JAB
John Abbott Bank®

Bank Management System & Personal Bank

Ksenia Studilina
Saulo Moreira da Silva

Customer relationship management

Software solutions for businesses that facilitate interaction with customers

Bank CRM

- Manage customers data
- Manage accounts
- Keep track of transactions
- Generate reports

Customer personal systems

Software solutions for customers that help to keep track of personal information and use business services

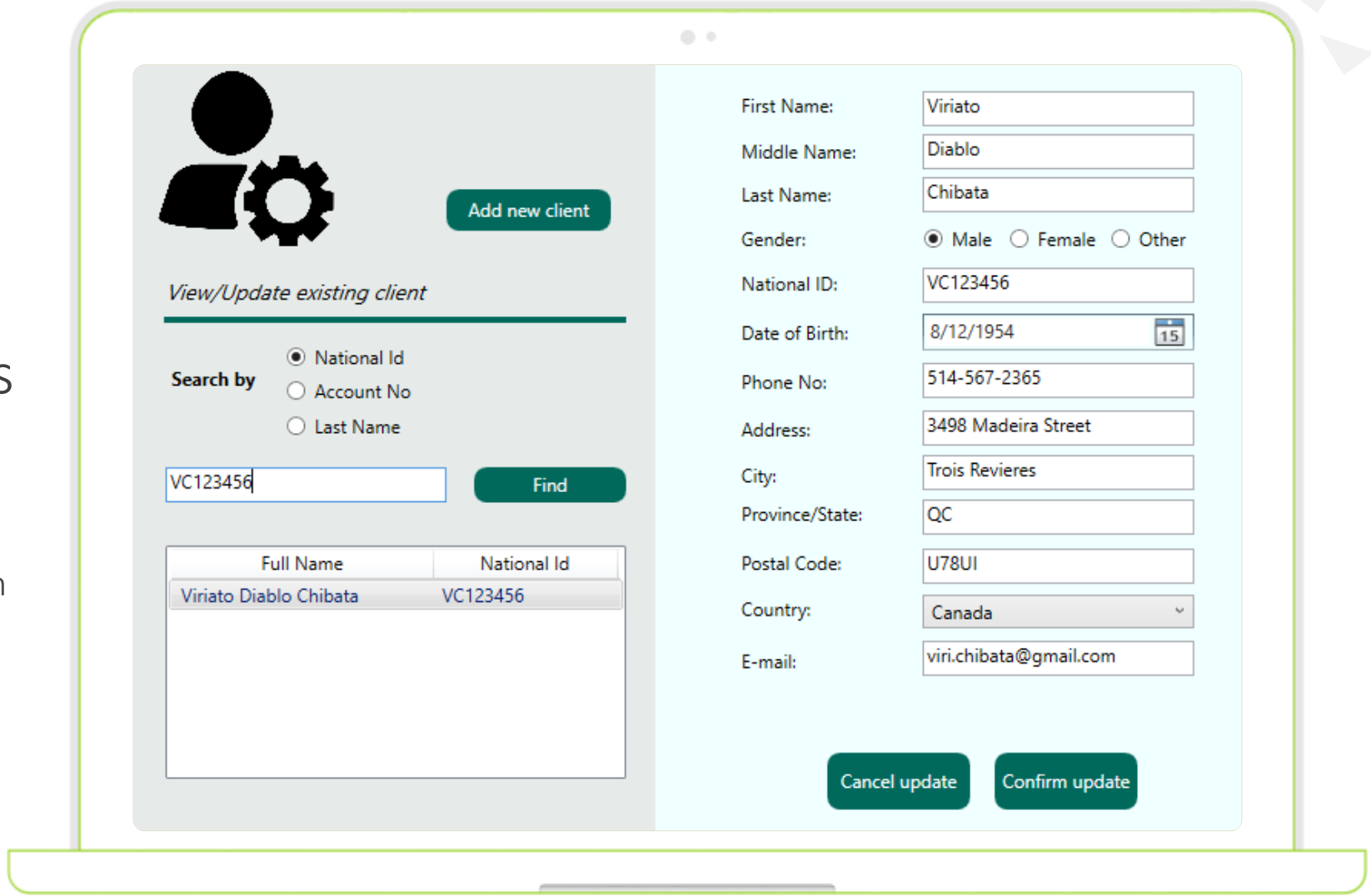
Personal banking

