Learning software for basic accounting

	Vanessa Schmelzer *WI21ZAS* *12.06.2023*

Introduction

.In view of the degree from the higher vocational school for business informatics, I was allowed to decide on a final project. I decided to use a basic accounting learning software because more and more people are having trouble finding their way around.

This should make it easier for customers to get used to accounting and help to refresh their knowledge.

Customers can train their knowledge with the help of easily understandable theory and tasks and the language can be changed from German to English.

For each question there is a detailed solution that knows how to fill in the gaps.

Criteria not implemented (see specifications)

- Intro video that plays at the beginning. Should serve as a little "fun" to bring more life into the project.

- Easy to use, clear dropp menu. After consideration and task creation, the menu could not be realized as shown on the wireframe.

The parts not produced due to time constraints will be reconsidered.

Creation of the program

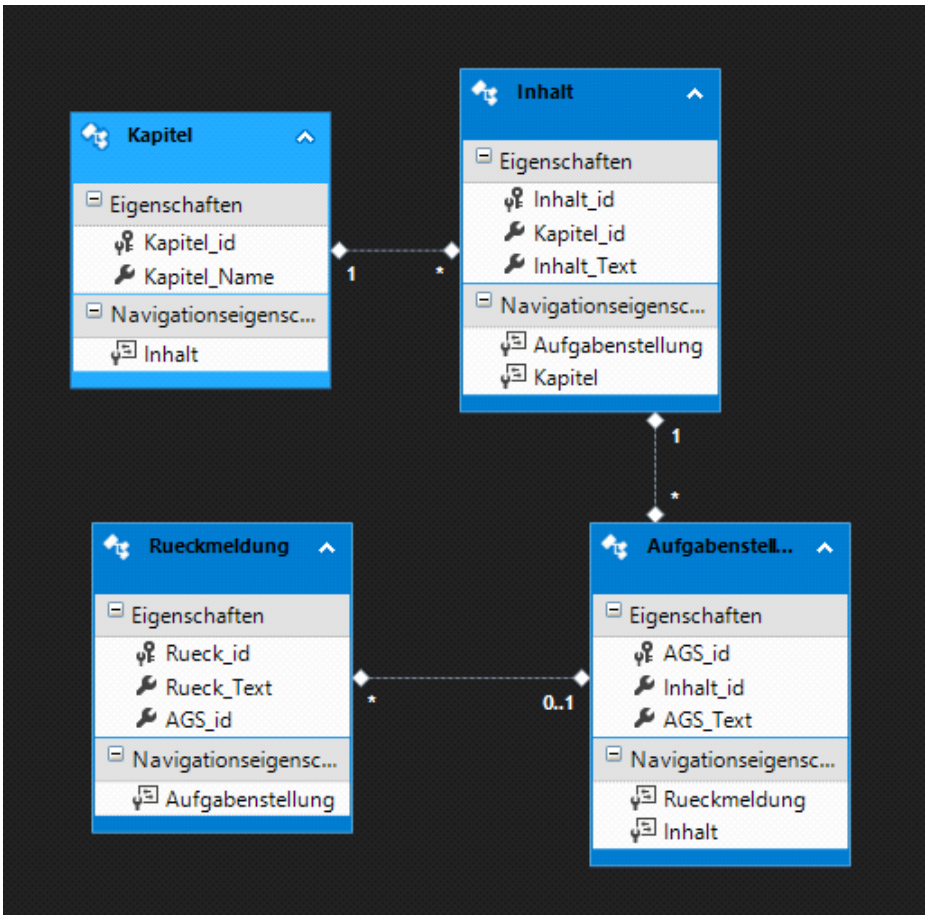
Framework

For the framework was used:

- Entity Framework: “Entity Framework Core is a modern object database mapper for .NET. It supports LINQ queries, change tracking, updates, and schema migrations. EF Core works with many databases, including SQL Database (on-premises and Azure), SQLite, MySQL, PostgreSQL, and Azure Cosmos DB.” (Microsoft, 2022)

Entity Framework is used in this project for data traffic between the database and the queries of the code

