

# **SAVINGS BANK ACCOUNT**

## **A PROJECT REPORT**

*Submitted by*

Akansha Kumari (18BCA001A)

*Submitted in partial fulfillment of the requirements for the  
award of the degree of*

**Bachelor of Computer Application**



**DEPARTMENT OF COMPUTER SCIENCE  
Jyoti Nivas College, Autonomous  
Bengaluru - 5600095**

## **ACKNOWLEDGEMENT**

With a deep sense of gratitude, we acknowledge all those who have contributed significantly towards the successful completion of this project.

We would like to express our gratitude to **Dr. Sr. Elizabeth C.S** **principal Jyoti Nivas college** for the giving us the opportunity to complete our project successfully by providing us with endless enthusiastic support in improving our creative skills and for providing us the excellent facilities and constant encouragement throughout our project.

We owe profound thanks to **Mrs. Neetha Georphin** the head of the **Department of computer science**, who took keen interest on our project work and guided us and encouraged at every step all along till the completion of the project work by providing all needed information for developing a good system.

We express our heartfelt gratitude to our project guides, **Mrs. Ruby Peter** for their cordial and considerate attitude with all valuable suggestions, guidance support and cooperation right from the inception of the project, which inspired us in carrying out this project successfully within the short period of time.

Also we would like to extend our sincere regards to our lab staffs **Mr. Basil Aleyas** and **Mr.Praveen Marcarenhas** and support staff **Mr. Anand** for their helpful guidance and timely support.

Finally, we are grateful to our parents and friends who have helped us to complete this project successfully.

## CONTENTS

S.No	CONTENT	Page No.
1.	<b>SYNOPSIS</b>	6-33
2.	<b>SYSTEM SPECIFICATION</b> a) Hardware Specification b) Software Specification	
3.	<b>SYSTEM ANALYSIS</b> a) Existing System b) Proposed System c) Module Description d) Data Flow Diagram e)E-R diagrams	
4.	<b>SYSTEM DESIGN</b> a )Database Design b) User Interface Design c) Code Design	
5.	<b>SYSTEM TESTING</b>	34
6.	<b>FUTURE ENHANCEMENTS</b>	36
7.	<b>REFERENCES</b>	37

# **SAVINGS BANK ACCOUNT**

## **SYNOPSIS:**

Savings bank account is used to deposit money for the future use. This project will show the following things: admin login, modules into admin account i.e registering for the customer, removing a customer, transferring money, withdrawal, and deposit fund into their savings account.

## **SYSTEM SPECIFICATION**

### **Hardware specification**

- Standard PC
- Internet connect on with good enough speed
- 128MB or more RAM (256 recommended)
- Atleast 500MB hard disk space
- Smart mobile phone Software Requirements

### **Software specification**

- Frontend: Visual Studio
- Backend: SQL
- Application: Microsoft visual studio 2019. Microsoft SQL Server
- Web Browser: Microsoft Internet Explorer, Google chrome

## **SYSTEM ANALYSIS:**

### **Existing System:**

The existing savings bank account system has a manual way of handling the money transactions and procedures. The bank employees will do the money related calculations manually, which can lead to miscalculations sometimes. Also, it can be a little difficult to maintain a report of the customer's transactions in an organized way. Hence, the existing system in savings bank account is manually done.

### **Proposed System:**

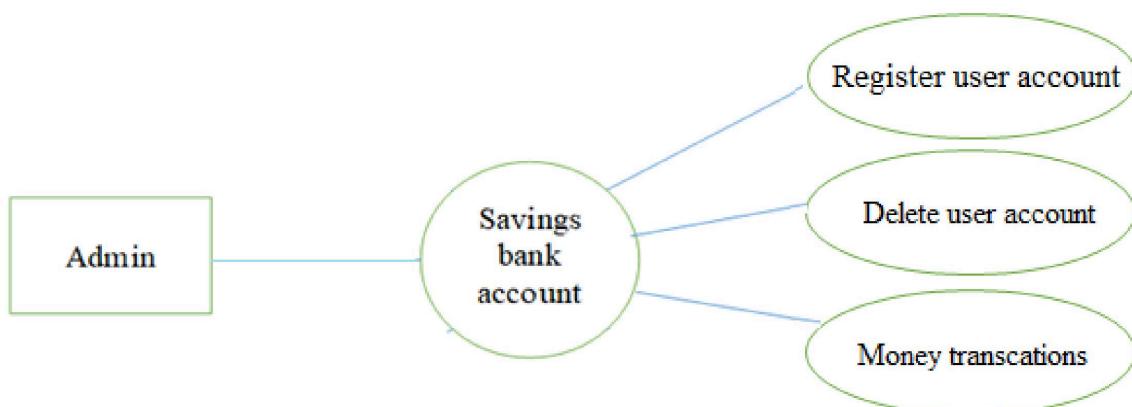
The savings bank system we have given in our project will provide an easier way to solve the difficulties in manual functions of the existing system. This system will let the bank employees to make money transactions easily. It will also enable them to keep a report of it in an organized way. The money transactions will have lesser chances of being miscalculated as the computer will perform the transactions.

## Modules Description:

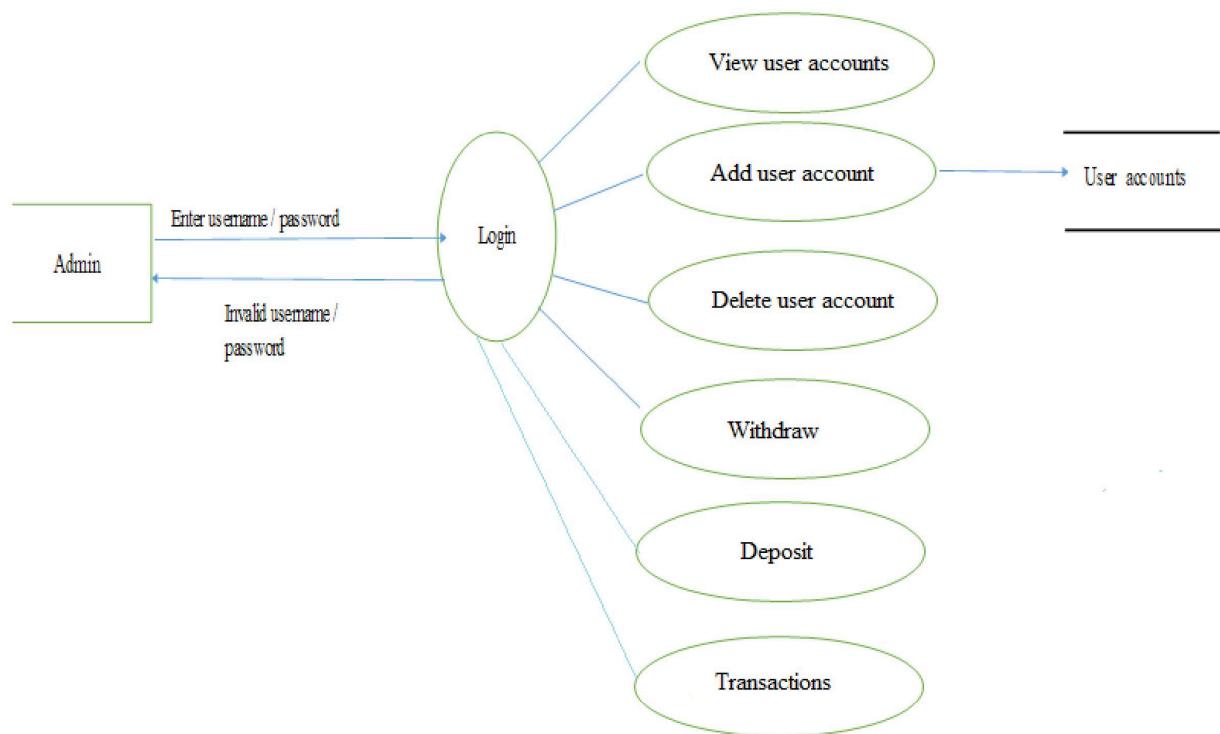
- Login : This will contain the login box for admin to login to their accounts. Admins can login to their account using admin login box.
- Modules into admin account:
  1. Add user accounts: This allows the admin to register user's account.
  2. Delete user accounts: This will allow the admin to delete the user accounts.
  3. Transfer money: It lets the admin to transfer fund from one user account to another user account.
  4. Withdrawal: Here the user can withdraw money for their expenses.
  5. Deposit: The user can deposit fund into their savings account by transferring money from another account.

