# Introduction

C# is a multi-paradigm programming language encompassing strong typing, imperative declarative, functional, generic, object-oriented and component-oriented programming disciplines. It was developed by Microsoft within its .NET initiative. It is intended to combine the computing power of C++ with the programming ease of Visual Basic. C# is designed to work with Microsoft’s .NET platform. Microsoft’s aim is to facilitate the exchange of information and services over the Web, and to enable developers to build highly portable application.

**My app for Stock Keeping** is one of the very simple applications that is designed and developed using C# platform. This application is developed for keeping the record of the stock which can be useful for future purposes. It is designed for those small companies whose volume of transaction of goods is very less. Similarly, for those companies who have very less number of debtors as well as creditors. Similarly, for those companies whose amount of purchase and sell are also less.

The main use of this application is to save the amount of received by sales and expensed by purchase by a company. Similarly, the user of this application is benefited by many things. Some of the basic things that the user can be benefitted by this application are mentioned below:

1. More Secure as our application has login function which will hide and make secure all the transaction that occur in the company.
2. Fast to check their total capital with a single click.
3. Users can check the total amount of debtors and creditors.
4. Users can check their drawings amount as well.

# Project Overview

My app for Stock Keeping is the application developed under C# platform. This application is created for those small companies who have all their transaction records in paper. It helps to reduce the paper record into electronic records. The records are fast and accurate. Similarly, this application helps to make the records safe and secure hence other cannot access to it and steal the records.

Though, the app is useful it has some limitation too. Following are some of the limitation that our program has:

1. There can be as many as same usernames and password for the application which may bring a conflict.
2. The account of debtors and creditors are not separate and the whole account is calculated in one account i.e. Debtors and creditors.
3. The Balance goes to negative also.
4. Email confirmation is not available.
5. Login is not as secure as other highly secured banking application of different institution.

# Methodology

For the development of application, different methods can be applied. For my application, I used following methods:

1. Analysis of the need of application.
2. Designing of Flowchart.
3. Designing E-R Diagram.
4. Coding and connecting Database.
5. Implementing.
6. Solving errors.

# Flowchart

Register

Fill fields

Forget password

Delete

Login Screen

Success

Click login

Enter ID and Password

Exit

Main Page

Update

Fill fields

Check balance

Back

Log out

Exit

Fill fields

Success

Click on Cash, Credit or Bank

Success

Purchase

Sales

Drawings

Load

Update message

# Main Function Code

namespace FinalProject

{

public partial class Form1 : MetroForm

{

Login l1 = new Login();

public Form1()

{

InitializeComponent();

}

public void values(String a)

{

label1.Text = a;

}

private void metroTile4\_Click(object sender, EventArgs e)

{

Loads l1 = new Loads();

l1.values(label1.Text);

l1.Show();

this.Hide();

}

private void metroButton1\_Click(object sender, EventArgs e)

{

try

{

SqlConnection con = new SqlConnection("Data Source=SAYAMI;Initial Catalog=bin;User ID=SA;Password=12345");

con.Open();//Making connection open

string query = "Select \* from Final where [UserName]='"+label1.Text+"'";//selecting the values from database Final where username is from label1.text

SqlCommand cmd = new SqlCommand(query, con);//making the query

SqlDataReader reader = cmd.ExecuteReader();//reading data from database

reader.Read();//reading function

Cash.Text = reader["Bank"].ToString();//reading cash from database

Bank.Text = reader["Cash"].ToString();//reading bank from database

reader.Close();//reader query close

con.Close();//conection close

double a = double.Parse(Cash.Text.ToString());//assing the value to a

double b = double.Parse(Bank.Text.ToString());//assigning the values to b

double cap = a + b;//cap = adding a and b that is bank and cash

query = "update Final set Capital='" + cap + "' where [UserName]='"+label1.Text+"'";//assigning the value to cap into Capital of database

con.Open();//making connection open

cmd = new SqlCommand(query, con);//making query

cmd.ExecuteNonQuery();//executing non query commannd

con.Close();//making ocnnection cloase

SqlConnection connection = new SqlConnection("Data Source=SAYAMI;Initial Catalog=bin;User ID=SA;Password=12345");