- Keep track of finances
- Perform transactions
- Update personal data

Bank Management System Functionality Overview

Managing customers

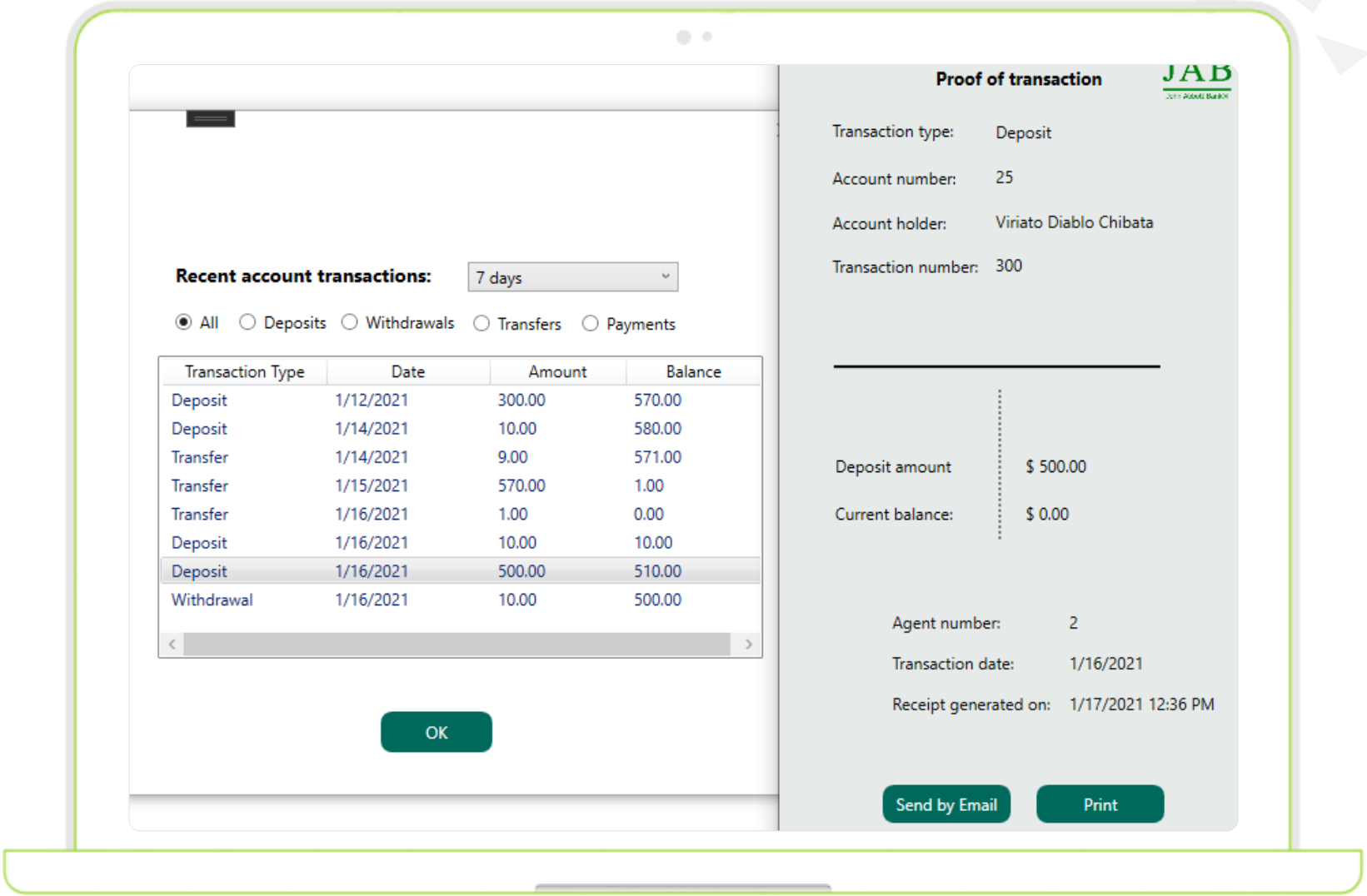
- Add new customers
- Update existing customers
- Gather customers' information



Bank Management System Functionality Overview

Managing accounts and transactions

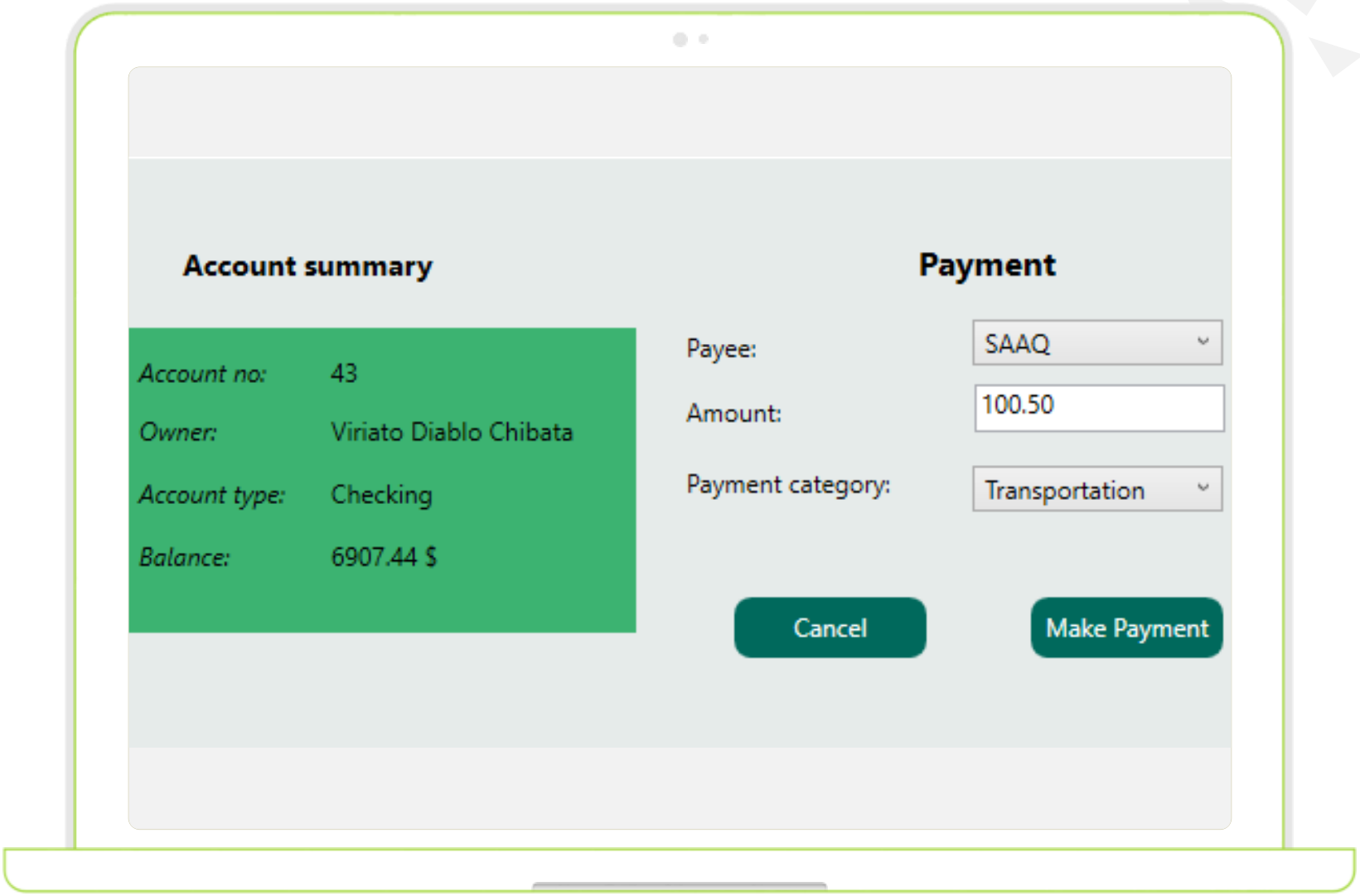
- Add/Update/Close account
- Perform and keep record of transactions
- Generate receipts
- Generate monthly statements