Database


Kapitel
Eigenschaften
Kapitel_id
Kapitel_Name
Navigationseigensc...
Inhalt
1
*
Inhalt
Eigenschaften
Inhalt_id
Kapitel_id
Inhalt_Text
Navigationseigensc...
Aufgabenstellung
Kapitel
1
*
Rueckmeldung
Eigenschaften
Rueck_id
Rueck_Text
AGS_id
Navigationseigensc...
Aufgabenstellung
*
0..1
Aufgabenstell...
Eigenschaften
AGS_id
Inhalt_id
AGS_Text
Navigationseigensc...
Rueckmeldung
Inhalt

```
CREATE TABLE Kapitel (
  Kapitel_id INT IDENTITY (1, 1) PRIMARY KEY,
  Kapitel_Name nvarchar(100) not null,
);

CREATE TABLE Inhalt (
  Inhalt_id INT IDENTITY (1, 1) PRIMARY KEY,
  Kapitel_id int not null,
  Inhalt_Text ntext not null,
  constraint fk_KapitelInhalt foreign key (Kapitel_id) references Kapitel(Kapitel_id)
);

CREATE TABLE Aufgabenstellung (
  AGS_id INT IDENTITY (1, 1) PRIMARY KEY,
  --Kapitel_id int not null,
  Inhalt_id int not null,
  AGS_Text ntext not null,
 -- CONSTRAINT fk_KapitelAufgabenstellung FOREIGN KEY (Kapitel_id) REFERENCES Kapitel(Kapitel_
    CONSTRAINT fk_InhaltAufgabenstellung FOREIGN KEY (Inhalt_id) REFERENCES Inhalt(Inhalt_id)
);

CREATE TABLE Rueckmeldung (
  Rueck_id INT IDENTITY (1, 1) PRIMARY KEY,
  Rueck_Text ntext not null,
  AGS_id int foreign key references Aufgabenstellung(AGS_id)
```

The necessary tables are created here and are required for the learning software.

The tables are linked together to form a chain for simplified retrieval.

Back-End-Code

Language setting

```
2 Verweise
private void LanguageChange(object sender, RoutedEventArgs e)
{
    Button clickedbutton = sender as Button;
    if (clickedbutton.Content.Equals("Deutsch")) {
        MyDataContext.switchCulture("de-DE");
    }
    else
    {
        MyDataContext.switchCulture("en-US");
    }

    Window m = new MainWindow();
    m.Show();
    this.Close();
}
```

Hier wird eine Abfrage auf die Sprache des Programmes abgerufen entweder zu Deutsch oder durch Knopfdruck wird sie mit der englischen “Englisch” ersetzt. Nachdem die Abfrage erfolgt,

wird das Window1 geschlossen.Here a query for the language of the program is called either in German or by pressing a button it is replaced with the English "English". After the query is made, Window1 is closed.

```
3 Verweise
public static string myculturеinfo { get; private set; } = "de-DE";
0 Verweise
public MyDataContext()
```

The database default is set to the German database “Deutsch”.

```
2 Verweise
private void chosenKapitel(object sender, RoutedEventArgs e)
{
    MenuItem m = (MenuItem)sender;
    // MessageBox.Show((string)m.Header);
    Window w = new SubWindow(k.FirstOrDefault(x => x.Kapitel_Name.Equals(m.Header)).Kapitel_id,this);
    w.Show();
    w.Focus();
    this.Hide();

}
```

This figure describes the jump from the MainWindow to the SubWindow.

After selecting the desired chapter, the SubWindow is called and placed in focus while the MainWindow is hidden.

SubWindow – Task Window

```
if (MyDataContext.myCultureInfo == "de-DE")
{
    hausarbeit_DEEntities ctx = new hausarbeit_DEEntities();

    k = ctx.Kapitel.ToList();

    foreach (Kapitel kitem in k)
    {
        MenuItem m = new MenuItem();
        m.Header = kitem.Kapitel_Name;
        m.Click += new RoutedEventHandler(chosenKapitel);
        menuItems.Add(m);
    }
    menuItems.ForEach(x => menü.Items.Add(x));

}
else
{
    hausarbeit_ENEntitie ctx = new hausarbeit_ENEntitie();

    k = ctx.Kapitel.ToList();

    foreach (Kapitel kitem in k)
    {
        MenuItem m = new MenuItem();
        m.Header = kitem.Kapitel_Name;
        m.Click += new RoutedEventHandler(chosenKapitel);
        menuItems.Add(m);
    }
    menuItems.ForEach(x => menü.Items.Add(x));
}
```

In this picture you can see the menu being created and using the chapter parts drawn from the database, each menu item is given the name of the chapter.

```
List<int> AgsId = new List<int>();
selectedAufgaben.ForEach(x => AgsId.Add(x.AGS_id));
MessageBox.Show(rueckmeldungen.FirstOrDefault(x => x.AGS_id == selectedAufgaben[_currentQuestionPosition].AGS_id).Rueck_Text);
```

