

Hausarbeit
im Bildungsgang
„Staatlich geprüfter Wirtschaftsinformatiker/in“

Thema der Arbeit
Entwurf und Implementierung eines Texteditors

Vorgelegt von: Khaled Kalo
Klasse: HBFS-WI20Z1A
Adresse: Mainzer St.33
Ort: 66333 Saarbrücken
E-Mail: Khaledkalo17@gmail.com
Abgabetermin: 30.05.2022
Betreuer/in: Herr Friedrich

Inhaltsverzeichnis

1	Einleitung	2
1.1	Beschreibung des Projektes	2
2	Einsatz	2
2.1	Anwendung	2
2.2	Zielgruppen	2
3	Muss- und Kann-Kriterien mit Aufwandsabschätzung	2
3.1	Muss-Kriterien	2
3.2	Kann-Kriterien	3
4	Funktionen	3
5	Produktübersicht	4
5.1	Use-Case Diagramm	4
5.2	Datenbank Entwurf	5
5.3	Git-Repository	5
5.4	Gantt-Diagramm	5
5.5	Skizze von GUI	6
6	Entwicklungsumgebung	6
6.1	Software	6
6.2	Hardware	6
6.3	Orgware	6
7	Dokumentation	6
8	Benutzerhandbuch	7
8.1	Texteditor Beschreibung	8
8.1.1	Menu Datei	9
8.1.2	Menu Einfügen	11
8.1.3	Menu Bearbeiten	12
8.1.4	Menu Ansicht	13
8.1.5	Menu About	13
8.1.6	Formatierung	14
8.1.7	Statusleiste	18
8.2	Zeichnen Beschreibung	20
9	Implementierung	21
9.1	Abweichungen vom Pflichtenheft	21
9.2	Beschreibung wichtiger Quellcode Ausschnitte	22
9.2.1	MainWindow:	22
9.2.2	UserControl_Texteditor	23
9.2.3	SizeChanged Ereignis	23
9.2.4	Windows-Resources:	23

9.2.5	Kopieren und Einfügen	24
9.2.6	Open File	25
9.2.7	Save File	25
9.2.8	Speech Funktion	26
9.2.9	UserControl_Zeichnen	27
9.2.10	Select Pen Click:	27
9.3	Installationsanleitung	27
9.4	Verwendete NuGet-Pakete	28
10	Testplan	28
11	Anhang / Ressourcen	29
11.1	Ressourcen	29
11.2	Demo Video	29

1 Einleitung

1.1 Beschreibung des Projektes

Für die Hausarbeit soll ein Texteditor in C# Sprache und WPF-GUI entwickelt werden. Die Software wird nach Kundenwünschen angepasst. Je nach Bedarf können Features (Funktionen) hinzugefügt werden. Auch kann die grafische Oberfläche, je nach Favorisierung der Befehle verändert werden. Einige Funktionen die der Texteditor bietet und die von Kunden unterschiedlich gewünscht ist Z. B die Datei in verschiedenen Format zu speichern oder Dark Mode zu benutzen.

2 Einsatz

2.1 Anwendung

Der Texteditor dient dazu, Text Dateien zu erstellen, bearbeiten und speichern.

2.2 Zielgruppen

Der Texteditor wird von Benutzer, die spezifische Wünsche und Features haben, benutzt. Sowie die alle anderen Benutzer.

3 Muss- und Kann-Kriterien mit Aufwandsabschätzung

3.1 Muss-Kriterien

Das System enthält folgende Eigenschaften:

- Neue Datei erstellen. 3 Stunden
- Texte in der Datei schreiben. 10 Stunden
- Nach Text suchen. 5 Stunden
- Text bearbeiten, die Bearbeitung enthält folgende Features. 30 Stunden
 - Text markieren

- Text kopieren
- Text ausschneiden
- Kopierter Text einfügen
- Schriftart ändern
- Schriftgrad ändern
- Schriftfarbe ändern
- Text vergrößern
- Text verkleinern
- Text ausrichten
- Datei speichern. 10 Stunden
- Datei öffnen. 4 Stunden
- Datei drucken. 4 Stunden
- Hilfe anfordern (Kontakt und Info). 2 Stunden

C# Code: Die Implementierung muss in C# erfolgen.

WPF: Die Implementierung muss in GUI-Oberfläche WPF erfolgen.