Personal Bank Functionality Overview

Perform account functions

- View balance
- Make transactions
- Generate receipts

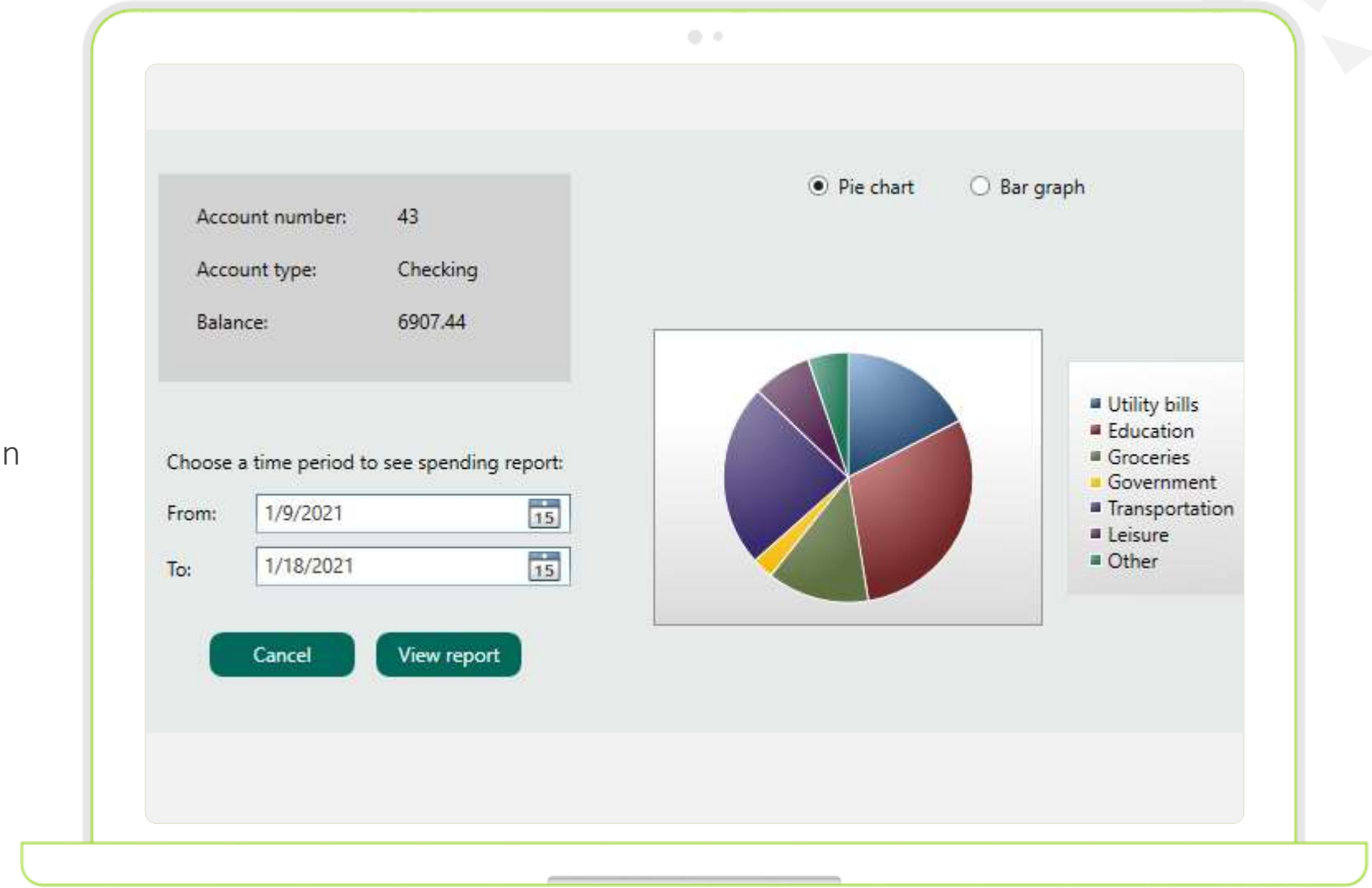


Personal Bank Functionality Overview

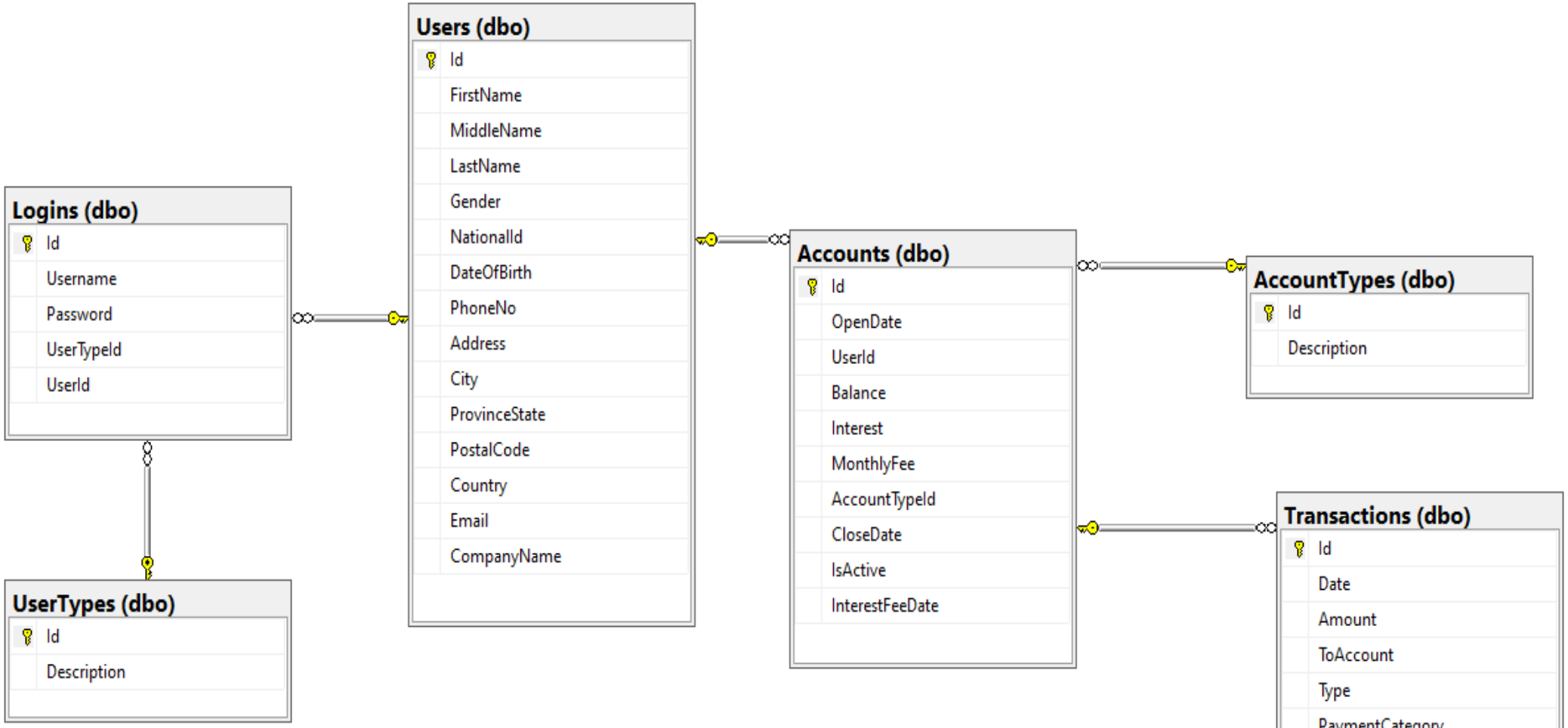
Keep track of spending

- Visual spending representation

over any time period



Database structure



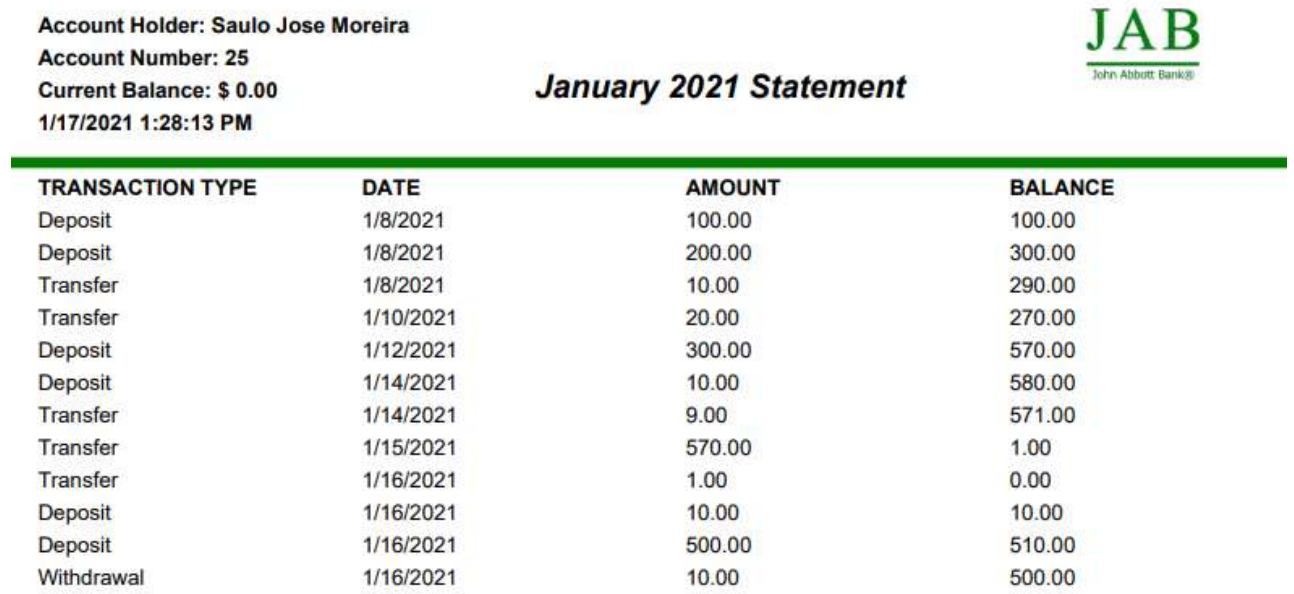
Challenges and Solutions

PDF creation

How to turn a ListView of transactions with a GridView into a PDF document ?

PDFSharp

- PDF library creates empty PDF document
- Offers solutions to draw graphical elements on it



Account Holder: Saulo Jose Moreira