This image describes the output of the solutions through a message box that is filled with the data set that has been dragged.

```
1 Verweis
private void zuruck(object sender, RoutedEventArgs e)
{
    _mainWindow.Show();
    this.Close();
}
```

Here you can see the connection of a button to navigate back to the MainWindow after successfully declaring a task set as done.

```
private void Back(object sender, RoutedEventArgs e)
{
    if (_currentInhalt > 0)
    {
        _currentInhalt--;
    }

    Inhalt.Text = selectedInhalt[_currentInhalt].Inhalt_Text;
}

1 Verweis
private void Forth(object sender, RoutedEventArgs e)
{
    if (_currentInhalt < selectedInhalt.Count()-1)
    {
        _currentInhalt++;
    }

    Inhalt.Text = selectedInhalt[_currentInhalt].Inhalt_Text;
}
```

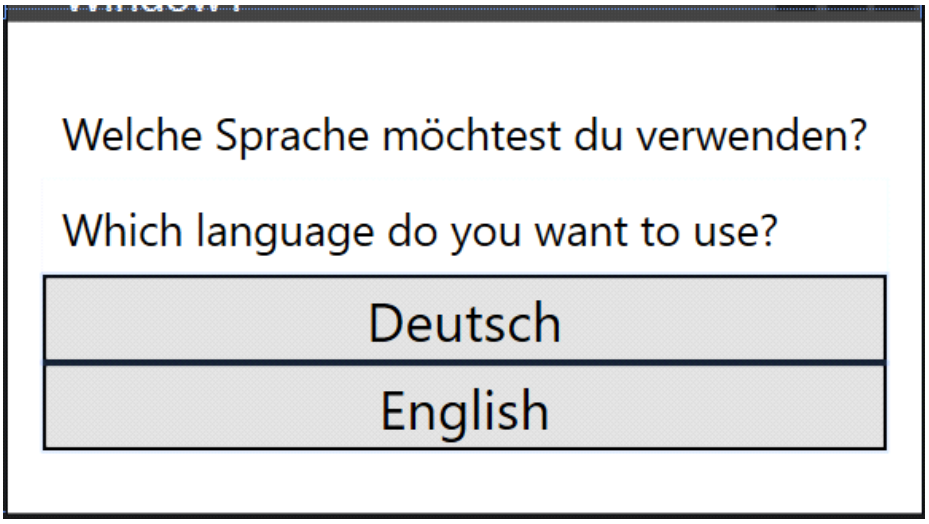
This picture describes the coding of the navigation in the SubWindow. It can be moved forwards and backwards until there are no more data records for the respective chapter and stops.

```
1 Verweis
private void NextQuestion(object sender, RoutedEventArgs e)
{

    if (_currentQuestionPosition < selectedAufgaben.Count()-1)
    {
        _currentQuestionPosition++;
    }
    else
    {
        _currentQuestionPosition = 0;
    }

    Frage.Text = selectedAufgaben.ElementAt(_currentQuestionPosition).AGS_Text;
```

As with the picture above, this describes the navigation forwards and backwards of the tasks, but this is repeated with each click.

Front-End-Code

Window1 – language selection




```
<StackPanel VerticalAlignment="Center" HorizontalAlignment="Center">
    <Label Content="Welche Sprache möchtest du verwenden?" />
    <Label Content="Which language do you want to use? "/>
    <Button Content="Deutsch" BorderThickness="1" BorderBrush="Black"
            Width="auto" Height="auto" FontSize="15" Click="LanguageChange" />
    <Button Content="English" BorderThickness="1" BorderBrush="Black"
            Width="auto" Height="auto" FontSize="15" Click="LanguageChange" />
</StackPanel>
```

The questions shown here to query the language were asked with labels and buttons could be used as an answer option with a simple click.


```
<Grid.ColumnDefinitions>
    <ColumnDefinition Width="90"/>
    <ColumnDefinition Width="*"/>
    <ColumnDefinition Width="90"/>
</Grid.ColumnDefinitions>

<Grid Grid.Column="1">
    <Grid.RowDefinitions>
        <RowDefinition Height="50"/>
        <RowDefinition Height="350"/>
        <RowDefinition Height="50"/>
        <RowDefinition Height="250"/>
        <RowDefinition Height="*"/>
    </Grid.RowDefinitions>
```

