.Focus();
                        txtParentsname.SelectAll();
                    }
                    else
                    {
                        //Using error handler tool
                        try
                        {
                            //set the update statement
                            sql = "insert into TableLogin
(Studentid,Password,Name,Address,TelNo,Gender,DOB,Parentsname) values ('" +
txtStudentid.Text + "','" + txtPassword.Text + "','" + txtSdudentname.Text + "','" +
txtAddress.Text + "','" + txtTelno.Text + "','" + cmbGender.Text + "','" + dtpDOB.Text + "','"
+ txtParentsname.Text + "')";

                            //open the connection
                            sqlconn.Open();
                            //Assigning the statement in command control
                            sqlcomm = new SqlCommand(sql, sqlconn);
                            //Write the record
                            sqlcomm.ExecuteNonQuery();

                            //Display message
                            MessageBox.Show("Details added successfully!", "Student details",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                        }

                        catch (Exception ex)
                        {
                            //Disply error
                            MessageBox.Show("Error! " + ex.Message, "Student Details",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                        }
                        finally
                        {

                            //Close the connection
```

```
            sqlconn.Close();
        }

        //Call the clear procedure
        clear();
```

3.1.5 Student Details Form



3.1.5.1 Form Load

```
    private void frmEditStudentDetails_Load(object sender, EventArgs e)
    {
        //disable delete button
        btnDelete.Enabled = false;
        //Set the connection statement
        sql = "Data Source=Ashan\\SQLEXPRESS;Initial Catalog=Login;Integrated
Security=True";
        //Assigning the connection
        sqlconn = new SqlConnection(sql); }
```

3.1.5.2 Search Button

```
private void btnSearch_Click(object sender, EventArgs e)
    {
        if (txtStudentid.TextLength < 6)
        {
            MessageBox.Show("Enter valid Student ID", "Edit student details",
MessageBoxButtons.OK, MessageBoxIcon.Information);
            txtStudentid.Select();
            clear();
        }
        else
        {
            //Using the error handler tool
            try
            {
```

```
            //set the SQL statement
            sql = "select * from TableLogin where Studentid='" + txtStudentid.Text + "'";

            //open the connection
            sqlconn.Open();
            //set the statement in command control
            sqlcomm = new SqlCommand(sql, sqlconn);
            //read the command
            sqlread = sqlcomm.ExecuteReader();

            //Using the error handler tool
            try
            {

               //Load the records
               sqlread.Read();
               {
                  //Assigning the Values
                  txtSdudentname.Text = sqlread[2].ToString();
                  txtAddress.Text = sqlread[4].ToString();
                  dtpDOB.Text = sqlread[7].ToString();
                  cmbGender.Text = sqlread[6].ToString();
                  txtTelno.Text = sqlread[5].ToString();
                  txtParentsname.Text = sqlread[8].ToString();

               }
            }
            catch (Exception ex)
            {
               //Disply error
               MessageBox.Show("Error! " + ex.Message, "Student Details",
MessageBoxButtons.OK, MessageBoxIcon.Error);
               clear();
            }
         }
         catch (Exception ex)
         {
            //Disply error
            MessageBox.Show("Error! " + ex.Message, "Student Details",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            clear();
         }
         finally
         {
            //close the connection
            sqlconn.Close();
         }
         //Enable buttons if data available
         if (txtSdudentname.Text != "")
         {
            btnDelete.Enabled = true;
            btnUpdate.Enabled = true;
         }
         else
         {
```