Account Number: 25
Current Balance: $ 0.00
1/17/2021 1:28:13 PM

January 2021 Statement

JAB
John Abbott Bank®

TRANSACTION TYPE	DATE	AMOUNT	BALANCE
Deposit	1/8/2021	100.00	100.00
Deposit	1/8/2021	200.00	300.00
Transfer	1/8/2021	10.00	290.00
Transfer	1/10/2021	20.00	270.00
Deposit	1/12/2021	300.00	570.00
Deposit	1/14/2021	10.00	580.00
Transfer	1/14/2021	9.00	571.00
Transfer	1/15/2021	570.00	1.00
Transfer	1/16/2021	1.00	0.00
Deposit	1/16/2021	10.00	10.00
Deposit	1/16/2021	500.00	510.00
Withdrawal	1/16/2021	10.00	500.00

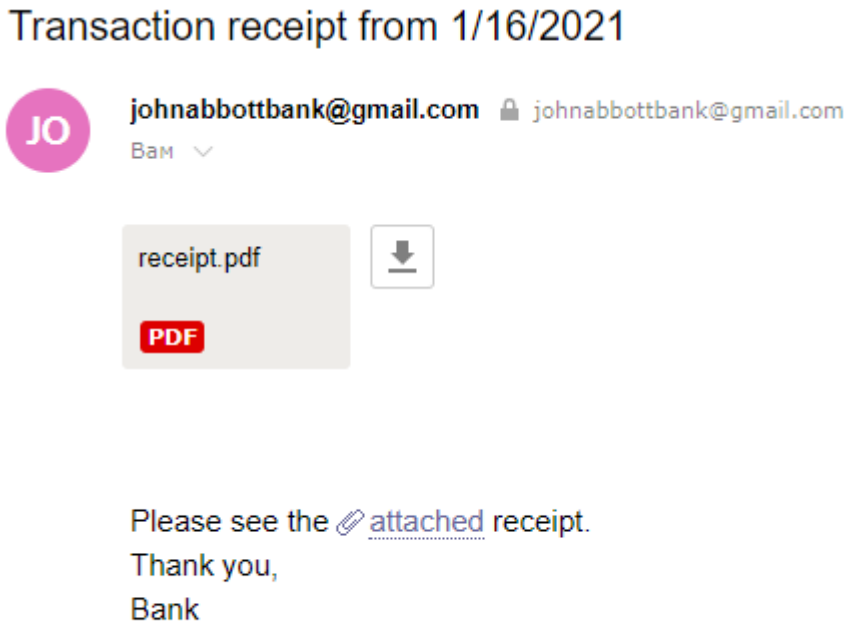
Challenges and Solutions

Sending E-mail

How to send generated PDF by E-mail ?

