



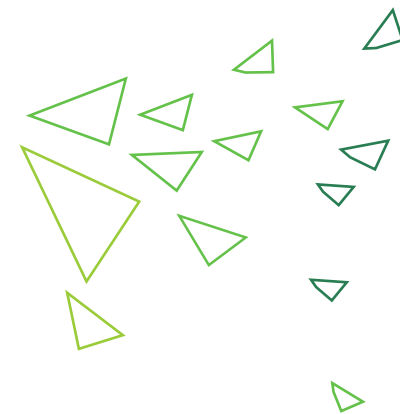
# Bank Management System & Personal Bank

Ksenia Studilina  
Saulo Moreira da Silva



## Customer relationship management

Software solutions for businesses that facilitate interaction with customers



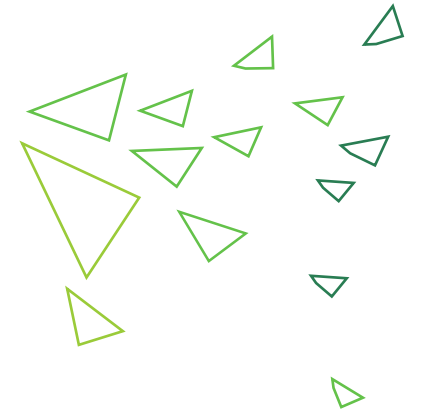
## Bank CRM

- 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



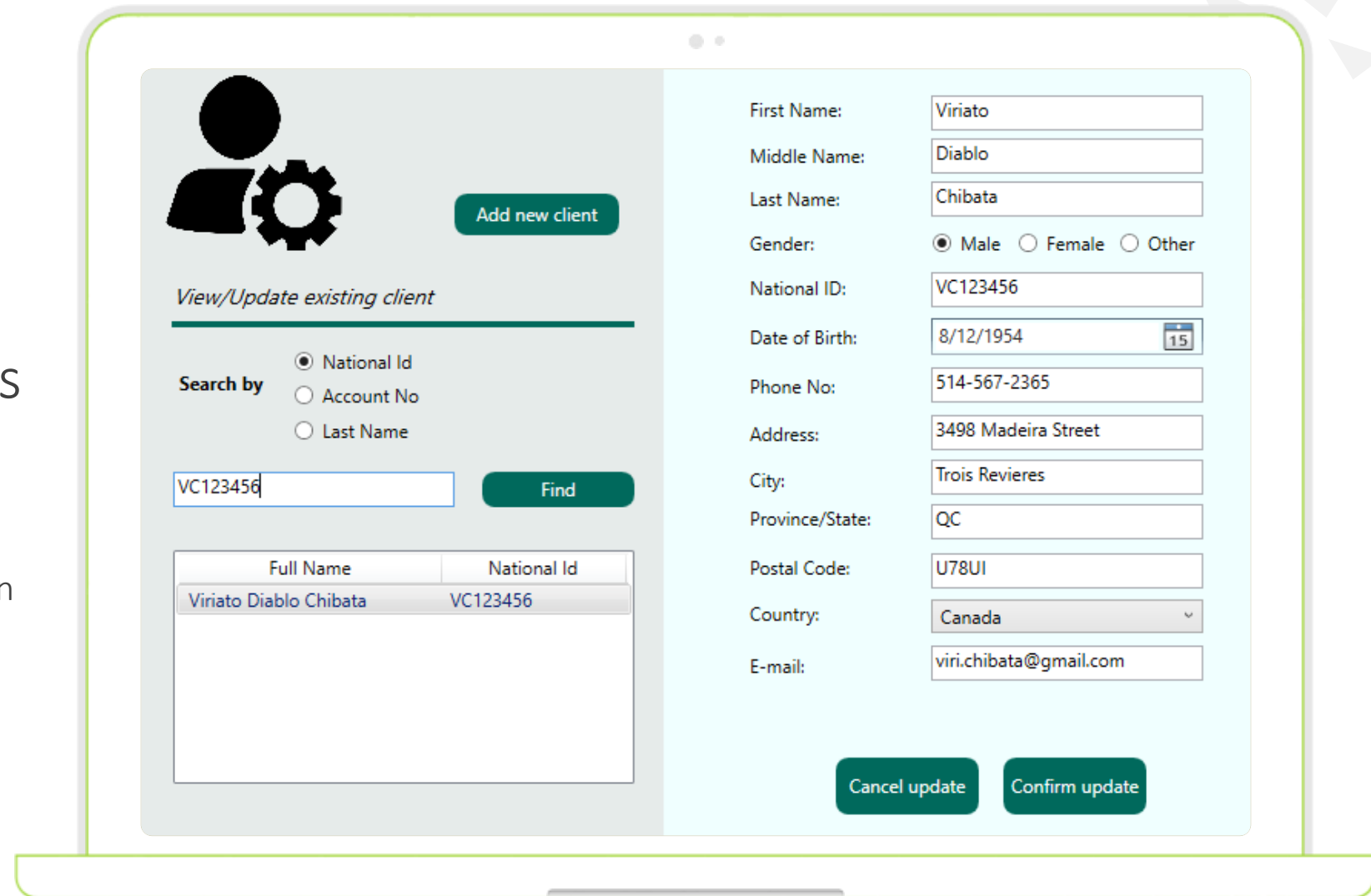
## Personal banking

- 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



The image shows a web application interface for managing customers. It is divided into two main sections: a left sidebar for navigation and a main content area for data entry and search.

**Left Sidebar:**

- Icon of a person and a gear.
- Button: **Add new client**
- Section: *View/Update existing client*
- Search by options:
  - ☒ National Id
  - ☐ Account No
  - ☐ Last Name
- Search input field containing **VC123456**
- Button: **Find**
- Table showing search results:

Full Name	National Id
Viriato Diablo Chibata	VC123456

**Main Content Area:**

Form fields for client information:

- First Name:
- Middle Name:
- Last Name:
- Gender: ☒ Male ☐ Female ☐ Other
- National ID:
- Date of Birth:  (Calendar icon)
- Phone No:
- Address:
- City:
- Province/State:
- Postal Code:
- Country:  (Dropdown arrow)
- E-mail:

Buttons at the bottom right: **Cancel update** and **Confirm update**

# Bank Management System Functionality Overview

## Managing accounts and transactions

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

The screenshot displays a web application interface for a bank management system. On the left, a panel titled 'Recent account transactions:' shows a table of transactions for the last 7 days. The table has columns for Transaction Type, Date, Amount, and Balance. The transactions include deposits, transfers, and a withdrawal. A green 'OK' button is at the bottom of this panel. On the right, a 'Proof of transaction' receipt is shown for a deposit of 500.00. The receipt includes the account number (25), account holder (Viriato Diablo Chibata), transaction number (300), agent number (2), transaction date (1/16/2021), and the receipt generation time (1/17/2021 12:36 PM). The receipt also shows the current balance as 0.00. At the bottom of the receipt are buttons for 'Send by Email' and 'Print'.

**Recent account transactions:** 7 days

☒ All ☐ Deposits ☐ Withdrawals ☐ Transfers ☐ Payments

Transaction Type	Date	Amount	Balance
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