## Data Flow Diagram

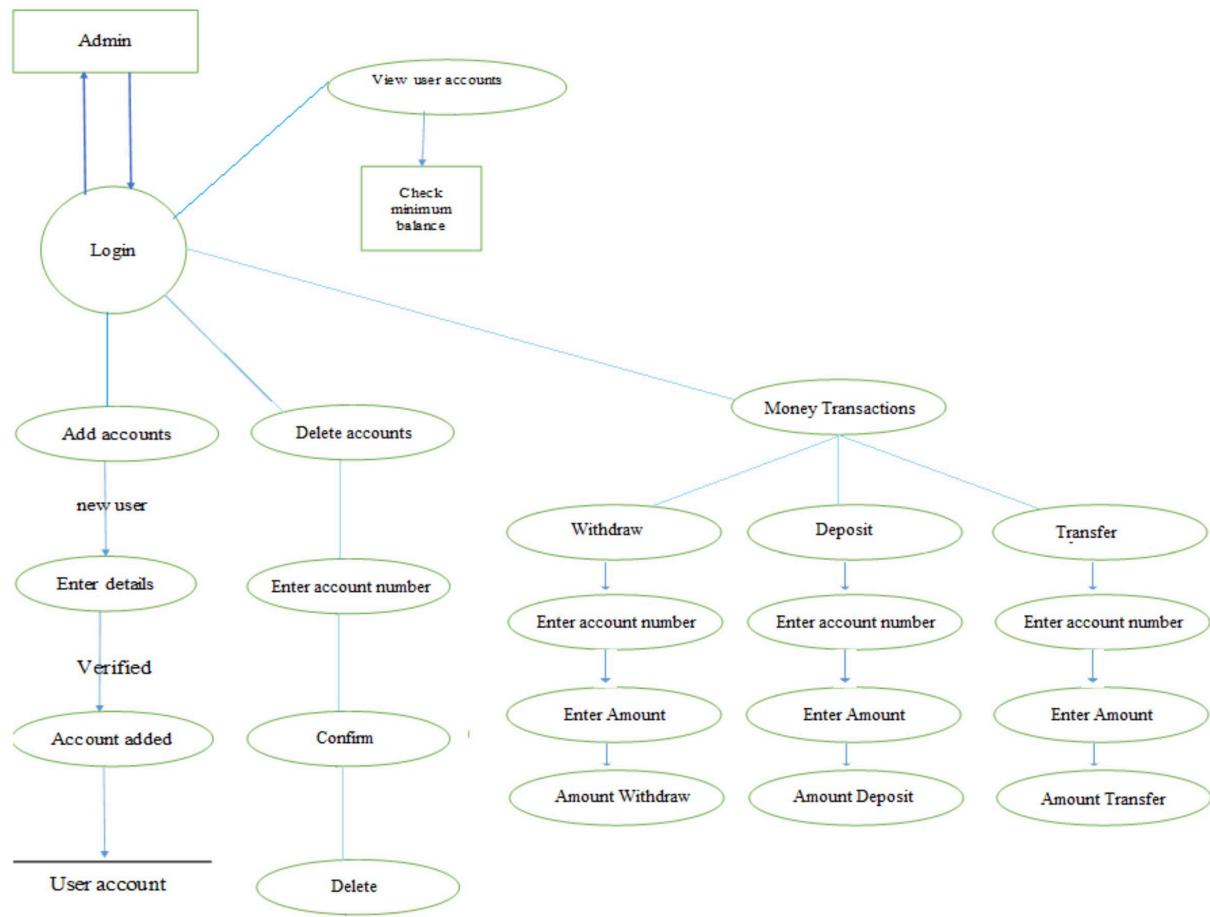
### Level 0



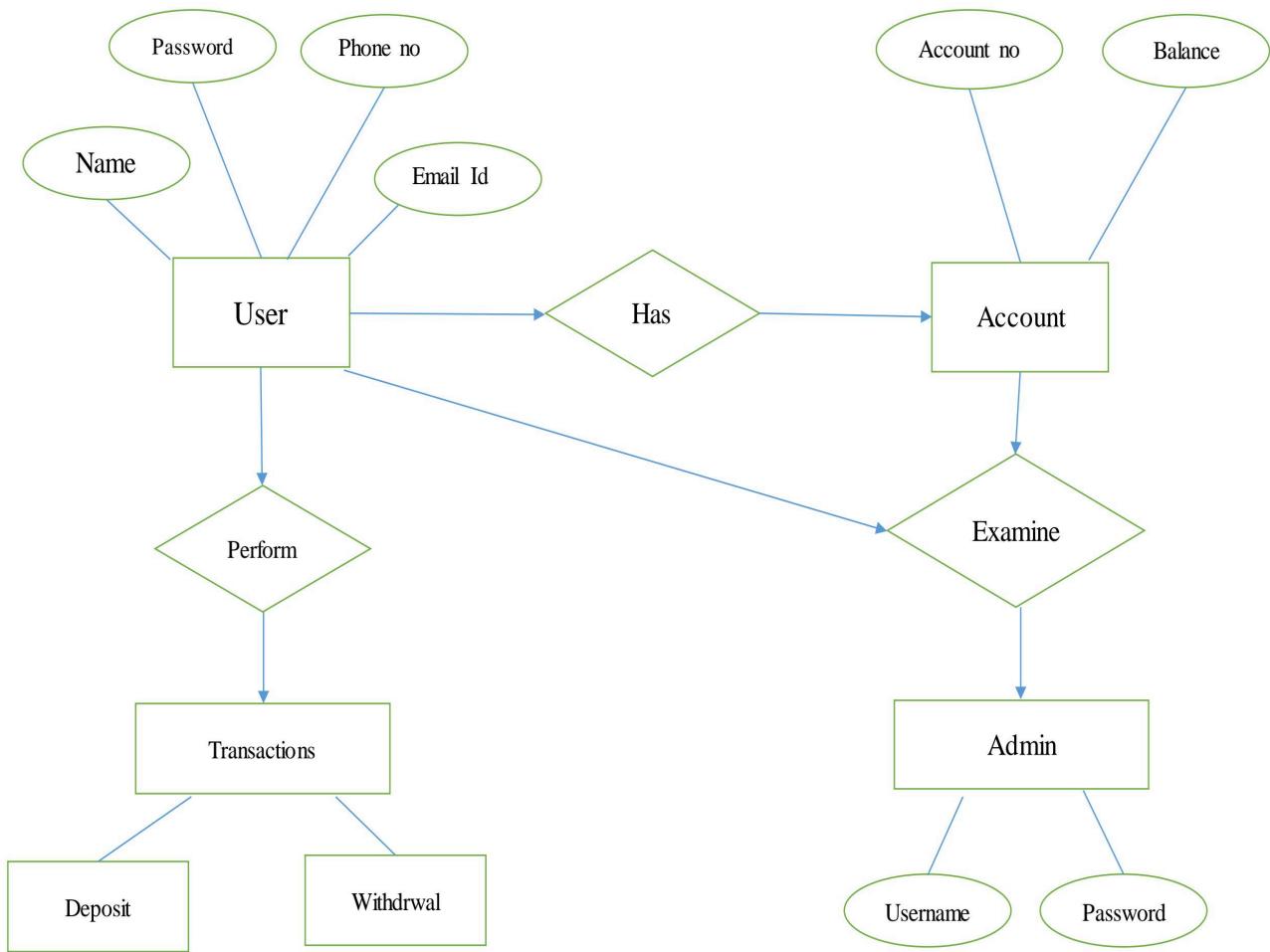
## Level 1



## Level 2



## ER DIAGRAM



## SYSTEM DESIGN

### DATABASE DESIGN

#### Login

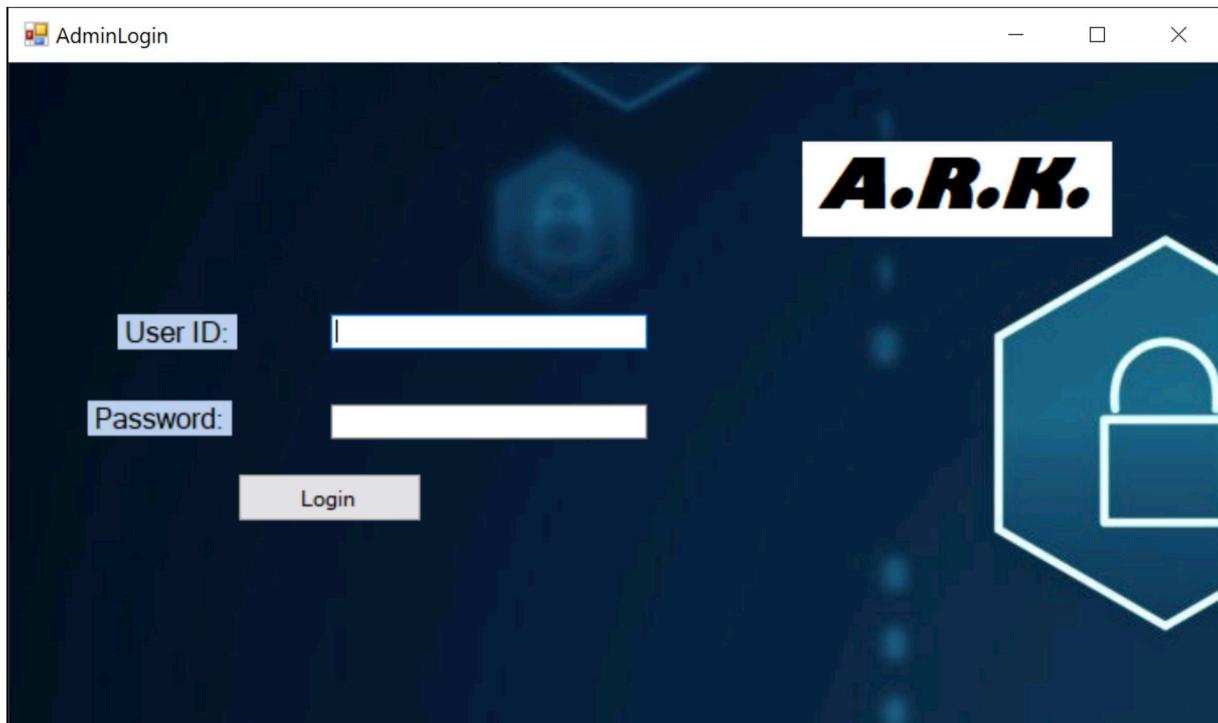
Field Name	Type	Description
username	Varchar(30)	Stores the username
password	Varchar(10)	Stores the password