SmtpClient Class

- Allows applications to send email by using the Simple Mail Transfer Protocol
- **Attachment Class** allows to attach existing file to email



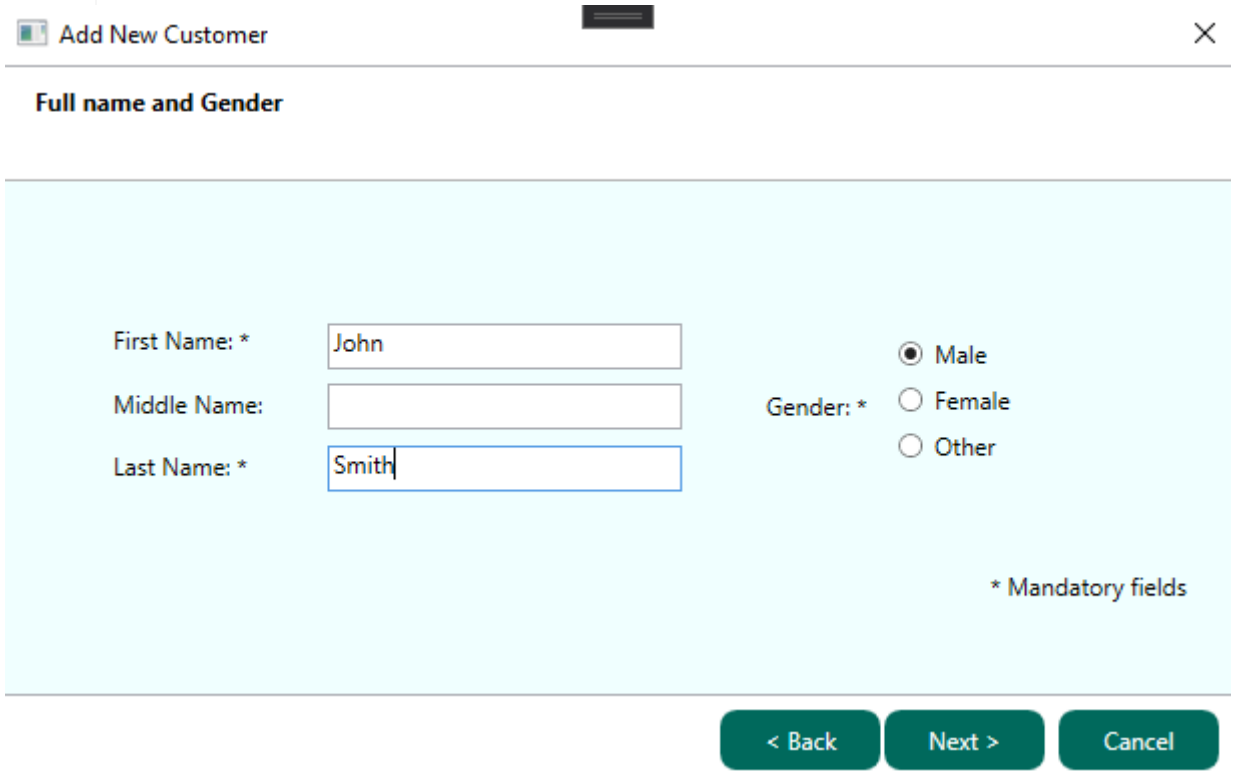
Challenges and Solutions

Adding customer with the help of wizard

How to create a wizard that will require to enter new customer information step by step?

WFP Extended Toolkit

- Allows to create multipage wizard



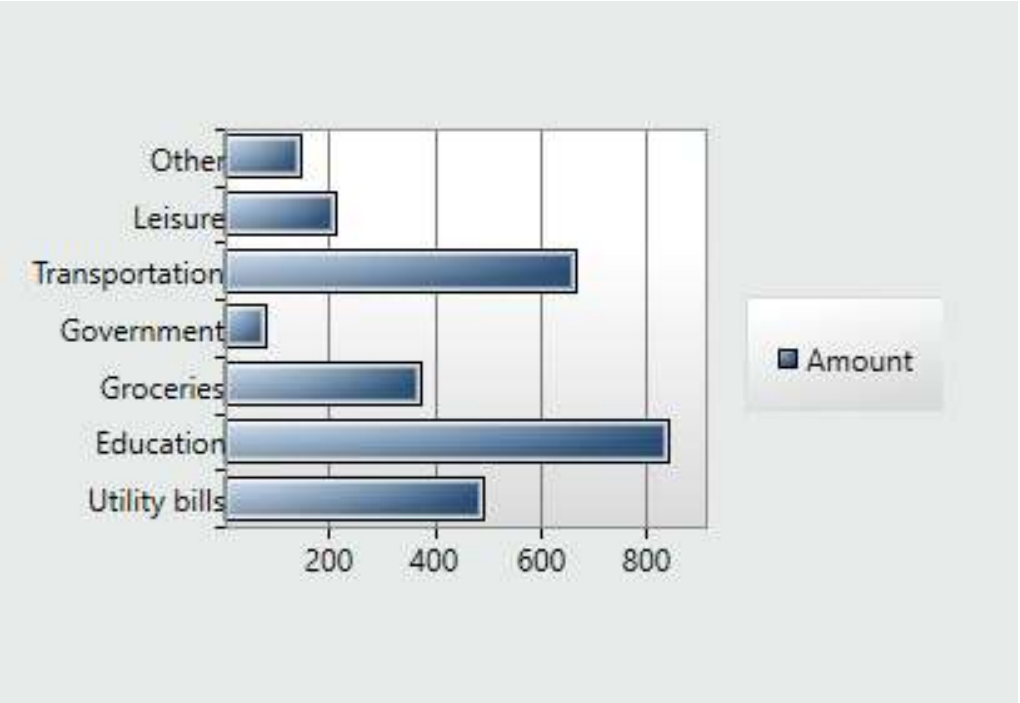
Challenges and Solutions

Visualize data

How to create a graph that will visually represent information about transactions?