Here you can see that the program was split into 2 and in the middle column it could be divided into 5 more parts. Estimates were made to approximate the wireframe.

```
<Frame Grid.Column="1" Grid.Row="1" BorderBrush="Purple"
 Margin="100,0" >
    <Frame.Background...>
</Frame>
<!--<MediaElement HorizontalAlignment="Left" Height="350" Grid.Row="1" Margin="100,0,0,0" VerticalAlignment="Top
<StackPanel Grid.Column="1" Grid.Row="1" HorizontalAlignment="Center" >
    <Label FontSize="26">Hallo und herzlich willkommen zu meiner Lernsoftware für Basis-Rechnungswesen.</Label>
    <TextBlock TextWrapping="Wrap" Margin="100,10,100,0" FontSize="15">In diesem Programm findest du eine Vielfalt an Theorie mit einer Misc
    <Label FontSize="20" Margin="100,40,100,0" HorizontalAlignment="Center">Hello and welcome to my learning sof
    <TextBlock TextWrapping="Wrap" Margin="100,10,100,0" FontSize="15">In this program, you will find a variety of theory along with a mixture o
    <Label FontSize="20" HorizontalAlignment="Center" >Viel Glück! Good luck!</Label>
</StackPanel>
```

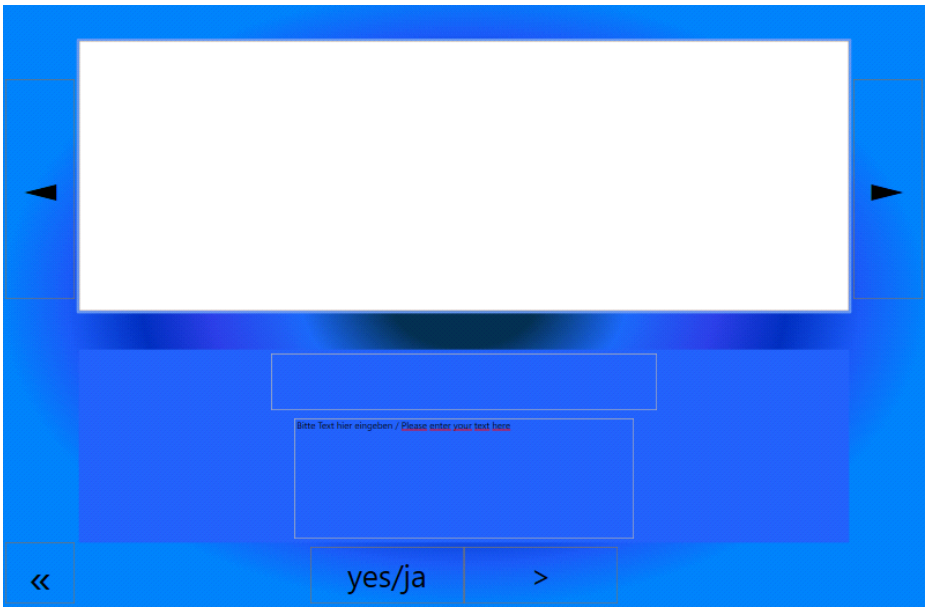
With the help of a frame, a help bridge could be built that could be colored at the end. It was intended

as a placeholder for the commented-out actual video. Instead, the video content could be made clear and recognizable with the help of labels and text blocks.

```
<Grid Grid.Row="3">
    <Grid.Background>
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="*"/>

    </Grid.ColumnDefinitions>
    <Grid.RowDefinitions>
        <RowDefinition Height="*"/>

    </Grid.RowDefinitions>

    <Menu x:Name="menü"
        Width="auto"
        Height="50"
        VerticalAlignment="Top"
        HorizontalAlignment="Center"
        Background="White"
        BorderBrush="Blue"
        BorderThickness="1"
          FontSize="10"
          Grid.Column="0"
          >

        <Menu.ItemsPanel>
            <ItemsPanelTemplate>
                <VirtualizingStackPanel Orientation="Horizontal" />
            </ItemsPanelTemplate>
        </Menu.ItemsPanel>
```

In the middle row, the 4th splate was initially broken down into a grid in which the menu is implemented, which is created by the required back-end code. With this front end code, the menu is

aligned horizontally and each column gets the space it needs.




```
<Grid>
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="100"/>
        <ColumnDefinition Width="*"/>
        <ColumnDefinition Width="100"/>
    </Grid.ColumnDefinitions>

    <Button Margin="5,700,5,5" Background="Transparent" FontSize="60" Click="zuruck"  >«</Button>
    <Button Content="◀" Margin="5,100,5,400" Background="Transparent" FontSize="60" Click="Back" />
    <Button Content="▶" Margin="5,100,5,400" Background="Transparent" FontSize="60" Grid.Column="2" Click="Forth" />
```

