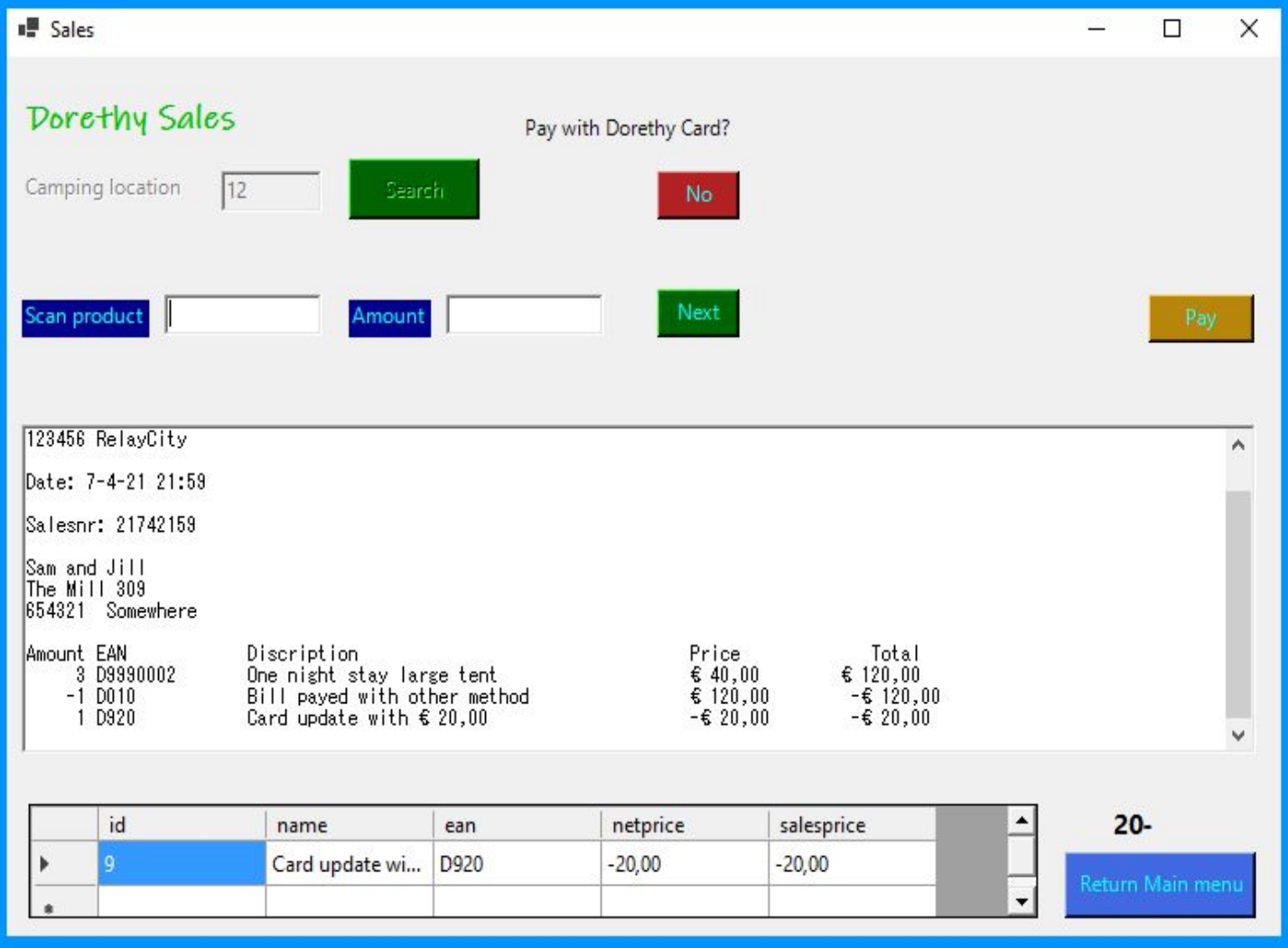
Dorethy,

Ideas in Visual Basic

Design a mini payment system

Source code free available via Nuget



DORETHY, IDEAS IN VISUAL BASIC

by

Johan van Hal

Introduction & Legal

It was for nearly 40 years I was introduced to computers. The company I was hired by, bought its first mainframe. It was an IBM which was nearly as big as the room where is was installed. Along with that computer, IBM delivered a huge amount of books with in. It was allowed by the company to learn to make yourself confident with the system. For me the system felt logical and in no time I was confident with it. Soon I concluded that computers would be the tool of tomorrow. That's how I started with my first personal computer. After following a course in machine learning for the 65xx processor, my first software program was published in a computer magazine. So computers and programming always kept my interest. In my work I've had a lot of advantage in the knowledge I built. Writing this book has two reasons. First of all it's a present for the youngest member of our family. He cleared his VWO exam (which is a pre academic scholar). And after the summer vacation he start with his academic education program. Years ago I told him about my published software. He was so curious about it that I promised him to write a book that encourage people to free up there mind and write a program. The time is now to publish this book an I truly hope that with the information in this book, you'll start with programming yourself. Step by step I'll guide you to the program Dorethy. Dorethy is a fictive person with a fictive camping site. Dorethy needs a payment system for her camping shop and she made it herself. Enjoy!

Legal

To complete the software you'll need some third party software. This software needs to be installed on your personal computer. All the software used is freeware and or public domain. Respect the terms and conditions! If you want to make a profit of the programs you make or even with selling "Dorethy" you need to buy licenses! Contact the publishers in this case.

The software Dorethy is based on:

- Microsoft Visual Studio community edition
- PostgreSQL
- Nuget UIDtoKeyBoard
- Nuget Fastreports community

About the name Dorethy. The name Dorethy is written as a Dutch girls name.

June 2021/January 2022

Index

Dorethy basics

As You've read in the introduction, Dorethy is a fictive person with a fictive camping site. So from now, I'll only talk about Dorethy and her camping. Probably you've asked yourself why would Dorethy make her own payment software, since there are so many payment providers who will solve her problem. You're probably right about that, but each payment will cost you money. That's the reason Dorethy want to do it different. She has a small camping site with a shop near the entry. She buys fruits, vegetables and milk form local farmers. Bread she buys from the local bakery who gets his flower from the local miller. In her shop she has a lot of products, but the price per item is pretty low. Sometimes she lose money, because the fee she has to pay for digital payment is higher than the profit on the product. So Dorethy was thinking about her time at school and the additional lessons she followed in computer science. She got her notebook and wrote down everything she could think of she would need. Her wish list is as follows:

Easy to learn programming software;

A SQL database, so she could expand her business when needed;

A reporting tool;

RFID reader and badges for the payments;

A barcode scanner for the shop.

Now her wish list was complete but where should she start? Her choice for the programming software was the most easy step. During her additional computer science lessons, she had some experience with Visual Studio. For this project she chose for Visual basic. But what should be the next step? Her free time she spent on learning the Visual Basic language. While learning the language she was already making some steps in her software development

project. Although her learning curve was pretty steep, she wasn't convinced about successfully completing her goal. Somewhere in the process she must have made a mistake. On an evening she took her wish list and looked over and over it. What she did not oversee? The software was the most important part of the process? And just when she got in bed and tried to sleep, she asked herself is the software the most important part? The next she kept asking herself over and over. Is the software the most important part? Later that day one of the local farmers delivered some fruit and asked Dorethy to sign the delivery list. And the moment she put her signature, she looked at the delivery list and say. This list is impossible the make, without a proper database. I never can develop a software program if I have no idea which data I need and how it is stored. From that moment the development of the database was the only thing she could think of.

Dorethy database

Now Dorethy was convinced on the fact that the development was her top priority. But she must keep in mind that she has customers who stayed at her camping ground for one or more days. Does the customer want a Dorethy payment card and how do I keep track on the sales and the debit amount? I need information of all the products I sell. Then I need to protect my data and limit the users. There is a lot to think of. That is why she decided to make a wish list for the database.

Later on in this chapter we build the database with PostgreSQL. So if you're not familiar with the terms used, don't worry. You'll learn it soon.

Her wish list for the database is as follows:

The table users, which keeps track on the users, there login and authority;

The table products, in which she could add products she sales, with the pricing and VAT;

The table sales, in which she could store all the sales per card and payments;

The table cards, in which she could store the card number and the active user of the card;

The table customer, in which she could store the visitor information;

The table payments, in which we store the way of payment and the actual amount of the Dorethy nfc.

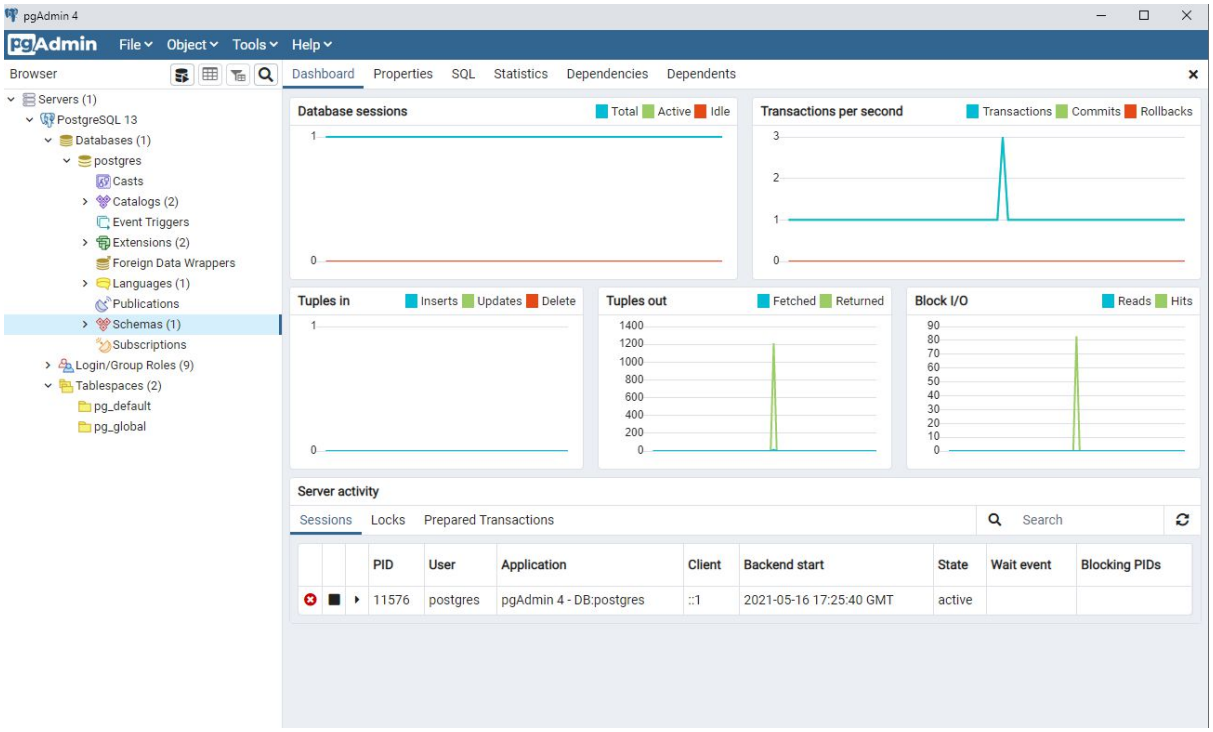
Now it's time for you to download PostgreSQL. The link to the software is:

https://www.postgresql.org/download/

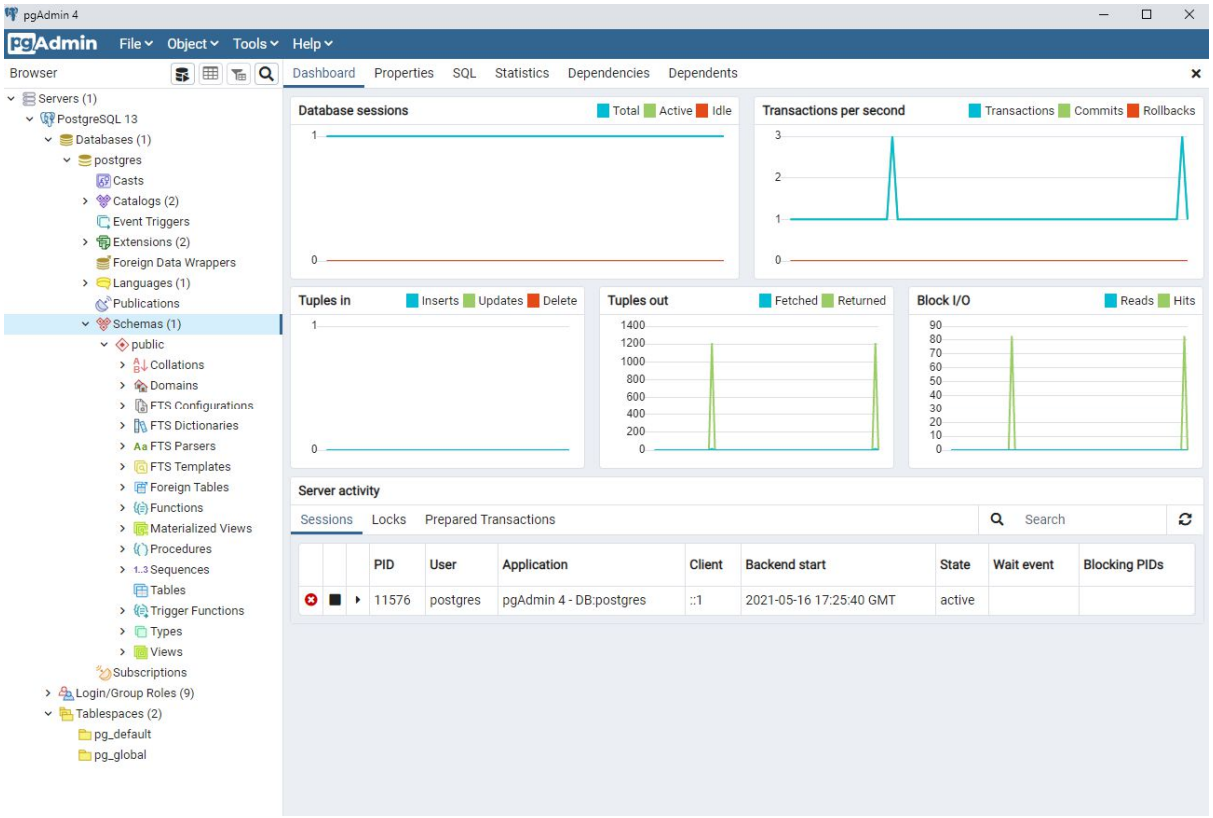
Follow the instructions as shown on the screen and install the software. During the installation you will be asked to enter a password. **Remember the password you entered!** You'll need it several of times. In Dorethy we use the password 4Dor=Access!

Start the program pgAdmin. During the startup of the software you will be asked for the first time to enter the password of the database. If you have used another password the we did in this example, please enter your password. If you used the same we used enter 4Dor=Access!

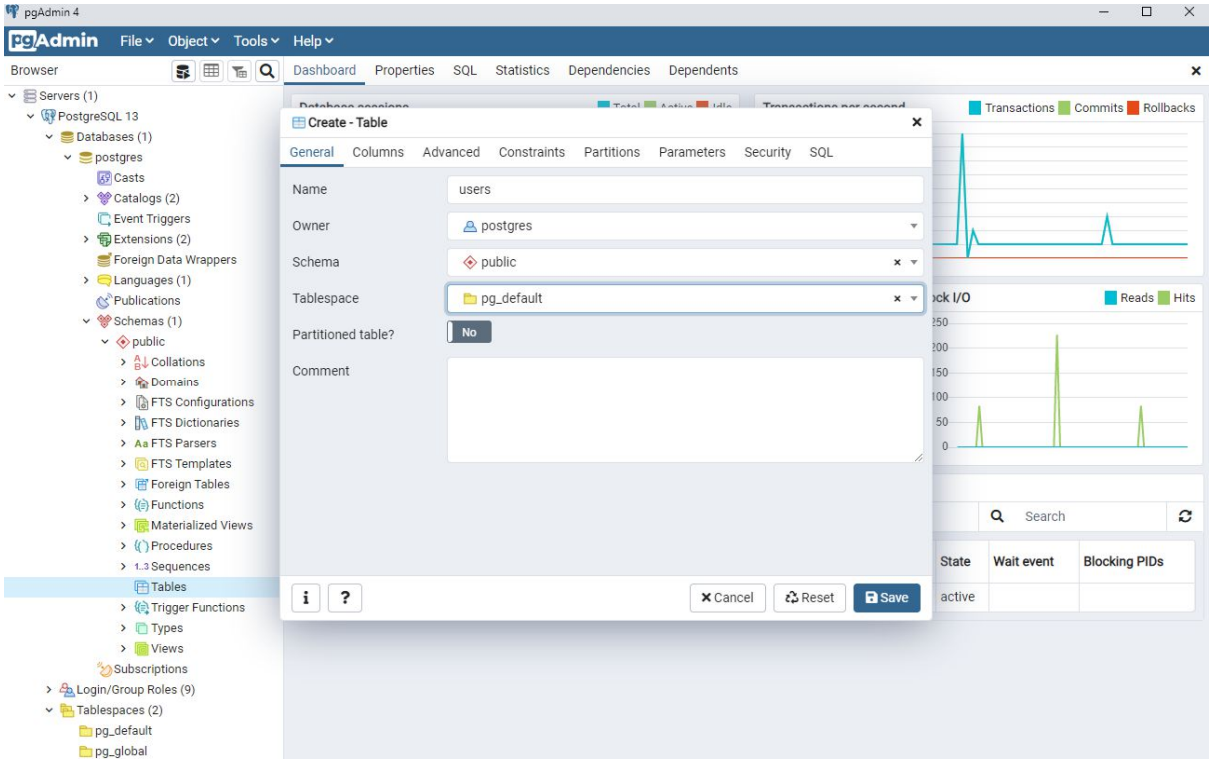
Your screen looks like:

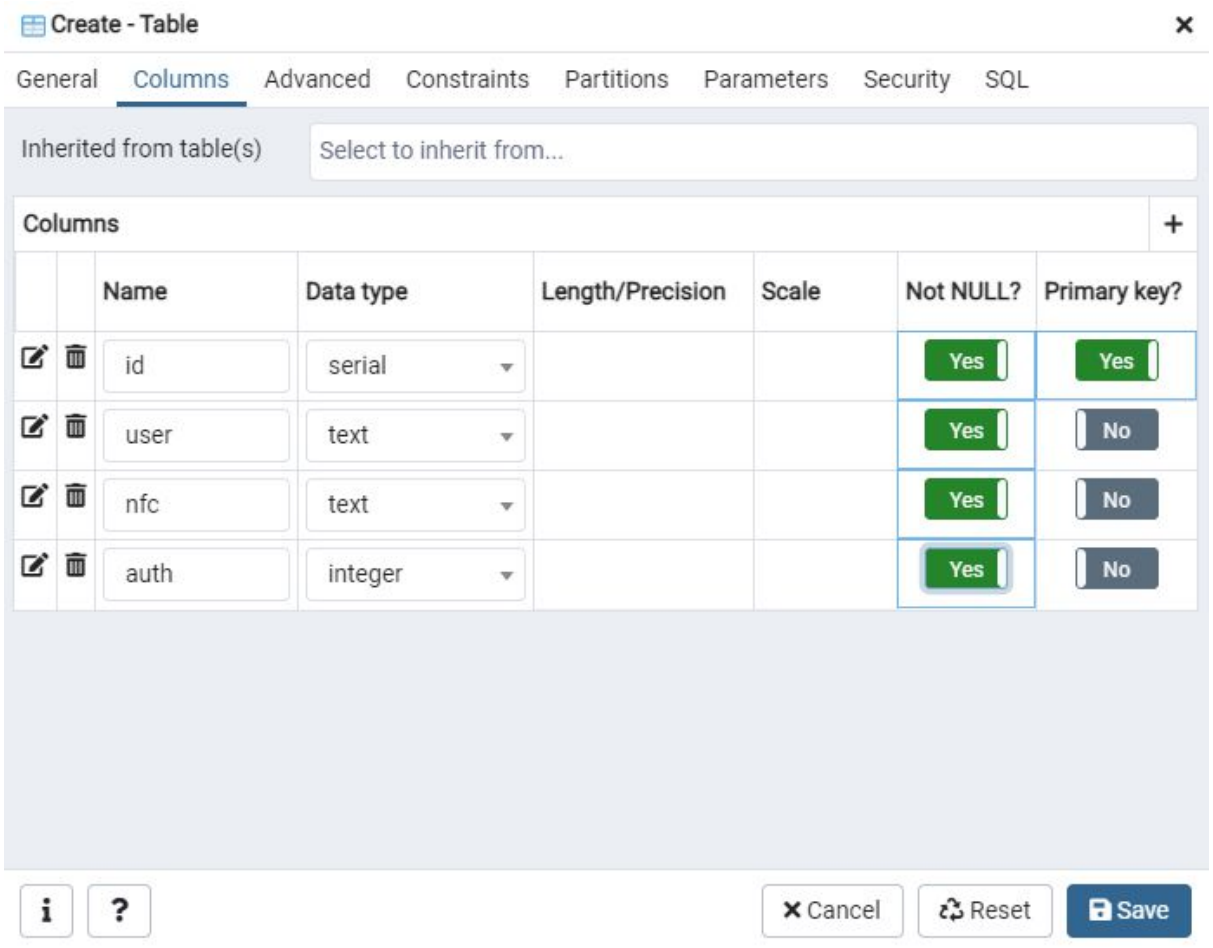


Dorethy used the database created during the installation of the postGres database. Click on the PostgreSQL and postgres to expand the tree. Click on Schemas and the tree under Schemas expands:



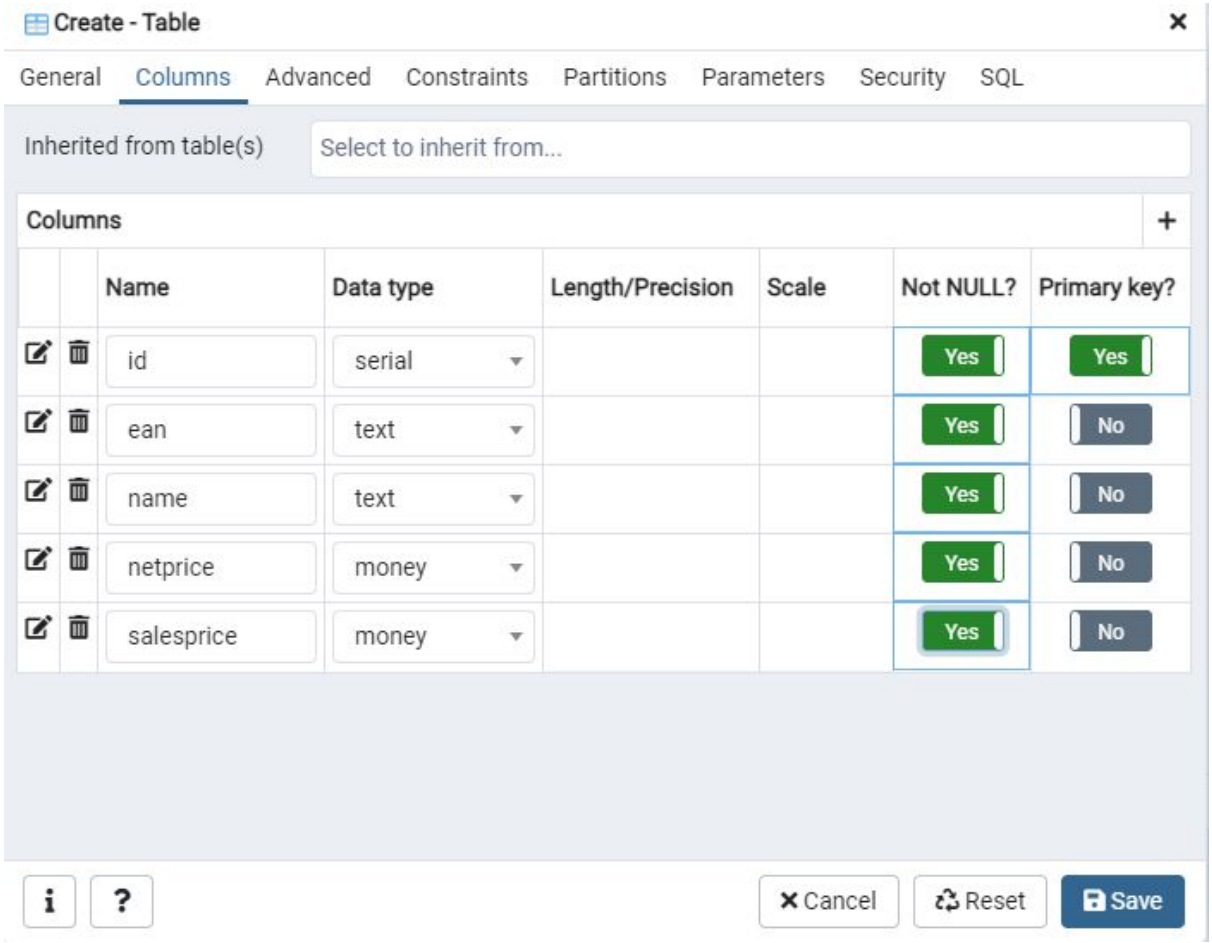
At Tables we are going to add the tables we are going to use in this example. Move your mouse to Tables, click on the right button and select Create Table. The first table we are going to enter is users. We will store it in pg_default under Table space. Don't enter Save, because we will first add the Columns in the table. Click on Columns and enter the information as shown on the next page.


pgAdmin 4
pgAdmin
File
Object
Tools
Help
Browser
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Servers (1)
PostgreSQL 13
Databases (1)
postgres
Casts
Catalogs (2)
Event Triggers
Extensions (2)
Foreign Data Wrappers
Languages (1)
Publications
Schemas (1)
public
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Procedures
Sequences
Tables
Trigger Functions
Types
Views
Subscriptions
Login/Group Roles (9)
Tablespaces (2)
pg_default
pg_global
Create - Table
General
Columns
Advanced
Constraints
Partitions
Parameters
Security
SQL
Name
users
Owner
postgres
Schema
public
Tablespace
pg_default
Partitioned table?
No
Comment
Cancel
Reset
Save
Transactions
Commits
Rollbacks
Reads
Hits
Search
State
Wait event
Blocking PIDs
active

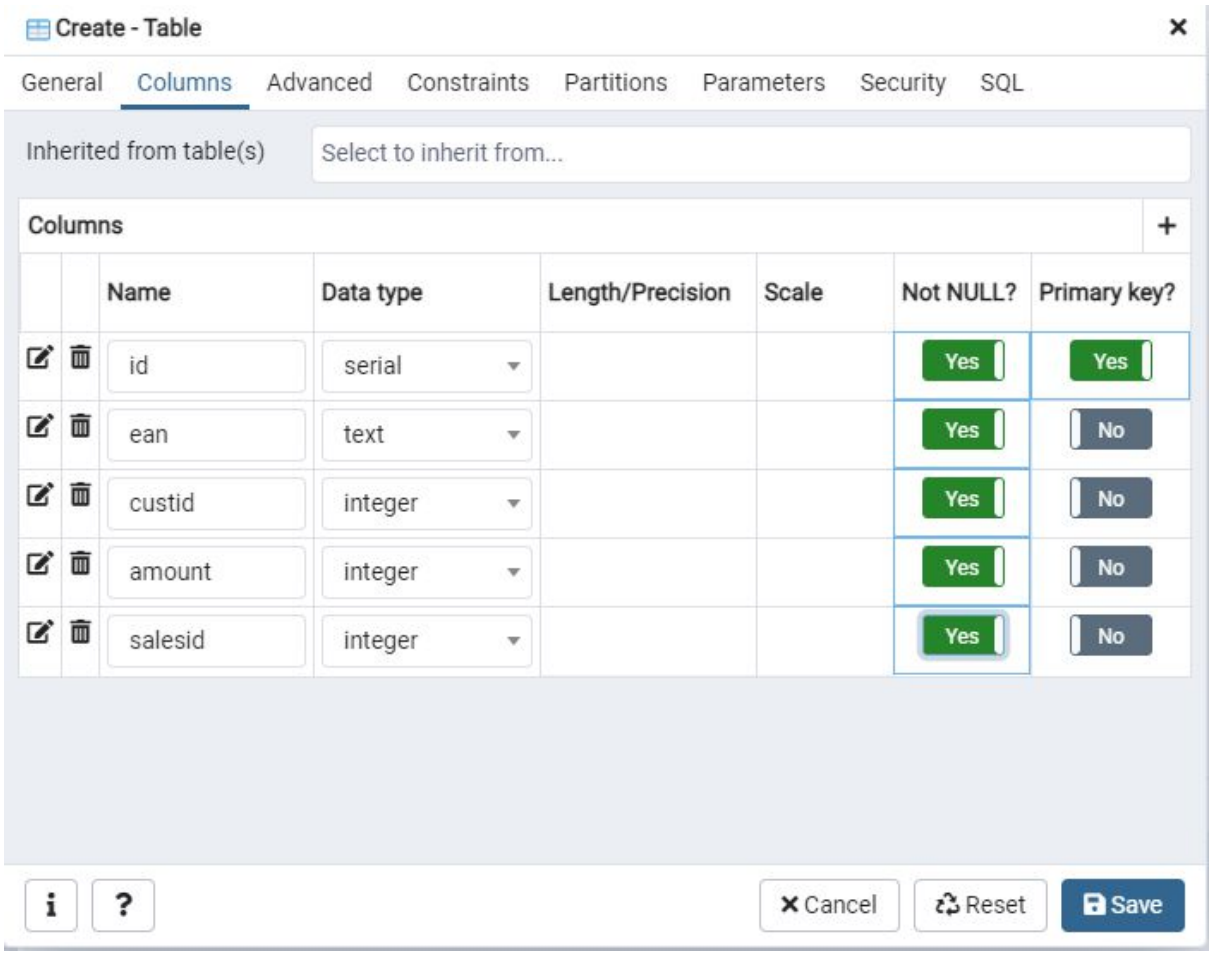

Create - Table
General
Columns
Advanced
Constraints
Partitions
Parameters
Security
SQL
Inherited from table(s)
Select to inherit from...
Columns
Name
Data type
Length/Precision
Scale
Not NULL?
Primary key?
id
serial
Yes
Yes
user
text
Yes
No
nfc
text
Yes
No
auth
integer
Yes
No
Cancel
Reset
Save

Enter Save to store this table. During the creation of the software it will be clear where the columns will be used for.

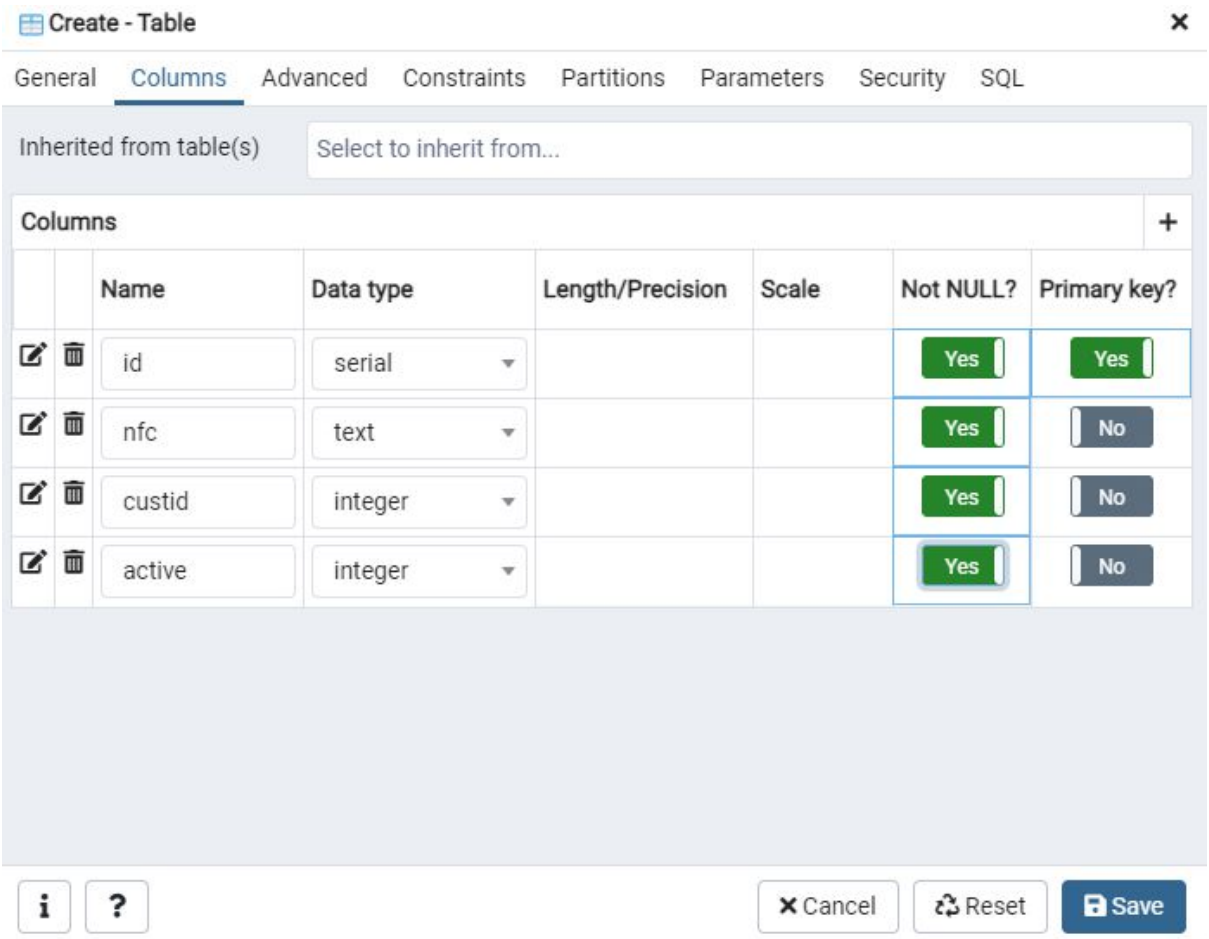
Now we do the same for the tables products, sales, cards, customer and payments. The columns for the table products:



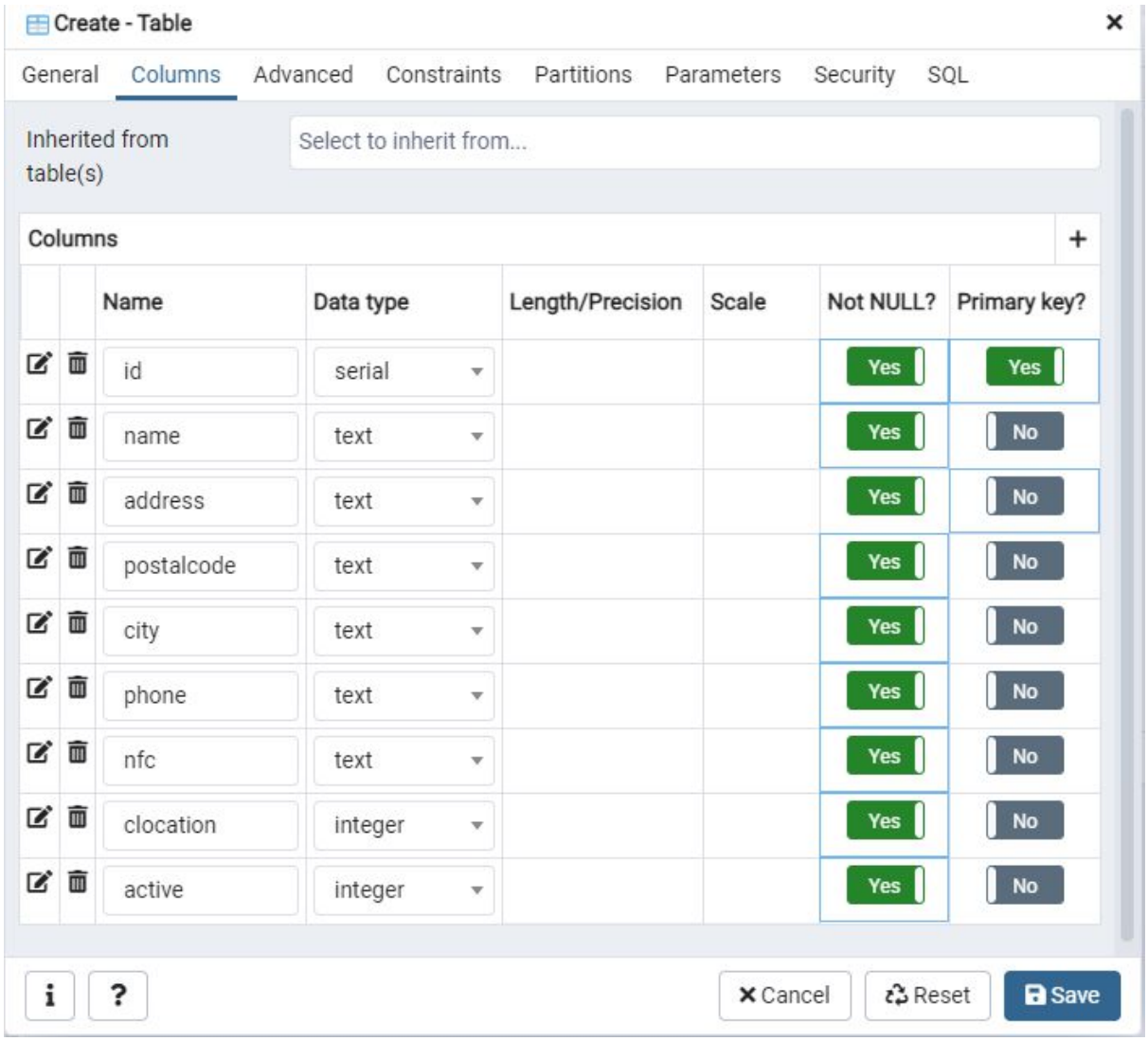
The columns for the table sales:



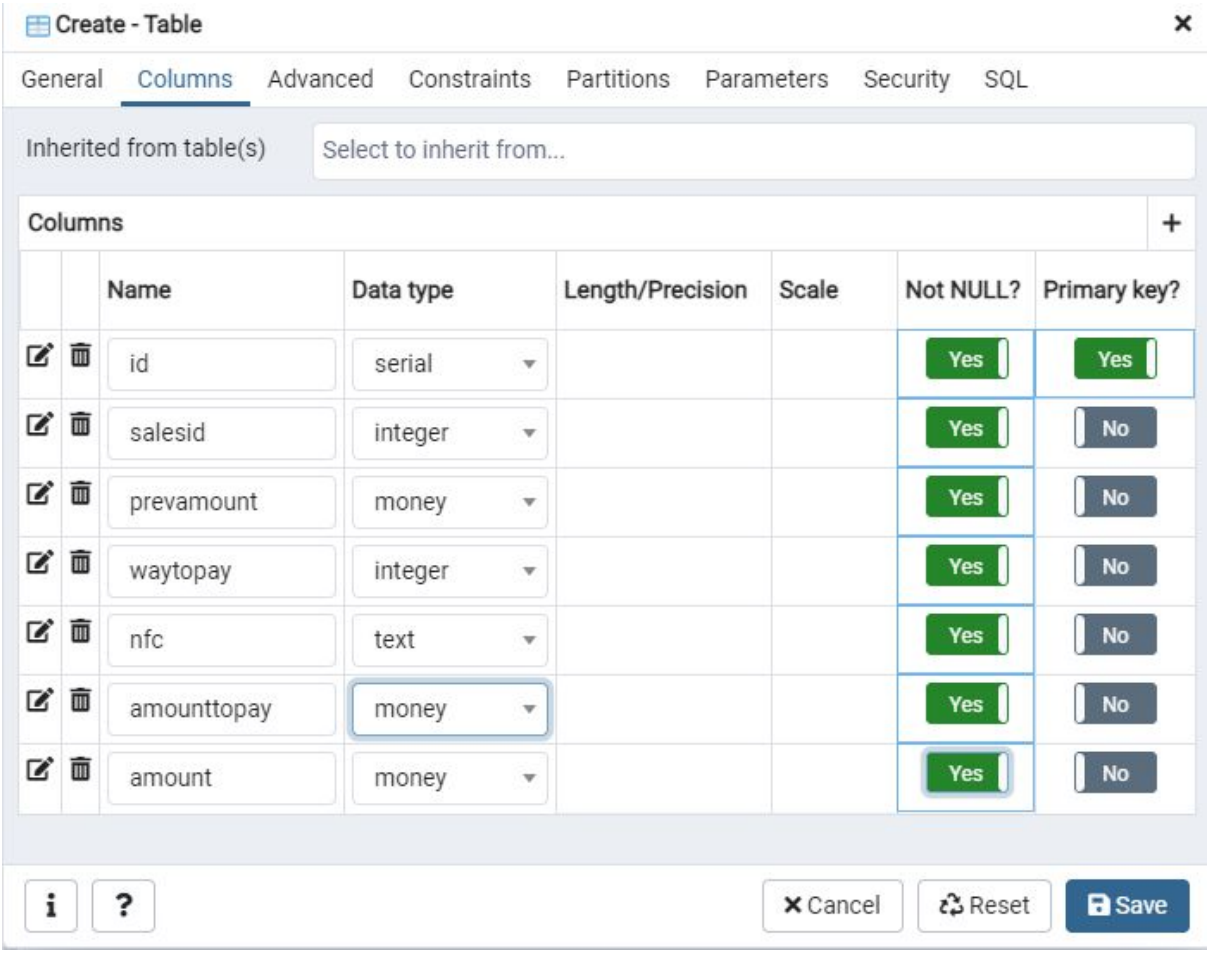
The columns for the table cards:



The columns for the table customer:



The columns for the table payments:



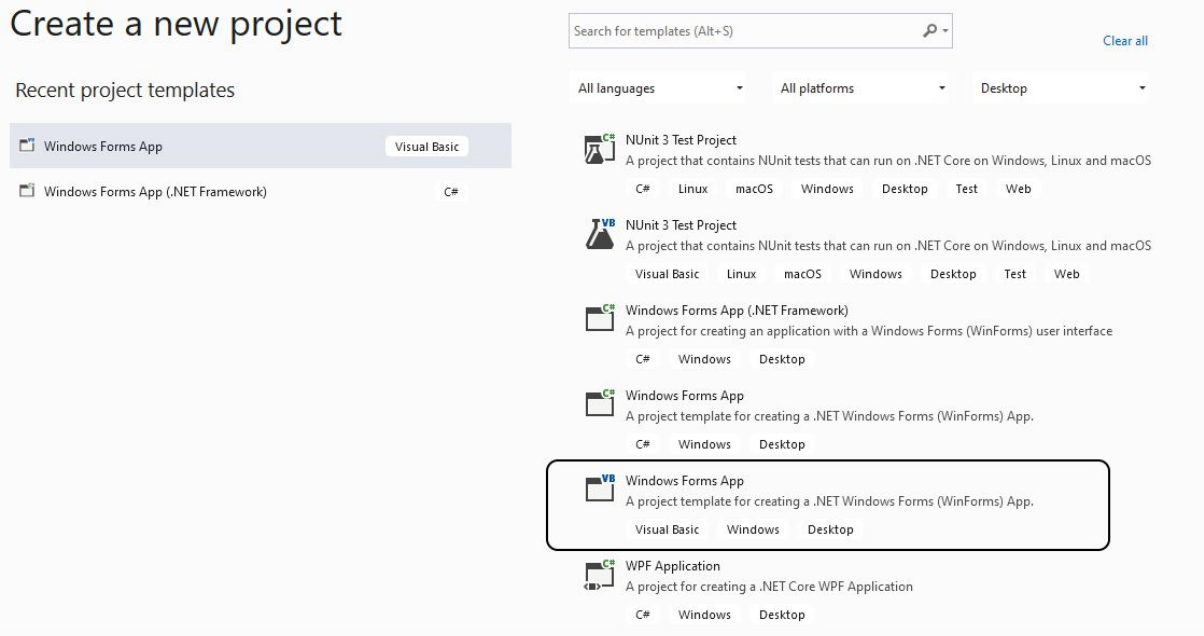
You have successfully created the Dorethy database. In the next chapter we'll start with the development of the software Dorethy has in mind.