#### Register

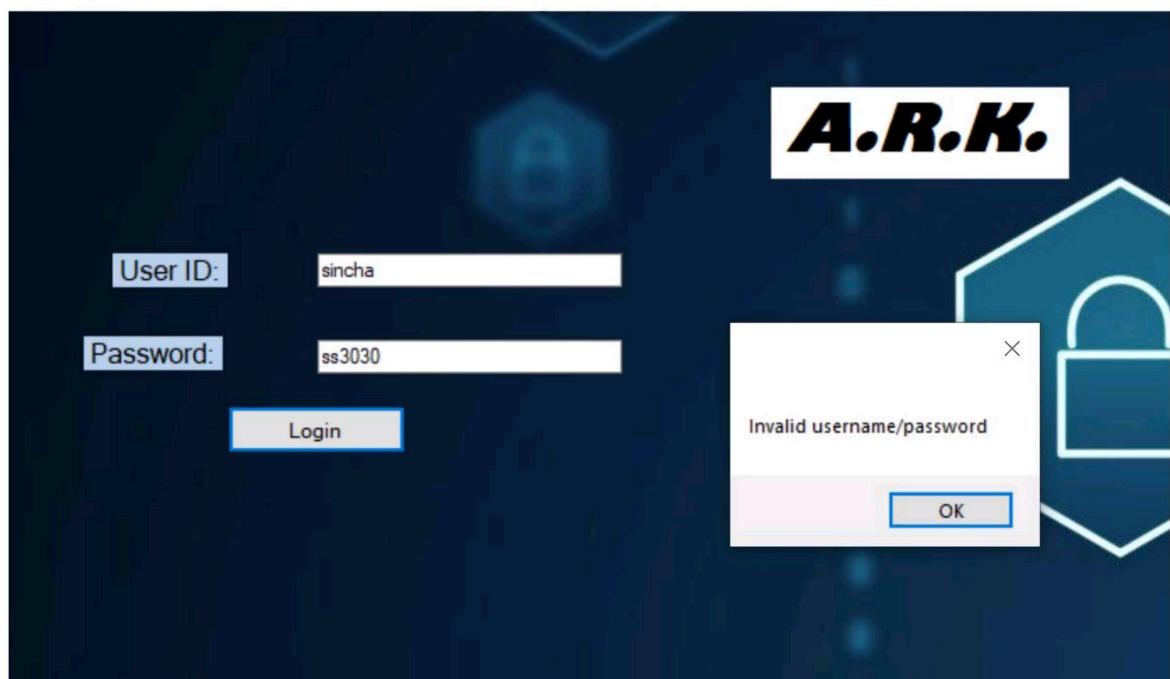
Field Name	Type	Description
name	Varchar(30)	Stores the name
dob	Varchar(10)	Stores the date of birth
gender	Varchar(1)	Stores the gender
age	Numeric(18,0)	Stores the age
balance	Numeric(18,0)	Stores the balance
accountno	Varchar(15)	Stores the account number
email	Varchar(40)	Stores the email id
phoneno	Numeric(18,0)	Stores the phone number
aadharno	Numeric(18,0)	Stores the aadhar number

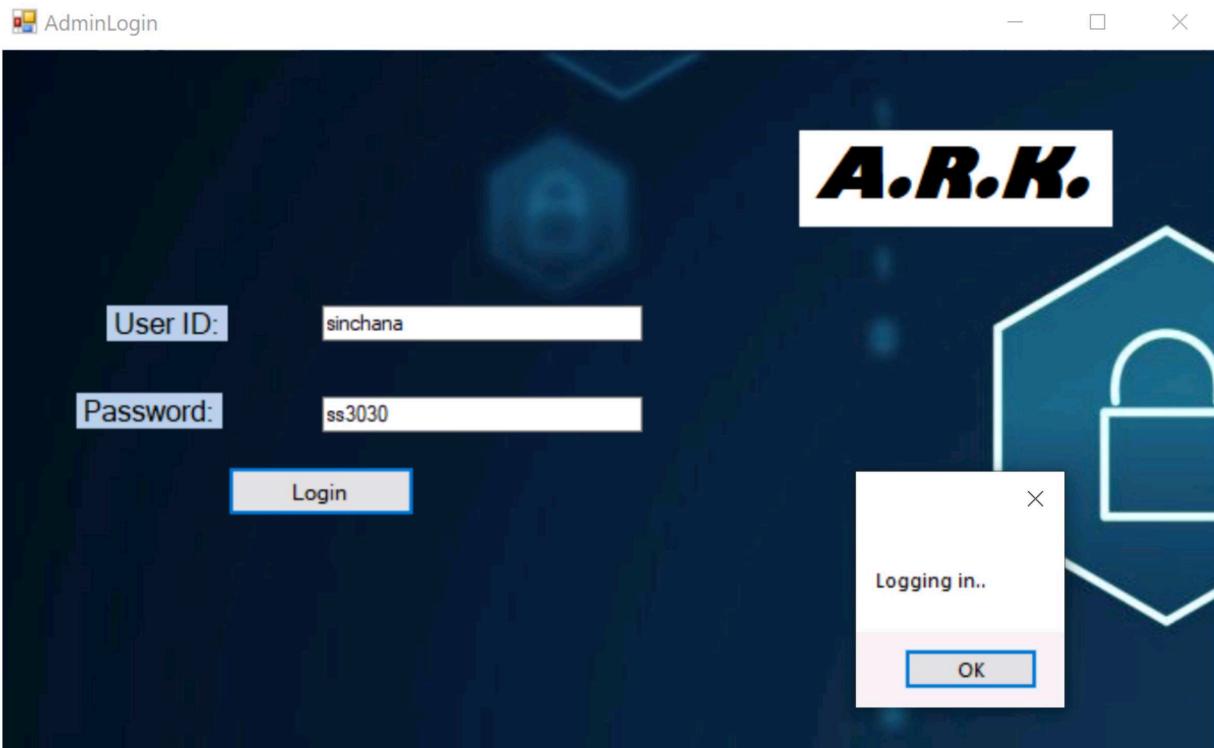
## USER INTERFACE DESIGN

AdminLogin

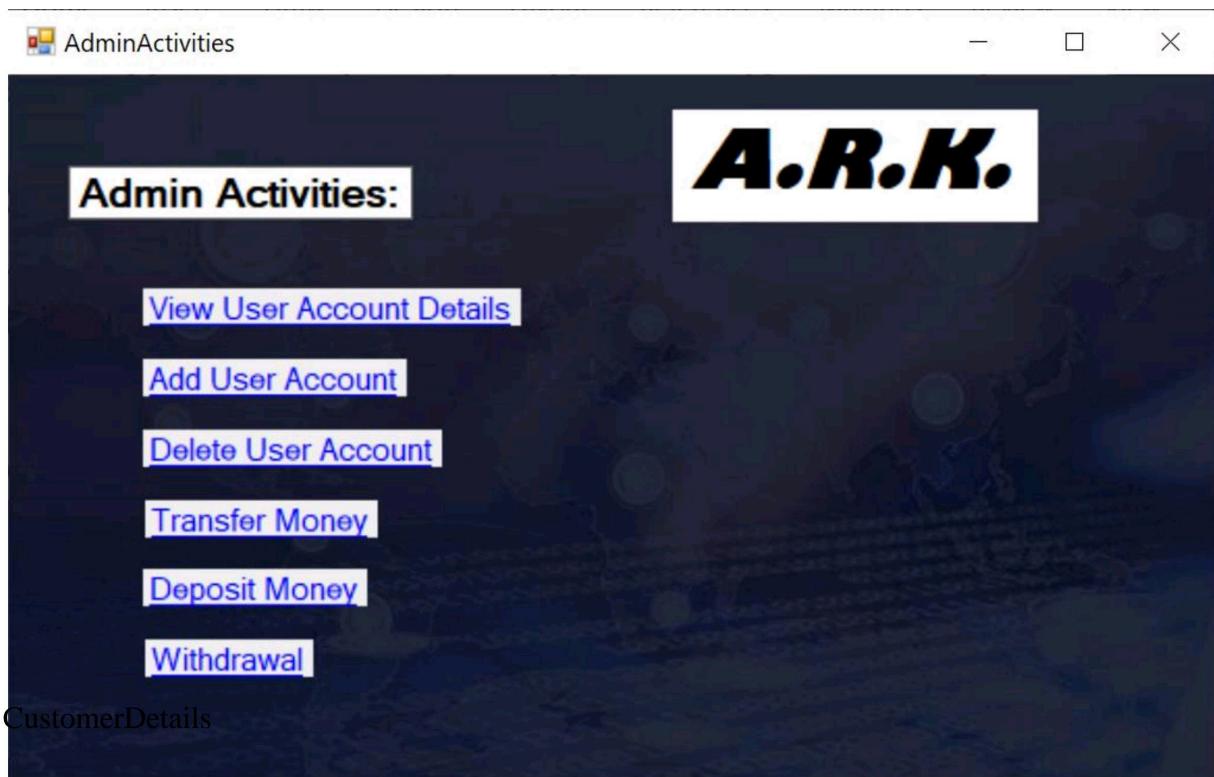


AdminLogin





## AdminActivities



## CustomerDetails

CustomerDetails

	name	dob	gender	age	balance	accountno
▶	Abi	03/05/1992	M	29	7450	A1234567
	aarthi	10/11/1990	F	30	8377	AC231245
	Sujitha	24/03/1975	F	45	10500	AK240324
	Suriya	10/08/1989	M	31	7523	AK474747
	Ravi	14/11/1982	M	38	27500	AP231455
*	Sakthi	30/08/1979	F	41	54600	AS344563