An outer field can be set here by setting the buttons for navigating back and forth and content in the middle.

```
<Grid Grid.Column="1">
    <Grid.RowDefinitions>
        <RowDefinition Height="50"/>
        <RowDefinition Height="350"/>
        <RowDefinition Height="50"/>
        <RowDefinition Height="250"/>
        <RowDefinition Height="*"/>
    </Grid.RowDefinitions>

    <Frame Grid.Column="1" Grid.Row="1" BorderBrush="Purple" Background="PeachPuff"  ></Frame>
    <Frame Grid.Column="1" Grid.Row="3" BorderBrush="Purple" Background="#FF225FF5" ></Frame>

    <Button Content="yes/ja" Grid.Row="4" Margin="300,5,500,5"
            Background="Transparent" FontSize="40" Click="QuestionHelp" />

    <TextBox Grid.Row="1"  x:Name="Inhalt"></TextBox>

    <Button Content=">" Grid.Row="4" Margin="500,5,300,5"
            Background="Transparent" FontSize="40" Click="NextQuestion" />
```

As with MainWindow, the middle row could be used in frames. In the bottom row are the buttons for submitting the answer, which calls up the event and the MessageBox with the solutions stands out.

Another button is next door to change the tasks. The TextBox is formed in the upper part of the middle row and in it the dragged content records are visualized.

```
<Grid Grid.Row="3">
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="*"/>
        <ColumnDefinition Width="2*"/>
        <ColumnDefinition Width="*"/>
    </Grid.ColumnDefinitions>
    <Grid.RowDefinitions>
        <RowDefinition Height="*"/>
        <RowDefinition Height="2*"/>
    </Grid.RowDefinitions>

    <TextBox Grid.Column="1" Background="Transparent" Margin="0,5" IsReadOnly="True" x:Name="Frage" ></TextBox>

    <TextBox TextWrapping="Wrap" SpellCheck.IsEnabled="True" Name="myTextBox" Background="Transparent"
    Grid.Column="1" Grid.Row="1" Margin="30,5" >Bitte Text hier eingeben / Please enter your text here</TextBox>
```

This last picture shows a division in the middle column and in the 3rd row. In this, the text boxes are subdivided and defined. The "Question" text box is the display of the task set which cannot be changed. The "myTextBox" text box is the field in which the user's answers are written.

User manual

When starting the program, please select your preferred language.

After selecting the language, you will be greeted with a friendly user guide. For first use please read.

By clicking on the menu, you can refer to a particular chapter, which opens in a new window.

Upper middle box represents information material. Information sets can be changed using the side arrows. Side-directed arrows stop at the end and beginning of a record Please navigate forwards or backwards.

Middle narrow box represents the task field which can be navigated forward with the “>” button, the task set repeats itself by clicking the “>” button repeatedly.

Own answers can be written in the field under the task set and the "yes/ja" button can be used to open another window with a detailed solution for comparison.

The button placed at the lower left corner leads back to the main window to select a new chapter.

Citation

-Schmolke Deitermann, Industrielles Rechnungswesen IKR , Westermann Gruppe

-Deitermann Rückwart, Rechnungswesen für Berufsfachschulen, Winklers

- https://stackoverflow.com

- https://www.steuertipps.de/lexikon/a/abschreibung-linear

- https://studyflix.de/wirtschaft/lineare-abschreibung-1192

- https://www.microtech.de/ HYPERLINK "https://www.microtech.de/erp-wiki/bilanz/"erp HYPERLINK "https://www.microtech.de/erp-wiki/bilanz/"-wiki/ HYPERLINK "https://www.microtech.de/erp-wiki/bilanz/"bilanz HYPERLINK "https://www.microtech.de/erp-wiki/bilanz/"/#:~:text=Die%20Bilanz%20ist%20die%20Gegenüberstellung,die%20Bilanz%20Hauptbestandteil%20eines%20Jahresabschlusses.

Statutory declaration

"I certify that I have completed the work independently and without using any tools other than those specified. I have clearly marked all passages that are taken literally or analogously from publications or other sources (including the Internet) as such. The work has not yet been published and has not yet been submitted as a coursework for recognition or evaluation.

I am aware that violations of these requirements will result in the work being graded as "insufficient" and in the denial of the desired performance record."

Datum	Unterschrift