System.Windows.Controls.DataVisualization.Toolkit

- Allows to create various types of graphs



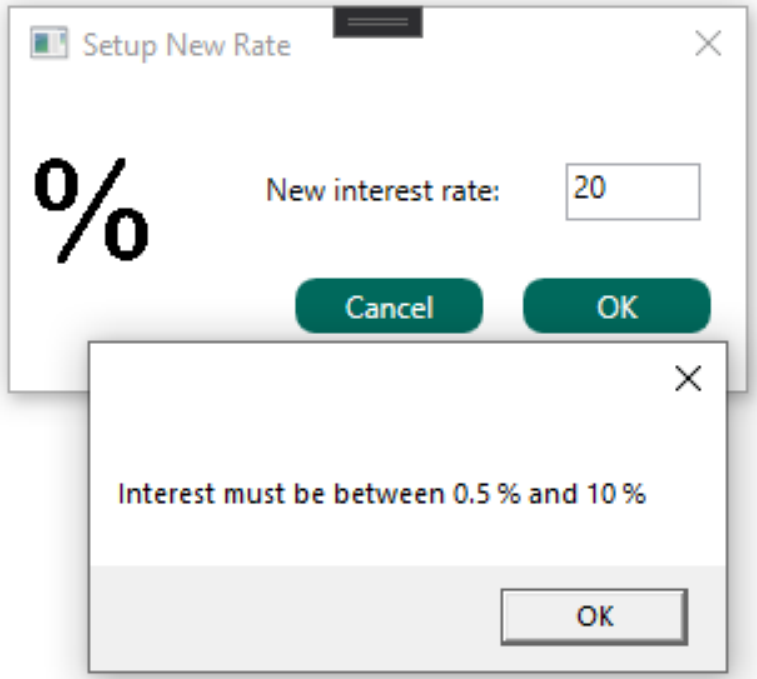
Challenges and Solutions

Entities validation

How to validate data before inserting it to database?

IValidatableObject Interface

- Provides a way for an object to be validated when method SaveChanges() is called



Challenges and Solutions

Installer

How to an installer for the program?

MS Visual Studio Installer Projects

- Official Microsoft extension for Visual Studio that allows to create WPF Setup Installer

In Details: How to create PDF

1. Install Pdfsharp library

2. Create an empty PDF document and add first page

```
PdfDocument document = new PdfDocument();
PdfPage page = document.AddPage();
```

3. Create a font

```
XFont fontReg = new XFont("Arial", 10, XFontStyle.Regular);
XFont fontBold = new XFont("Arial", 10, XFontStyle.Bold);
XFont fontItalic = new XFont("Arial", 10, XFontStyle.Italic);
XFont fontBoldItalic = new XFont("Arial", 15, XFontStyle.BoldItalic);
```

4. Drawing is done with XGraphics object

```
XGraphics graphics = XGraphics.FromPdfPage(page);
```

5. Draw string or line with the help of XGraphics and font

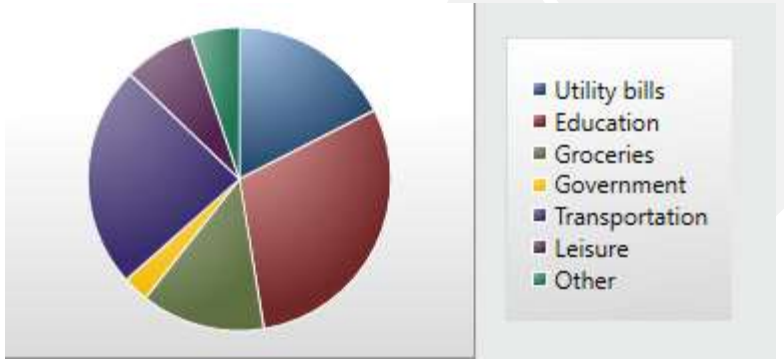
```
graphics.DrawString("Account Number: " + currentAccount.Id, fontBold, XBrushes.Black, 20, 45);
```

```
graphics.DrawLine(lineRed, pt1, pt2);
```

6. Draw a table with the help of XGraphics and font

```
for (int i = 0; i < tr.Count; i++)
{
    Transaction t = tr[i];
    graphics.DrawString(t.Type, fontReg, XBrushes.Black, 20, ind);
    graphics.DrawString(t.Date.ToShortDateString(), fontReg, XBrushes.Black, 170, ind);
    graphics.DrawString(t.Amount.ToString(), fontReg, XBrushes.Black, 320, ind);
    ind = ind + 15;
}
```

In Details: How to create a pie chart



1. Install System.Windows.Controls.DataVisualization.Toolkit

↓

2. In XAML add **Chart** and **Series** (depend on the type of desired chart)

```
<DVC:Chart Canvas.Top="80" Canvas.Left="10" Name="mcChart" Width="400" Height="250" BorderThickness="0">
    <DVC:Chart.Series>
        <DVC:PieSeries BorderThickness="0" IndependentValueBinding="{Binding Path=Key}" DependentValueBinding="{Binding Path=Value}">
        </DVC:PieSeries>
    </DVC:Chart.Series>
</DVC:Chart>
```

↓

3. To load data to pie chart

```
List<decimal> amounts = new List<decimal>();


foreach (string pc in Utils.paymentCategories)
{
    var transacByCat = transactions.FindAll(t => t.PaymentCategory == pc);
    decimal sum = 0;
    foreach (Transaction t in transacByCat)
    {
        sum = sum + t.Amount;
    }
    amounts.Add(sum);
}
```

→

```
var CategoryAmount = Enumerable.Range(0, Utils.paymentCategories.Count)
.Select(i => new KeyValuePair<string, decimal>(Utils.paymentCategories[i], amounts[i]))
.ToList();

((PieSeries)mcChart.Series[0]).ItemsSource = CategoryAmount;
```