< >

Click here to go back

## AddCustomer

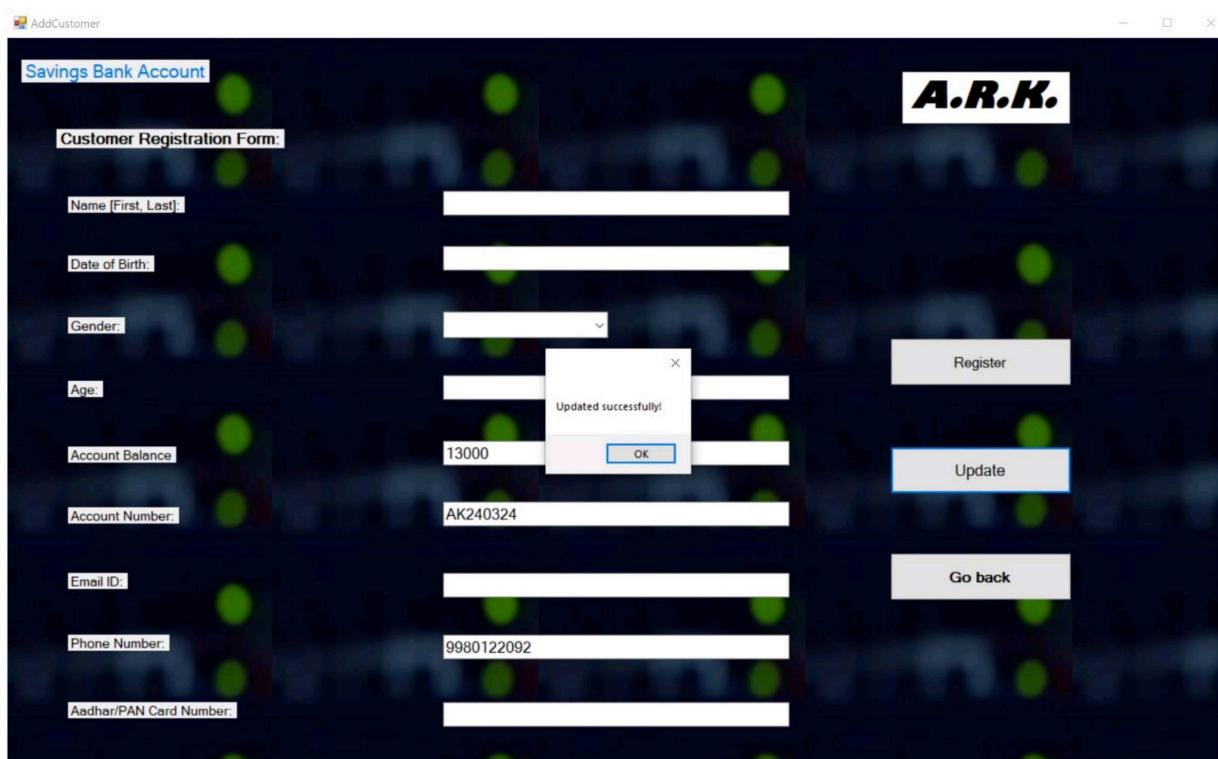
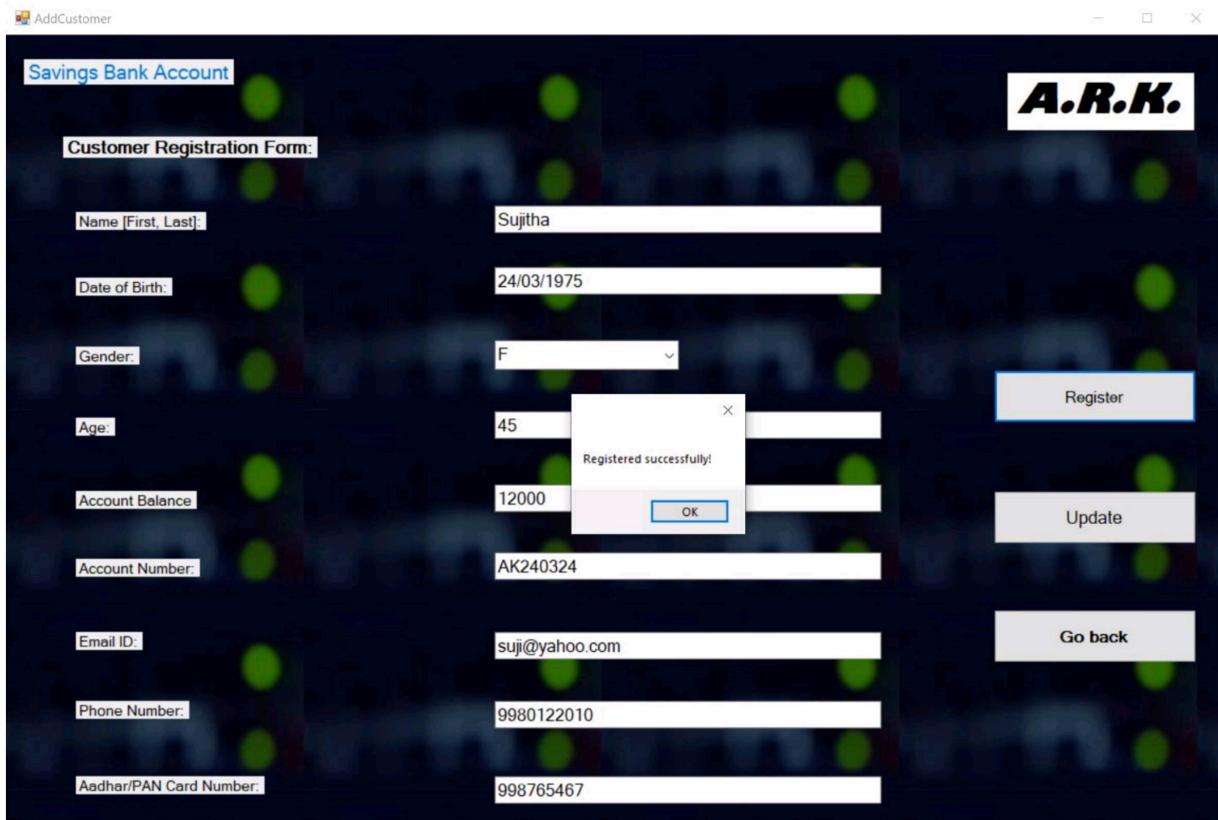
AddCustomer