SqlDataAdapter dataadapater = new SqlDataAdapter("Select [CompanyName],[FirstName],[Capital] from Final where [UserName]='"+label1.Text+"'", connection);//using dataadapter for data tabale

DataSet ds = new DataSet();//dataset for data in data view or let say table

connection.Open();//making the conneciton open

dataadapater.Fill(ds, "f");//data adapter filled

connection.Close();//making connection cloase

dataGridView1.DataSource = ds;//data in datagrid view

dataGridView1.DataMember = "f";//data member

dataGridView1.ReadOnly = true;//read only

}

catch (Exception err)

{

MessageBox.Show("Error 404: " + err.Message, "Error");

}

}

private void exitToolStripMenuItem2\_Click(object sender, EventArgs e)

{

DialogResult r1 = MessageBox.Show("Do you want to Exit", "Exit", MessageBoxButtons.YesNo);

if (r1 == DialogResult.Yes)

{

Application.Exit();

}

}

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

{

Loads l1 = new Loads();

l1.values(label1.Text);

l1.Show();

this.Hide();

}

private void metroTile1\_Click(object sender, EventArgs e)

{

Purchase p1 = new Purchase();

p1.values(label1.Text);

p1.Show();

this.Hide();

}

private void metroTile2\_Click(object sender, EventArgs e)

{

Sales s1 = new Sales();

s1.values(label1.Text);

s1.Show();

this.Hide();

}

private void metroTile3\_Click(object sender, EventArgs e)

{

drawings d1 = new drawings();

d1.values(label1.Text);

d1.Show();

this.Hide();

}

private void metroButton2\_Click(object sender, EventArgs e)

{

try//the same process as of capital in database for debtors/account recivable

{

SqlConnection connection = new SqlConnection("Data Source=SAYAMI;Initial Catalog=bin;User ID=SA;Password=12345");

SqlDataAdapter dataadapater = new SqlDataAdapter("Select [CompanyName],[FirstName],[Debitors] from Final where [UserName]='" + label1.Text + "'", connection);

DataSet ds = new DataSet();

connection.Open();

dataadapater.Fill(ds, "f");

connection.Close();

dataGridView1.DataSource = ds;

dataGridView1.DataMember = "f";

dataGridView1.ReadOnly = true;

}

catch (Exception err)

{

MessageBox.Show("Error 404: " + err.Message,"Error");

}

}

private void metroButton3\_Click(object sender, EventArgs e)

{

try//the same process as of capital in database for Creditors/ account payable

{

SqlConnection connection = new SqlConnection("Data Source=SAYAMI;Initial Catalog=bin;User ID=SA;Password=12345");

SqlDataAdapter dataadapater = new SqlDataAdapter("Select [CompanyName],[FirstName],[Creditors] from Final where [UserName]='" + label1.Text + "'", connection);

DataSet ds = new DataSet();

connection.Open();

dataadapater.Fill(ds, "f");

connection.Close();

dataGridView1.DataSource = ds;

dataGridView1.DataMember = "f";

dataGridView1.ReadOnly = true;

}

catch (Exception err)

{

MessageBox.Show("Error 404: " + err.Message,"Error");

}

}

private void metroTile6\_Click(object sender, EventArgs e)

{

}

private void metroTile5\_Click(object sender, EventArgs e)

{

Updats s1 = new Updats();

s1.values(label1.Text);

s1.Show();

this.Hide();

}

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

{

this.Hide();

Login l1 = new Login();

l1.Show();

}

private void exitToolStripMenuItem1\_Click(object sender, EventArgs e)

{

Application.Exit();//exit application

}

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

{

Updats s1 = new Updats();

s1.values(label1.Text);

s1.Show();

this.Hide();

}

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

{

Register r1 = new Register();

r1.Show();

this.Hide();

}

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

{

Application.Exit();

}

private void notifyIcon1\_MouseDoubleClick(object sender, MouseEventArgs e)

{

MessageBox.Show("My App for Stock Keeping Designed and Developed by Bibesh Manandhar","Bibesh Manandhar");