**Proof of transaction** JAB

Transaction type: Deposit

Account number: 25

Account holder: Viriato Diablo Chibata

Transaction number: 300

---

Deposit amount: \$ 500.00

Current balance: \$ 0.00

Agent number: 2

Transaction date: 1/16/2021

Receipt generated on: 1/17/2021 12:36 PM

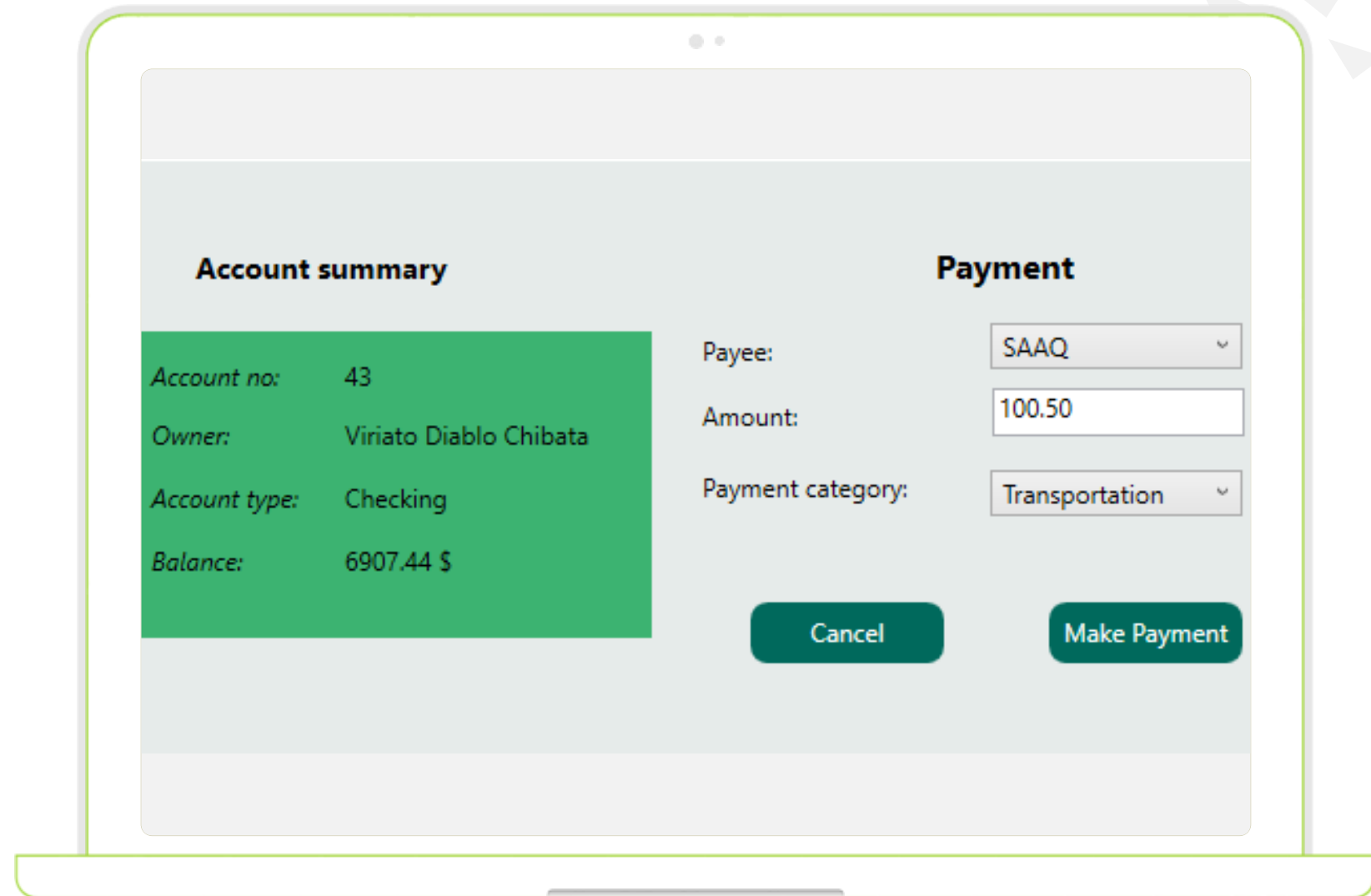
[Send by Email](#) [Print](#)



# Personal Bank Functionality Overview

## Perform account functions

- View balance
- Make transactions
- Generate receipts



The image shows a laptop screen with a web application interface for a personal bank. The interface is divided into two main sections: 'Account summary' and 'Payment'.

**Account summary**

Account no:	43
Owner:	Viriato Diablo Chibata
Account type:	Checking
Balance:	6907.44 \$