Savings Bank Account

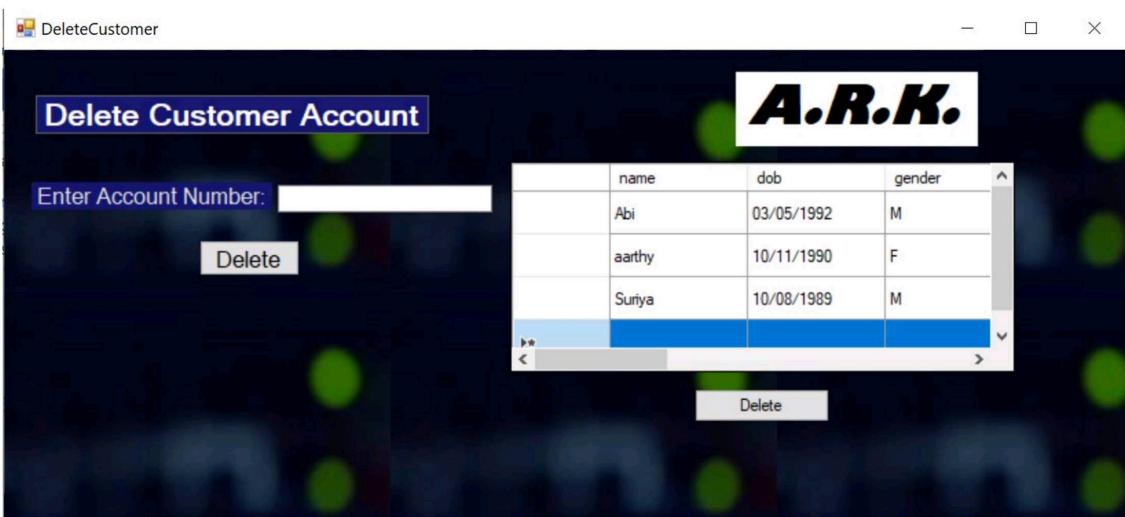
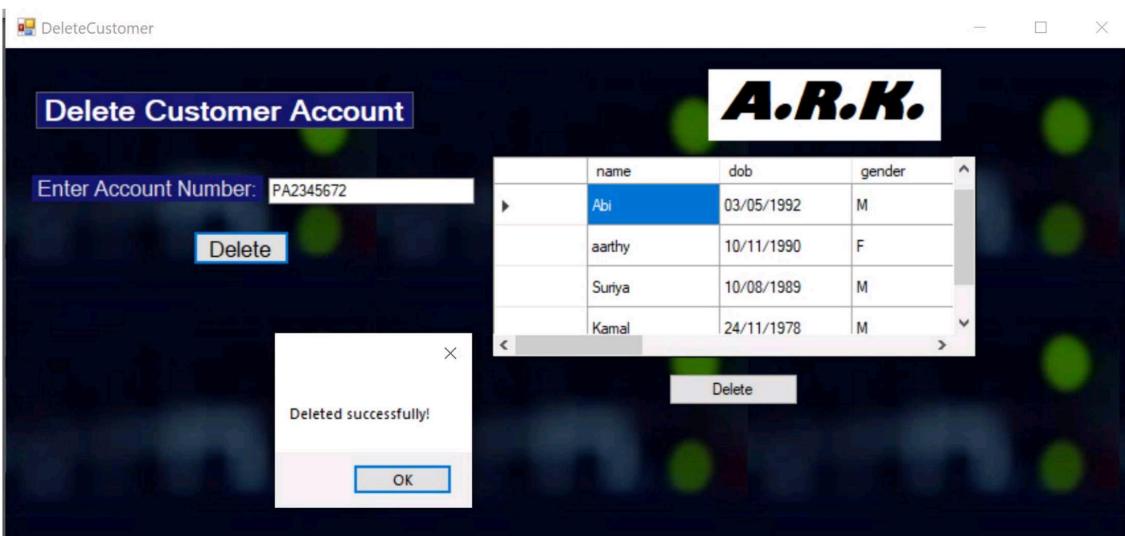
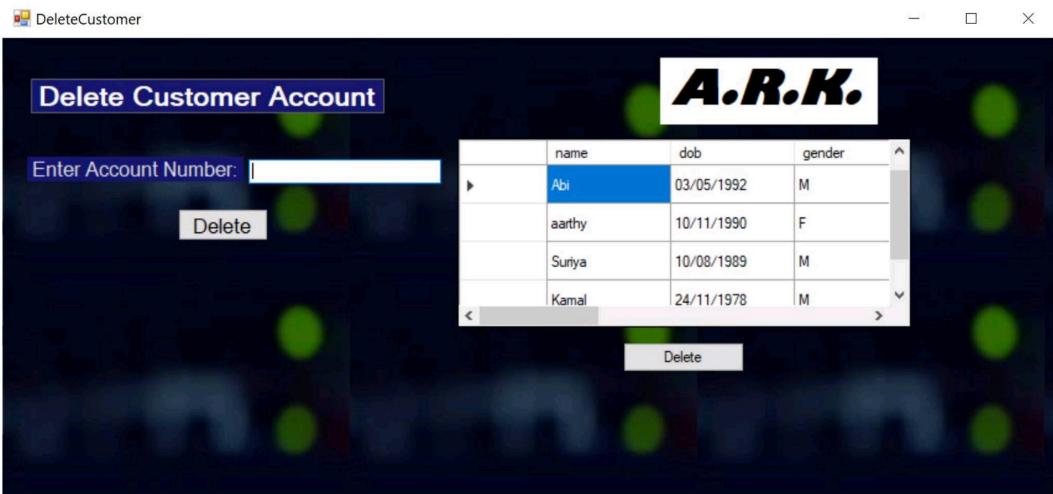
**A.R.K.**

**Customer Registration Form:**

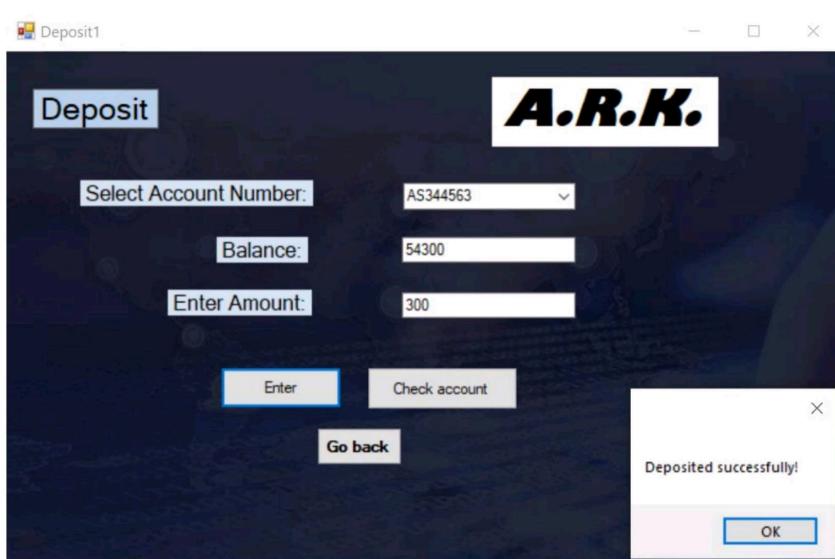
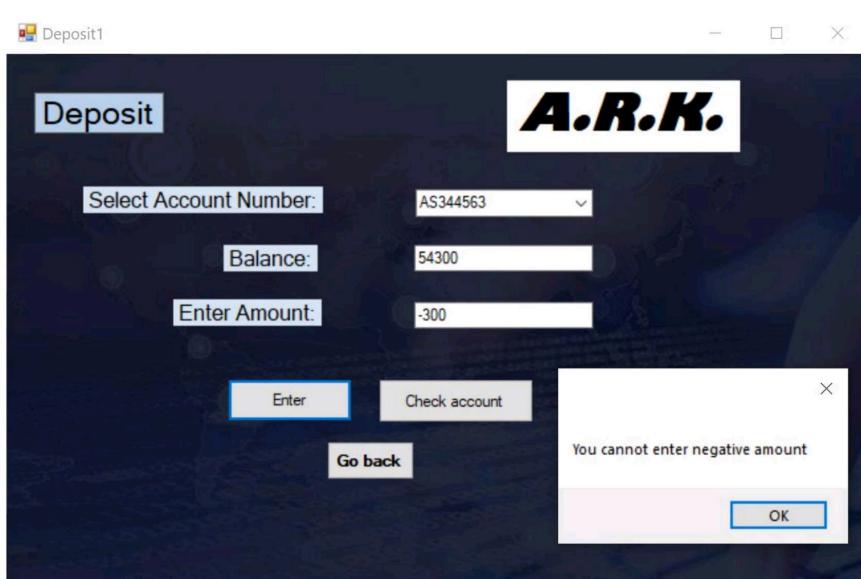
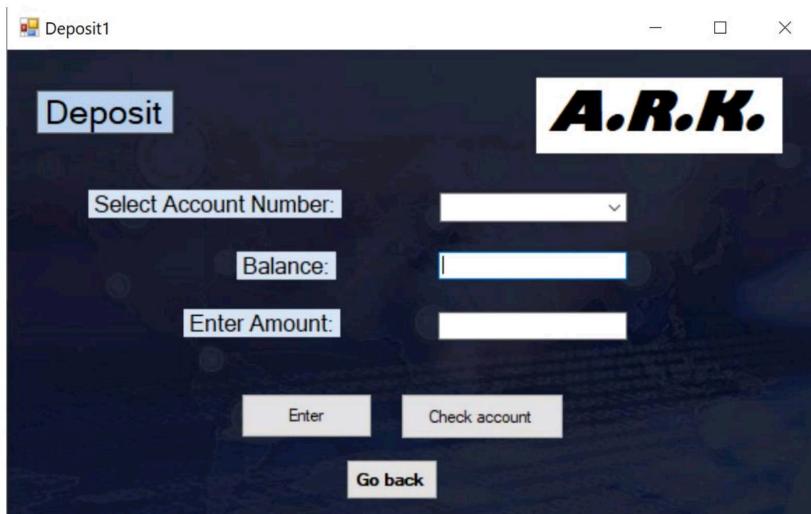
Name [First, Last]:	<input type="text"/>	Register
Date of Birth:	<input type="text"/>	Update
Gender:	<input type="text"/>	Go back
Age:	<input type="text"/>	
Account Balance:	<input type="text"/>	
Account Number:	<input type="text"/>	
Email ID:	<input type="text"/>	
Phone Number:	<input type="text"/>	
Aadhar/PAN Card Number:	<input type="text"/>	