```
                btnDelete.Enabled = false;
                btnSave.Enabled = false;
                btnUpdate.Enabled = false;
            }
        }
    }
```

3.1.5.3 Save Button

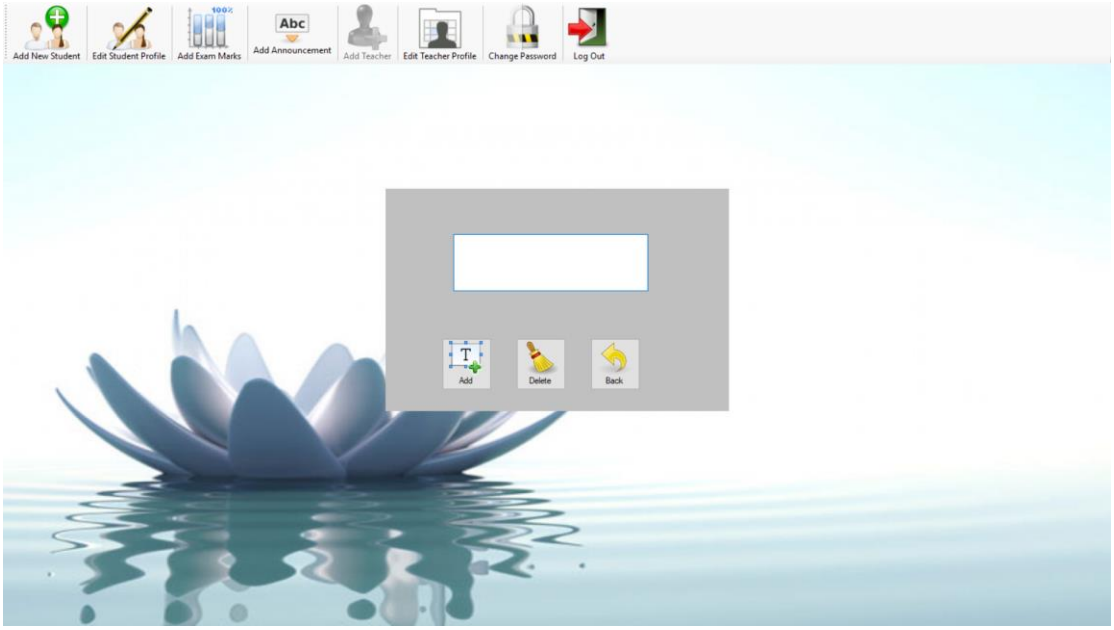
```
private void btnSave_Click(object sender, EventArgs e)
    {
        if ((txtSdudentname.Text == "") || (txtAddress.Text == "") || (cmbGender.Text == "") ||
(txtTelno.Text == "") || (txtParentsname.Text == ""))
        {
            MessageBox.Show("Please fill all details", "Update Details",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        else
        {
            if (!System.Text.RegularExpressions.Regex.IsMatch(txtSdudentname.Text, "^[a-
zA-Z ]+$"))
            {
                MessageBox.Show("Please enter a valid Student Name", "Update Details",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                txtSdudentname.Focus();
                txtSdudentname.SelectAll();
            }
            else
            {

                if ((!System.Text.RegularExpressions.Regex.IsMatch(txtTelno.Text, "^[0-
9]+$")) || (txtTelno.Text.Length <= 9))
                {
                    MessageBox.Show("Please enter a valid Telephone Number", "Update
Details", MessageBoxButtons.OK, MessageBoxIcon.Information);
                    txtTelno.Focus();
                    txtTelno.SelectAll();
                }
                else
                {
                    if (!System.Text.RegularExpressions.Regex.IsMatch(txtParentsname.Text,
"^[a-zA-Z ]+$"))
                    {
                        MessageBox.Show("Please enter a valid Parents' Name", "Update Details",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                        txtParentsname.Focus();
                        txtParentsname.SelectAll();
                    }
                    else
                    {
                        //Display message
                        DialogResult diRes;
                        diRes = MessageBox.Show("Do you want to save your new records?",
"Update details", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
```

```
                    if (diRes == DialogResult.Yes)
                    {
                        //Using error handler tool
                        try
                        {
                            //set the update statement
                            sql = "update TableLogin set Name='" + txtSdudentname.Text +
"',Address='" + txtAddress.Text + "',TelNo='" + txtTelno.Text + "',Gender='" +
cmbGender.Text + "',DOB='" + dtpDOB.Text + "',Parentsname='" + txtParentsname.Text + "'
where Studentid='" + txtStudentid.Text + "'";

                            //open the connection
                            sqlconn.Open();
                            //Assigning the statement in command control
                            sqlcomm = new SqlCommand(sql, sqlconn);
                            //Write the record
                            sqlcomm.ExecuteNonQuery();

                            //Display message
                            MessageBox.Show("Details updated successfully!", "Student details",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                        }

                        catch (Exception ex)
                        {
                            //Disply error
                            MessageBox.Show("Error! " + ex.Message, "Student Details",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                        }
                        finally
                        {

                            //Close the connection
                            sqlconn.Close();
                        }
```

3.1.5.4 Save Button automatic Enable Codes