**Payment**

Payee: SAAQ

Amount: 100.50

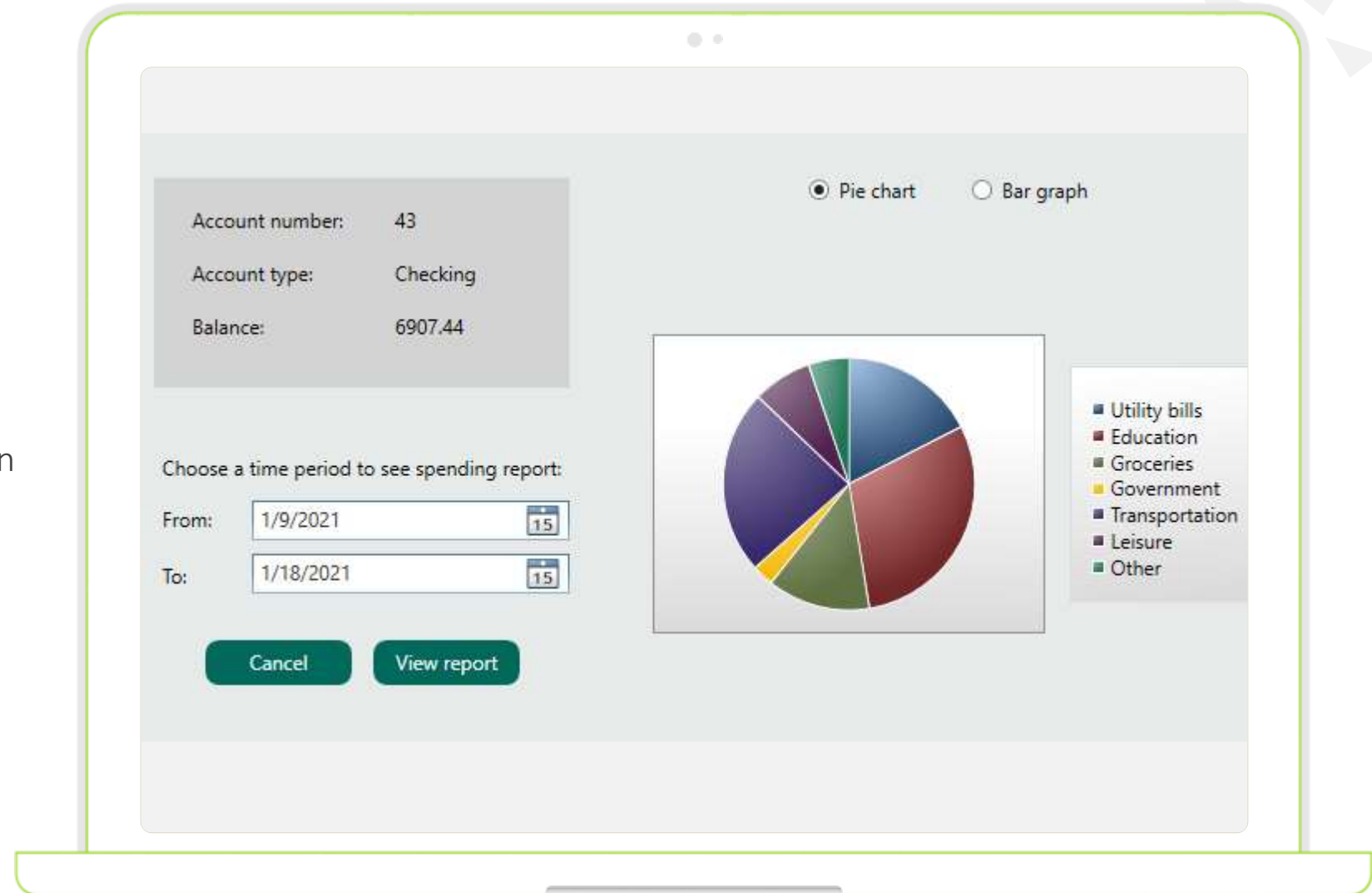
Payment category: Transportation

Buttons: Cancel, Make Payment

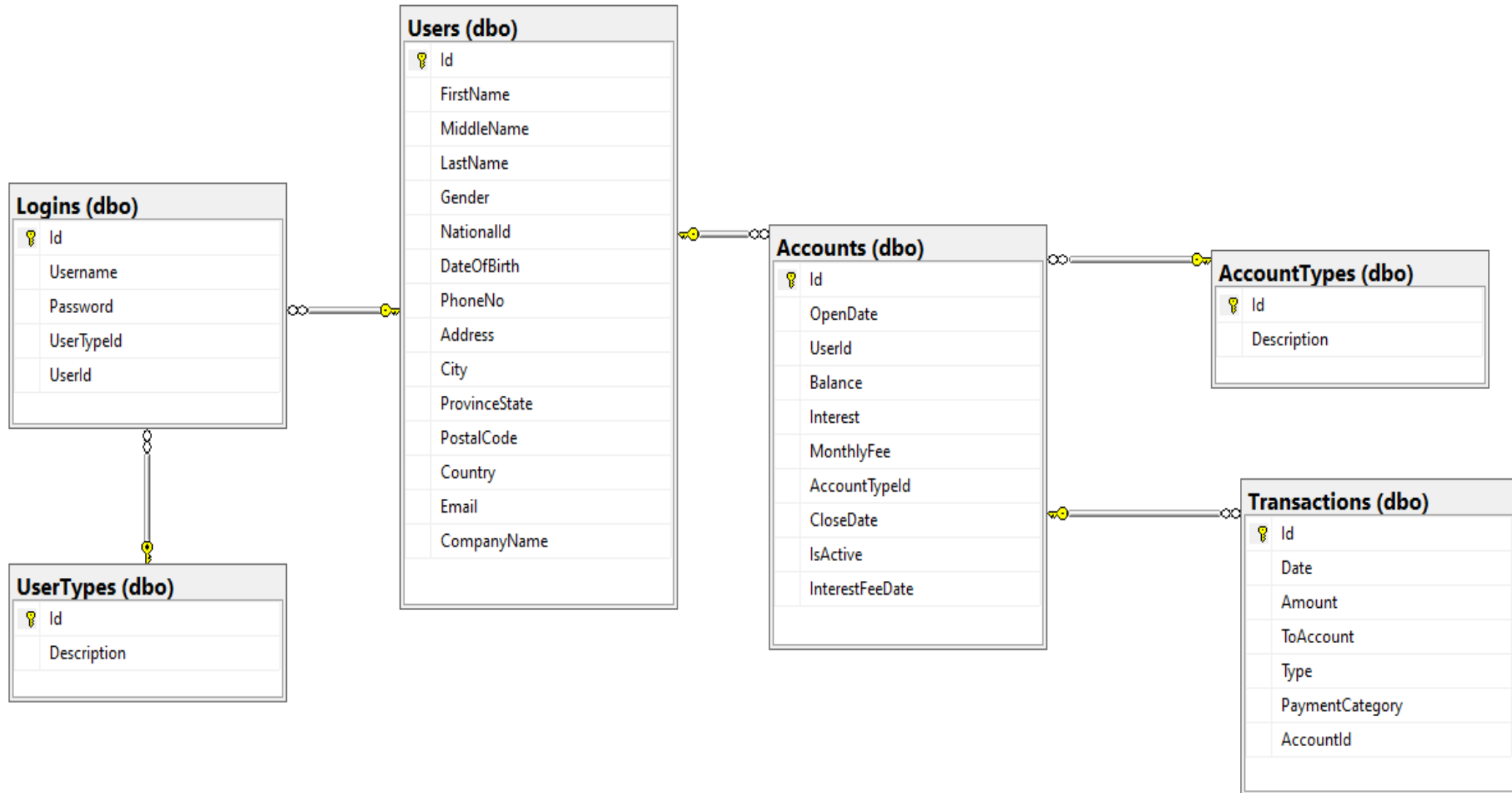
# Personal Bank Functionality Overview

## 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

**JAB**  
John Abbott Bank

### January 2021 Statement

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 ?



## Smtplib Class

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

Transaction receipt from 1/16/2021



johnabbottbank@gmail.com

johnabbottbank@gmail.com

Bank

receipt.pdf



PDF

Please see the [attached](#) receipt.

Thank you,

Bank

# 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

Add New Customer

Full name and Gender

First Name: \* John

Middle Name:

Last Name: \* Smith

Gender: \* ☒ Male ☐ Female ☐ Other

\* Mandatory fields

< Back Next > Cancel

# Challenges and Solutions



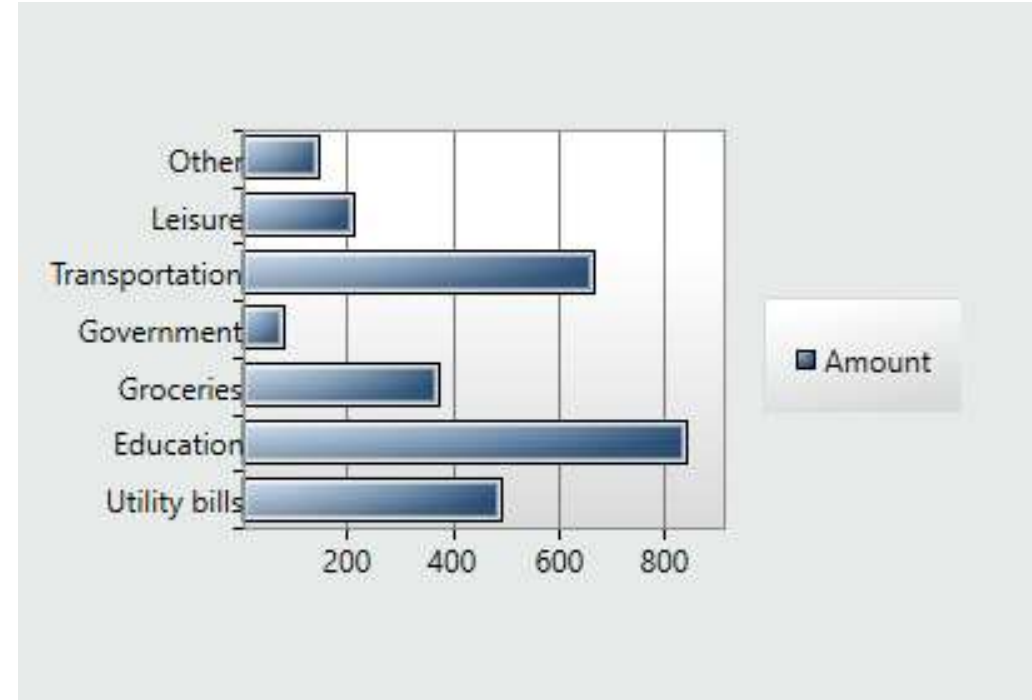
## Visualize data

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



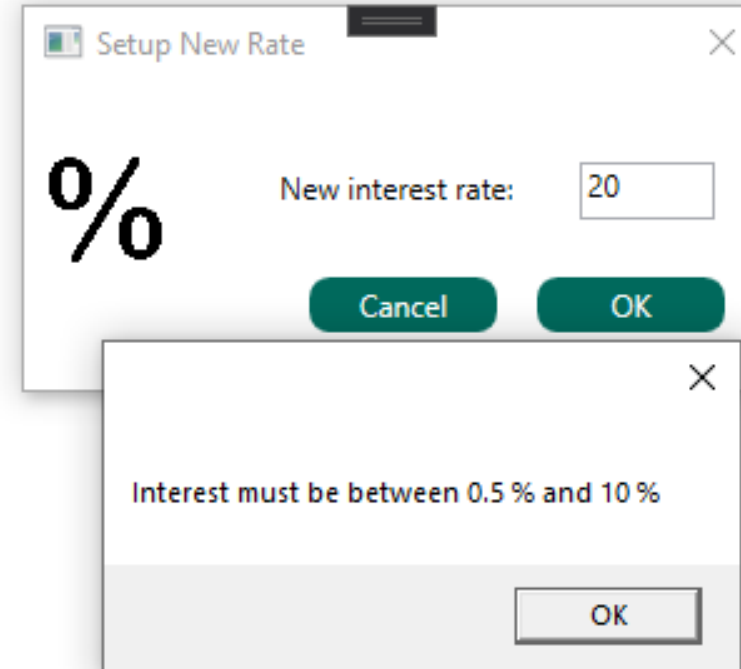
## Entities validation

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);
```



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;  
}
```

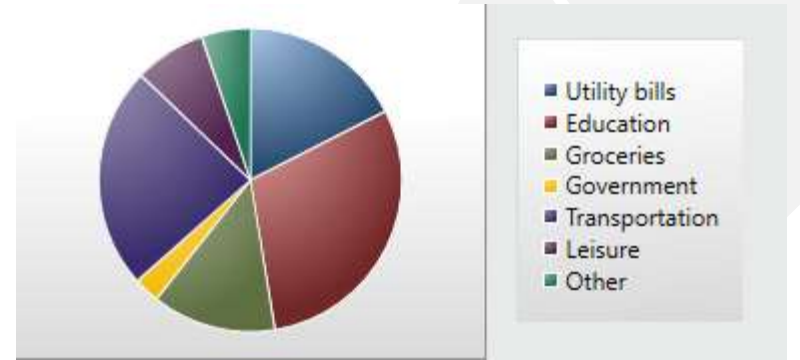


5. Draw string or line with the help of XGraphics and font