UML: Für Modellierung und Dokumentation muss Use-Case-Diagramme genutzt werden.

3.2 Kann-Kriterien

- Implementierung und Verbindung mit Datenbank: Das System kann die Datei in Datenbank Speichern. 6 Stunden
- Bildschirm auf Dark Mode drehen. 6 Stunden

4 Funktionen

Geschäftsprozess:	Text schreiben
Akteur:	Alle Benutzer
Beschreibung:	Nach dem Starten des Editors kann Text geschrieben werden.
Geschäftsprozess:	Datei öffnen
Akteur:	Alle Benutzer
Beschreibung:	Nach dem Starten des Editors kann eine vorhandene Datei geöffnet werden
Geschäftsprozess:	Datei speichern
Akteur:	Alle Benutzer
Beschreibung:	Nach dem Ändern der Datei kann diese gespeichert werden
Geschäftsprozess:	Datei drucken
Akteur:	Alle Benutzer
Beschreibung:	Nach dem Schreiben der Datei kann diese gedruckt werden
Geschäftsprozess:	Datei bearbeiten, die Bearbeitung enthält folgenden Funktionen: (Ausschneiden, Kopieren, Löschen, suchen, alles markieren, Einfügen-Datum kurz, lang) (Änderung von: Schriftart, Schriftschnitt, Größe, Farbe) (Ansicht: vergrößern, verkleinern, Standardgröße, Ausrichtung)
Akteur:	Alle Benutzer
Beschreibung:	Nach dem Öffnen des Editors kann einen vorhandenen Text bearbeitet werden

Geschäftsprozess: Datei in Datenbank speichern

Akteur: Registrierte Benutzer

Beschreibung: Nach dem Ändern der Datei kann diese in Datenbank gespeichert werden