```
private void txtAddress_TextChanged(object sender, EventArgs e)
      {
         if (txtSdudentname.Enabled)
         {
            btnSave.Enabled = true;
         }
      }
```

3.1.6 Add Announcement Form



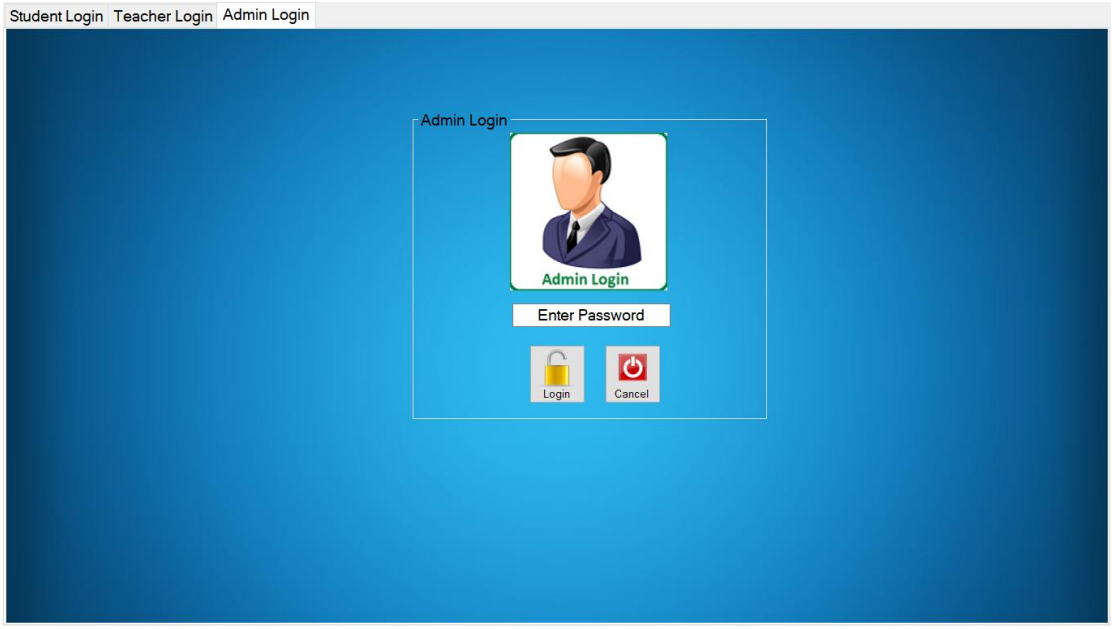
3.1.6.1 Add Button

```
private void btnAdd_Click(object sender, EventArgs e)
      {
         if (txtType.Text == string.Empty)
         {
            MessageBox.Show("Please add message first", "Add Announcement",
MessageBoxButtons.OK, MessageBoxIcon.Information);
         }
         else
         {
            Announcement = txtType.Text;
            MessageBox.Show("Announcement added successfully", "Add Announcement",
MessageBoxButtons.OK, MessageBoxIcon.Information);
         }
      }
```

3.1.6.2 Timer Codes

```
private void timer1_Tick(object sender, EventArgs e)
      {
         lblAnnounce.Left -= 5;
         if (lblAnnounce.Left <= -Width)
         {
            lblAnnounce.Left = Width;
         }
      }
```

3.1.7 Admin Login Form



3.1.7.1 Password Label Hide Codes

```
private void txtpw_TextChanged(object sender, EventArgs e)
      {
         txtpw.UseSystemPasswordChar = true;
      }