```
graphics.DrawString("Account Number: " + currentAccount.Id, fontBold, XBrushes.Black, 20, 45);  
graphics.DrawLine(lineRed, pt1, pt2);
```



# 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 += 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

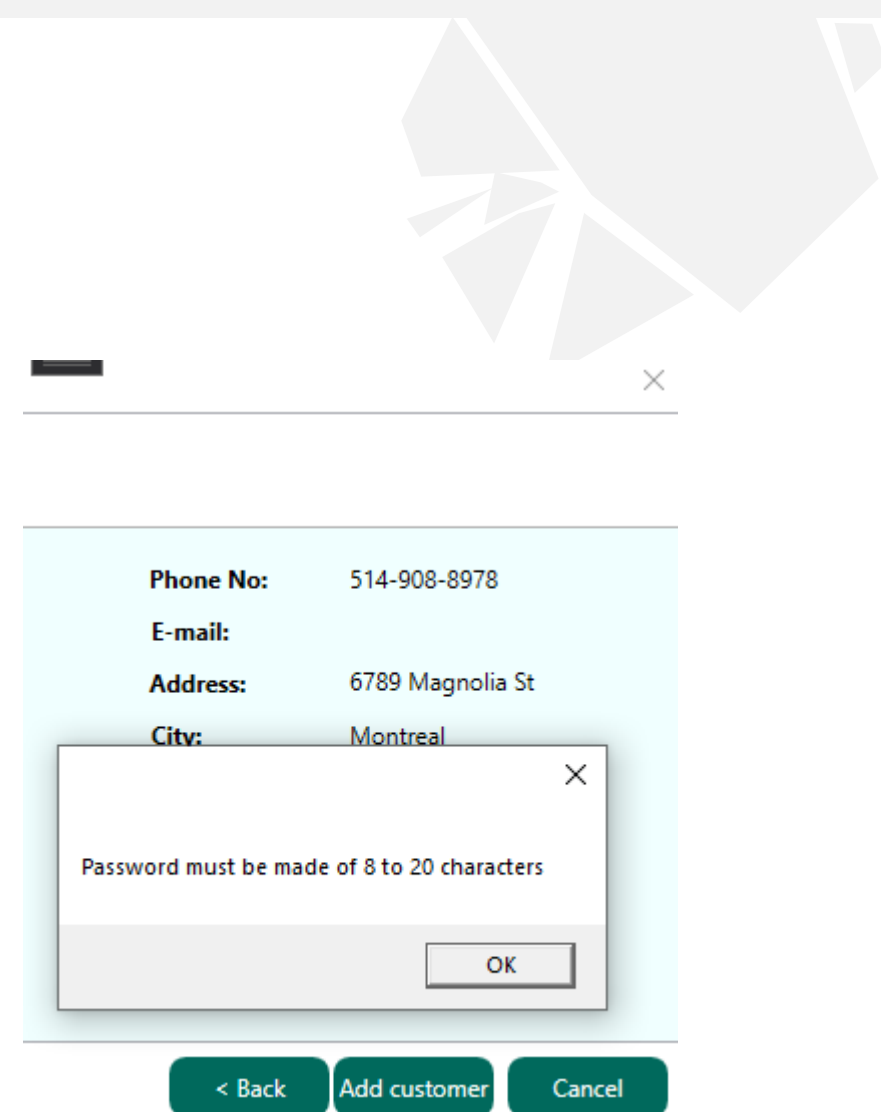
1. Add Ivalidatable Object Interface to class

```
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,
        UserId = 3,
        UserType = 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;
    }
}
```