Dorethy the software

Now the time has come to explain the software program Dorethy has in mind. If you are familiar with Visual Basic, you can rename every single item so it will fit your purpose. If you are not that familiar with Visual Basic. We suggest you don't change anything, until you're more familiar with Visual Basic.

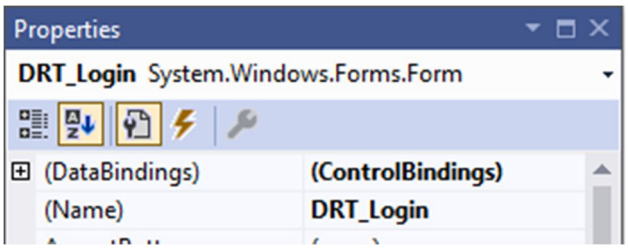
As mentioned before, the software depends on some GitHub code. The reason is, that we use a AR122U rfid reader. For Dorethy she decided to make use of the available VB.net code named UIDtoKeybord. We suggest you download it now form GitHub, connect your RFID reader and make yourself familiar with it.

We start with the setup of the programming environment. Run your Visual Studio program and click on "Create a new project".

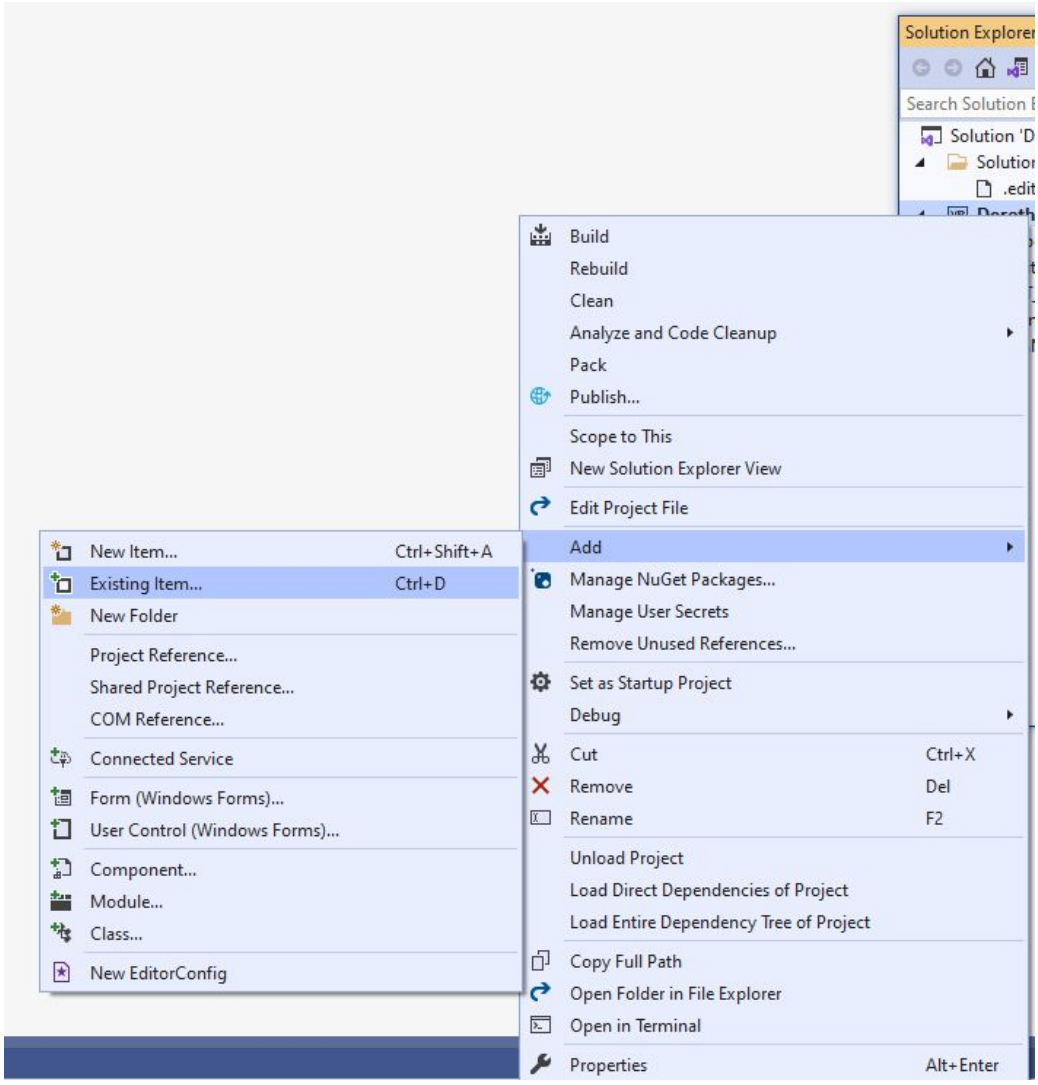


And chose Windows Forms App Visual Basic. Give your project a name. We used the name Dorethy for it.

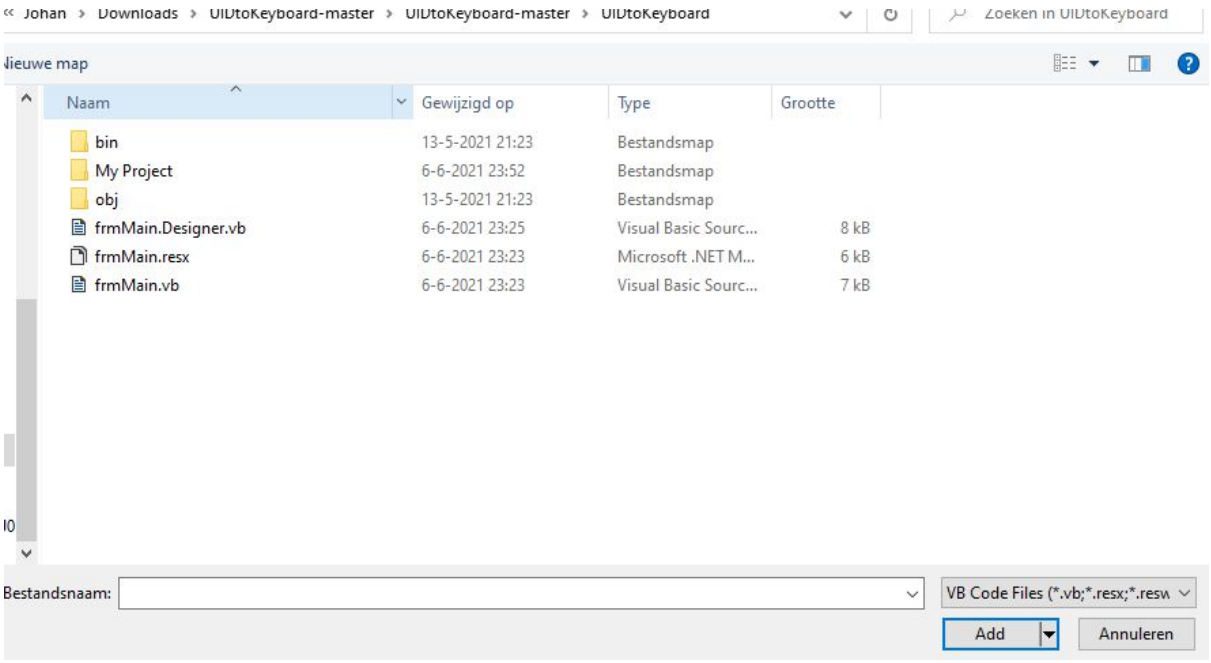
The system automatically creates Form1.vb. We are going to rename it as DRT_Login. Click somewhere on the form with your mouse and change the name under (Name).



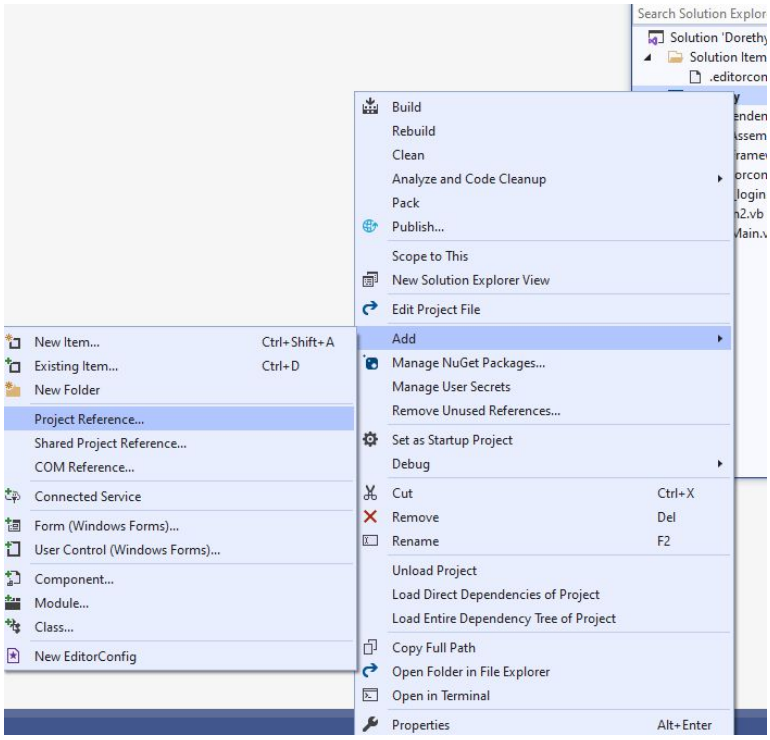
The next thing we do is to add the downloaded GitHub UIDtoKeybord solution to our project. On the Solution Explorer, move your mouse to Solution ‘Dorethy’ and click on the right button. Select Add>>Existing Item..



Go to the folder where the UIDtoKeyboard is stored and select the *.vb and *.resx files and click Add.

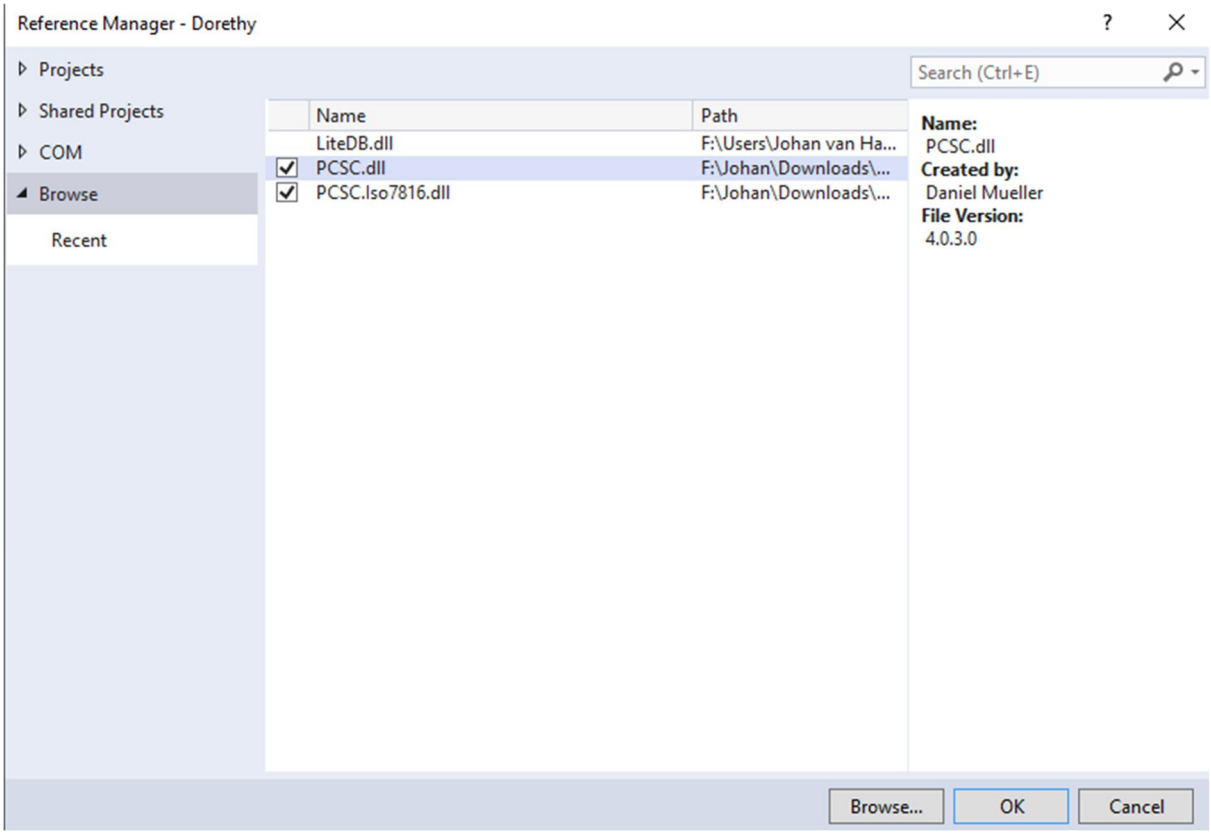


The UIDtoKeyboard depends on 2 DLL's. We have to add these DLL's to the Dependencies of this project. In the Solution Explorer move to Dorethy and right click your mouse. Select Add >> Project reference

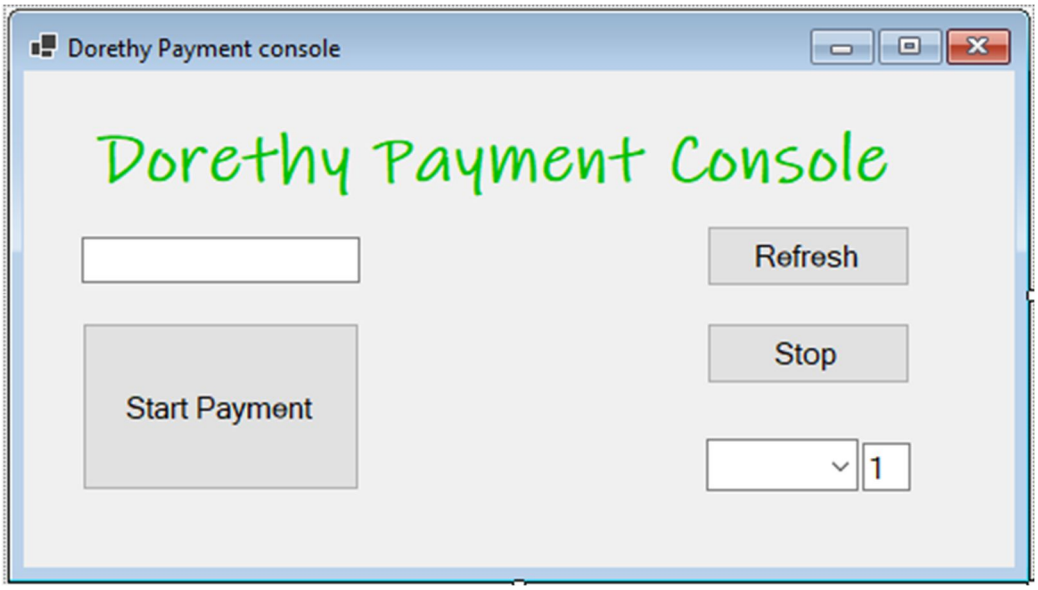


Browse to the location where your UIDtoKeyboard download is stored and select both DLL files.

After adding you have to select both DLL's in your reference manager as shown below.



Click on OK. Since Dorethy has tested the UIDtoKeyboard solution before, she decided that it has to be changed before she could use it in her Payment solution. But she decided after several trials that the main part of the software will remain. First she changed the original interface so it would be easy to intergrade it with the Dorethy solution.



First all text boxes and labels are removed from this form. The next items are changed:

Label1 is added and renamed. The name Label1 is changed to Dorethy_brand in the Proporties items. The font is changed as Ink Free; 26.25 pt; style=Bold. The ForeColor is changed to 0;192;0. The text is changed to Dorethy Payment Console.

The **txtInputspace** is changed in size and the MaxLength is changed to 8.

The **btnStartMonitor** is changed in size and the text is changed in Start Payment.

The **btnRefreshReader** is changed in size and is disabled for use. (Enable=false)

The **btnStopMonitor** is changed in size.

The **cbxReaderList** is changed in size.

The **TxtReadingMode** is changed in size and the text is changed in 1. The reason for this is that Dorethy has decided to use this reading mode in her project.

These changes has effect in the underlying software. The changed version is:

frmMain:

```
Imports PCSC
Imports PCSC.Iso7816
Imports PCSC.Monitoring
Imports System.Threading


<DebuggerDisplay("{GetDebuggerDisplay(),nq}")>
Public Class FrmMain
    Private Shared ReadOnly _contextFactory As IContextFactory =
ContextFactory.Instance
    Private _hContext As ISCardContext
    Dim readerName As String
    Dim readingMode As String
    Dim VoTis As String
    Dim isstart As Boolean = False

    Function LoadReaderList()
        Dim readerList As String()
        Try
            cbxReaderList.DataSource = Nothing

            _hContext = _contextFactory.Establish(SCardScope.System)
            readerList = _hContext.GetReaders()
            _hContext.Release()

            If readerList.Length > 0 Then
                cbxReaderList.DataSource = readerList
            Else
                MessageBox.Show("No card reader detected!", "Message",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
            End If

            Return True
        Catch ex As Exceptions.PCSCException
            MessageBox.Show("Error: getReaderList() : " & ex.Message & " (" &
ex.SCardError.ToString() & ")")
            Return False
        End Try

    End Function

    Dim monitor

    Private Sub StartMonitor()
        Dim monitorFactory As MonitorFactory = MonitorFactory.Instance
        monitor = monitorFactory.Create(SCardScope.System)
        AttachToAllEvents(monitor)
        monitor.Start(cbxReaderList.Text)

        readerName = cbxReaderList.Text
        readingMode = txtReadingMode.Text

    End Sub

    Private Sub AttachToAllEvents(monitor As ISCardMonitor)
        AddHandler monitor.CardInserted, AddressOf CardInit
    End Sub
```

```
    Sub CardInit(eventName As SCardMonitor, unknown As CardStatusEventArgs)
        If readingMode = 1 OrElse readingMode = 2 Then
            SendUID4Byte()
        ElseIf readingMode = 3 OrElse readingMode = 4 Then
            SendUID7Byte()
        End If
    End Sub

    Private Sub FrmMain_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        LoadReaderList()

    End Sub

    Private Sub BtnRefreshReader_Click(sender As Object, e As EventArgs) Handles
btnRefreshReader.Click
        LoadReaderList()
    End Sub

    Private Sub BtnStartMonitor_Click(sender As Object, e As EventArgs) Handles
btnStartMonitor.Click

        If txtReadingMode.Text <> 1 AndAlso txtReadingMode.Text <> 2 AndAlso
txtReadingMode.Text <> 3 AndAlso txtReadingMode.Text <> 4 Then
            MessageBox.Show("Error: Reading mode not macth the preset.")
        Else
            If isstart = True Then
                monitor.Cancel()
            End If
            StartMonitor()
            isstart = True
            TxtInputspace.Text = ""
            TxtInputspace.Focus()
        End If

    End Sub

    Private Sub BtnStopMonitor_Click(sender As Object, e As EventArgs) Handles
btnStopMonitor.Click
        If isstart = True Then
            monitor.Cancel()
            DRT_Login.GetFromRFID = TxtInputspace.Text
            DRT_Login.ReturnCodeRFID = "ok"
        ElseIf TxtInputspace.Text <> "" Then
            DRT_Login.GetFromRFID = TxtInputspace.Text
            DRT_Login.ReturnCodeRFID = "ok"
        End If
        TxtInputspace.Text = ""
        If DRT_Login.GetFromRFID = "" Then
            DRT_Login.ReturnCodeRFID = "canceled"
        End If

        If DRT_Login.ReturnTo = "1" Then
            DRT_Login.ActiveForm.Activate()
            DRT_Login.Show()
        End If
        If DRT_Login.ReturnTo = "2" Then
            AddNewUser.ActiveForm.Activate()
            AddNewUser.Show()
        End If
```

```
        If DRT_Login.ReturnTo = "3" Then
            Customer.ActiveForm.Activate()
            Customer.Show()
        End If
        Me.Close()

    End Sub

    Function SendUID4Byte()
        Try
            Using context = _contextFactory.Establish(SCardScope.System)
                Using rfidReader = context.ConnectReader(readerName,
SCardShareMode.Shared, SCardProtocol.Any)
                    Using rfidReader.Transaction(SCardReaderDisposition.Leave)

                        Dim apdu As Byte() = {&HFF, &HCA, &H0, &H0, &H4}
                        Dim sendPci = SCardPCI.GetPci(rfidReader.Protocol)
                        Dim receivePci = New SCardPCI()

                        Dim receiveBuffer = New Byte(255) {}
                        Dim command = apdu.ToArray()
                        Dim bytesReceived = rfidReader.Transmit(sendPci, command,
command.Length, receivePci, receiveBuffer, receiveBuffer.Length)
                        Dim responseApdu = New ResponseApdu(receiveBuffer,
bytesReceived, IsoCase.Case2Short, rfidReader.Protocol)

                        If readingMode = 1 Then
                            Dim uid As String =
BitConverter.ToString(responseApdu.GetData())
                            uid = uid.Replace("-", "")

                            SendKeys.SendWait(uid + "{ENTER}")
                        ElseIf readingMode = 2 Then
                            Dim uid As Byte() = New Byte(3) {}
                            Dim revuid As Byte() = New Byte(3) {}
                            Array.Copy(responseApdu.GetData(), uid, 4)
                            Array.Copy(uid, revuid, 4)
                            Array.Reverse(revuid, 0, 4)

                            Dim uid2 As String = BitConverter.ToString(revuid)
                            uid2 = uid2.Replace("-", "")

                            SendKeys.SendWait(uid2 + "{ENTER}")
                        End If
                    End Using
                End Using
            End Using
        Catch
            'Error Handling should be developed
        End Try

        Return True
    End Function

    Function SendUID7Byte()
        Try
            Using context = _contextFactory.Establish(SCardScope.System)
                Using rfidReader = context.ConnectReader(readerName,
SCardShareMode.Shared, SCardProtocol.Any)
                    Using rfidReader.Transaction(SCardReaderDisposition.Leave)
```

```
                        Dim apdu As Byte() = {&HFF, &HCA, &H0, &H0, &H7}
                        Dim sendPci = SCardPCI.GetPci(rfidReader.Protocol)
                        Dim receivePci = New SCardPCI()

                        Dim receiveBuffer = New Byte(255) {}
                        Dim command = apdu.ToArray()
                        Dim bytesReceived = rfidReader.Transmit(sendPci, command,
command.Length, receivePci, receiveBuffer, receiveBuffer.Length)
                        Dim responseApdu = New ResponseApdu(receiveBuffer,
bytesReceived, IsoCase.Case2Short, rfidReader.Protocol)

                        If readingMode = 3 Then
                            Dim uid As String =
BitConverter.ToString(responseApdu.GetData())
                            uid = uid.Replace("-", "")

                            SendKeys.SendWait(uid + "{ENTER}")
                        ElseIf readingMode = 4 Then
                            Dim uid As Byte() = New Byte(6) {}
                            Dim revuid As Byte() = New Byte(6) {}
                            Array.Copy(responseApdu.GetData(), uid, 7)
                            Array.Copy(uid, revuid, 7)
                            Array.Reverse(revuid, 0, 7)

                            Dim uid2 As String = BitConverter.ToString(revuid)
                            uid2 = uid2.Replace("-", "")

                            SendKeys.SendWait(uid2 + "{ENTER}")
                        End If
                    End Using
                End Using
            End Using
        Catch
            'Error Handling should be developed
        End Try

        Return True
    End Function

    Private Function GetDebuggerDisplay() As String
        Return ToString()
    End Function


End Class
```

Now we have successfully completed the works to be done to get the RFID reader working.

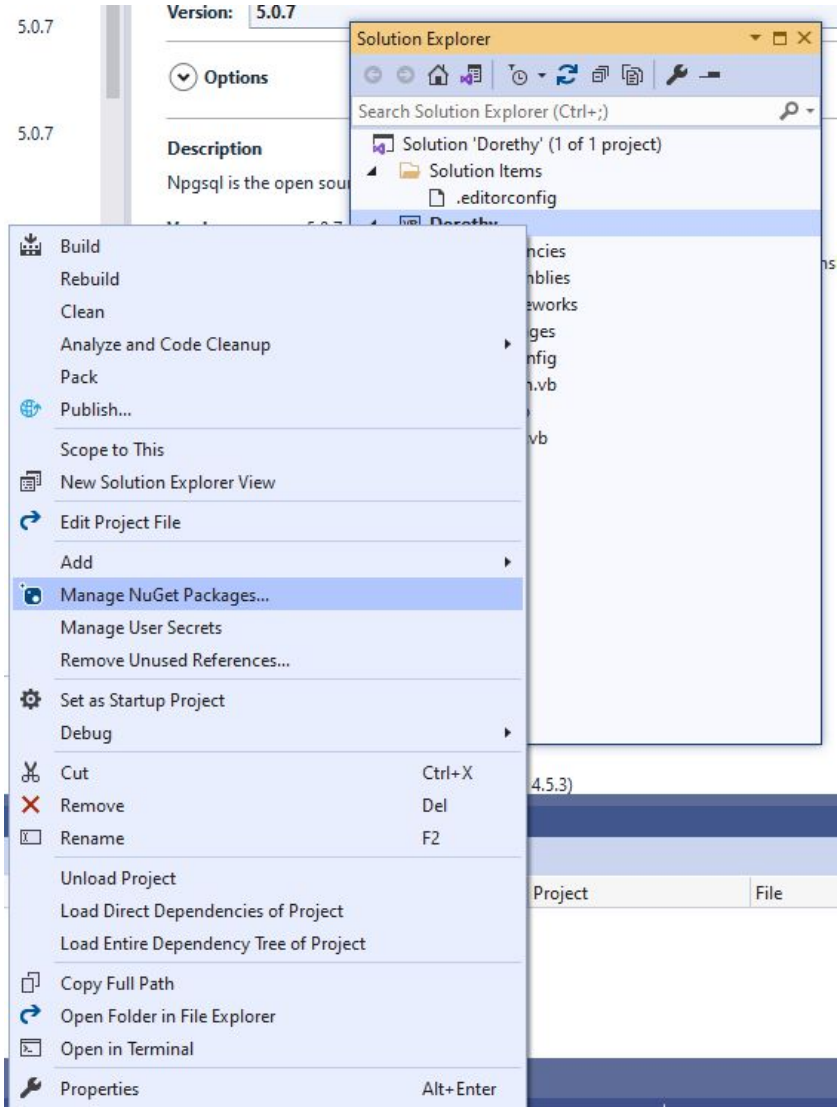
For the following software code, we have to make a connection to the PostgreSQL database.

We are going to make use of the NPGSQL driver for .net. This driver is also available on

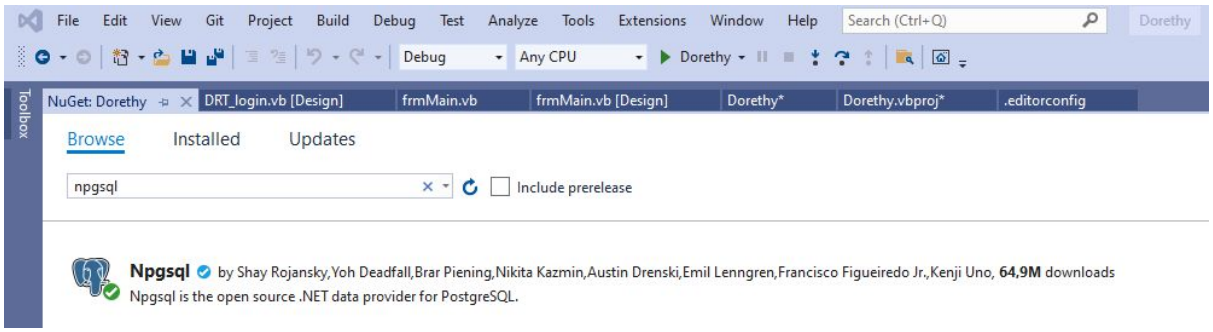
NuGet. The best way to add the files to Dorethy is to download them via the solution which

is provided in Visual Studio. But before we continue we suggest that you save the work you've done by now. Click on File >> Save all and your project will be saved.

We continue with adding the NPGSQL driver to Dorethy. In the Solution Explorer move to Dorethy and right click your mouse. Select Manage NuGet Packages…



Visual Studio connects to the NuGet page. Click on Browse and surge for Npgsql.



Click on Npgsql and follow the instructions to add the driver to Dorethy. If everything is completed you'll find it under Dorethy >> Dependencies >> Packages in the Solution Explorer.

At last we install from Nuget the packages:

Microsoft.VisualBasic and

Microsoft.Windows.Compatibility

The installation of the above packages will make your implementation of the Dorethy software a success. To understand the software better, we start with an explanation of the Dorethy project.

Dorethy how it should work?

Dorethy has made a simple payment solution. Since her camping is located far away of a city. Dorethy needed a solution for small payments on het camping. The idea is that visitors can opt for a Dorethy Card. Top it up with money and spent it in the local store of the camping.

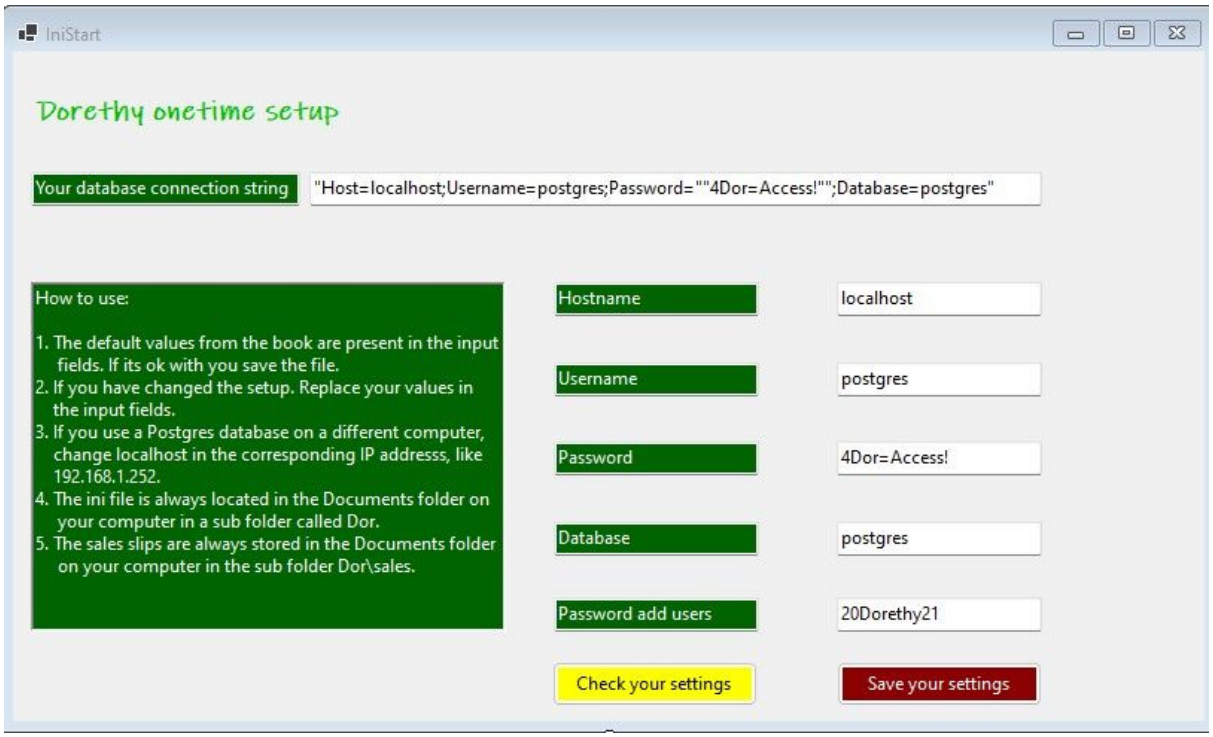
To store products, customers, payments, sales, users and cards. Dorethy uses a PostgreSQL database. You'll find the setup of this database in earlier chapters. The Dorethy software "speaks" with the database via the Npgsql adapter. Since there are several ways to connect to the database, we used different solutions to get the result we need. It's up to you which

solution suits you best. If we need information out of the database, we put the information first in a Datagridview. For further use of the data, we copy the information we need in strings, integers and singles. This is used in every form of the software. If we want to store information in the database, we collect the information from input fields and calculated values. For shared data between forms we used Public Shared Properties. In most cases we handle all the work what need to be done within the active form. This gives you a better overview on how the software operates.

Will the software run smooth out of the box?,… Uhm not everything works as a charm. First the interface with the rfid reader is pretty slow. Furthermore the interface don't like changes. That's the reason you have to click on several buttons to get things done. This prevents malfunction of the interface but it looks like clumsy. Secondly reporting. Our aim with this book and software is to get you motivated to start making your own programs. Reporting software implemented within Dorethy would cost at least several hundreds of Euro's, Dollars or whatever currency you are paying with. Investing that kind of money doesn't fit by the aim of this software. Although you're free to implement it yourself. Before we start with the software code, when you have connected your rfid reader and running the software it works as follows.

Running Dorethy for the very first time.

Each time Dorethy is started, it's looking for the file dps.ini. If it doesn't exist. Dorethy will start up with this page:

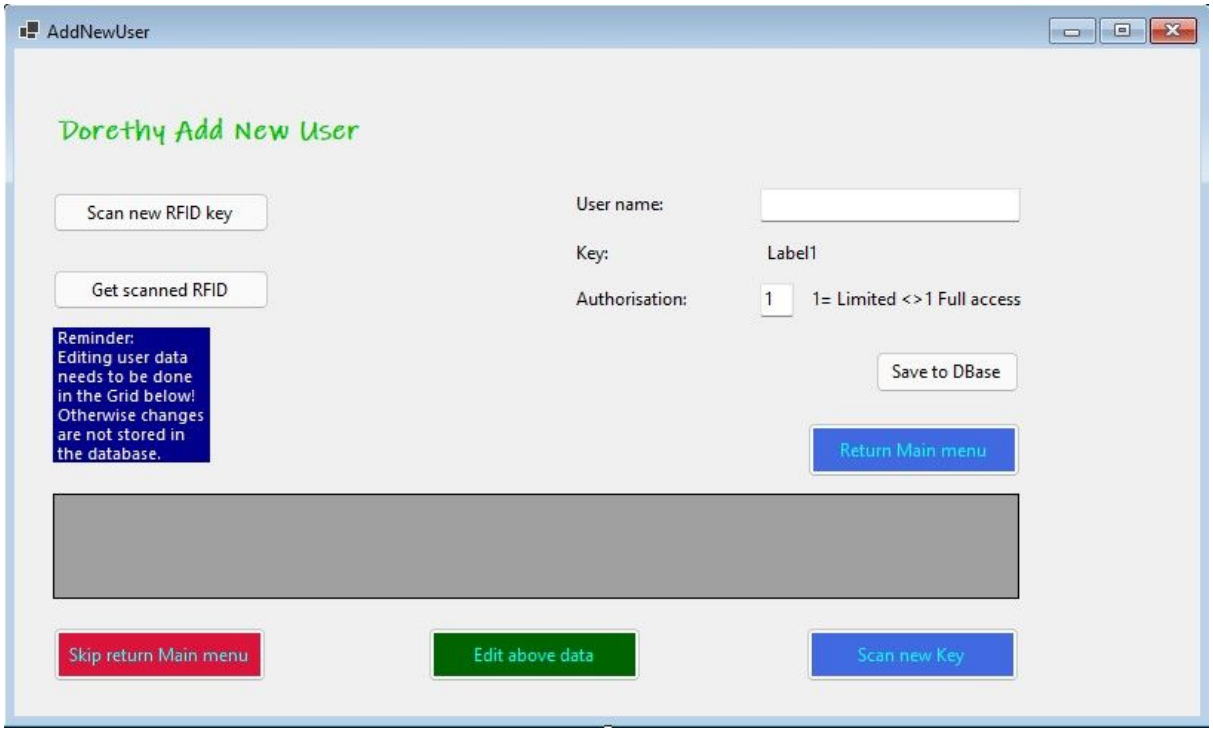


Here you must enter the values required in your situation. For example. If your Postgres database is running on an Raspberry with IP address 192.168.2.252. Then the text localhost should be replaced with the IP address. The text as shown in the input fields are the values we used during creating the software. When you save your settings, the standard login screen is presented.

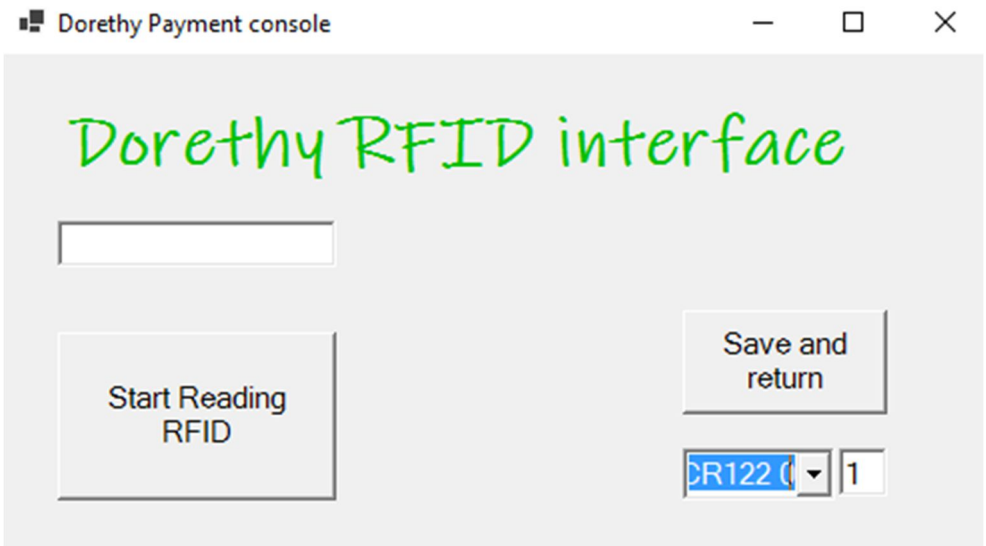
This is the startup screen. When it's the first time to use the software, you need the green field on the screen. Clicking on this field brings you to the form where you can add new users with their onnected rfid chip. The use of this form is password protected. In our example the password is written in the software itself on this location:

```
Private Const V12 As String = "20Dorethy21"
```

After entering the password you'll be transferred to the Add new user form:

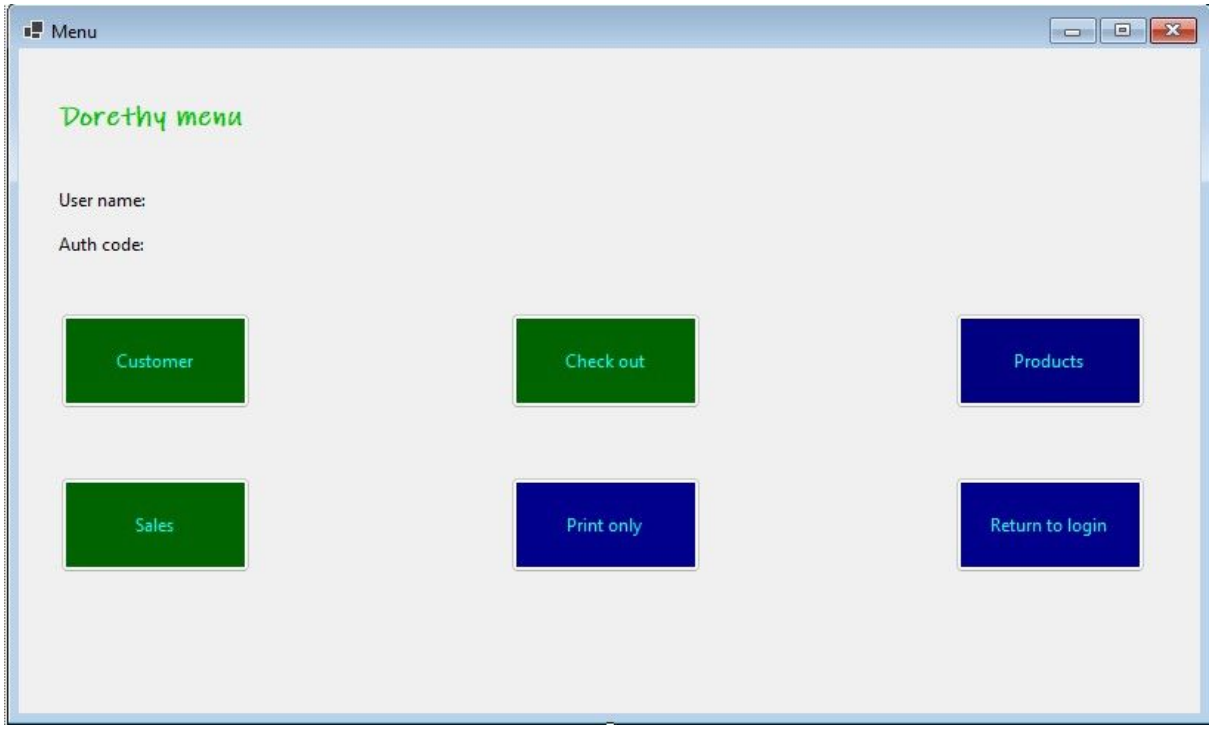


In all forms where we use the RFID reader there are 2 buttons available. The first, in this example Scan new RFID key starts the interface with the rfid scanner. In all situations the interface with the rfid scanner is the same:

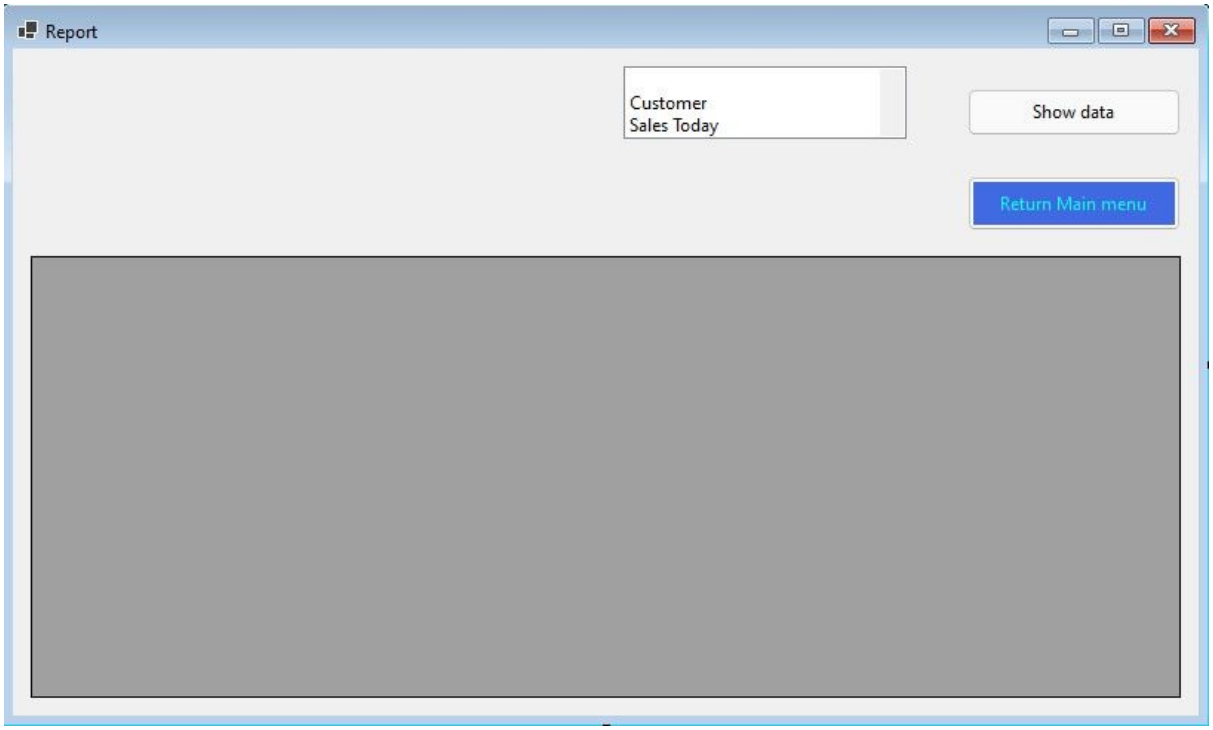


Click on Start reading RFID and wait for the blinking cursor in the field below Dorethy. After a successful scan, the ID of the card is displayed in the field. Press on Save and return to copy the value of the card in one of the constant string and you'll return to the Add new

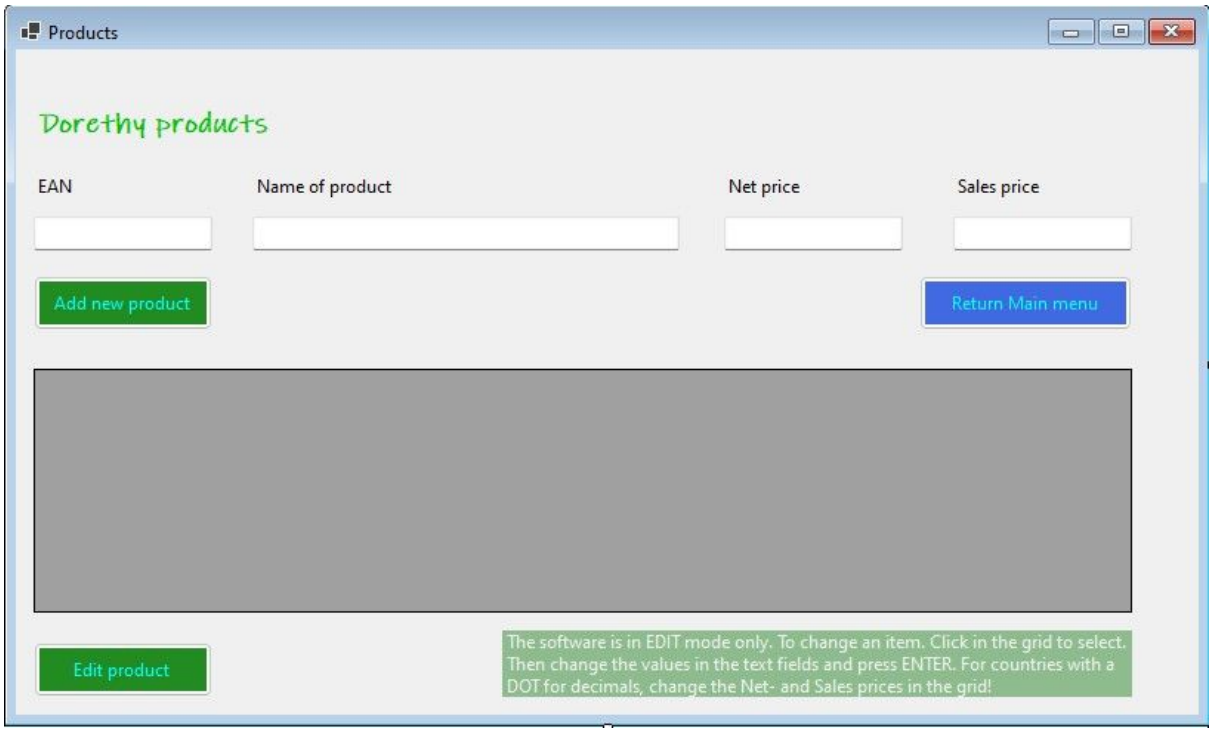
user form. In this form you need to press on the button Get scanned RFID, to copy the value in the field label 1. Add the name and authorization code and press on Save to Dbase to add the user. Within each form we hide fields which are not mandatory for the current process. When users are already in the database you'll start on the login screen with the button Scan new RFID key. Follow the above mentioned procedure for scanning your rfid card and after returning to the login screen press on Get scanned RFID. With a successful match you'll be transferred to the main menu:



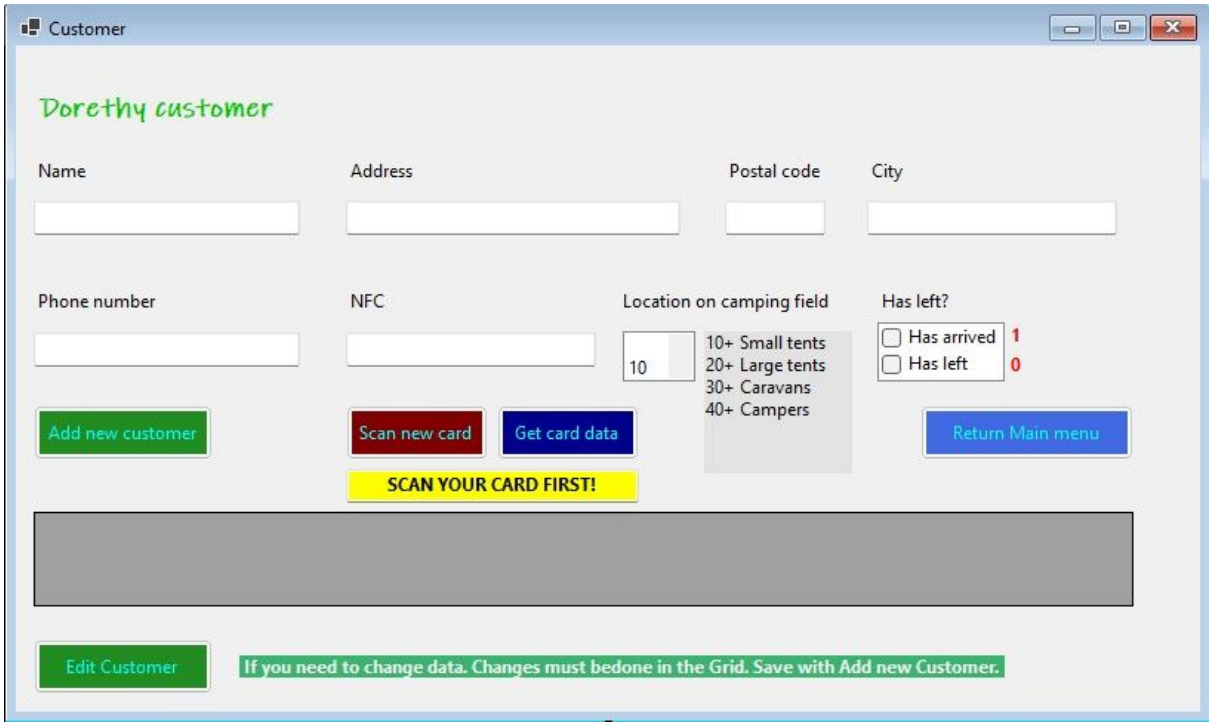
In Dorethy 's opinion, the green buttons will be mostly used. The Print only button shows you 2 options.



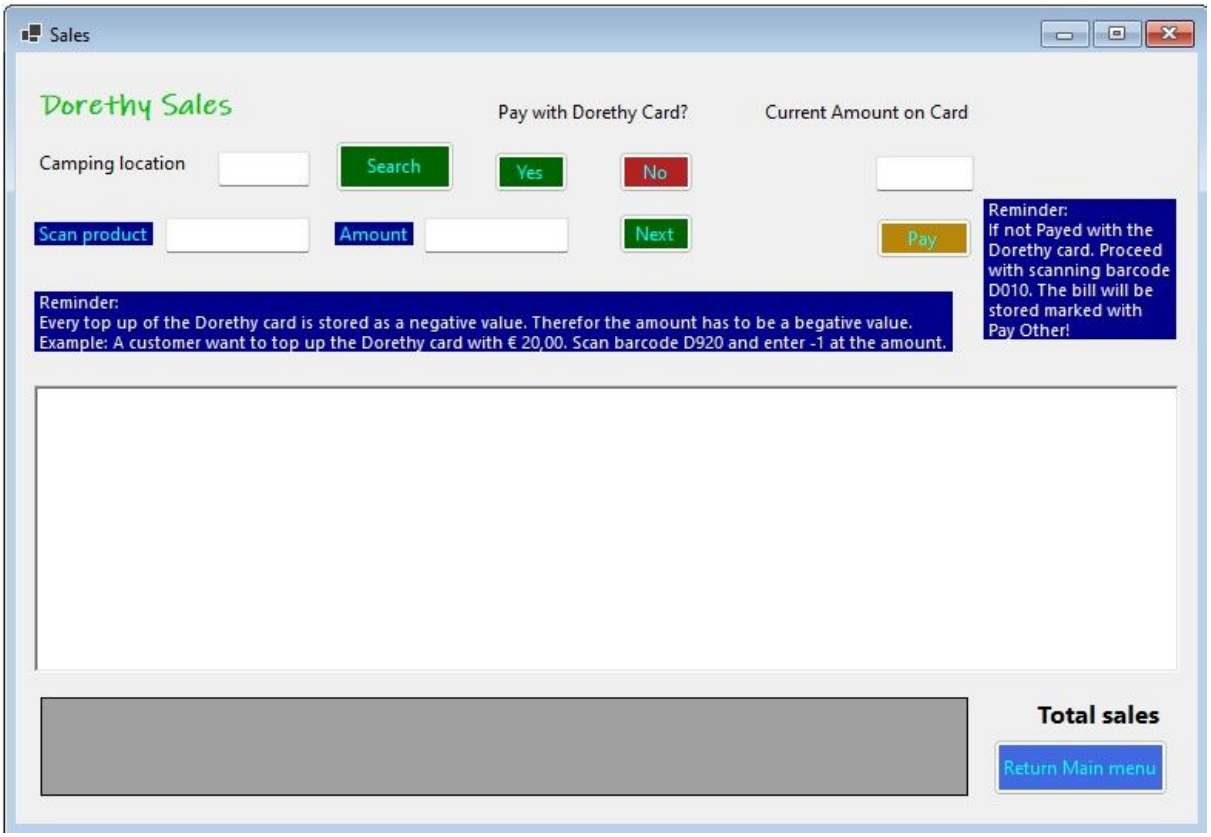
The first one is Customer. If you click on show data, a table is generated with a list of all your customers. The second option is Sales Today. When clicking on it a table with your day sales is generated. Due to the missing report generator this is all what you can do with it. In the form Products your able to add or change the products available within Dorethy camping:



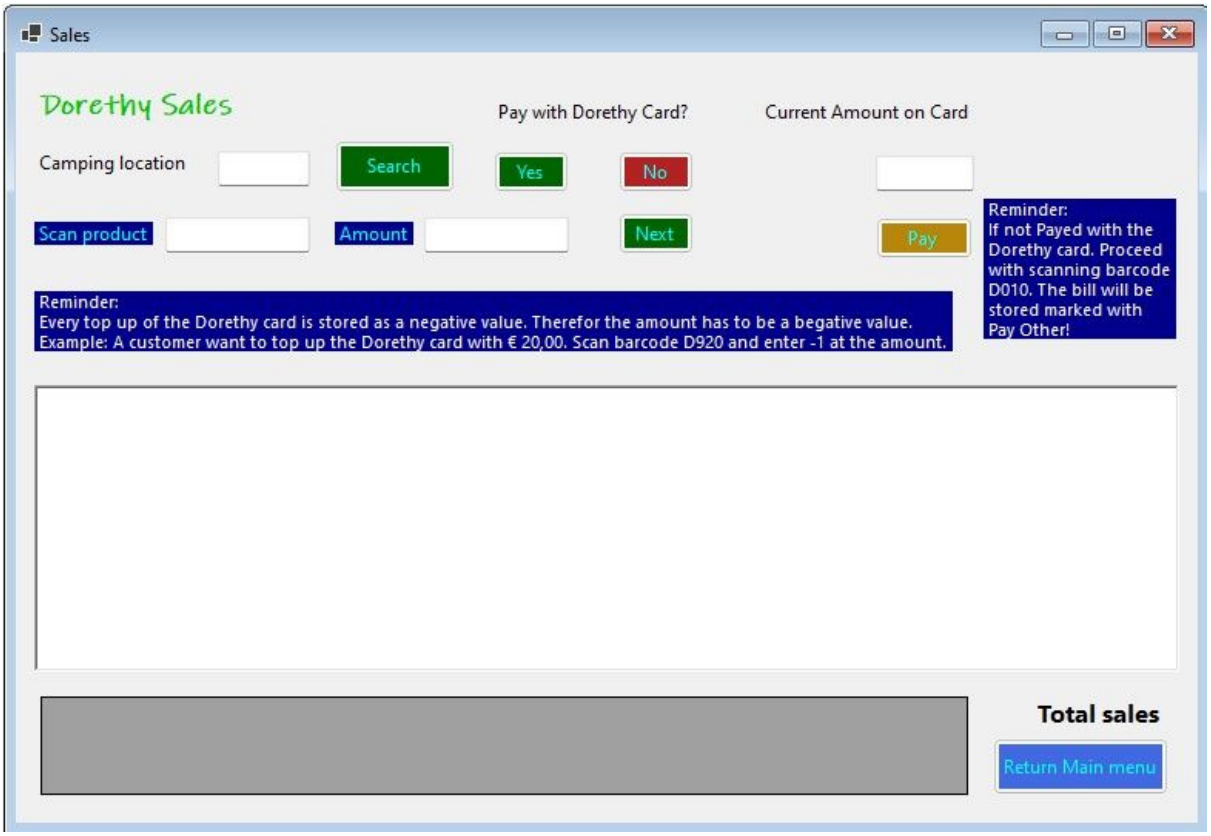
If a product is already in the database then it's possible to change the data and store it. If you enter a new product a text appears when the data is successfully stored. When you're done return to the main menu. The table customers stores the information from a customer who is staying on the camping ground.



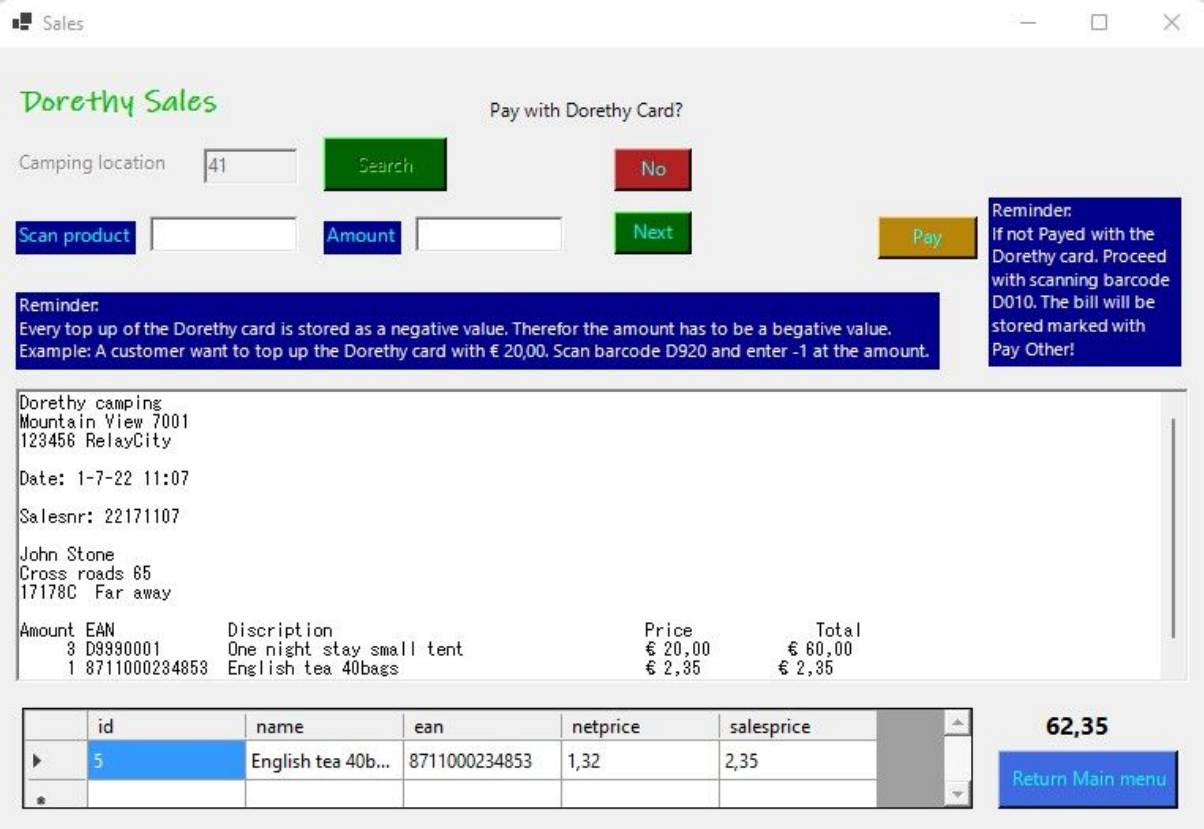
You always need to start with scanning the rfid belonging to the customer. That's the reason why all the others fields are blocked. This is due to the fact that the form is cleared at the moment you return from scanning the rfid. After a successful scan of the rfid all other fields will be available for storing the requested information. Make sure you click on the location number of the camping field. The number is stored and you won't be able to add the same location on another customer. The sales form. Within this form you can store the selling's made by the customers and store the information in the database:



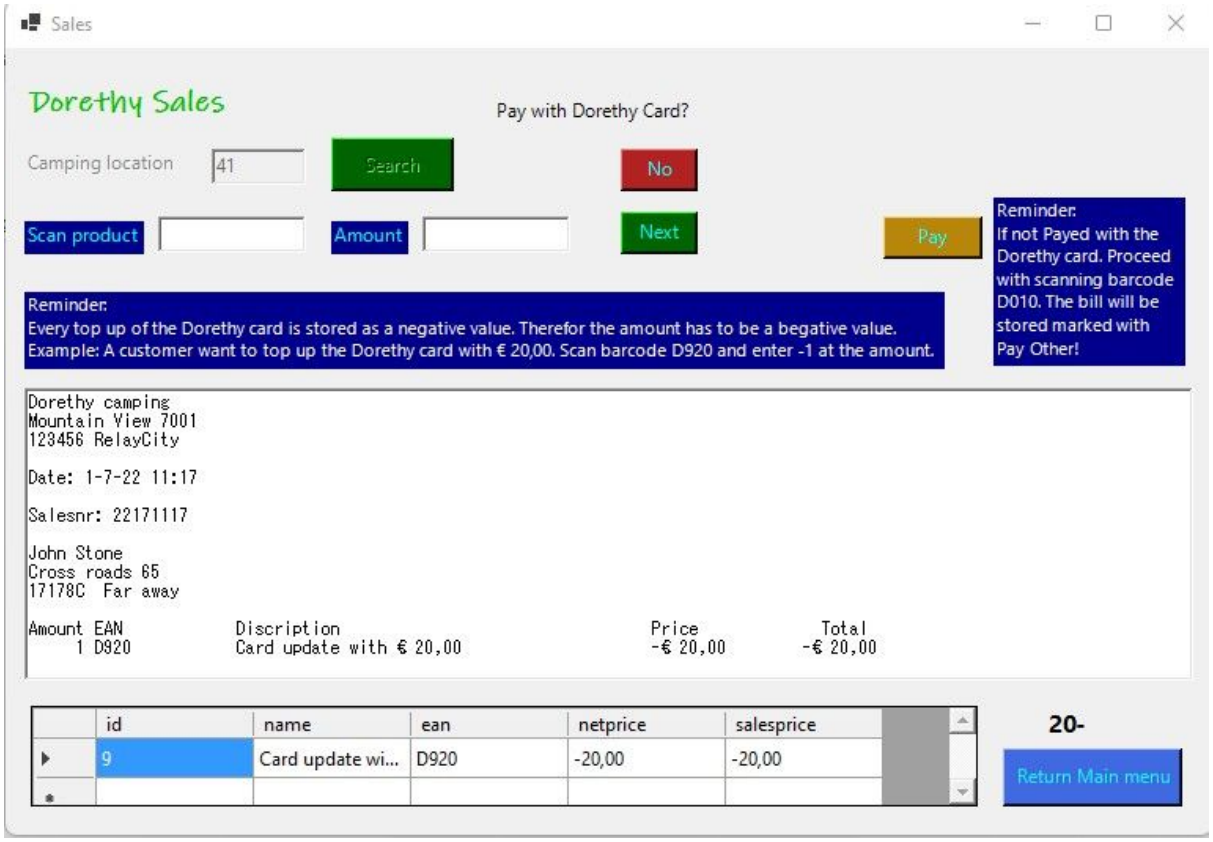
Ask he customer for his or her location on the camping field, enter it and click on search.



