Student Database Software (Documentation)

UBIT | Karachi University

Muhammad shoaib khan

B12101087

2014

Name: Muhammad Shoaib Khan

Seat Number: B12101087

Class: BSCS-2nd Year

Section: A

Shift: Morning

**Documentation of Database Management System Program**

(***S*tudent Database Management Software)**

Dedicated to Miss Shaista Raees

(Assistant Professor in Department of Computer Science

Umair Basha Institute of Technology

Karachi University)

Date: 25-June-2014

**Index**

**S. No Topic Page No.**

1) Short project description 03

2) List of important requirements 03

3) Possible risks in making a database 04

Program

4) System Architecture 04

5) Technologies 04

6) Documentation 05

7) Quality Control and Testing 05

8) Compatibility 05

9) Possibility 05

10) Budget Detail 06

11) Budget Calculation 06

12) Payment terms 07

13) Warranty 08

14) Program Source Code 09

**Technical Part**

1. **Short Project Description:**

Database system is important to ensure the data can be stored, updated and retrieved for future use. Data modelling using the Entity Relationship Model has been introduced more than thirty years. However, designing a good database system is still an attractive issue particularly in system analysis and design because of very hard to do consistency checking between system design and database design. In this paper, a proposal for designing a relational database system based on Object Oriented Analysis and Design is presented. The database system is created by the schema table that extract from class diagram. The rules applying in this paper is following the object oriented concept. It is based on the relationships among the classes, multiplicity, attributes name, class name, data type and the behaviors of the classes. Beside that the user is required to insert record to accomplish a good designing the schema tables to avoid redundancy data. Finally, an automatic editor is proposed in order to automate the process.

***S*tudent Data Management Software (SDMS)** is a comprehensive software solution targeted at Educational Institutions, facilitating the management and analysis of a student's life cycle throughout his / her tenure at the Institution. Starting from getting admitted, to passing out and collecting certificates, the system is capable of generating useful reports at the designated levels.

**Student Data Management Software** enables Educational Institutions to enhance efficiency, eliminate human errors and provide quick access to relevant data.

**Student Data Management Software** uses C#**.Net** for the front end and **SQL Server** 2013 for the backend.

**Student Data Management Software** is the data repository of students, starting from getting admitted, to passing out and collecting certificates; the system is capable of generating useful reports at the designated levels. **Student Data Management Software** manages Attendance, Fees, Examination Scheduler, Marks Entry etc.

1. **List of Important Requirements:**
2. Microsoft Visual Studio 2013
3. Microsoft SQL Server 2013
4. **Possible Risks in making a Database Program:**

|  |  |
| --- | --- |
| Physical | Your computers must be physically inaccessible to unauthorized users. This means that you must keep them in a secure physical environment. |
| Personnel | The people responsible for system administration and data security at your site must be reliable. You may need to perform background checks on DBAs before making hiring decisions. |
| Procedural | The procedures used in the operation of your system must assure reliable data. For example, one person might be responsible for database backups. Her only role is to be sure the database is up and running. Another person might be responsible for generating application reports involving payroll or sales data. His role is to examine the data and verify its integrity. It may be wise to separate out users' functional roles in data management. |
| Technical | Storage, access, manipulation, and transmission of data must be safeguarded by technology that enforces your particular information control policies. |

1. **System Architecture:**

The system will consist of the following major sections:

1. A SQL Server Database Program
2. A C# program for accessing and retrieving the database information
3. **Technologies:**

The system is built by using the following technologies:

1. SQL Server 2012
2. Microsoft Visual Studio 2013
3. **Documentation:**

It will provide:

1. Connected Source Code
2. Technical Documentation

1. **Quality Control and Testing:**

It will implement the following tests:

1. Functional Testing
2. Performance Testing
3. Destructive Testing
4. Usability Testing
5. Security Testing
6. **Compatibility:**

.Net Framework 4.0 required

1. **System Requirements:**

**Minimum System Requirements:**

Intel Pentium 4, 2.66 GHz Processor, 512 MB RAM, 80 GB Hard Disk, Keyboard, Mouse.

**Recommended System Requirements:**

Intel Core I7 Extreme, 2.66 GHz Processor, 6 GB RAM, 1 TB Hard Disk Keyboard Mouse.

1. **Possibility:**

By using this program you can add, delete, edit and update student database management system.