The screenshot shows a user registration form with the following fields and values:

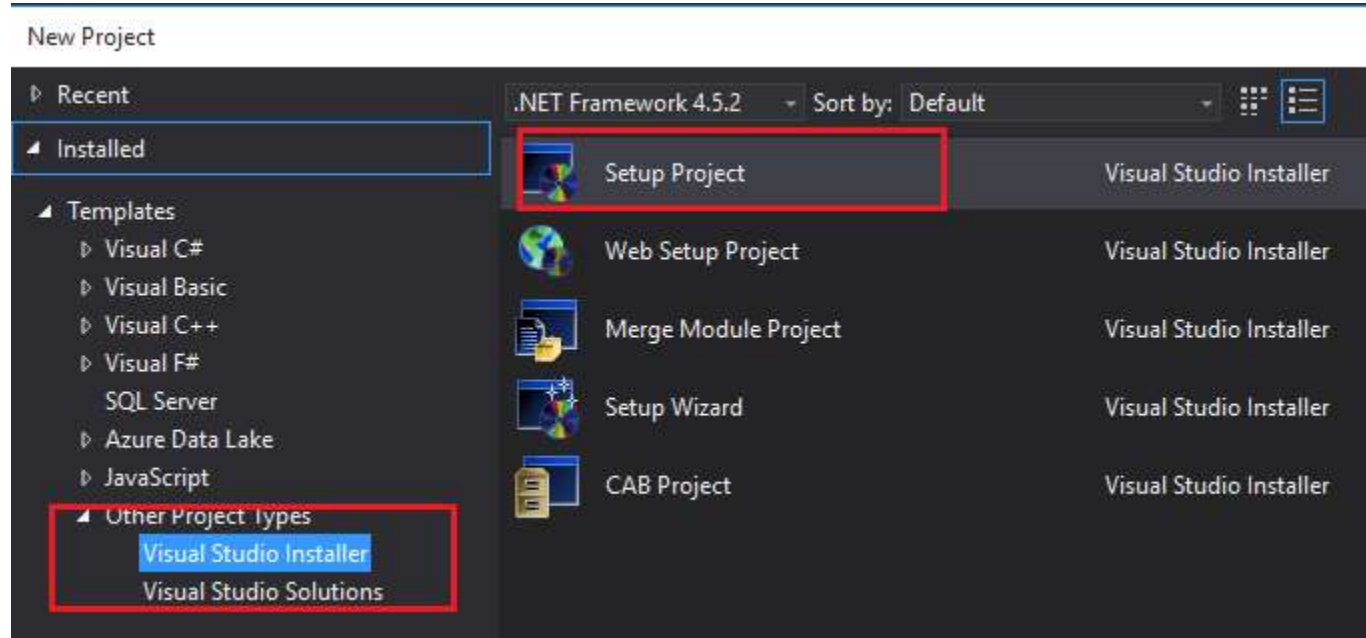
- Phone No: 514-908-8978
- E-mail: (empty)
- Address: 6789 Magnolia St
- City: Montreal

A modal dialog box is displayed in the foreground with the message: "Password must be made of 8 to 20 characters". The dialog has an "OK" button.