Geschäftsprozess: Hilfe anfordern

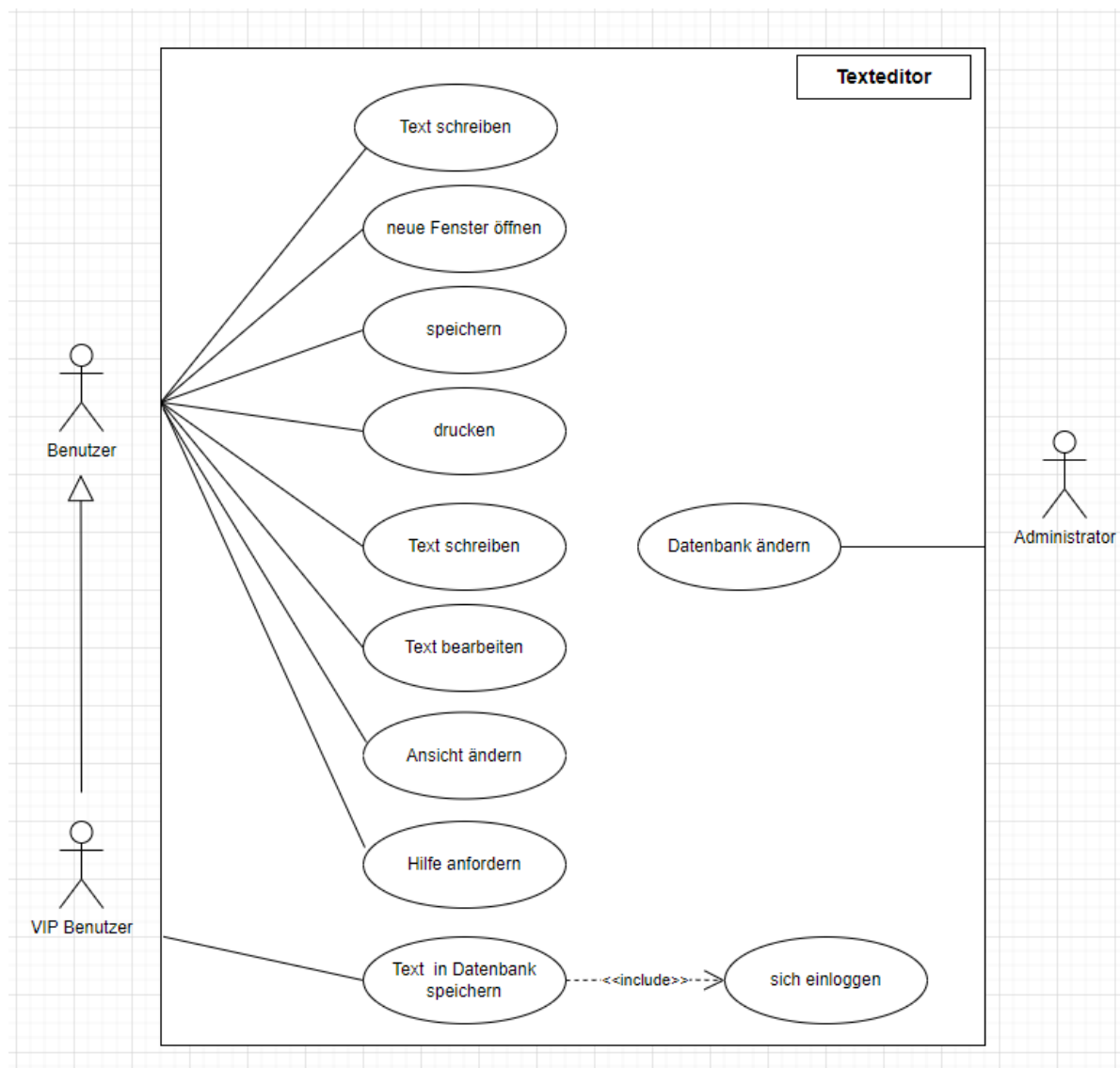
Akteur: Alle Benutzer

Beschreibung: Benutzer dieser Applikation können nach Hilfe fragen oder wenn sie Vorschläge haben um bestimmte Funktionen zu verbessern

5 Produktübersicht

5.1 Use-Case Diagramm

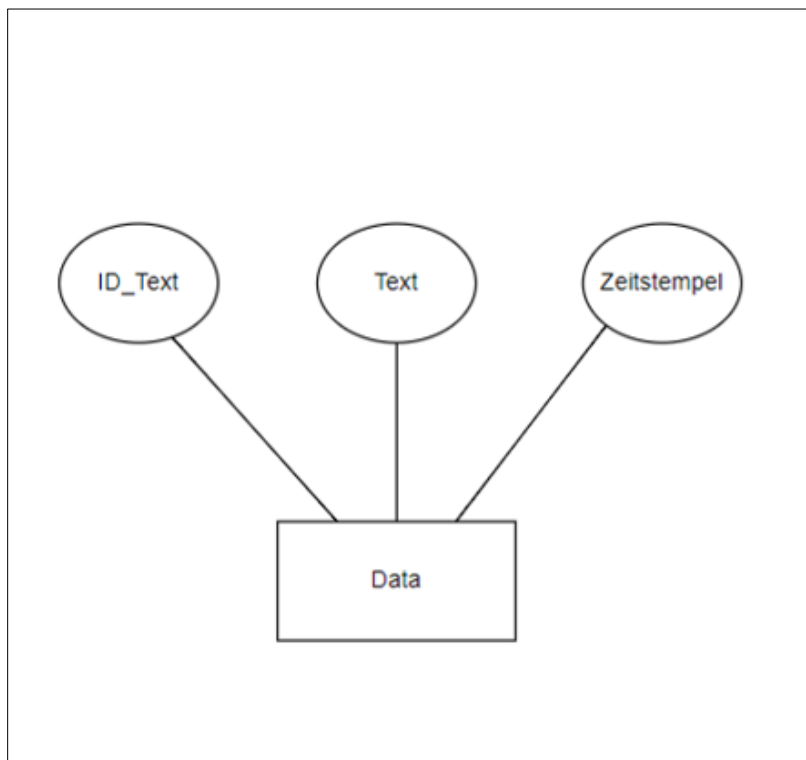
Die vereinbarten Funktionalitäten werden in (Abbildung 1) Use-Case Diagramm grafisch dargestellt.