}

}

}

# ScreenShots

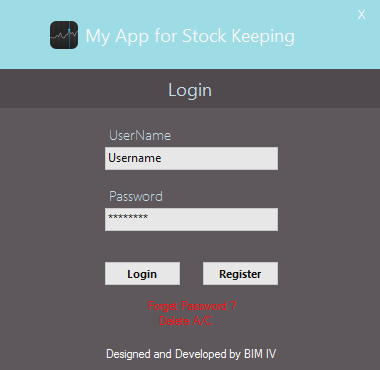


Fig: Login Form

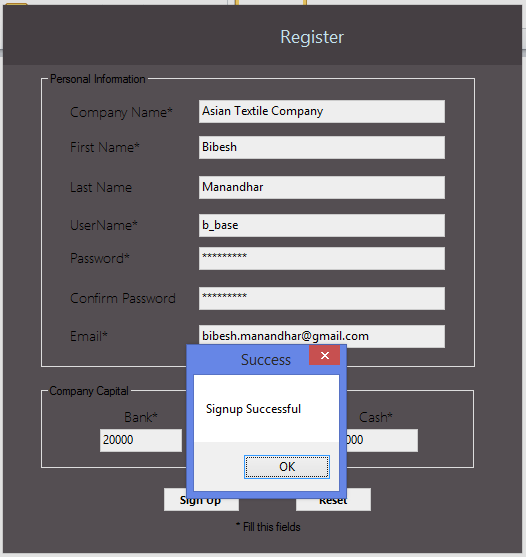


Fig: Register form and successfully registering.

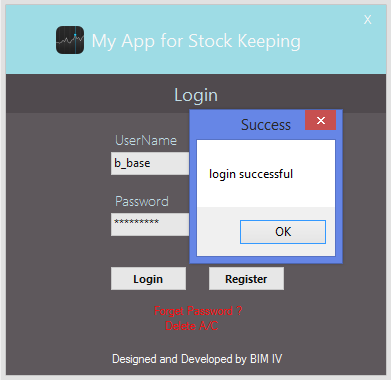


Fig: Login form with Successful login

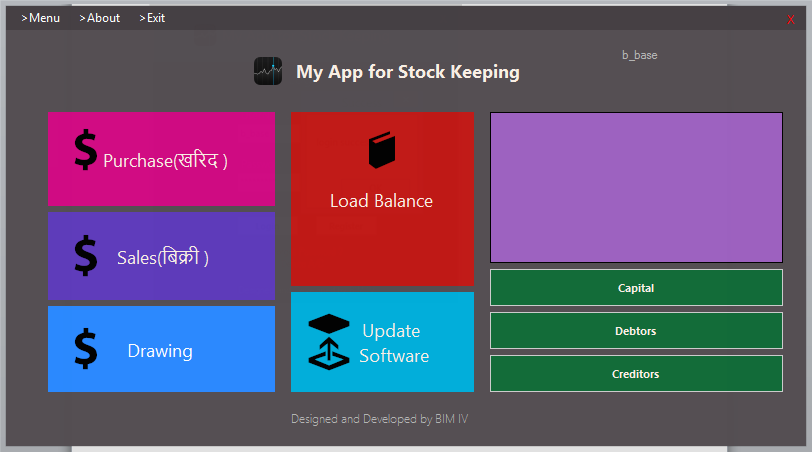


Fig: Main form of the application.

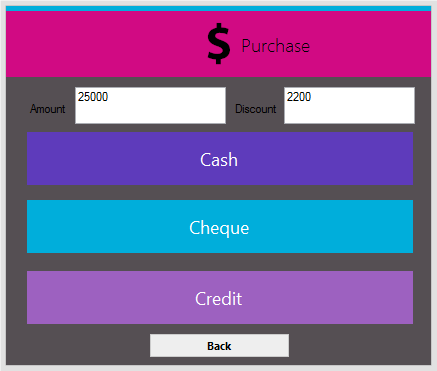


Fig: Purchase of goods using different methods.