**Financial Part:**

1. **Budget Detail:**

Senior Software Engineer: 30 USD per hour

Software Engineer: 18 USD per hour

Database Administrator: 20 USD per hour

1. **Budget Calculation:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Labor Category** | **Duration, man-hours** | **Cost per Category** | **Total cost per category** |
| Senior Software Engineer | 29 | 30 | 500 |
| Software Engineer | 20 | 18 | 300 |
| Database Administrator | 10 | 20 | 100 |
| Total (USD) |  |  | 900 |

1. **Payment Terms:**
2. Pre-payment : 30% Price 1833 USD
3. After Completion of stage 2: 30% Price 1833 USD
4. After Completion of stage 4: 40% Price 2444 USD

Total Price: 6110 USD

Methods of Payment:

1. Payment via PayPal
2. Payment via Wire Transfer
3. Payment via Check

**14.** **Warranty:**

All errors that are discovered within the first three months of system operation will be fixed at no additional cost.

**Program Source Code:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace Database\_Project4

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

SqlConnection con = new SqlConnection("Data Source=PAKINFO-PC;Initial Catalog=Login;Integrated Security=True;Pooling=False");

SqlDataAdapter sda = new SqlDataAdapter("Select count(\*) From Login where Username='" + textBox1.Text + "' and Password ='"+ textBox2.Text+ "'",con);

DataTable dt = new DataTable();

sda.Fill(dt);

if (dt.Rows[0][0].ToString() == "1")

{

this.Hide();

main ss = new main();

ss.Show();

}

else

{

MessageBox.Show("Please check your Username or Password");

}

}

private void button2\_Click(object sender, EventArgs e)

{

this.Close();

}

private void fontDialog1\_Apply(object sender, EventArgs e)

{

}

private void label7\_Click(object sender, EventArgs e)

{

}

private void groupBox2\_Enter(object sender, EventArgs e)

{

}

private void pictureBox2\_Click(object sender, EventArgs e)

{

}

private void pictureBox3\_Click(object sender, EventArgs e)

{

}

}

}

public partial class main : Form

{

public main()

{

InitializeComponent();

}

private void main\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'database1DataSet.Student' table. You can move, or remove it, as needed.

this.studentTableAdapter.Fill(this.database1DataSet.Student);

timer1.Start();

}

private void studentBindingNavigatorSaveItem\_Click(object sender, EventArgs e)

{

this.Validate();

this.studentBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.database1DataSet);

}

private void label2\_Click(object sender, EventArgs e)

{

}

private void idTextBox\_TextChanged(object sender, EventArgs e)

{

}

private void studentDataGridView\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

this.studentBindingSource.AddNew();

}

private void button2\_Click(object sender, EventArgs e)

{

this.Validate();

this.studentBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.database1DataSet);

}

private void button3\_Click(object sender, EventArgs e)

{

this.studentBindingSource.RemoveCurrent();

}

int count = 0;

private void timer1\_Tick(object sender, EventArgs e)

{

count = studentBindingSource.Count;

label\_status.Text = " There are "+ count.ToString()+" rows in your database ";

if(count<2)

{

Next\_btn.Visible = false;

Previous\_btn.Visible = false;

}

else

{

Next\_btn.Visible = true;

Previous\_btn.Visible = true;

}

}

private void button4\_Click(object sender, EventArgs e)

{

studentBindingSource.MoveNext();

}

private void Previous\_btn\_Click(object sender, EventArgs e)

{

studentBindingSource.MovePrevious();

}

private void searchNameToolStripButton\_Click(object sender, EventArgs e)

{

try

{

this.studentTableAdapter.SearchName(this.database1DataSet.Student, nameToolStripTextBox.Text);

}

catch (System.Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

}

}

private void search\_SeatNumberToolStripButton\_Click(object sender, EventArgs e)

{

try

{

this.studentTableAdapter.Search\_SeatNumber(this.database1DataSet.Student, seatNumberToolStripTextBox.Text);

}

catch (System.Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

}

}

private void search\_CLassToolStripButton\_Click(object sender, EventArgs e)

{

try

{

this.studentTableAdapter.Search\_CLass(this.database1DataSet.Student, classToolStripTextBox.Text);

}

catch (System.Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

}

}

private void main\_FormClosing(object sender, FormClosingEventArgs e)

{

DialogResult dialog = MessageBox.Show("Do you really want to close the program?", "Exit", MessageBoxButtons.YesNo);

if(dialog==DialogResult.Yes)

{

e.Cancel = false;

}

else if(dialog==DialogResult.No)

{

e.Cancel = true;

}}}}