(Abbildung 1: Use-Case Diagramm)

Erstellt mit: draw.io Tool

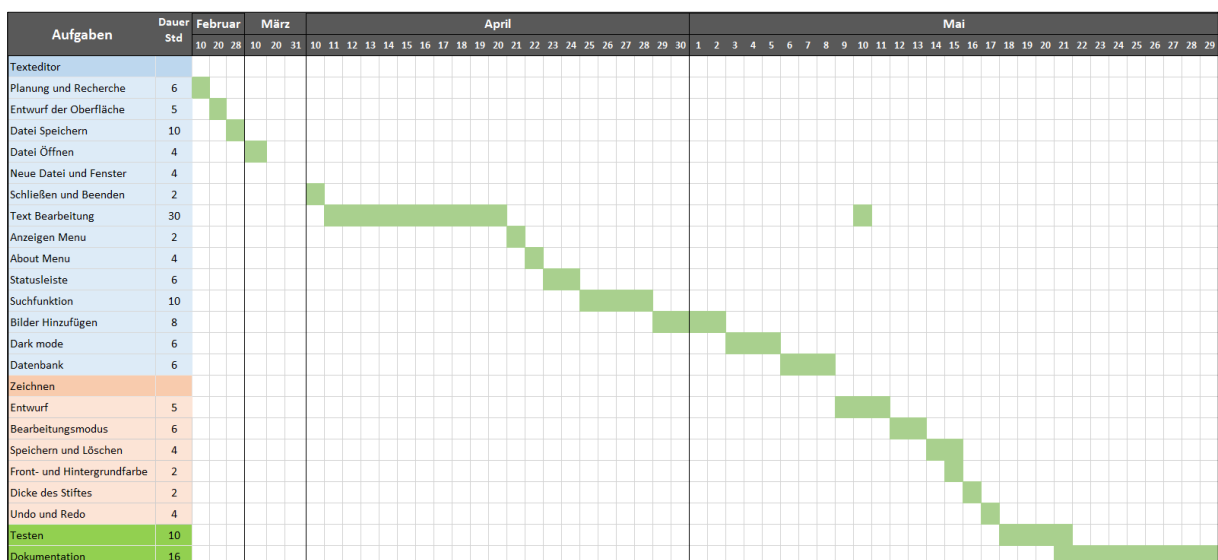
5.2 Datenbank Entwurf



5.3 Git-Repository

Link zu Git-Repository: <https://github.com/Khaledkalo/Texteditor>

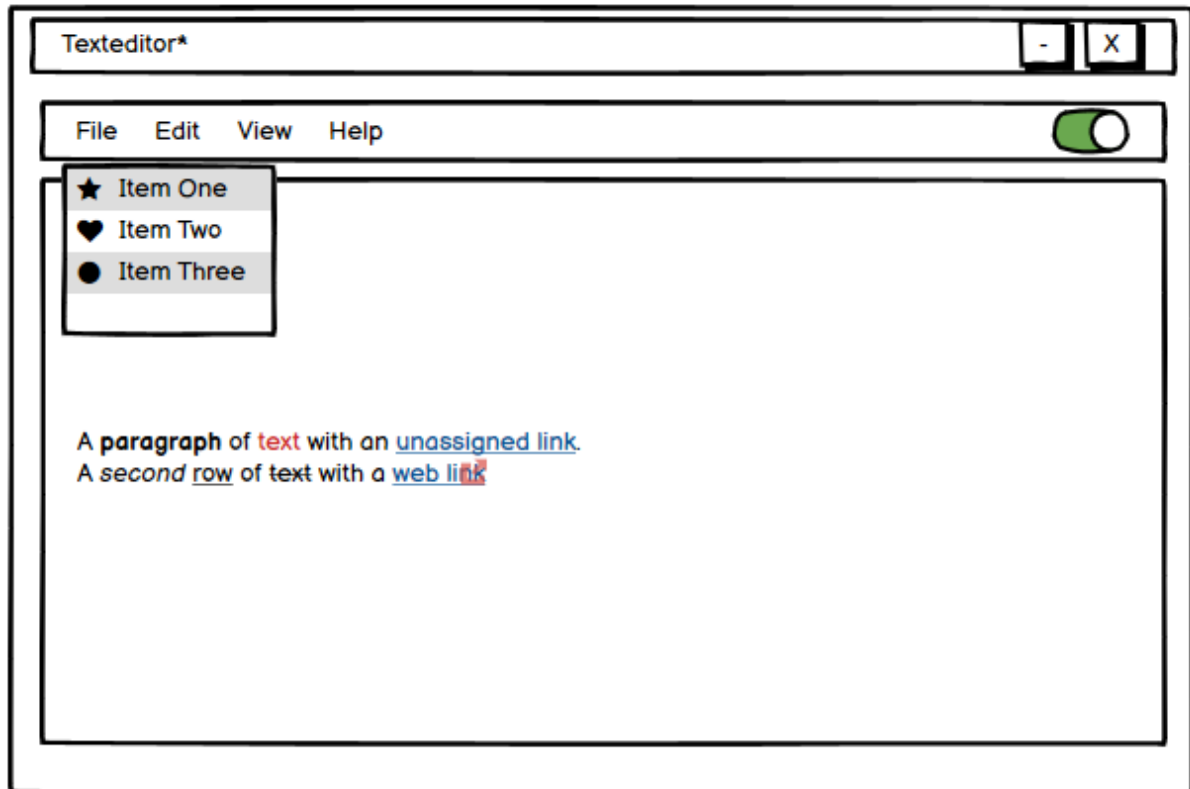
5.4 Gantt-Diagramm



Es wurde circa zwei Stunden pro Tag außer Wachende auf dem Projekt gearbeitet.
Im Februar und März sind die Tage als Zeitraum gedacht.
Am Beginn von März wurde nicht auf Projekt wegen Vorbereitung für Prüfung gearbeitet.
Erst ab 10. April wurde es wieder mit der Arbeit auf Projekt begonnen.