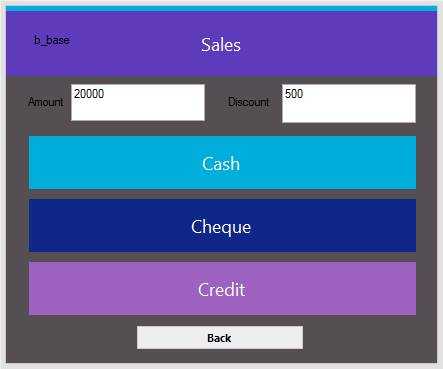


Fig: Sales of goods using different methods.

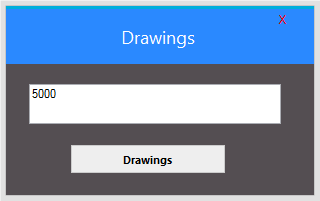


Fig: Drawing amount to be entered

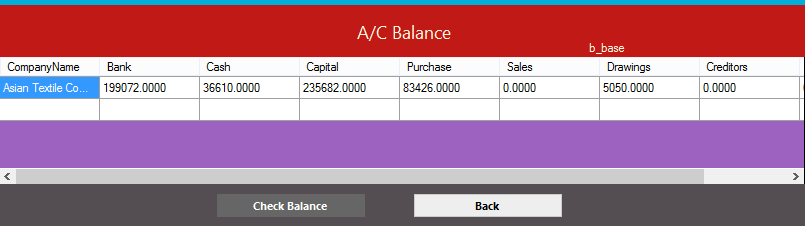


Fig: Total balance of the user.

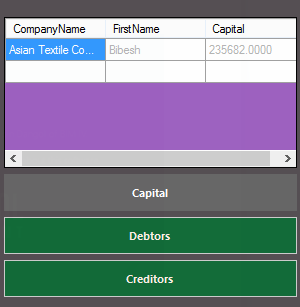


Fig: Brief capital amount of the user

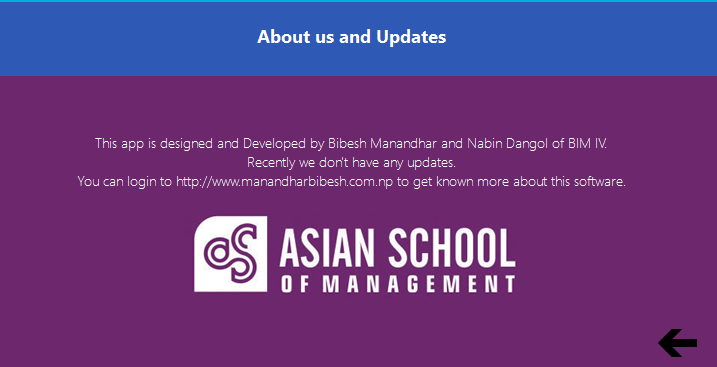


Fig: Update information and about the developer

# Conclusion

Last but not the least, we would like to conclude our project report with lots of thanks to our teacher cum helper, Mr. Sudan Prajapati, for giving me such a great opportunity with fully confident. After this project we had learned a lot of things about the software develop.

After a long struggle finally we had completed our project. We had tried to give up our best for the successive completion of our project. We had tried to make my project unique, easy understanding, simple, and effective. For the project we had gone through many websites, books, and also consulted with other students projects and I had gone through with our own logic also. We had tried to present as perfect as we can. We had visited many companies and many co-operative limited and finally I choose this **My App for Stock Keeping** for my project.

For the terms we had tried to present our own views and ideas. We had given all the power to make our project “**The** **Best**”. We had tried to use perfect new technologies so that the company may not have to bear problems. Our project is efficient and economic.

# Reference

1. <http://searchwindevelopment.techtarget.com/definition/C>
2. <https://en.wikipedia.org/wiki/C_Sharp_(programming_language)>
3. <https://en.wikipedia.org/wiki/Systems_development_life_cycle>
4. <https://github.com/viperneo/winforms-modernui>
5. <https://metroui.org.ua/>

# Bibliography

1. <https://www.youtube.com/watch?v=mM4fLnRDVto>
2. <https://www.youtube.com/watch?v=aoFDyt8oG0k>
3. <https://www.youtube.com/watch?v=ujMVHPRLGxc> ( ADO.NET)