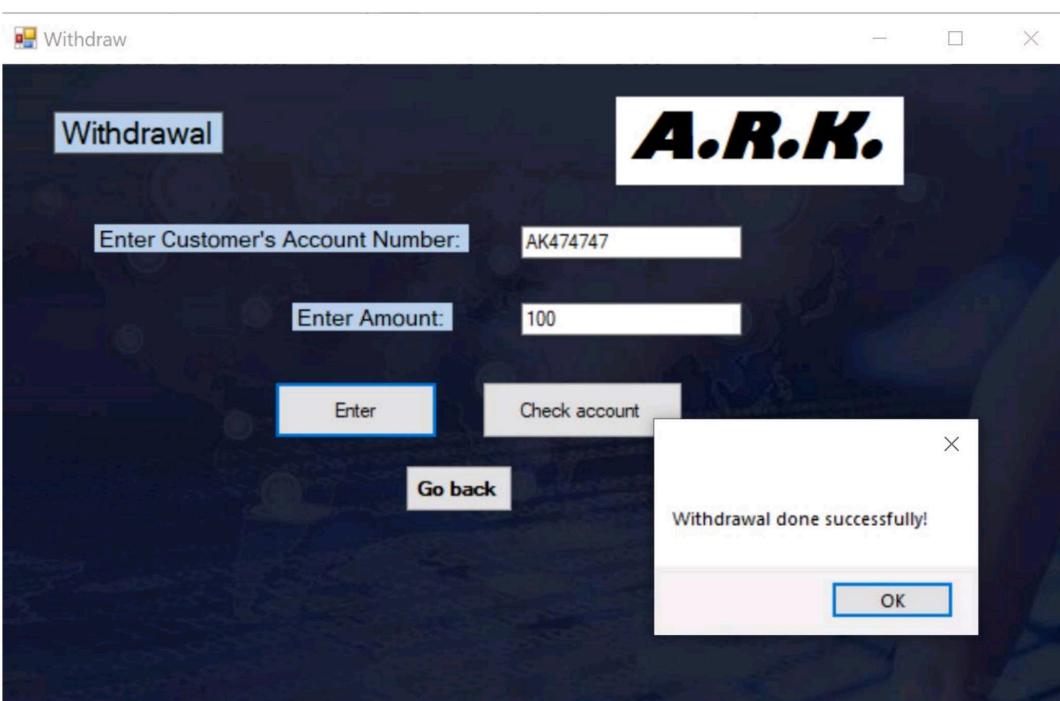
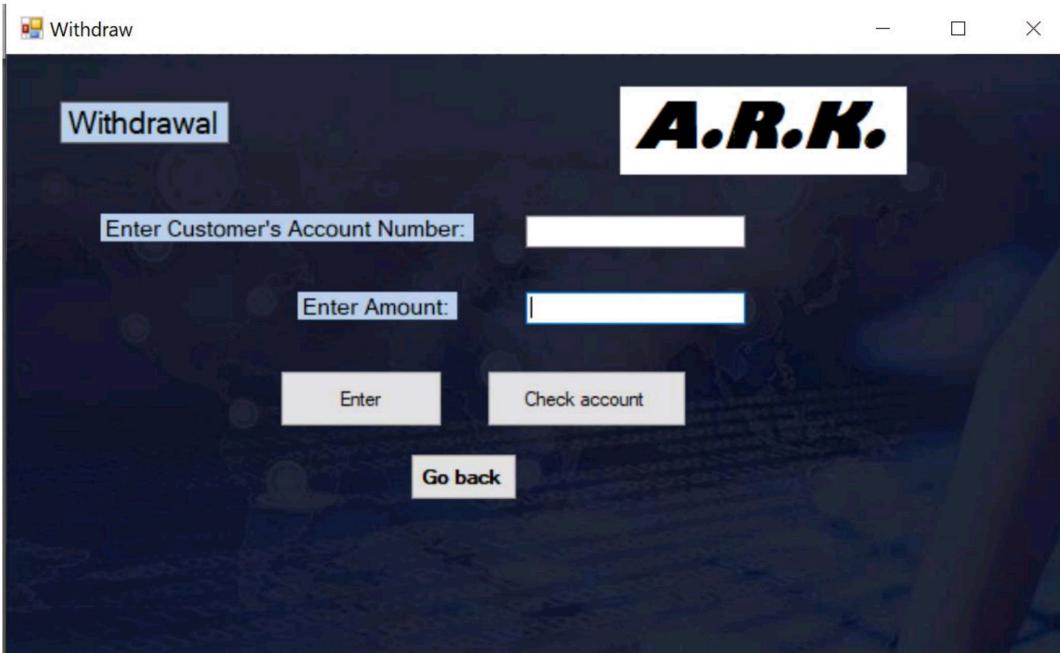
## DeleteCustomer



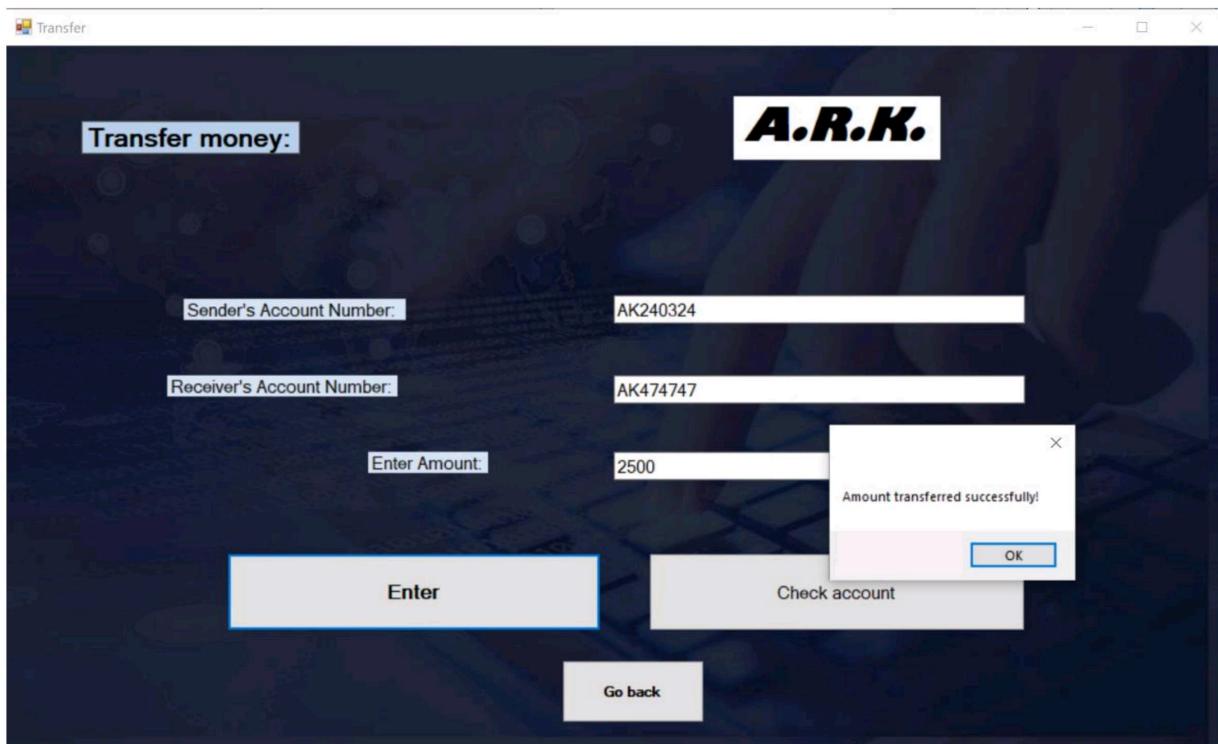
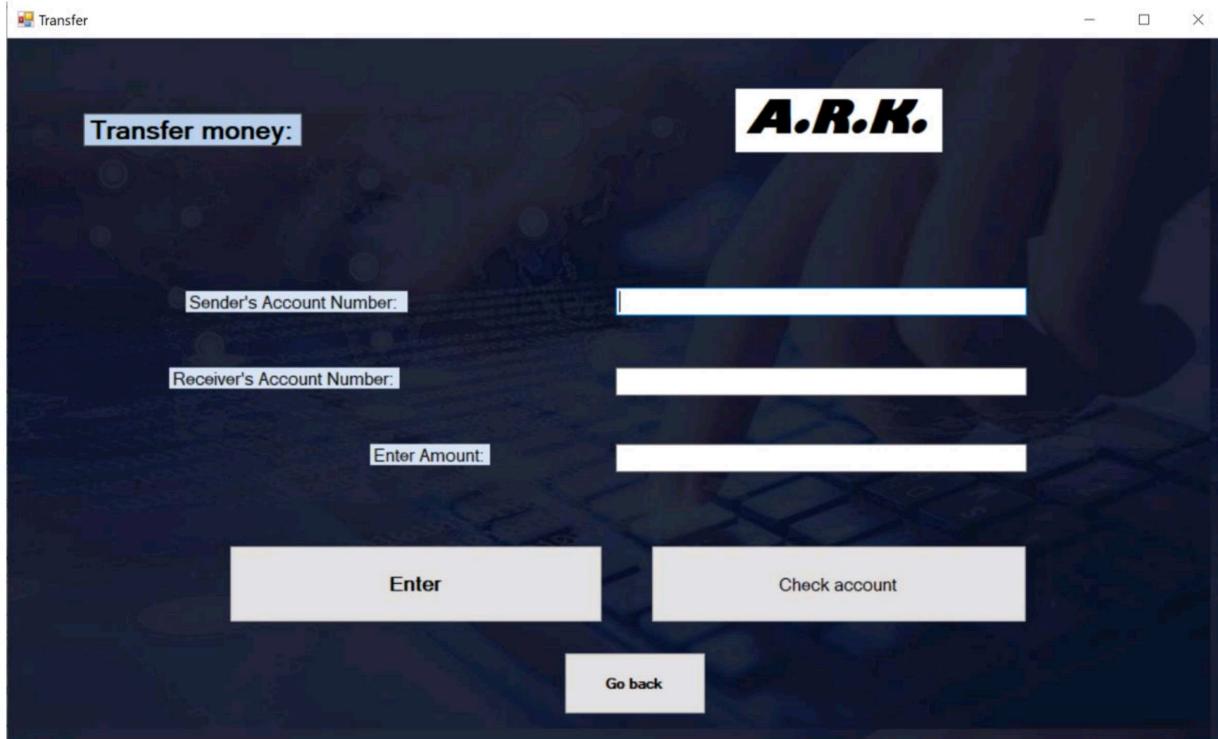
## Deposit1