The information of the customer is copied in a Richtextbox. This box is used to create a sales ticket. Start by choosing if the customer will pay with the Dorethy card or with another payment method. When a customers comes for the first time on the camping ground you'll need to follow the following procedure to make the software automatically fill all the tables. Let's assume John Stone just arrived at the camping ground. He want to stay for 2 days. The screen looks like below, when he also buys tea. The tables payments and cards will be filled automatically.



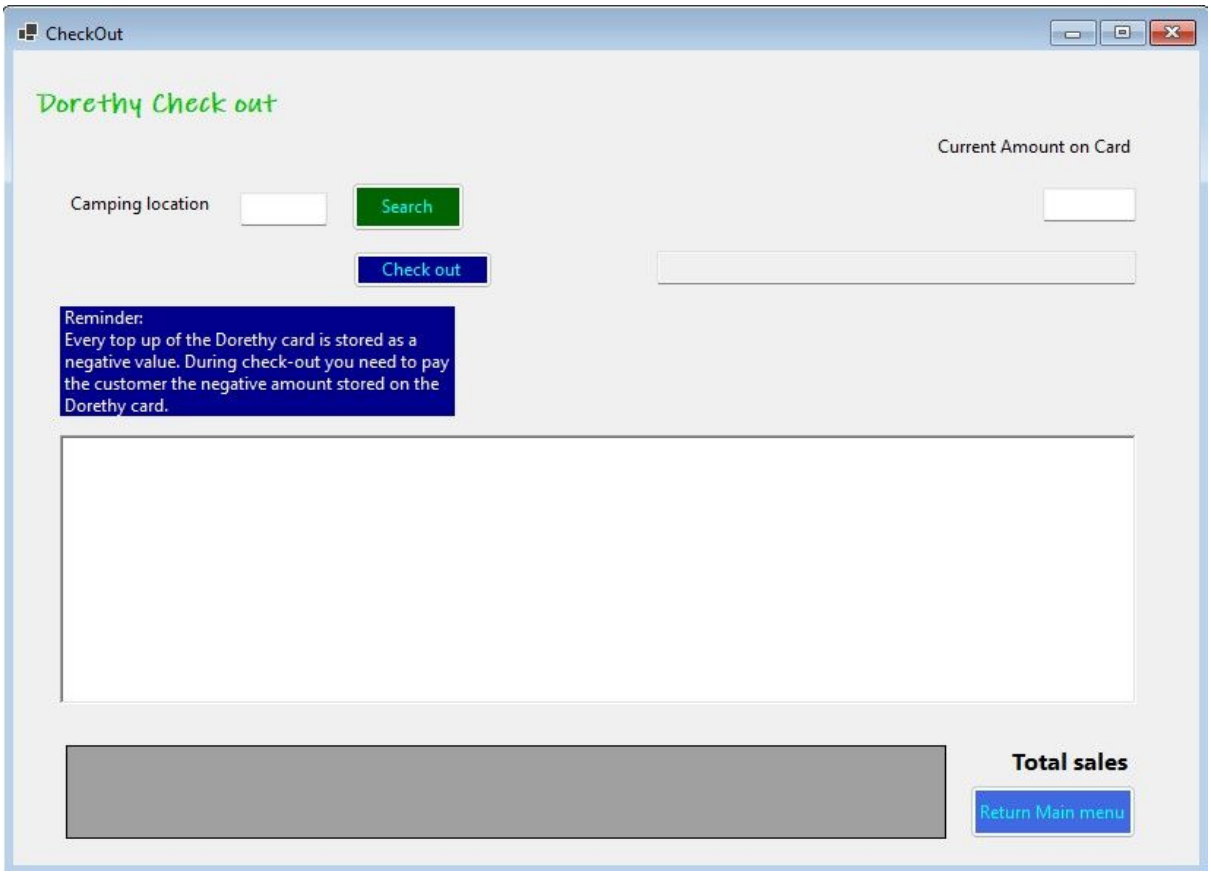
If he also wants to make use of the Dorethy card. A top up needs to be handled as a separate booking. See below when John want a top up of € 20,00 on his Dorethy card:



The next time he want to buy something is the store is will be automatically deducted from the Dorethy card.

After pressing the Pay button all data will be stored. The information within the Richtextbox is stores as RTF file on your hard disk. This makes it easy to print or email the sales ticket.

We end with the customer CheckOut form. Entering the camping location will check if the customer must be paid for the balance on his or her Dorethy card. On the other hand you've maybe allowed to pay at checkout. In that situation the customer has to pay you.



Completing DRT_login

We started our programming with adding the form DRT_login to our solution. Dorethy is keen on following the law. Since DRT_login is the first page which will be shown when the software starts. It seems Dorethy the perfect place to show here the legal content. Dorethy renamed all the field, buttons and so on to make the software more readable. Looking at the screen prints above it will be an easy job to combine the program code with the fields and buttons.

DRT_login:

```
Imports Microsoft.VisualBasic
Imports Microsoft.VisualBasic.Interaction
Imports System.Data.SqlClient
Imports System.Data.Odbc
Imports Npgsql
```

```
Public Class DRT_Login

    Private Const V As String = ""
    Private Const V12 As String = "20Dorethy21"
    ReadOnly path As String =
Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments)
    Public Shared Property GetFromRFID As String
    Public Shared Property ReturnCodeRFID As String
    Public Shared Property UserName As String
    Public Shared Property Auth As String
    Public Shared Property ReturnTo As String

    Private Sub DRT_Login_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        EIF.Visible = False

        Dim file As System.IO.StreamReader

        Try
            file = My.Computer.FileSystem.OpenTextFileReader(path & "\Dor\dps.ini")
            IniStart.DBstr = file.ReadLine
            IniStart.AUpw = file.ReadLine
            IniStart.SAsl = file.ReadLine
            file.Close()

        Catch ex As Exception
            EIF.Visible = True
            StrtRFIDr.Visible = False
            BtnGetRFID.Visible = False
        End Try

    End Sub
    Private Sub Chk()

        Dim oForm As IniStart
        oForm = New IniStart()
        oForm.Show()

        Me.Hide()


    End Sub
    Private Sub BtnGetRFID_Click(sender As Object, e As EventArgs) Handles
BtnGetRFID.Click
        Actions()
    End Sub

    Private Sub Actions()
        If ReturnCodeRFID = "ok" And ReturnTo = "1" Then
            RFID_scanned.Text = GetFromRFID
            StrtRFIDr.Visible = True

            Dim Conn As NpgsqlConnection
            Dim Adpt As NpgsqlDataAdapter

            'Dim cmd As NpgsqlCommand

            Dim npgsqlConnection As New NpgsqlConnection
            Conn = npgsqlConnection
            Conn.ConnectionString = IniStart.DBstr
            Conn.Open()
```

```
            Dim sql As String = "Select * from users where users.nfc = '" &
RFID_scanned.Text & "'"

            Adpt = New NpgsqlDataAdapter(sql, Conn)
            Dim RS As New DataTable
            Try
                Adpt.Fill(RS)
                Dim ds As New DataSet("DataSetOne")
                ds.Tables.Add(RS)
                DataGridView1.DataSource = RS
                Conn.Close()
            Catch ex As Exception
                MsgBox("Some database error ocurred!",, "Dorethy message")
            End Try

            Conn.Close()

            If DataGridView1.Rows(0).Cells(1).Value <> "" Then
                UserName = DataGridView1.Rows(0).Cells(1).Value
                Auth = DataGridView1.Rows(0).Cells(3).Value

                Dim oForm As Menu
                oForm = New Menu()
                oForm.Show()

                Hide()
            End If
            StrtRFIDr.Visible = True
        End If
    End Sub



    Private Sub StrtRFIDr_Click(sender As Object, e As EventArgs) Handles
StrtRFIDr.Click
        StrtRFIDr.Visible = False
        BtnGetRFID.Visible = True

        ReturnTo = "1"
        GetFromRFID = ""

        Dim oForm As FrmMain
        oForm = New FrmMain()
        oForm.Show()

        Hide()
    End Sub

    Private Sub AddNewUser_Click(sender As Object, e As EventArgs) Handles
AddNewUser.Click
        RFID_scanned.Text = GetFromRFID
        Dim EnterYourPW As String
        EnterYourPW = InputBox("Dorethy password", "Dorethy payment solution")
        Try
            If EnterYourPW = IniStart.AUpw Then
                RFID_scanned.Text = EnterYourPW
                GetFromRFID = V
                Dim oForm As AddNewUser
                Dim form2 As New AddNewUser()
```

```
                oForm = form2
                oForm.Show()
                Hide()
                Return
            ElseIf RFID_scanned.Text <> EnterYourPW Then
                MsgBox("Your entered password in incorrect",, "Dorethy error message")
                Return
            End If

        Catch ex As Exception

        End Try
    End Sub

    Private Sub BtnExit_Click(sender As Object, e As EventArgs) Handles BtnExit.Click
        End
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles EIF.Click
        Chk()
    End Sub
End Class
```

Within DRT_login there are several variables which are shared within Dorethy. These are:

```
    Private Const V As String = ""
    Private Const V12 As String = "20Dorethy21"
    ReadOnly path As String =
Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments)
    Public Shared Property GetFromRFID As String
    Public Shared Property ReturnCodeRFID As String
    Public Shared Property UserName As String
    Public Shared Property Auth As String
    Public Shared Property ReturnTo As String
```

All Public Shared Property ‘s. If you are going to use fixed values within the software use it in the way we use it at V12.

To handle the most of the data within the form. We use the same connection with the database on local sub level. Look in the program code for these lines:

```
Dim Conn As NpgsqlConnection
            Dim Adpt As NpgsqlDataAdapter

            'Dim cmd As NpgsqlCommand

            Dim npgsqlConnection As New NpgsqlConnection
            Conn = npgsqlConnection
            Conn.ConnectionString = V1
            Conn.Open()

            Dim sql As String = "Select * from users where users.nfc = '" &
RFID_scanned.Text & "'"
```

Don’t forget to close the database at the moment the data is stored or retrieved. The command is: `Conn.Close()`

The program code is in all forms the same way build. That is the reason why all program code follows after each other.

AddNewUser:

```
Imports Microsoft.VisualBasic
Imports Microsoft.VisualBasic.Interaction
Imports System.Data.SqlClient
Imports System.Data.Odbc
Imports Npgsql

Public Class AddNewUser

    Private Const V11 As String = "Select * from users, cmb"
    Private Const V As String = ""
    Dim V12 As String
    Private Sub ScanNewRFID_Click(sender As Object, e As EventArgs) Handles
ScanNewRFID.Click
        DRT_Login.ReturnTo = "2"
        DRT_Login.GetFromRFID = ""
        GetscannedRFID.Visible = True
        ScanNewRFID.Visible = False
        Refresh()
        Dim oForm As FrmMain
        oForm = New FrmMain()
        oForm.Show()

        Hide()
    End Sub

    Private Sub GetscannedRFID_Click(sender As Object, e As EventArgs) Handles
GetscannedRFID.Click
        If DRT_Login.ReturnCodeRFID = "ok" And DRT_Login.ReturnTo = "2" Then
            ID_FS.Text = DRT_Login.GetFromRFID
        End If
        ScanNewRFID.Visible = False
        Save2DB.Visible = True
        GetscannedRFID.Visible = False
        Save2DB.PerformClick()

    End Sub

    Private Sub Save2DB_Click() Handles Save2DB.Click

        If V12 <> "EDT" Then
            Dim Conn As NpgsqlConnection
            Dim Adpt As NpgsqlDataAdapter

            'Dim cmd As NpgsqlCommand

            Dim npgsqlConnection As New NpgsqlConnection
            Conn = npgsqlConnection
```

```
            Conn.ConnectionString = IniStart.DBstr
            Conn.Open()

            Dim sql As String = "Select * from users where users.nfc = '" & ID_FS.Text
& "'"

            Adpt = New NpgsqlDataAdapter(sql, Conn)
            Dim RS As New DataTable
            Try
                Adpt.Fill(RS)
                Dim ds As New DataSet("DataSetOne")
                ds.Tables.Add(RS)
                DataGridView1.DataSource = RS
                Conn.Close()
            Catch ex As Exception
                MsgBox("Some database error ocurred!",, "Dorethy message")
            End Try

            Conn.Close()

            If DataGridView1.Rows(0).Cells(1).Value = "" And Name2DB.Text <> "" Then
                AddNU2DB()
            ElseIf DataGridView1.Rows(0).Cells(1).Value <> "" Then
                Name2DB.Text = DataGridView1.Rows(0).Cells(1).Value
                ID_FS.Text = DataGridView1.Rows(0).Cells(2).Value
                Auth2DB.Text = DataGridView1.Rows(0).Cells(3).Value
                MsgBox("RFID is already stored in the database, select your option
below",, "Dorethy message")
                DRT_Login.ReturnCodeRFID = "exist"
                ViewEKS()
            End If

        End If
        If V12 = "EDT" Then
            AddNU2DB()
        End If
    End Sub
    Private Sub AddNU2DB()

        If Name2DB.Text <> "" And Auth2DB.Text <> "" Then
            Dim Conn As NpgsqlConnection
            Dim Adpt As NpgsqlDataAdapter
            Dim cmd As NpgsqlCommand

            Conn = New NpgsqlConnection With {
                .ConnectionString = IniStart.DBstr
            }
            Conn.Open()

            Dim sql As String = V11

            Adpt = New NpgsqlDataAdapter(sql, Conn)

            If DRT_Login.ReturnCodeRFID = "ok" And V12 <> "EDT" Then
                sql = ("insert into public.users (" & Chr(34) & "user" & Chr(34) & ",
nfc, auth) values (" & Chr(39) & Name2DB.Text & Chr(39) & ", " & Chr(39) & ID_FS.Text
& Chr(39) & "," & Chr(39) & Auth2DB.Text & Chr(39) & ")")
            End If

            If V12 = "EDT" Then
```

```
                sql = "update public.users set ""user"" =" & Chr(39) &
DataGridView1.Rows(0).Cells(1).Value & Chr(39) & ", auth = " & Chr(39) &
DataGridView1.Rows(0).Cells(3).Value & Chr(39) & " where id=" & Chr(39) &
DataGridView1.Rows(0).Cells(0).Value & Chr(39) & " "
            End If

            cmd = New NpgsqlCommand(sql, Conn)
            cmd.ExecuteNonQuery()

            Conn.Close()
            ID_FS.ForeColor.R.ToString()
            ID_FS.Text = "Key successfuly saved"
        End If
        If Name2DB.Text = "" And Auth2DB.Text = "" Then
            MsgBox("Name and authorisation need to be filled in!",, "Dorethy error
message")
        End If
        ScanNewRFID.Visible = True
        Save2DB.Visible = False
        V12 = ""
    End Sub
    Private Sub BtnReturn_Click(sender As Object, e As EventArgs) Handles
BtnReturn.Click
        Name2DB.Text = ""
        Auth2DB.Text = ""
        HideEKS()
        ScanNewRFID.Visible = True
        GetscannedRFID.Visible = True
        DataGridView1.Columns.Clear()
        Dim oForm As DRT_Login
        oForm = New DRT_Login()
        oForm.Show()

        Hide()
    End Sub

    Private Sub AddNewUser_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Save2DB.Visible = False
        DataGridView1.Enabled = False
        HideEKS()
    End Sub
    Private Sub ViewEKS()
        SNI.Visible = True
        EDT.Visible = True
        SRM.Visible = True
    End Sub
    Private Sub HideEKS()
        SNI.Visible = False
        EDT.Visible = False
        SRM.Visible = False
    End Sub

    Private Sub SNI_Click(sender As Object, e As EventArgs) Handles SNI.Click
        HideEKS()
        ScanNewRFID.Visible = True
        Save2DB.Visible = False
        Name2DB.Text = ""
        Auth2DB.Text = ""
        ID_FS.Text = ""
    End Sub
```

```
    Private Sub EDT_Click(sender As Object, e As EventArgs) Handles EDT.Click
        V12 = "EDT"
        Name2DB.Text = DataGridView1.Rows(0).Cells(1).Value
        HideEKS()
        DataGridView1.Enabled = True
    End Sub


    Private Sub SRM_Click(sender As Object, e As EventArgs) Handles SRM.Click
        BtnReturn.PerformClick()

    End Sub
End Class
```

Menu:

```
Public Class Menu
    Private Sub Menu_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        DB_user.Text = DRT_Login.UserName
        Auth_C.Text = DRT_Login.Auth
        If Auth_C.Text = "1" Then
            BtnProducts.Visible = False
            BtbPO.Visible = False
        End If
    End Sub

    Private Sub BtnReturn_Click(sender As Object, e As EventArgs) Handles
BtnReturn.Click
        Dim oForm As DRT_Login
        oForm = New DRT_Login()
        oForm.Show()

        Hide()
    End Sub

    Private Sub BtnProducts_Click(sender As Object, e As EventArgs) Handles
BtnProducts.Click
        Dim oForm As Products
        oForm = New Products()
        oForm.Show()

        Hide()
    End Sub

    Private Sub BtnCust_Click(sender As Object, e As EventArgs) Handles BtnCust.Click
        Dim oForm As Customer
        oForm = New Customer()
        oForm.Show()

        Hide()
    End Sub

    Private Sub BtbPO_Click(sender As Object, e As EventArgs) Handles BtbPO.Click
        Dim oForm As Report
        oForm = New Report()
        oForm.Show()

        Hide()
```

```
    End Sub

    Private Sub BtnSales_Click(sender As Object, e As EventArgs) Handles
BtnSales.Click
        Dim oForm As Sales
        oForm = New Sales()
        oForm.Show()

        Hide()
    End Sub

    Private Sub BtnChkO_Click(sender As Object, e As EventArgs) Handles BtnChkO.Click
        Dim oForm As CheckOut
        oForm = New CheckOut()
        oForm.Show()

        Hide()
    End Sub
End Class
```