In Details: How to validate entities

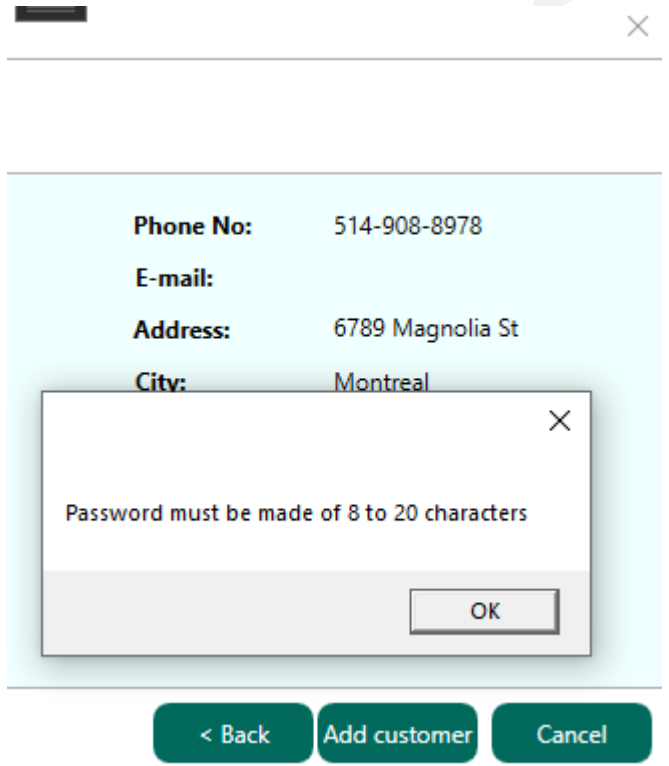
1. Add Ivalidatable Object Interface to class

```
public partial class Login : IValidatableObject
{
    3 references
    public IEnumerable<ValidationResult> Validate(ValidationContext validationContext)
    {
        if (Username.Length < 5 || Username.Length > 20)
        {
            yield return new ValidationResult(
                    "Username must be made of 5 to 20 characters",
                    new[] { nameof(Username) });
        }
        if (Password.Length < 8 || Username.Length > 20)
        {
            yield return new ValidationResult(
                    "Password must be made of 8 to 20 characters",
                    new[] { nameof(Password) });
        }
    }
}
```

In Details: How to validate entities

2. Catch DbEntityValidationException

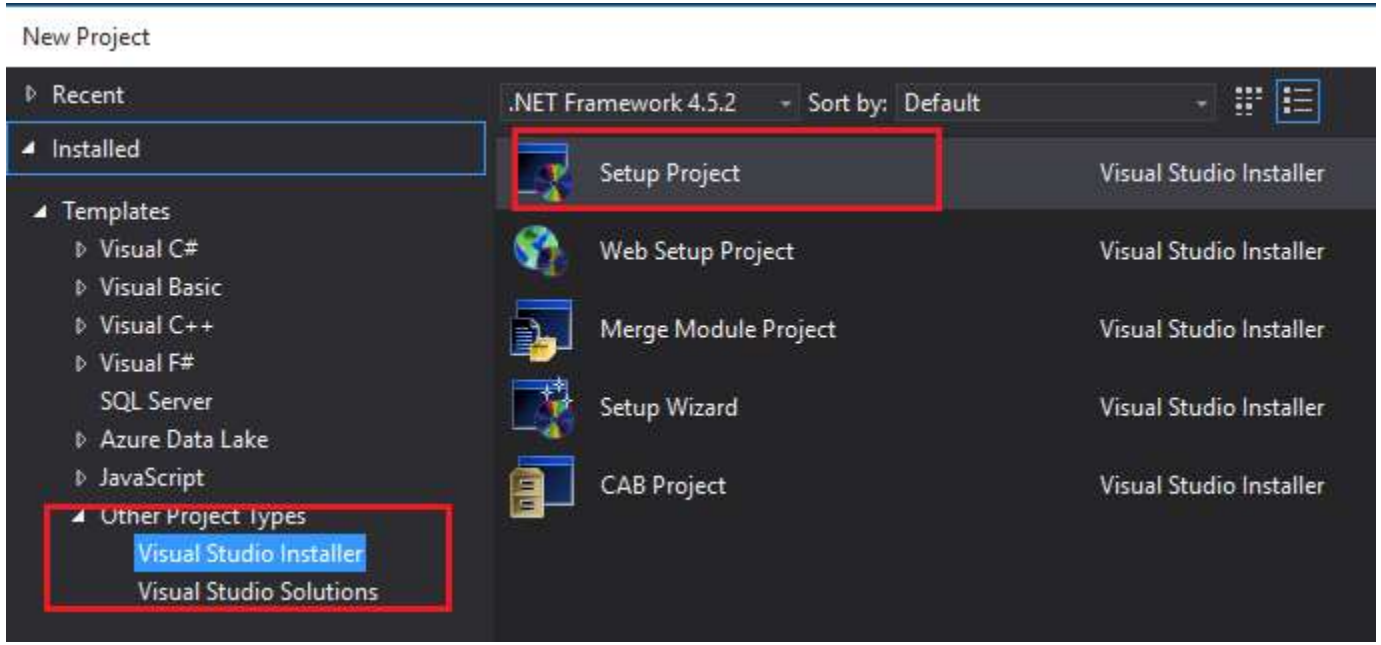
```
private void AddLogin()
{

    Login login = new Login
    {
        Username = tbUsername.Text,
        Password = tbPassword.Text,
        UserTypeId = 3,
        UserId = selectedUser.Id
    };
    try
    {
        EFData.context.Logins.Add(login);
        EFData.context.SaveChanges();
    }
    catch (DbEntityValidationException ex)
    {
        var error = ex.EntityValidationErrors.First().ValidationErrors.First();
        MessageBox.Show(error.ErrorMessage);
        EFData.context.Entry(login).State = EntityState.Detached;
    }
}
```