## Withdraw



## Transfer



## CRYSTAL REPORT OF USER ACCOUNTS

<u>name</u>	<u>dob</u>	<u>gend</u>	<u>age</u>	<u>balance</u>	<u>accountno</u>	<u>email</u>	<u>phoneno</u>	<u>sadham</u>
Abi	03/05/1992	M	29.00	#####	A1234567	abi@gmail.com	#####	#####
earthy	10/11/1990	F	30.00	#####	AC231245	earthy@yahoo.com	#####	#####

<u>name</u>	<u>dob</u>	<u>gendr</u>	<u>age</u>	<u>balance</u>	<u>accountno</u>	<u>email</u>	<u>phoneno</u>	<u>sadham</u>
Sunya	10/08/1989	M	31.00	#####	AK474747	sun@gmail.com	#####	#####

## CODE DESIGN

### Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WindowsApp2
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new AdminLogin());
        }
    }
}
```

### AdminLogin.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WindowsApp2
{
    public partial class AdminLogin : Form
    {
        public AdminLogin()
        {
            InitializeComponent();
        }
    }
}
```

```

        }
        SqlConnection con = new SqlConnection("Data Source=DESKTOP-
UDH7VMN\SQLEXPRESS01;Initial Catalog=SavingsBank;Integrated Security=True");
        private void label4_Click(object sender, EventArgs e)
        {
        }

        private void button1_Click(object sender, EventArgs e)
        {
            con.Open();
            SqlCommand cmd = new SqlCommand("select * from admin_login where username='" +
adminid.Text + "' and password='" + adminpwd.Text + "'", con);
            SqlDataReader dr = cmd.ExecuteReader();
            if (dr.Read())
            {
                AdminActivities adminact = new AdminActivities();
                adminact.Show();
            }
            else
            {
                MessageBox.Show("Invalid username");
            }
            dr.Close();
            con.Close();
        }
    }
}

```

### AdminActivities.cs

```

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;

namespace WindowsApp2
{
    public partial class AdminActivities : Form
    {
        public AdminActivities()
        {
            InitializeComponent();
        }
    }
}

```

```

        private void LinkLabel2_LinkClicked(object sender, LinkLabelLinkClickedEventArgs
e)
{
    this.Hide();
    AddCustomer addcustomer = new AddCustomer();
    addcustomer.Show();
}

private void LinkLabel3_LinkClicked(object sender, LinkLabelLinkClickedEventArgs
e)
{
    this.Hide();
    DeleteCustomer dc = new DeleteCustomer();
    dc.Show();
}

private void LinkLabel4_LinkClicked(object sender, LinkLabelLinkClickedEventArgs
e)
{
    this.Hide();
    Transfer tr = new Transfer();
    tr.Show();
}

private void LinkLabel6_LinkClicked(object sender, LinkLabelLinkClickedEventArgs
e)
{
    this.Hide();
    Withdraw wd = new Withdraw();
    wd.Show();
}

private void LinkLabel5_LinkClicked(object sender, LinkLabelLinkClickedEventArgs
e)
{
    this.Hide();
    Deposit1 dep = new Deposit1();
    dep.Show();
}

private void Label1_Click(object sender, EventArgs e)
{

}

private void LinkLabel1_LinkClicked(object sender, LinkLabelLinkClickedEventArgs
e)
{
    this.Hide();
    CustomerDetails cd = new CustomerDetails();
    cd.Show();
}
}

```

### AddCustomer.cs

```

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 WindowsApp2
{
    public partial class AddCustomer : Form
    {
        public AddCustomer()
        {
            InitializeComponent();
        }

        SqlConnection con = new SqlConnection("Data Source=DESKTOP-UDH7VMN\\SQLEXPRESS01;Initial Catalog=SavingsBank;Integrated Security=True");

        private void button1_Click(object sender, EventArgs e)
        {
            con.Open();
            SqlCommand cmd = new SqlCommand("insert into register
(name,dob,gender,age,balance,accountno,email,phoneno,aadharno) values ('" + tname.Text +
"', '" + tdob.Text + "','" + cbgender.Text + "','" + tage.Text + "," + tbalance.Text + "," +
+ taccno.Text + "','" + temail.Text + "," + tphone.Text + "," + taadhar.Text + ")");
            cmd.ExecuteNonQuery();
            MessageBox.Show("Registered successfully!");
            con.Close();
        }

        private void TextBox4_TextChanged(object sender, EventArgs e)
        {

        }

        private void TextBox6_TextChanged(object sender, EventArgs e)
        {

        }

        private void tdob_TextChanged(object sender, EventArgs e)
        {

        }

        private void Label12_Click(object sender, EventArgs e)
        {

        }

        private void AddCustomer_Load(object sender, EventArgs e)
        {

        }

        private void button2_Click(object sender, EventArgs e)

```

```

{
    con.Open();
    SqlCommand cmd = new SqlCommand("update register set phoneno=" + tphone.Text
+ ", balance=" + tbalance.Text + " where accountno='" + tacctno.Text + "'", con);
    cmd.ExecuteNonQuery();
    MessageBox.Show("Updated successfully!");
    con.Close();
}

private void button3_Click(object sender, EventArgs e)
{
    this.Hide();
    AdminActivities ad = new AdminActivities();
    ad.Show();
}

private void label3_Click(object sender, EventArgs e)
{

}

private void label13_Click(object sender, EventArgs e)
{

}

private void Label4_Click(object sender, EventArgs e)
{

}

private void Label5_Click(object sender, EventArgs e)
{

}

private void Label6_Click(object sender, EventArgs e)
{

}

private void Label7_Click(object sender, EventArgs e)
{

}

private void Label9_Click(object sender, EventArgs e)
{

}

private void Label10_Click(object sender, EventArgs e)
{

}

private void Label11_Click(object sender, EventArgs e)
{
}

```

```
}

private void taadhar_TextChanged(object sender, EventArgs e)
{
}

private void temail_TextChanged(object sender, EventArgs e)
{
}

private void tacccno_TextChanged(object sender, EventArgs e)
{
}

private void tbalance_TextChanged(object sender, EventArgs e)
{
}

private void tage_TextChanged(object sender, EventArgs e)
{
}

private void cbgender_SelectedIndexChanged(object sender, EventArgs e)
{
}

private void tname_TextChanged(object sender, EventArgs e)
{
}

}
```

## DeleteCustomer.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WindowsApp2
{
    public partial class DeleteCustomer : Form
    {
        public DeleteCustomer()
        {
```

```

{
    InitializeComponent();
}
SqlConnection con = new SqlConnection("Data Source=DESKTOP-
UDH7VMN\SQLEXPRESS01;Initial Catalog=SavingsBank;Integrated Security=True");
private void button1_Click(object sender, EventArgs e)
{
    con.Open();
    SqlCommand cmd = new SqlCommand("delete from register where accountno='"
+ textBox1.Text + "'", con);
    cmd.ExecuteNonQuery();
    MessageBox.Show("Deleted successfully!");
    con.Close();
}

private void DeleteCustomer_Load(object sender, EventArgs e)
{
    Refresh();
}

private void button2_Click(object sender, EventArgs e)
{
    int rowindx = dgv_deletecustomer.SelectedCells[0].RowIndex;
    con.Open();
    SqlCommand cmd = new SqlCommand("delete from register where accountno='"
+ dgv_deletecustomer.Rows[rowindx].Cells[5].Value.ToString() + "'", con);
    cmd.ExecuteNonQuery();
    MessageBox.Show("Deleted successfully!");
    con.Close();
    Refresh();
}
void Refresh()
{
    con.Open();
    SqlDataAdapter ada = new SqlDataAdapter("select * from register", con);
    DataSet ds = new DataSet();
    ada.Fill(ds, "register");
    dgv_deletecustomer.DataSource = ds;
    dgv_deletecustomer.DataMember = "register";
    con.Close();
}

private void label2_Click(object sender, EventArgs e)
{
}

private void dgv_deletecustomer_CellContentClick(object sender,
DataGridViewCellEventArgs e)
{
}

private void button3_Click(object sender, EventArgs e)
{
    this.Hide();
    AdminActivities ad = new AdminActivities();
    ad.Show();
}

```