      private void txtpw_Click(object sender, EventArgs e)
      {
         txtpw.Text = string.Empty;
      }
```

4. TESTING AND DEPLOYMENT

4.1 Testing

Testing is a process of checking the software performance. The testing phase is debugging the errors encountered after the system functions as a whole. Therefore, software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation.

When doing testing, the focus was given to unit testing, white box testing, black box testing, and system testing respectively.

4.1.1 White Box Testing

In White Box Testing (WBT), the logical pathway of the system is checked to ensure the nitty-gritty of the system is complied with the designed criteria.

The SMS underwent through a WBT and the tester did it with the collaboration of system programmer. The realized issues were discussed at the same time to assure the things were going well.

4.1.2 Unit Testing

In Unit Testing, the entire system is divided into smaller entities, each entity is known as a unit, and each unit is tested at once. The purpose is to validate that each unit of the software performs as designed and expected.

When doing the unit testing with respect to the SMS,each module such as student and teacher was taken into account separately by giving more priority to check;

- whether the text fields accept the respective data types
- filling capability of data into data base
- functionality of the buttons in each form

The system tester did this testing phase individually and identified errors were corrected at the same time.

4.1.3 Black Box Testing

This is the test, which is done without giving priority to the internal logic of the system. The purpose it do a test in a user' point of view to identify any discrepancies with the system.

The incorrect and missing functions, designed interface errors, discrepancies in data structures and database and performance issues were taken into account in relation to the SMS, and the tester discussed it with the team to find out some ways to avoid them.

4.1.4 System Testing

This testing method evaluates the overall system at once. The aim is to review whether the system is complied with the initial requirements and designed scope.

This was the final stage of the testing with respect to SMS and the tester along with other team members paid attention on following:

- connectivity of the all modules in the system
- entire process and data communication within the system
- improvements that could be taken to improve the user-friendliness of the system

Thus, a separate test phase was followed up for this SMS by team tester and got a confirmation that the overall system was complied with the client's requirements and designed scope.

4.2 Test Case

A test case has components that describe an input, action/event, and an expected response, in order to determine if a feature of an application is working correctly. Accordingly, some of test cases, which did with respect to the SMS, are documented as following way

Test case ID	Description	Result Expected	Actual Result	Status
01	Check Student login with valid data	Student should login to the system	As expected	Pass
02	Check user login with invalid data	Display Error message	As expected	Pass
03	Try to enter non-numeric to Student ID	Display Error message	As expected	Pass
04	Enter Invalid names,TelNo, etc. in Edit Student Details Form	Display Error message	As expected	Pass
05	Enter valid details in Edit Student Details Form	Details should be updated	As expected	Pass
06	Try to change password with correct password	Password should be changed	As expected	Pass
07	Try to change password with wrong passwords	Display Error message	As expected	Pass
08	Try to logout from Student Form	A Message should be displayed & user should be logout from the system	As expected	Pass
09	Try to enter more than 4 digits in Teacher id text field	Digits mustn't be displayed	As expected	Pass
10	Check Teacher login with valid data	Teacher should login to the system	As expected	Pass
11	Check Teacher login with invalid data	Display Error message	As expected	Pass