Products:

```
Imports Microsoft.VisualBasic
Imports Microsoft.VisualBasic.Interaction
Imports System.Data.SqlClient
Imports Npgsql


Public Class Products

    Private Const V11 As String = "Select * from products, cmb"
    'Private Const V As String = ""
    Dim V12 As String

    Private Sub BtnAdd_Click(sender As Object, e As EventArgs) Handles BtnAdd.Click
        If V12 = "EDT" Then
            AddNU2DB()
        ElseIf V12 <> "EDT" Then
            Dim Conn As NpgsqlConnection
            Dim Adpt As NpgsqlDataAdapter

            'Dim cmd As NpgsqlCommand

            Dim npgsqlConnection As New NpgsqlConnection
            Conn = npgsqlConnection
            Conn.ConnectionString = IniStart.DBstr
            Conn.Open()

            Dim sql As String = "Select * from products where products.ean = '" &
EAN2DB.Text & "'"

            Adpt = New NpgsqlDataAdapter(sql, Conn)
            Dim RS As New DataTable
            Try
                Adpt.Fill(RS)
                Dim ds As New DataSet("DataSetOne")
                ds.Tables.Add(RS)
                DataGridView1.DataSource = RS
                Conn.Close()
```

```
            Catch ex As Exception
                MsgBox("Some database error ocurred!",, "Dorethy message")
            End Try

            Conn.Close()

            If DataGridView1.Rows(0).Cells(1).Value = "" Then
                AddNU2DB()
            ElseIf DataGridView1.Rows(0).Cells(1).Value <> "" Then
                MsgBox("EAN is already stored in the database, select your option
below",, "Dorethy message")
                DRT_Login.ReturnCodeRFID = "exist"
                V12 = "EDT"
                BtnEdit.Visible = True
            End If

        End If

    End Sub
    Private Sub AddNU2DB()
        If EAN2DB.Text <> "" And Name2DB.Text <> "" And NETPR2DB.Text <> "" And
SALESPR2DB.Text <> "" Then
            Dim Conn As NpgsqlConnection
            Dim Adpt As NpgsqlDataAdapter
            Dim cmd As NpgsqlCommand

            Conn = New NpgsqlConnection With {
                .ConnectionString = IniStart.DBstr
            }
            Conn.Open()

            Dim sql As String = V11

            Adpt = New NpgsqlDataAdapter(sql, Conn)

            If V12 <> "EDT" Then
                sql = ("insert into public.products (ean, name, netprice, salesprice)
values (" & Chr(39) & EAN2DB.Text & Chr(39) & ", " & Chr(39) & Name2DB.Text & Chr(39)
& "," & Chr(39) & NETPR2DB.Text & Chr(39) & ", " & Chr(39) & SALESPR2DB.Text & Chr(39)
& ")")
            ElseIf V12 = "EDT" Then
                sql = "update public.products set name = " & Chr(39) &
DataGridView1.CurrentRow.Cells(2).Value & Chr(39) & ", netprice = " & Chr(39) &
DataGridView1.CurrentRow.Cells(3).Value & Chr(39) & ", salesprice = " & Chr(39) &
DataGridView1.CurrentRow.Cells(4).Value & Chr(39) & " where id=" & Chr(39) &
DataGridView1.CurrentRow.Cells(0).Value & Chr(39) & " "

            End If


            cmd = New NpgsqlCommand(sql, Conn)
            cmd.ExecuteNonQuery()

            Conn.Close()
            If ID_FS.Visible = False Then ID_FS.Visible = True
            ID_FS.ForeColor.R.ToString()
            ID_FS.Text = "Key successfuly saved"
            V12 = ""
            Application.DoEvents()
            System.Threading.Thread.Sleep(500)
```

```
            Application.DoEvents()
            ID_FS.Visible = False
            ClearFields()
            GoTo SKIPPED
        End If
        If V12 <> "EDT" And EAN2DB.Text = "" And Name2DB.Text = "" And NETPR2DB.Text =
"" And SALESPR2DB.Text = "" Then
            MsgBox("All fields are mandatory!",, "Dorethy error message")
        End If
SKIPPED:
    End Sub

    Private Sub BtnEdit_Click(sender As Object, e As EventArgs) Handles BtnEdit.Click
        IF_SWN.Visible = True
        Dim Conn As NpgsqlConnection
        Dim Adpt As NpgsqlDataAdapter

        'Dim cmd As NpgsqlCommand

        Dim npgsqlConnection As New NpgsqlConnection
        Conn = npgsqlConnection
        Conn.ConnectionString = IniStart.DBstr
        Conn.Open()

        Dim sql As String = "Select * from products"

        Adpt = New NpgsqlDataAdapter(sql, Conn)
        Dim RS As New DataTable
        Try
            Adpt.Fill(RS)
            Dim ds As New DataSet("DataSetOne")
            ds.Tables.Add(RS)
            DataGridView1.DataSource = RS
            Conn.Close()
        Catch ex As Exception
            MsgBox("Some database error ocurred!",, "Dorethy message")
        End Try

        Conn.Close()
        BtnAdd.Text = "Save changes"

        V12 = "EDT"
        MsgBox("Edit your data in the datagrid and save your changed by clicking the
Add new product button. Remember! You have to save settings per line!",, "Dorethy
message")
    End Sub

    Private Sub BtnReturn_Click(sender As Object, e As EventArgs) Handles
BtnReturn.Click
        DataGridView1.Columns.Clear()
        IF_SWN.Visible = False
        Dim oForm As Menu
        oForm = New Menu()
        oForm.Show()

        Hide()
    End Sub
    Private Sub ClearFields()
        EAN2DB.Text = ""
        Name2DB.Text = ""
```

```
        NETPR2DB.Text = ""
        SALESPR2DB.Text = ""
    End Sub

    Private Sub DataGridView1_CellContentClick(sender As Object, e As
DataGridViewCellEventArgs) Handles DataGridView1.CellContentClick
        EAN2DB.Text = DataGridView1.CurrentRow.Cells(1).Value
        Name2DB.Text = DataGridView1.CurrentRow.Cells(2).Value
        NETPR2DB.Text = DataGridView1.CurrentRow.Cells(3).Value
        SALESPR2DB.Text = DataGridView1.CurrentRow.Cells(4).Value
        V12 = "EDT"

    End Sub

    Private Sub Products_Load(sender As Object, e As EventArgs) Handles MyBase.Load

        IF_SWN.Visible = False
    End Sub

    Private Sub NETPR2DB_Keypress(sender As Object, e As KeyPressEventArgs) Handles
NETPR2DB.KeyPress

        If (e.KeyChar = ".") Then
            e.KeyChar = ","
        End If
        If (e.KeyChar = Chr(13)) Then
            DataGridView1.CurrentRow.Cells(3).Value = NETPR2DB.Text
        End If
    End Sub

    Private Sub EAN2DB_Keypress(sender As Object, e As KeyPressEventArgs) Handles
EAN2DB.KeyPress
        If (e.KeyChar = Chr(13)) Then
            DataGridView1.CurrentRow.Cells(1).Value = EAN2DB.Text
        End If
    End Sub

    Private Sub Name2DB_Keypress(sender As Object, e As KeyPressEventArgs) Handles
Name2DB.KeyPress
        If (e.KeyChar = Chr(13)) Then
            DataGridView1.CurrentRow.Cells(2).Value = Name2DB.Text
        End If
    End Sub

    Private Sub SALESPR2DB_Keypress(sender As Object, e As KeyPressEventArgs) Handles
SALESPR2DB.KeyPress

        If (e.KeyChar = ".") Then
            e.KeyChar = ","
        End If
        If (e.KeyChar = Chr(13)) Then
            DataGridView1.CurrentRow.Cells(4).Value = SALESPR2DB.Text
        End If
    End Sub
End Class
```

Customer:

```
Imports Microsoft.VisualBasic
```

```
Imports Microsoft.VisualBasic.Interaction
Imports System.Data.SqlClient
Imports System.Data.Odbc
Imports Npgsql


Public Class Customer

    Private Const V11 As String = "Select * from customer, cmb"
    'Private Const V As String = ""
    Dim V12 As String

    Private Sub BtnScanCrd_Click(sender As Object, e As EventArgs) Handles
BtnScanCrd.Click

        BtnScanCrd.Visible = False
        BtnGetCardData.Visible = True
        DRT_Login.ReturnTo = "3"
        DRT_Login.GetFromRFID = ""
        NFC2DB.Text = ""
        Dim oForm As FrmMain
        oForm = New FrmMain()
        oForm.Show()

        Hide()
    End Sub

    Private Sub Customer_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        LockFields()
        DataGridView1.Enabled = False
        If DRT_Login.ReturnTo = "3" Then
            BtnGetCardData.Visible = True
        End If
    End Sub

    Private Sub BtnGetCardData_Click(sender As Object, e As EventArgs) Handles
BtnGetCardData.Click
        NFC2DB.Text = DRT_Login.GetFromRFID
        BtnGetCardData.Visible = False
        BtnScanCrd.Visible = False
        OpenFields()
        DispBAnfc()
    End Sub

    Private Sub DispBAnfc()
        Dim Conn As NpgsqlConnection
        Dim Adpt As NpgsqlDataAdapter

        'Dim cmd As NpgsqlCommand

        Dim npgsqlConnection As New NpgsqlConnection
        Conn = npgsqlConnection
        Conn.ConnectionString = IniStart.DBstr
        Conn.Open()

        Dim sql As String = "Select * from customer where customer.nfc = '" &
NFC2DB.Text & "'"

        Adpt = New NpgsqlDataAdapter(sql, Conn)
        Dim RS As New DataTable
```

```
        Try
            Adpt.Fill(RS)
            Dim ds As New DataSet("DataSetOne")
            ds.Tables.Add(RS)
            DataGridView1.DataSource = RS
            Conn.Close()
        Catch ex As Exception
            MsgBox("Some database error ocurred!",, "Dorethy message")
        End Try

        Conn.Close()

    End Sub
    Private Sub BtnAdd_Click() Handles BtnAdd.Click
        If V12 = "EDT" Then
            AddNU2DB()
        ElseIf V12 <> "EDT" Then
            Dim Conn As NpgsqlConnection
            Dim Adpt As NpgsqlDataAdapter

            'Dim cmd As NpgsqlCommand

            Dim npgsqlConnection As New NpgsqlConnection
            Conn = npgsqlConnection
            Conn.ConnectionString = IniStart.DBstr
            Conn.Open()

            Dim sql As String = "Select * from customer where customer.nfc = '" &
NFC2DB.Text & "'"

            Adpt = New NpgsqlDataAdapter(sql, Conn)
            Dim RS As New DataTable
            Try
                Adpt.Fill(RS)
                Dim ds As New DataSet("DataSetOne")
                ds.Tables.Add(RS)
                DataGridView1.DataSource = RS
                Conn.Close()
            Catch ex As Exception
                MsgBox("Some database error ocurred!",, "Dorethy message")
            End Try

            Conn.Close()

            If DataGridView1.Rows(0).Cells(1).Value = "" Then
                AddNU2DB()
            ElseIf DataGridView1.Rows(0).Cells(1).Value <> "" Then
                MsgBox("Campinglocation is in use, correct the error",, "Dorethy
message")
                DRT_Login.ReturnCodeRFID = "exist"
                V12 = "EDT"
                BtnEdit.Visible = True
            End If

        End If
    End Sub

    Private Sub AddNU2DB()
```

```
        If Name2DB.Text <> "" And Address2DB.Text <> "" And PostalC2DB.Text <> "" And
City2DB.Text <> "" And PHONE2DB.Text <> "" And NFC2DB.Text <> "" And CLOC2DB.Text <>
"" Then
            Dim Conn As NpgsqlConnection
            Dim Adpt As NpgsqlDataAdapter
            Dim cmd As NpgsqlCommand

            Conn = New NpgsqlConnection With {
                .ConnectionString = IniStart.DBstr
            }
            Conn.Open()

            Dim sql As String = V11
            Dim Cloc As Integer = Convert.ToInt32(CLOC2DB.Text)
            Dim LBL As Integer
            If LBleft.Text = "Has left" Then LBL = 0
            If LBleft.Text = "Has arrived" Then LBL = 1
            Adpt = New NpgsqlDataAdapter(sql, Conn)

            If V12 <> "EDT" Then
                sql = ("insert into public.customer (name, address, postalcode, city,
phone, nfc, clocation, active) values (" & Chr(39) & Name2DB.Text & Chr(39) & ", " &
Chr(39) & Address2DB.Text & Chr(39) & "," & Chr(39) &
                    PostalC2DB.Text & Chr(39) & ", " & Chr(39) & City2DB.Text & Chr(39)
& "," & Chr(39) &
                    PHONE2DB.Text & Chr(39) & ", " & Chr(39) & NFC2DB.Text & Chr(39) &
"," & Chr(39) & Cloc & Chr(39) & "," & Chr(39) & LBL & Chr(39) & ")")
            ElseIf V12 = "EDT" Then
                sql = "update public.customer set name = " & Chr(39) &
DataGridView1.Rows(0).Cells(1).Value & Chr(39) & ", address = " & Chr(39) &
DataGridView1.Rows(0).Cells(2).Value & Chr(39) & ", postalcode = " & Chr(39) &
DataGridView1.Rows(0).Cells(3).Value & Chr(39) &
                    ", city = " & Chr(39) & DataGridView1.Rows(0).Cells(4).Value &
Chr(39) & ", phone = " & Chr(39) & DataGridView1.Rows(0).Cells(5).Value & Chr(39) & ",
nfc = " & Chr(39) & DataGridView1.Rows(0).Cells(6).Value & Chr(39) & ", clocation = "
& Chr(39) & DataGridView1.Rows(0).Cells(7).Value & Chr(39) &
                    ", active = " & Chr(39) & DataGridView1.Rows(0).Cells(8).Value &
Chr(39) & " where id=" & Chr(39) & DataGridView1.Rows(0).Cells(0).Value & Chr(39) & "
"

            End If


            cmd = New NpgsqlCommand(sql, Conn)
            cmd.ExecuteNonQuery()

            Conn.Close()
            ID_FS.Visible = True
            ID_FS.ForeColor.R.ToString()
            ID_FS.Text = "Key successfuly saved"

        End If
        If V12 = "EDT" Then
            V12 = ""
        ElseIf Name2DB.Text = "" Or Address2DB.Text = "" Or PostalC2DB.Text = "" Or
City2DB.Text = "" Or PHONE2DB.Text = "" Or CLOC2DB.Text = "" Or LBleft.Text = "" Then
            MsgBox("All fields are mandatory!",, "Dorethy error message")
        End If
        ClearFields()
    End Sub
```

```
    Private Sub BtnReturn_Click(sender As Object, e As EventArgs) Handles
BtnReturn.Click
        DataGridView1.Columns.Clear()
        ClearFields()
        BtnGetCardData.Visible = True
        BtnScanCrd.Visible = True
        InfoHTC.Visible = False
        DataGridView1.Enabled = False
        Dim oForm As Menu
        oForm = New Menu()
        oForm.Show()

        Hide()
    End Sub
    Private Sub ClearFields()
        Name2DB.Text = ""
        Address2DB.Text = ""
        PostalC2DB.Text = ""
        City2DB.Text = ""
        PHONE2DB.Text = ""
        NFC2DB.Text = ""
    End Sub
    Private Sub LockFields()
        Name2DB.Enabled = False
        Address2DB.Enabled = False
        PostalC2DB.Enabled = False
        City2DB.Enabled = False
        PHONE2DB.Enabled = False
        InfoHTC.Visible = False
        ID_FS.Visible = False
    End Sub
    Private Sub OpenFields()
        Name2DB.Enabled = True
        Address2DB.Enabled = True
        PostalC2DB.Enabled = True
        City2DB.Enabled = True
        PHONE2DB.Enabled = True
    End Sub

    Private Sub BtnEdit_Click(sender As Object, e As EventArgs) Handles BtnEdit.Click
        Dim LBL As Integer
        V12 = "EDT"
        Name2DB.Text = DataGridView1.Rows(0).Cells(1).Value
        Address2DB.Text = DataGridView1.Rows(0).Cells(2).Value
        PostalC2DB.Text = DataGridView1.Rows(0).Cells(3).Value
        City2DB.Text = DataGridView1.Rows(0).Cells(4).Value
        PHONE2DB.Text = DataGridView1.Rows(0).Cells(5).Value
        NFC2DB.Text = DataGridView1.Rows(0).Cells(6).Value
        CLOC2DB.Text = DataGridView1.Rows(0).Cells(7).Value
        LBL = DataGridView1.Rows(0).Cells(8).Value
        If LBL = 0 Then LBleft.Text = "Has left"
        If LBL = 1 Then LBleft.Text = "Has arrived"
        InfoHTC.Visible = True
        DataGridView1.Enabled = True
    End Sub
End Class
```