5.5 Skizze von GUI

Eine Darstellung, wie die Applikation aussehen wird.



Erstellt mit: Balsamiq_Wireframes Tool

6 Entwicklungsumgebung

6.1 Software

Die Applikation ist eine Desktopanwendung. Für die Entwicklung dieser Applikation werden Visual Studio IDE, C# Sprache und WPF-GUI verwendet.

6.2 Hardware

PC, Laptop

6.3 Orgware

Als Betriebssystem wird Windows 10 und 11 eingesetzt.

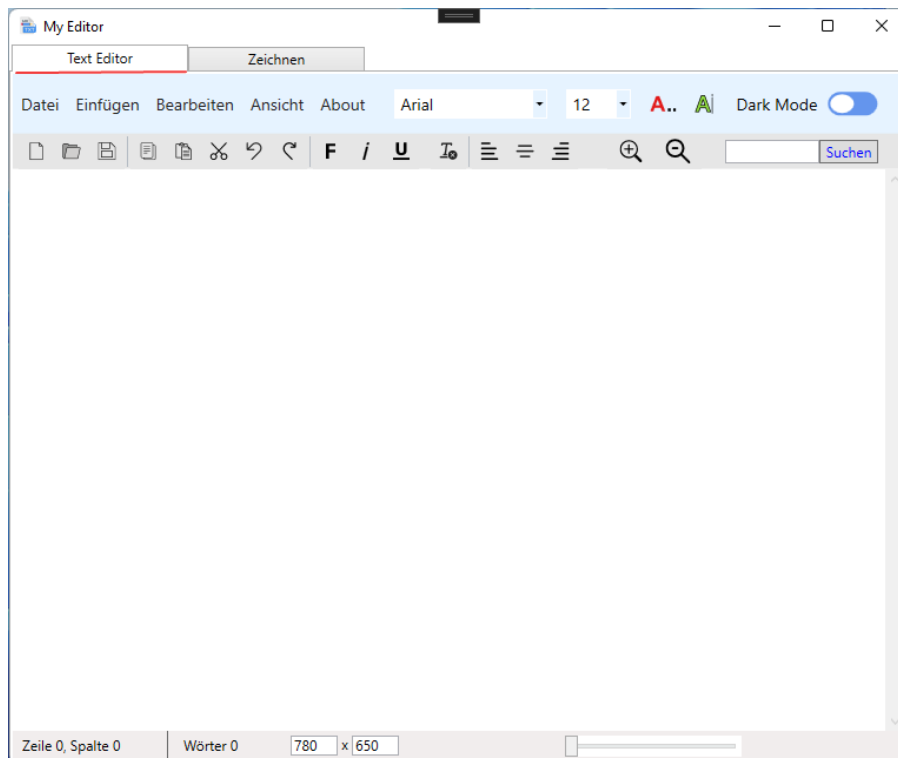
7 Dokumentation

Die Anwendungsdokumentation wird als folgende zur Verfügung gestellt:

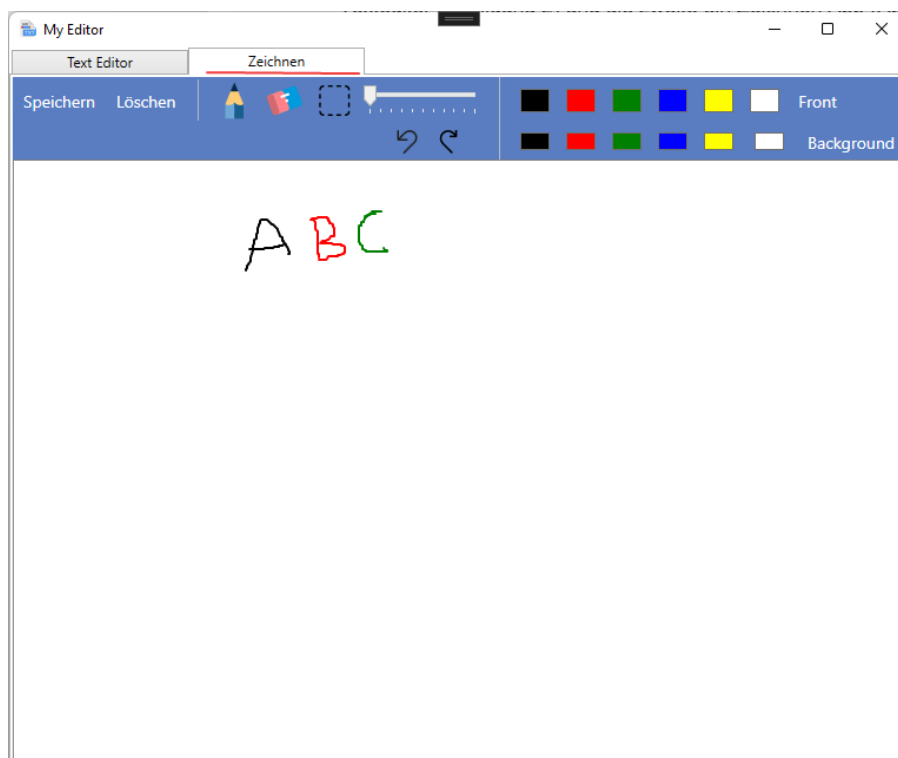
1. Quellcode des Programms in Git-Repository.
2. Dokumentation des ganzen Projekts als PDF-Datei.
3. Dokumentation und Quellcode in einem USB-Speicher-Stick.

8 Benutzerhandbuch

Dieses App besteht aus zwei Ansichten (Tab Control), die erste Ansicht enthält ein Texteditor (Abbildung 2) und die zweite ein Zeichnen App (Abbildung 3).



(Abbildung 2, Texteditor)