At the bottom of the form, there are three buttons: "< Back", "Add customer", and "Cancel".

# 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, Dividends



## **Loans and Credit**

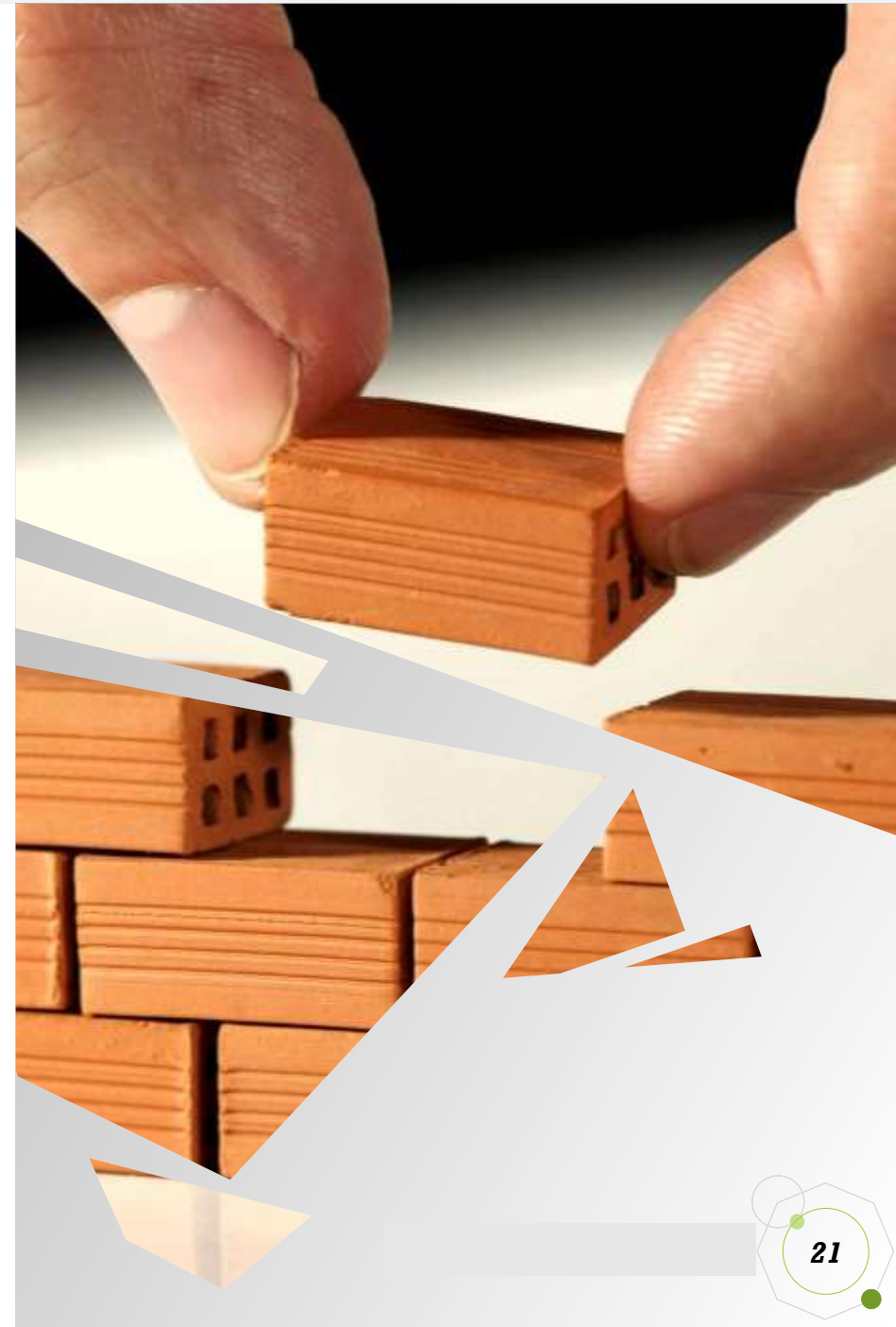
Issue loans and credit  
cards

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

A photograph of a man and a woman shaking hands with a banker in a modern office setting. The man is wearing a striped sweater and the woman is wearing a light blue shirt. The banker is wearing a light blue shirt and a patterned tie. They are all smiling and looking at each other. A semi-transparent dark banner is overlaid across the middle of the image, containing the text 'Thank You' and the JAB logo.

# Thank You

**JAB**  
John Abbott Bank®