Sales:

```
Imports Microsoft.VisualBasic
Imports Microsoft.VisualBasic.Interaction
Imports System.Drawing.Printing
Imports System.Data.SqlClient
Imports System.Data.Odbc
Imports System.Runtime.InteropServices
Imports System.Globalization
Imports Npgsql


Public Class Sales

    Private Const V11 As String = "Select * from customer, cmb"
    'Private Const V As String = ""
    Dim V12 As String
    Dim Pprice, Tprice, TSTR, FWS, W2P As Single
    Dim LOT As Integer
    Dim CID, CLOC, CACT, FChar, Adis, PAY, PREVA As String
    Dim CNAM, CADD, CPOSTAL, CCIT, CNFC, DAT2Day, BNR, CURRA As String


    Private Sub BtnReturn_Click() Handles BtnReturn.Click
        DataGridView1.Columns.Clear()
        Dim oForm As Menu
        oForm = New Menu()
        oForm.Show()

        Hide()
    End Sub

    ReadOnly time As DateTime = DateTime.Now
    ReadOnly format As String = "M/d/yy HH:mm"
    ReadOnly format2 As String = "yMdHHmm"


    Private Sub PayNow_Click(sender As Object, e As EventArgs) Handles PayNow.Click
        Dim Conn As NpgsqlConnection
        Dim Adpt As NpgsqlDataAdapter
        Dim cmd As NpgsqlCommand
        Dim LA, TTLBill As Single

        Conn = New NpgsqlConnection With {
                .ConnectionString = IniStart.DBstr
            }

        If PAY = "other" Then
            SalesSlip.Text = SalesSlip.Text & Chr(13) & "THE TOTAL AMOUNT OF THIS BILL
IS:  " & TTLsls.Text
            SalesSlip.Text = SalesSlip.Text & Chr(13) & Chr(13) & "OTHER PAYMENT
METHOD SELECTED"
            W2P = 1
        End If
        If PAY = "Dorethy Card" Then
            SalesSlip.Text = SalesSlip.Text & Chr(13) & "THE TOTAL AMOUNT OF THIS BILL
IS:  " & TTLsls.Text
            LA = TSTR + CURRA
```

```
            SalesSlip.Text = SalesSlip.Text & Chr(13) & Chr(13) & "PAYED WITH DORETHY
CARD. YOUR AVAILABLE SALDO IS: " & LA.ToString("€ #,###.00")
            W2P = 2
        End If
        SalesSlip.SaveFile(IniStart.SAsl & "\" & BNR & ".rtf")

        Conn.Open()

        Dim sql As String = V11
        Dim SID As String = BNR
        Dim Card As String = CNFC

        Adpt = New NpgsqlDataAdapter(sql, Conn)
        If W2P = 1 And TTLBill < 0 Then
            LA = TTLBill
            CURRA = TTLBill
        End If
        sql = ("INSERT INTO Public.payments (salesid, prevamount, waytopay, nfc,
amounttopay, amount) VALUES (" & Chr(39) & SID & Chr(39) & "::integer, " & Chr(39) &
CURRA & Chr(39) & "::money, " & Chr(39) & W2P & Chr(39) &
          "::integer, " & Chr(39) & Card & Chr(39) & "::text, " & Chr(39) & TTLBill &
Chr(39) & "::money, " & Chr(39) & LA & Chr(39) & "::money)")

        cmd = New NpgsqlCommand(sql, Conn)
        cmd.ExecuteNonQuery()

        Conn.Close()
        BtnReturn.PerformClick()
    End Sub


    Private Sub Sales_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        SCNPR.Visible = False
        ISCBpr.Visible = False
        AMT.Visible = False
        IMOU.Visible = False
        BtnNext.Visible = False
        CAC.Visible = False
        CA.Visible = False
        SalesSlip.Text = ""
        DataGridView1.Enabled = False
    End Sub
    Private Sub BtnNext_Click(sender As Object, e As EventArgs) Handles BtnNext.Click
        Dim Conn As NpgsqlConnection
        Dim Adpt As NpgsqlDataAdapter

        'Dim cmd As NpgsqlCommand

        Dim npgsqlConnection As New NpgsqlConnection
        Conn = npgsqlConnection
        Conn.ConnectionString = IniStart.DBstr
        Conn.Open()

        Dim sql As String = "Select * from products where products.ean = '" &
ISCBpr.Text & "'"

        Adpt = New NpgsqlDataAdapter(sql, Conn)
        Dim RS As New DataTable
        Try
            Adpt.Fill(RS)
```

```
            Dim ds As New DataSet("DataSetOne")
            ds.Tables.Add(RS)
            DataGridView1.DataSource = RS
            Conn.Close()
        Catch ex As Exception
            MsgBox("Some database error ocurred!",, "Dorethy message")
            ResetInput()
        End Try

        Conn.Close()
        GaDoor()
        If DataGridView1.Rows(0).Cells(1).Value = "" Then
            MsgBox("Can't find EAN, please retry",, "Dorethy message")

        End If
    End Sub
    Private Sub GaDoor()
        If ISCBpr.Text <> "" And IMOU.Text <> "" Then
            Dim Conn As NpgsqlConnection
            Dim Adpt As NpgsqlDataAdapter
            Dim cmd As NpgsqlCommand

            Conn = New NpgsqlConnection With {
                .ConnectionString = IniStart.DBstr
            }
            Conn.Open()

            Dim sql As String = V11
            Dim Amt As Integer = Convert.ToInt32(IMOU.Text)
            Dim SID As Integer = Convert.ToInt32(BNR)
            Dim iCID As Integer = Convert.ToInt32(CID)


            Adpt = New NpgsqlDataAdapter(sql, Conn)
            Pprice = Val(DataGridView1.Rows(0).Cells(4).Value)
            If ISCBpr.Text = "D010" Then
                ' Dim Cpp As Integer = Convert.ToDouble(TTLsls.Text)
                Dim Cpp = TTLsls.Text
                Pprice = Cpp
            End If

            Adis = DataGridView1.Rows(0).Cells(2).Value
            Tprice = Amt * Pprice
            TSTR += Tprice
            TTLsls.Text = TSTR

            sql = ("INSERT INTO public.sales (ean, custid, amount, salesid) VALUES ("
& Chr(39) & ISCBpr.Text & Chr(39) & "::text, " & Chr(39) & iCID & Chr(39) &
"::integer, " & Chr(39) & Amt & Chr(39) & "::integer, " & Chr(39) & SID & Chr(39) &
"::integer)")

            cmd = New NpgsqlCommand(Sql, Conn)
            cmd.ExecuteNonQuery()

            Conn.Close()
            V12 = ""

            SalesSlipAdd()
        End If
        ResetInput()
```

```
    End Sub


    Private Sub ResetInput()
        ISCBpr.Text = ""
        IMOU.Text = ""
        ISCBpr.Focus()
    End Sub



    Private Sub PWDCN_Click() Handles PWDCN.Click
        PWDCY.Visible = False
        CAC.Visible = False
        CA.Visible = False
        ScanFields()
        PAY = "other"
    End Sub

    Private Sub PWDCY_Click(sender As Object, e As EventArgs) Handles PWDCY.Click
        Dim Conn As NpgsqlConnection
        Dim Adpt As NpgsqlDataAdapter

        'Dim cmd As NpgsqlCommand

        Dim npgsqlConnection As New NpgsqlConnection
        Conn = npgsqlConnection
        Conn.ConnectionString = IniStart.DBstr
        Conn.Open()

        Dim sql As String = "Select * from payments where payments.nfc = '" & CNFC &
"' order by id desc"

        Adpt = New NpgsqlDataAdapter(sql, Conn)
        Dim RS As New DataTable
        Try
            Adpt.Fill(RS)
            Dim ds As New DataSet("DataSetOne")
            ds.Tables.Add(RS)
            DataGridView1.DataSource = RS
            Conn.Close()
        Catch ex As Exception
            MsgBox("Some database error ocurred!",, "Dorethy message")
        End Try

        Conn.Close()
        CAC.Visible = True
        CA.Visible = True
        PWDCN.Visible = False
        ScanFields()
        PREVA = DataGridView1.Rows(0).Cells(2).Value
        CURRA = DataGridView1.Rows(0).Cells(6).Value
        CA.Text = CURRA
        PAY = "Dorethy Card"
    End Sub
    Private Sub ScanFields()
        SCNPR.Visible = True
        ISCBpr.Visible = True
        AMT.Visible = True
```

```
        IMOU.Visible = True
        BtnNext.Visible = True

    End Sub

    Private Sub SalesSlipAdd() Handles MyBase.Load
        LOT = Microsoft.VisualBasic.Len(IMOU.Text)
        FWS = 6 - LOT
        AW32()
        SalesSlip.Text = SalesSlip.Text & FChar & IMOU.Text & Chr(32)
        LOT = Microsoft.VisualBasic.Len(ISCBpr.Text)
        FWS = 14 - LOT
        AW32()
        SalesSlip.Text = SalesSlip.Text & ISCBpr.Text & FChar & Chr(32)
        LOT = Microsoft.VisualBasic.Len(Adis)
        FWS = 40 - LOT
        AW32()
        SalesSlip.Text = SalesSlip.Text & Adis & FChar
        LOT = Microsoft.VisualBasic.Len(Pprice)
        FWS = 8 - LOT
        AW32()
        SalesSlip.Text = SalesSlip.Text & FChar & Pprice.ToString("€ #,###.00")
        LOT = Microsoft.VisualBasic.Len(Tprice)
        FWS = 12 - LOT
        AW32()
        SalesSlip.Text = SalesSlip.Text & FChar & Tprice.ToString("€ #,###.00") &
Chr(13)
    End Sub
    Private Sub AW32()
        FChar = ""
        For i = 1 To FWS
            FChar += Chr(32)
        Next
    End Sub
    Private Sub S4LOC_Click(sender As Object, e As EventArgs) Handles S4LOC.Click
        Dim Conn As NpgsqlConnection
        Dim Adpt As NpgsqlDataAdapter

        'Dim cmd As NpgsqlCommand

        Dim npgsqlConnection As New NpgsqlConnection
        Conn = npgsqlConnection
        Conn.ConnectionString = IniStart.DBstr
        Conn.Open()

        Dim sql As String = "Select * from customer where customer.clocation = '" &
Val(S4CL.Text) & "'"

        Adpt = New NpgsqlDataAdapter(sql, Conn)
        Dim RS As New DataTable
        Try
            Adpt.Fill(RS)
            Dim ds As New DataSet("DataSetOne")
            ds.Tables.Add(RS)
            DataGridView1.DataSource = RS
            Conn.Close()
        Catch ex As Exception
            MsgBox("Some database error ocurred!",, "Dorethy message")
        End Try
```

```
        Conn.Close()

        If DataGridView1.Rows(0).Cells(1).Value = "" Then
            MsgBox("this location is not registred",, "Dorethy message")
        ElseIf DataGridView1.Rows(0).Cells(1).Value <> "" Then
            CID = DataGridView1.Rows(0).Cells(0).Value
            CNAM = DataGridView1.Rows(0).Cells(1).Value
            CADD = DataGridView1.Rows(0).Cells(2).Value
            CPOSTAL = DataGridView1.Rows(0).Cells(3).Value
            CCIT = DataGridView1.Rows(0).Cells(4).Value
            CNFC = DataGridView1.Rows(0).Cells(6).Value
            CLOC = DataGridView1.Rows(0).Cells(7).Value
            CACT = DataGridView1.Rows(0).Cells(8).Value
            DAT2Day = time.ToString(format)
            BNR = time.ToString(format2)
            SalesSlip.Text = "Dorethy camping" & Chr(13) & "Mountain View 7001" &
Chr(13) & "123456 RelayCity" & Chr(13) & Chr(13) & "Date: " & DAT2Day & Chr(13) &
Chr(13) &
                "Salesnr: " & BNR & Chr(13) & Chr(13) & CNAM & Chr(13) & CADD &
Chr(13) & CPOSTAL & "  " & CCIT & Chr(13) & Chr(13)
            SalesSlip.Text = SalesSlip.Text & "Amount EAN           Discription
Price            Total" & Chr(13)
            S4LOC.Enabled = False
            S4CL.Enabled = False
            CAMPLOC.Enabled = False
        End If

        If CACT <> 1 Then
            MsgBox("This location is not occupied. Sales is impossible!!",, "Dorethy
WARNING!!")
            BtnReturn.PerformClick()
            GoTo EndCorrect
        End If

        If Len(CNFC) < 3 Then
            PWDCN.PerformClick()
        End If

EndCorrect:
    End Sub

End Class
```

Check out:

```
Imports Microsoft.VisualBasic
Imports Microsoft.VisualBasic.Interaction
Imports System.Drawing.Printing
Imports System.Data.SqlClient
Imports System.Data.Odbc
Imports System.Runtime.InteropServices
Imports Npgsql


Public Class CheckOut
    Private Const V11 As String = "Select * from customer, cmb"
    Dim TSTR, W2P As Single
    Dim CID, CLOC, CACT, PAY, PREVA As String
```

```
    Private Sub CPO_Click(sender As Object, e As EventArgs) Handles CPO.Click
        Dim Conn As NpgsqlConnection
        Dim Adpt As NpgsqlDataAdapter
        Dim cmd As NpgsqlCommand
        Dim LA, TTLBill As Single

        Conn = New NpgsqlConnection With {
                .ConnectionString = IniStart.DBstr
            }

        If PAY = "other" Then
            W2P = 1
        End If
        If PAY = "Dorethy Card" Then
            SalesSlip.Text = SalesSlip.Text & Chr(13) & "Your Dorethy card has a value
of:  " & TTLsls.Text
            LA = TSTR + CURRA
            SalesSlip.Text = SalesSlip.Text & Chr(13) & Chr(13) & "If your card has a
negative value, you'll be payed by us. Every positive value is a debt to us: " &
LA.ToString("€ #,###.00")
            If LA <= 0 Then
                SalesSlip.Text = SalesSlip.Text & Chr(13) & Chr(13) & "We refund you:
" & LA.ToString("€ #,###.00")
            Else SalesSlip.Text = SalesSlip.Text & Chr(13) & Chr(13) & "You pay us: "
& LA.ToString("€ #,###.00")
                LA -= LA
            End If
            W2P = 2
        End If
        SalesSlip.SaveFile(IniStart.SAsl & "\" & BNR & ".rtf")

        Conn.Open()

        Dim sql As String = V11
        Dim SID As String = BNR
        Dim Card As String = CNFC

        Adpt = New NpgsqlDataAdapter(sql, Conn)
        If W2P = 1 And TTLBill < 0 Then
            LA = TTLBill
            CURRA = TTLBill
        End If
        sql = ("INSERT INTO Public.payments (salesid, prevamount, waytopay, nfc,
amounttopay, amount) VALUES (" & Chr(39) & SID & Chr(39) & "::integer, " & Chr(39) &
CURRA & Chr(39) & "::money, " & Chr(39) & W2P & Chr(39) &
          "::integer, " & Chr(39) & Card & Chr(39) & "::text, " & Chr(39) & TTLBill &
Chr(39) & "::money, " & Chr(39) & LA & Chr(39) & "::money)")

        cmd = New NpgsqlCommand(sql, Conn)
        cmd.ExecuteNonQuery()

        CNFC = ""
        CACT = 0
        CLOC = 0

        sql = "update public.customer set nfc = " & Chr(39) & CNFC & Chr(39) & ",
active = " & Chr(39) & CACT & Chr(39) & ", clocation = " & Chr(39) & CLOC & Chr(39) &
" where id=" & Chr(39) & CID & Chr(39) & " "

        cmd = New NpgsqlCommand(sql, Conn)
```

```
        cmd.ExecuteNonQuery()

        Conn.Close()
        BtnReturn.PerformClick()
    End Sub

    Private Sub BtnReturn_Click(sender As Object, e As EventArgs) Handles
BtnReturn.Click
        DataGridView1.Columns.Clear()
        Dim oForm As Menu
        oForm = New Menu()
        oForm.Show()

        Hide()
    End Sub

    Dim CNAM, CADD, CPOSTAL, CCIT, CNFC, DAT2Day, BNR, CURRA As String
    Private Sub CheckOut_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        DataGridView1.Enabled = False
    End Sub

    ReadOnly time As DateTime = DateTime.Now
    ReadOnly format As String = "M/d/yy HH:mm"
    ReadOnly format2 As String = "yMdHHmm"

    Private Sub S4LOC_Click(sender As Object, e As EventArgs) Handles S4LOC.Click
        Dim Conn As NpgsqlConnection
        Dim Adpt As NpgsqlDataAdapter

        'Dim cmd As NpgsqlCommand

        Dim npgsqlConnection As New NpgsqlConnection
        Conn = npgsqlConnection
        Conn.ConnectionString = IniStart.DBstr
        Conn.Open()

        Dim sql As String = "Select * from customer where customer.clocation = '" &
Val(S4CL.Text) & "'"

        Adpt = New NpgsqlDataAdapter(sql, Conn)
        Dim RS As New DataTable
        Try
            Adpt.Fill(RS)
            Dim ds As New DataSet("DataSetOne")
            ds.Tables.Add(RS)
            DataGridView1.DataSource = RS
            Conn.Close()
        Catch ex As Exception
            MsgBox("Some database error ocurred!",, "Dorethy message")
        End Try

        Conn.Close()

        If DataGridView1.Rows(0).Cells(1).Value = "" Then
            MsgBox("this location is not registred",, "Dorethy message")
        ElseIf DataGridView1.Rows(0).Cells(1).Value <> "" Then
            CID = DataGridView1.Rows(0).Cells(0).Value
            CNAM = DataGridView1.Rows(0).Cells(1).Value
            CADD = DataGridView1.Rows(0).Cells(2).Value
            CPOSTAL = DataGridView1.Rows(0).Cells(3).Value
```

```
            CCIT = DataGridView1.Rows(0).Cells(4).Value
            CNFC = DataGridView1.Rows(0).Cells(6).Value
            CLOC = DataGridView1.Rows(0).Cells(7).Value
            CACT = DataGridView1.Rows(0).Cells(8).Value
            DAT2Day = time.ToString(Format)
            BNR = time.ToString(format2)
            SalesSlip.Text = "Dorethy camping" & Chr(13) & "Mountain View 7001" &
Chr(13) & "123456 RelayCity" & Chr(13) & Chr(13) & "Date: " & DAT2Day & Chr(13) &
Chr(13) &
                "Salesnr: " & BNR & Chr(13) & Chr(13) & CNAM & Chr(13) & CADD &
Chr(13) & CPOSTAL & "  " & CCIT & Chr(13) & Chr(13)
            SalesSlip.Text = SalesSlip.Text & "CHECKOUT" & Chr(13)
            If Len(CNFC) < 3 Then
                SalesSlip.Text = SalesSlip.Text & "No Dorethy card used" & Chr(13) &
Chr(13) & "On behalf of the Dorethy team, THANK YOU for visiting us!" & Chr(13) &
Chr(13) & "Have a save trip home!"
                'Hier nog toevoegen Bewaren afschrift, opslaan en de juiste waarden op
0 zetten.
                PAY = "other"
                GoTo EndCheckout
            End If

            S4LOC.Enabled = False
            S4CL.Enabled = False
            CAMPLOC.Enabled = False
            GetValueCard()
EndCheckout:
        End If
    End Sub

    Private Sub GetValueCard()
        Dim Conn As NpgsqlConnection
        Dim Adpt As NpgsqlDataAdapter
        Dim SaldoP As Single
        'Dim cmd As NpgsqlCommand

        Dim npgsqlConnection As New NpgsqlConnection
        Conn = npgsqlConnection
        Conn.ConnectionString = IniStart.DBstr
        Conn.Open()

        Dim sql As String = "Select * from payments where payments.nfc = '" & CNFC &
"' order by id desc"

        Adpt = New NpgsqlDataAdapter(sql, Conn)
        Dim RS As New DataTable
        Try
            Adpt.Fill(RS)
            Dim ds As New DataSet("DataSetOne")
            ds.Tables.Add(RS)
            DataGridView1.DataSource = RS
            Conn.Close()
        Catch ex As Exception
            MsgBox("Some database error ocurred!",, "Dorethy message")
        End Try

        Conn.Close()
        CAC.Visible = True
        CA.Visible = True
        PREVA = DataGridView1.Rows(0).Cells(2).Value
```

```
        CURRA = DataGridView1.Rows(0).Cells(6).Value
        CA.Text = CURRA
        PAY = "Dorethy Card"
        SaldoP = (Val(CURRA) - Val(CURRA)) - Val(CURRA)
        'Saldo = (CA.Text - CA.Text) + CA.Text
        If SaldoP > 0 Then IN_pay.Text = "You owe the customer: " & SaldoP.ToString("€
#,###.00")
        If SaldoP <= 0 Then IN_pay.Text = "The customer has to pay: " &
SaldoP.ToString("€ #,###.00")
    End Sub
End Class
```

Inistart:

```
Imports Microsoft.VisualBasic
Imports Microsoft.VisualBasic.Interaction
Imports System.Data.SqlClient
Imports System.Data.Odbc
Imports Npgsql

Public Class IniStart
    ReadOnly path As String =
Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments)

    Public Shared Property DBstr As String
    Public Shared Property AUpw As String
    Public Shared Property SAsl As String


    Private Sub IniStart_Load(sender As Object, e As EventArgs) Handles MyBase.Load

        EnterLocation()

    End Sub

    Private Sub EnterLocation()

        My.Computer.FileSystem.CreateDirectory(path & "\Dor")
            My.Computer.FileSystem.CreateDirectory(path & "\Dor\salesslips")
            BtnWRT.Visible = False


    End Sub

    Private Sub BtnCHK_Click(sender As Object, e As EventArgs) Handles BtnCHK.Click
        GLDBstr.Text = "Host=" & IPHN.Text & ";Username=" & IPUN.Text & ";Password=" &
Chr(34) & IPPW.Text & Chr(34) & ";Database=" & IPDB.Text
        BtnWRT.Visible = True
    End Sub

    Private Sub BtnWRT_Click(sender As Object, e As EventArgs) Handles BtnWRT.Click
        Dim file As System.IO.StreamWriter
        Try
            DBstr = GLDBstr.Text
            AUpw = IPPWAU.Text
            SAsl = path & "\Dor\salesslips"
            file = My.Computer.FileSystem.OpenTextFileWriter(path & "\Dor\dps.ini",
True)
            file.WriteLine(DBstr)
```

```
            file.WriteLine(AUpw)
            file.WriteLine(SAsl)
            file.Close()
            MsgBox("Dorethy will be closed. Restart for loading your settings!")
            End
        Catch ex As Exception
            MsgBox("Failed to write ini file.")
        End Try
    End Sub

End Class
```

Report:

```
Imports Microsoft.VisualBasic
Imports Microsoft.VisualBasic.Interaction
Imports System.Data.SqlClient
Imports System.Data.Odbc
Imports Npgsql
Public Class Report
    Private Const V11 As String = "0000"
    Private Const V12 As String = "2359"
    'Dim V12 As String
    Dim A1, Dat2Day, RepSLSst, RepSLSsp As String
    ReadOnly time As DateTime = DateTime.Now

    ReadOnly format3 As String = "yMd"

    Private Sub BtnShow_Click(sender As Object, e As EventArgs) Handles BtnShow.Click
        Dim Conn As NpgsqlConnection
        Dim Adpt As NpgsqlDataAdapter

        'Dim cmd As NpgsqlCommand

        Dim npgsqlConnection As New NpgsqlConnection
        Conn = npgsqlConnection
        Conn.ConnectionString = IniStart.DBstr
        Conn.Open()
        Dim sql As String
        Dat2Day = time.ToString(format3)
        RepSLSst = Dat2Day & V11
        RepSLSsp = Dat2Day & V12
        If LBTable.Text = "Customer" Then
            A1 = "Select * from customer"
        End If
        If LBTable.Text = "Products" Then
            A1 = "Select * from products"
        End If
        If LBTable.Text = "Sales Today" Then
            A1 = "Select * from sales, payments, products where payments.salesid > '"
& RepSLSst & "' and sales.salesid = payments.salesid and sales.ean=products.ean"
            'and payments.salesid <= '" & RepSLSsp & "'"
        End If

        sql = A1

        Adpt = New NpgsqlDataAdapter(sql, Conn)

        Dim RS As New DataTable
```

```
        Adpt.Fill(RS)
            Dim ds As New DataSet("DataSetOne")
            ds.Tables.Add(RS)
            DataGridView1.DataSource = RS
            Conn.Close()
    End Sub

    Private Sub BtnReturn_Click(sender As Object, e As EventArgs) Handles
BtnReturn.Click
        DataGridView1.Columns.Clear()
        Dim oForm As Menu
        oForm = New Menu()
        oForm.Show()

        Hide()
    End Sub
End Class
```

Sales slip

Within the form Sales we save the entered data in a richtextbox and store it on the hard drive.

It's done with the line: `SalesSlip.SaveFile("f:\Dorethy\Sales\" & BNR & ".rtf")`

Make sure the hard drive location is available. Unique sales numbers are generated by the software based on date and time. The generated document looks like this:

```
Dorethy camping
Mountain View 7001
123456 RelayCity

Date: 7-4-21 21:12

Salesnr: 21742112

Sam and Jill
The Mill 309
654321  Somewhere

Amount EAN          Discription                              Price            Total
     1 23033344     Peanut butter                            € 1,95       € 1,95

THE TOTAL AMOUNT OF THIS BILL IS:  1,95

PAYED WITH DORETHY CARD. YOUR AVAILABLE SALDO IS: -€ 16,46
```

If you wat to change the layout, you have to change the lines starting with `SalesSlip.Text =`

`SalesSlip.Text & …`

If you need more reports without paying for reporting tools, you can make them on the same way as Dorethy ‘s sales slip.

The data we used

For testing the software we used some product codes for testing. These codes are presented in the following table:

EAN	Barcode	Name	Net price	Sales price
D9990001		One night stay small tent	€ 12.33	€ 20.00
D9990002		One night stay large tent	€ 29.66	€ 40.00
7300400481823		Wasa knackebrot	€ 2.15	€ 3.08
23033344		Peanut butter	€ 1.35	€ 1.95
8711000234853		English tea 40 bags	€ 1.32	€ 2.35
80023357		Pesto 90gr	€ 1.07	€ 1.88
8717333715090		Hand soap 500ml	€ 0.98	€ 1.55
D010		Bill payed with other method	€ -	€ -
D910		Card update with € 10,00	€ -10.00	€ -10.00
D920		Card update with € 20,00	€ -20.00	€ -20.00
D930		Card update with € 30,00	€ -30.00	€ -30.00
D940		Card update with € 40,00	€ -40.00	€ -40.00

On behalf of

Dorethy, Sebastian and myself. Many thanks for buying this book and using this project to educate yourself. The complete software package is available for download on Nuget.

We wish you all the best and happy coding. Take care of yourself and your relatives.

Johan van Hal

Dear Sebastian,

Congratulations with passing successfully your VWO exam. I wish you a great time during your Study at Tilburg University!

Johan van Hal