12	Add a student with valid data	The new student details will be added to the system & display a message	As expected	Pass
13	Try to type invalid data in Add new student form	Display error message and data should be cleared	As expected	Pass
14	Add a new announcement	Announcement should be display in Login form	As expected	Pass

15	Delete added announcement	Announcement should be cleared from Login form	As expected	Pass
16	Add valid marks in Add Exam Marks Form	New Exam Marks should be added to the system	As expected	Pass
17	Add invalid marks in Add Exam Marks Form	Display error message	As expected	Pass
18	Update added marks in Add Exam Marks Form	New Exam Marks should be added to the system	As expected	Pass
19	Admin login with wrong password	Display error message	As expected	Pass
20	Admin login with correct password	Admin should login to the system	As expected	Pass
21	Add new teacher with valid data	The new teacher details will be added to the system & display a message	As expected	Pass
22	Add new teacher with invalid data	Display error message & highlight the wrong information	As expected	Pass

4.3 Deployment

In this phase, it gives the attention on releasing the system to the intended user. Accordingly, some computer system specifications and end user manual development were taken into consideration under this stage.

4.3.1 System Requirements

This section explains the hardware and software requirements in relation to this SMS.

4.3.1.1 System Software Tools

- Visual C # to run the system
- Microsoft SQL Server 2014 to function the database

4.3.1.2 System Hardware Tools

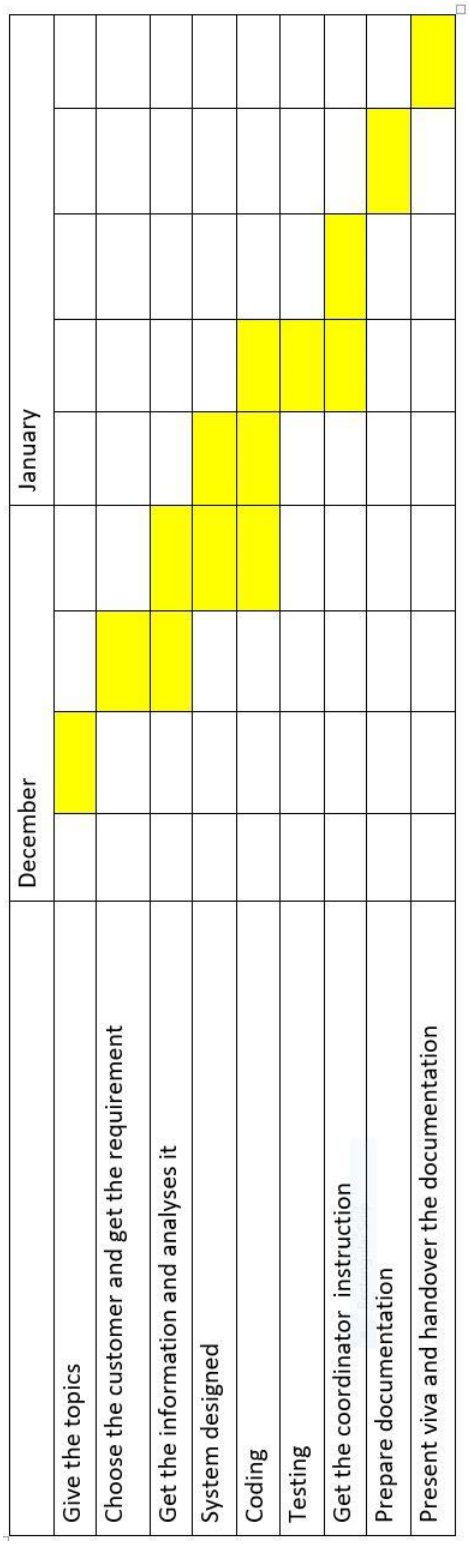
- Microprocessor: Intel (R) Pentium-4 CPU @ 3.0 GHz
- RAM: 1 GB of RAM
- Hard Disk: 1GB on system installation and 10 GB maximum capacity

4.3.1.3 Operating System

- Windows XP, Windows Vista, Windows 7, Window 8/8.1, Windows 10

5. GANTT CHART

The Gantt chart provides a breakdown of the whole system, from beginning to end of the project. It gives a realistic view on how well the project is going according to the plan and if any changes necessary. The below table 5-1, shows the Gantt Chart with respect to SMS.



6. CONCLUSION

SMS is very useful to each education institute irrespective its nature. Because, there is no paper work in this proposed system and anyone could use this without minimum human effort. According to this system, it uses highly user-friendly software like Visual Studio and Visual C # respectively. The system allows the teacher to insert, update, and delete her student's data effectively and efficiently. However, it also provides the decentralized function of search to both the teacher and student. Accordingly, it is viable that this kind of a system is very useful in effective management of the daily functions in the institution.

7. REFERENCES

- Diploma in Information Technology. (2018). In E. M. Campus, Diploma in Information Technology,Part Two,2018 Revised Syllabus.
- Entity Relationship Diagram Symbols | Professional ERD Drawing. (n.d.). Retrieved from www.conceptdraw.com: https://www.conceptdraw.com/How-To-Guide/erd-entityrelationship-diagram-symbols
- ERD Diagrams. (2018, 07 14). Retrieved from www.erdplus.com: https://erdplus.com/#/editdiagram/617249
- Student Management Use Case. (n.d.). Retrieved from www.edrawsoft.com: https://www.edrawsoft.com/template-student-management-usecase.php
- UML Use Case Diagrams. (2018, 07 14). Retrieved from www.uml-diagrams.org: https://www.umldiagrams.org/use-case-diagrams.html
- Waterfall Model - Testing Phase. (2018, 7 14). Retrieved from www.kpsprofession.blogspot.com: http://kpsprofession.blogspot.com/2011/10/waterfall-model-testing-phase.html
- What is Class Diagram? (n.d.). Retrieved from www.visual-paradigm.com: https://www.visualparadigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/
- What is Class Diagram? (2014, 07 14). Retrieved from www.visual-paradigm.com: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-classdiagram/
- What is SDLC Waterfall Model? — Software Testing Help. (n.d.). Retrieved from www.softwaretestinghelp.com: https://www.softwaretestinghelp.com/what-is-sdlcwaterfall-model/