```
    }
}
```

### Deposit1.cs

```
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 WindowsApp2
{
    public partial class Deposit1 : Form
    {
        public Deposit1()
        {
            InitializeComponent();
        }

        SqlConnection con = new SqlConnection("Data Source=DESKTOP-UDH7VMN\SQLEXPRESS01;Initial Catalog=SavingsBank;Integrated Security=True");
        private void button1_Click(object sender, EventArgs e)
        {

            int test = int.Parse(depositamount.Text);
            if (test <= 0)
            {
                MessageBox.Show("You cannot enter negative amount");

            }
            else
            {
                MessageBox.Show("amount is > 0");
                con.Open();

                SqlCommand cmd = new SqlCommand("update register set balance=balance + " +
+ depositamount.Text + " where accountno=' " + cbaccno.SelectedItem.ToString() + "'",
                con);
                cmd.ExecuteNonQuery();
                SqlCommand comd = new SqlCommand("insert into deposit
                (accountno,depamount,balance) values ('" + cbaccno.SelectedItem.ToString() + "','" +
depositamount.Text + "," + depositbalance.Text + "')", con);
                comd.ExecuteNonQuery();
                MessageBox.Show("Deposited successfully!");
                con.Close();
            }
        }

        private void cbaccno_SelectedIndexChanged(object sender, EventArgs e)
        {
            con.Open();
```

```

        SqlCommand cmd = new SqlCommand("select balance from register where
accountno=" + cbaccno.SelectedItem.ToString() + "'", con);
        SqlDataReader dr = cmd.ExecuteReader();
        while (dr.Read())
        {
            depositbalance.Text = dr.GetValue(0).ToString();
        }
        dr.Close();
        con.Close();
    }

    private void Deposit1_Load(object sender, EventArgs e)
    {
        con.Open();
        SqlCommand cmd = new SqlCommand("select * from register", con);
        SqlDataReader dr = cmd.ExecuteReader();
        while (dr.Read())
        {
            cbaccno.Items.Add(dr.GetValue(5).ToString());
        }
        dr.Close();
        con.Close();
    }

    private void button2_Click(object sender, EventArgs e)
    {
        this.Hide();
        CustomerDetails cd = new CustomerDetails();
        cd.Show();
    }

    private void button3_Click(object sender, EventArgs e)
    {
        this.Hide();
        AdminActivities ad = new AdminActivities();
        ad.Show();
    }
}
}

```

### Withdraw.cs

```

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 WindowsApp2
{
    public partial class Withdraw : Form

```

```

{
    public Withdraw()
    {
        InitializeComponent();
    }
    SqlConnection con = new SqlConnection("Data Source=DESKTOP-
UDH7VMN\SQLEXPRESS01;Initial Catalog=SavingsBank;Integrated Security=True");
    private void button1_Click(object sender, EventArgs e)
    {
        int test = int.Parse(wdamount.Text);
        if (test <= 0)
        {
            MessageBox.Show("You cannot enter negative amount");

        }
        else
        {
            con.Open();
            SqlCommand cmd = new SqlCommand("update register set balance=balance - "
+ wdamount.Text + " where accountno='" + wdaccno.Text + "'", con);
            cmd.ExecuteNonQuery();
            MessageBox.Show("Withdrawal done successfully!");
            con.Close();
        }
    }

    private void button2_Click(object sender, EventArgs e)
    {
        this.Hide();
        CustomerDetails cd = new CustomerDetails();
        cd.Show();
    }

    private void button3_Click(object sender, EventArgs e)
    {
        this.Hide();
        AdminActivities ad = new AdminActivities();
        ad.Show();
    }
}
}

```

### Transfer.cs

```

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 WindowsApp2
{

```

```

public partial class Transfer : Form
{
    public Transfer()
    {
        InitializeComponent();
    }
    SqlConnection con = new SqlConnection("Data Source=DESKTOP-
UDH7VMN\SQLEXPRESS01;Initial Catalog=SavingsBank;Integrated Security=True");

    private void Transfer_Load(object sender, EventArgs e)
    {

    }

    private void button1_Click(object sender, EventArgs e)
    {
        int test = int.Parse(tamount.Text);
        if (test <= 0)
        {
            MessageBox.Show("You cannot enter negative amount");

        }
        else
        {
            con.Open();
            SqlCommand cmd = new SqlCommand("update register set balance=balance - "
+ tamount.Text + " where accountno='" + senderaccno.Text + "'", con);
            cmd.ExecuteNonQuery();
            SqlCommand comd = new SqlCommand("update register set balance=balance + "
+ tamount.Text + " where accountno='" + receiveraccno.Text + "'", con);
            comd.ExecuteNonQuery();
            MessageBox.Show("Amount transferred successfully!");
            con.Close();
        }
    }

    private void button2_Click(object sender, EventArgs e)
    {
        this.Hide();
        CustomerDetails cd = new CustomerDetails();
        cd.Show();
    }

    private void button3_Click(object sender, EventArgs e)
    {
        this.Hide();
        AdminActivities ad = new AdminActivities();
        ad.Show();
    }
}

```

### Report.cs

```

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;

namespace WindowsApp2
{
    public partial class Report : Form
    {
        public Report()
        {
            InitializeComponent();
        }
    }
}
```

## SYSTEM TESTING

System testing of software or hardware is testing conducted on a complete integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing and as such requires no knowledge of inner design of code or logic. System testing includes testing for bugs in the projects. Testing is useful to check syntax and logical errors. All texts should be traceable to the customer requirements. Test should be planned long before testing begins 80 percent all errors uncovered during testing will likely be traceable to 20 percent of all program components. The testing process begins with levels of testing followed by test plan and procedures. In further steps test case is analyzed and executed. During this process if errors are uncovered then testing is rolled back to test plan and all other steps are repeated. Otherwise, if there are no errors, testing is successful and finally the test project is given.

System testing is performed on entire system in the context of a Functional Requirement Specification (FRS) and/or a System Requirement Specification (SRS). System tests only the design, but also behaviour and even the believed expectations of the customers. It also intended to test upto and beyond the bounds defined in the software/hardware requirements specifications.

### White Box Testing

It is also called as glass box testing. It is a test case design method that uses structure of the procedural design to derive test cases. Using this method the software engineer can derive test cases.

- 1) Guarantee that all independent paths within a module have been exercised at least ones.
- 2) Exercise all logical decisions on their true and false sides.
- 3) Exercise also loops at their boundaries and within their operational bounds.
- 4) Exercise internal data structures to ensure their validity.

White box testing of software is predicated on close examination of procedural detail, the system was tested for the calculation matters were the data provided for giving the right output or not. If wrong data was provided then what it is throwing error or accepting.

### Black Box Testing

It is also called as behaviour testing which focus on the functional requirement of the software. That means it enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements of the program. Black box testing is a complimentary

approach that is likely to uncover a different class of errors than white box testing. This testing is used to demonstrate that

software functions are operational. That is, it ensures the input is properly accepted and output is correctly produced. The integrity of external information is maintained. It examines some fundamental aspects of a system. Black box testing attempts to find the following errors:

Incorrect or missing functions

Interface errors

Errors in external data space access

Behavior or performance errors

Initialization and termination errors

By applying black box techniques, we derive a set of test cases that satisfy the following criteria:

- a) Test cases that reduce, by a count that is greater than one, the number of additional test cases that must be designed to achieve reasonable testing.
- b) Test cases that tell us something about the presence or absence of classes of errors, rather than an error associated only with specific test at hand.

The attributes of white box and black box testing can be combined to provide an approach that validates the software interface and selectively ensures that internal working of software is correct.

Black box testing for this system was done to check the internal testing to check whether the system is working properly in each case or no and what kind of errors are there in database designs.

## FUTURE ENHANCEMENTS

### The future of Savings Bank System

The scope of the project includes all the future enhancements which can be done to make it more feasible.

New technologies have greatly improved the tools used to manage and make money transactions accurately. The features that will be available in the new technology will let us to add images of a customer. Powerful computer systems that are linked into the networks are now able to receive the information from handheld devices. The wireless handheld devices will help the admins to operate and send the data to a tracking database in real time.

Features which can be included for future enhancements of this project are:

Databases for each type of transactions can be provided to check that particular transaction.

Multilingual support can be provided so that it can be understandable by the persons of any language.

More graphics can be added to make it more user-friendly and understandable.

We can make a platform independent application by using HTML, JAVASCRIPT, JSP.

## REFERENCES

[www.google.com](http://www.google.com)

[www.tutorialspoint.com](http://www.tutorialspoint.com)

[www.youtube.com](http://www.youtube.com)