In Details: How to create a program installer

1. Install MS Visual Studio Installer Projects

2. Create new WPF Project of type **Visual Studio Installer**



3. Create project output and include all files and dependencies

In Details: How to create a program installer

4. Press ‘Batch Build’ To build the installer

5. Find .exe file and install it

Future development

Banking Automatization

Fees, Interest, Dividents

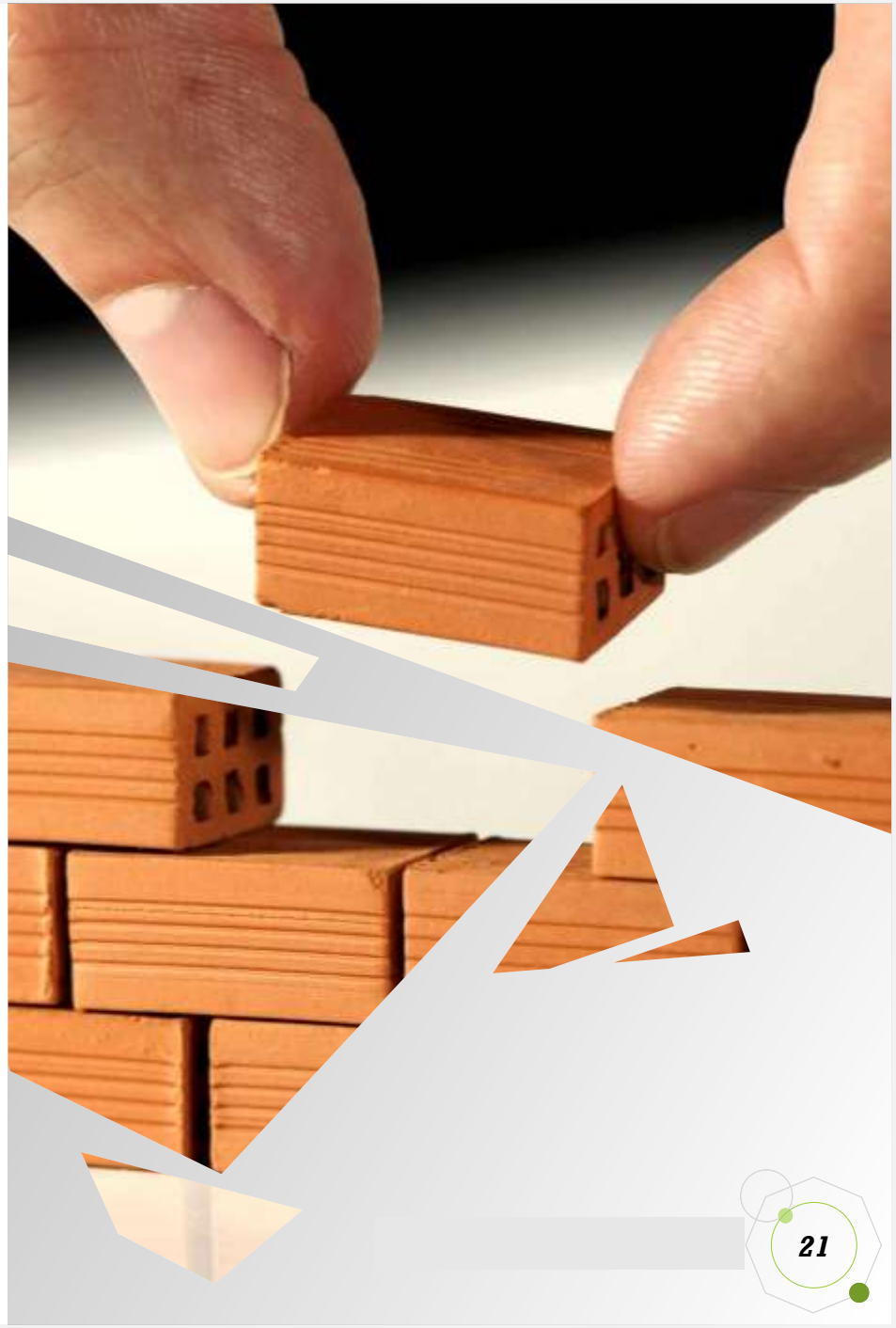
Loans and Credit

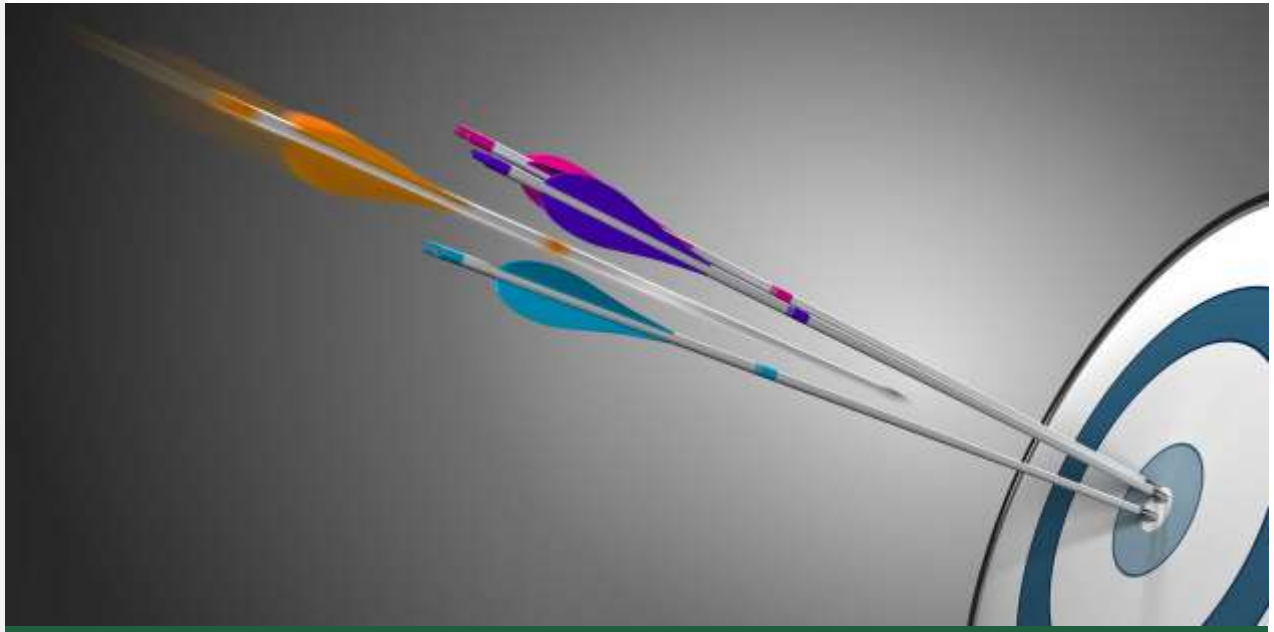
Issue loans and credit cards

Implement interest charge

Banking standards

Account number, branch number, institution number





Summary

Bank Management System

- Clients
- Accounts
- Transactions
- Reports
- Statements
- Receipts

Client Personal Banking

- Accounts
- Transactions
- Personal data
- Spending reports
- Monthly statements


Thank You
JAB
John Abbott Bank®