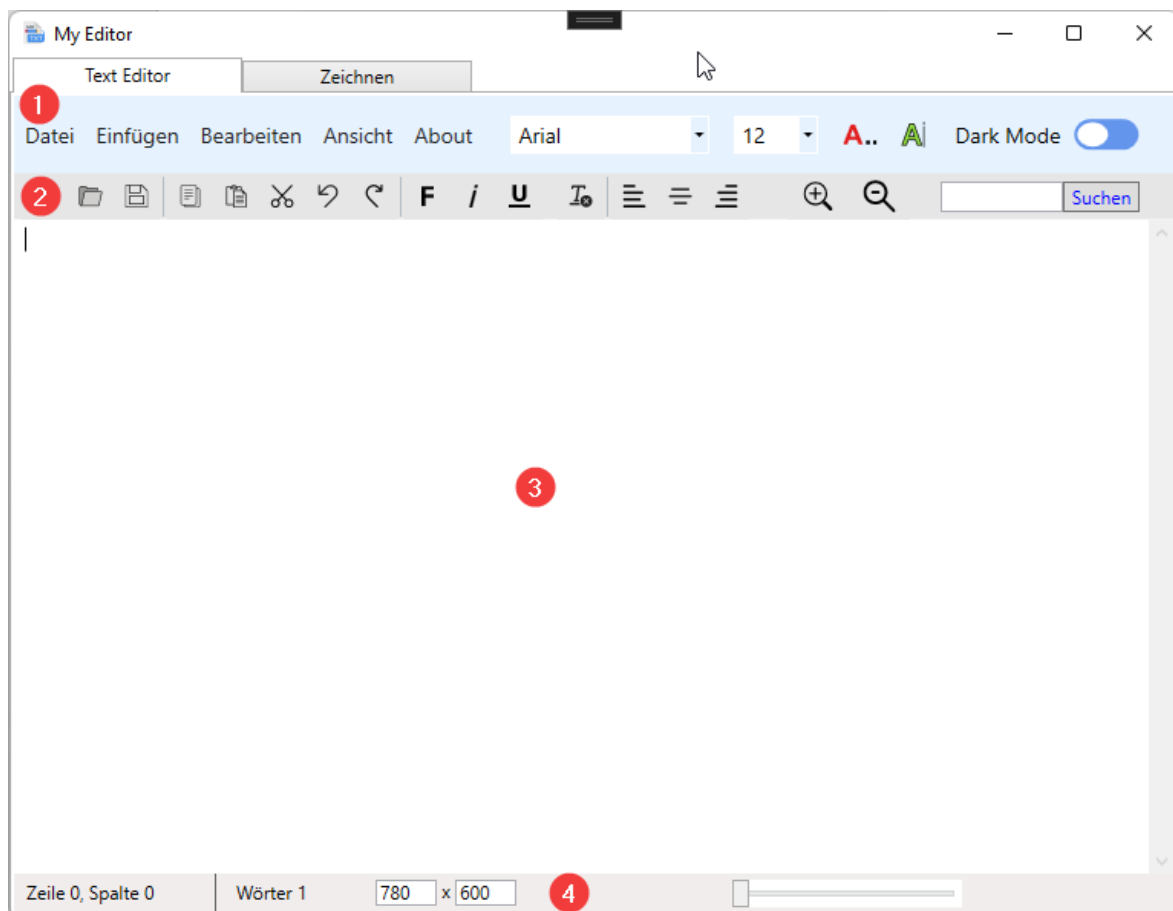


(Abbildung 3, Zeichnen)

Die Funktionen des Texteditors und des Zeichnens werden im Folgenden beschrieben.

8.1 Texteditor Beschreibung

Der Texteditor befindet sich in UserControl_Texteditor.xaml Datei und ist so entworfen, dass er durch vier Zeilen (Rows) geteilt (Abbildung 4): In der ersten Zeile (Schritt 1) steht folgende Menus und Buttons: Datei, Einfügen, Bearbeiten, Ansicht, About, Font Family, Font Size, Vordergrund- und Hintergrundfarbe, und Dark Mode. In der zweiten Zeile (Schritt 2) steht Symbolleiste für den Schnellzugriff und andere Formatierung Funktionen wie Textstyle ändern, Änderungen rückgängig machen oder wiederherstellen, Zoom und Suchen Funktion. In der dritten Zeile (Schritt 3) steht Textbereich, hier wird den Text geschrieben oder Bilder hinzugefügt. in der Letzen Zeile (Schritt 4) steht Statusbar, hier wird Zeilen-, Spalten- und Wörterzahl, Fenstergröße und Zoom angezeigt.



(Abbildung 3)

8.1.1 Menu Datei

In Menu Datei steht folgende Funktionen (Öffnen, Speichern, Drucken und Schließen) mit Beschreibung der Funktionsweise. (Abbildung 4)

Neu: um neue Datei zu erstellen, Klicken Sie auf Menu Datei dann auf Neu. Falls der Dateinhalt nicht leer ist, wird es den Benutzer gefragt ob die Änderungen zu speichern. (Abbildung 4, Schritt 1)

Neues Fenster: Neben dem geöffneten Fenster ein neues Fenster öffnen. Klicken Sie auf Menu Datei dann auf Neues Fenster.

Öffnen: Um eine bestehende Datei auf dem PC zu öffnen. Klicken Sie auf Menu Datei dann auf Öffnen (Abbildung 4, Schritt 3), es öffnet ein Dialog (Abbildung 5), Navigieren Sie zu Datei Ort wo der Datei auf Ihrem PC steht und Klicken Sie auf der Datei dann auf Button Öffnen oder Ok drücken.

Speichern: Um die bearbeitete Datei zu speichern, und zwar als TXT, RTF oder XAML-Datei. Klicken Sie auf Menu Datei dann auf Speichern (Abbildung 4, Schritt 4), es öffnet ein Dialog (Abbildung 6), wählen Sie ein Speicherort auf Ihrem PC, nennen Sie die Datei um, wählen Sie mit welchem Format Sie Datei zu Speichern TXT, RTF oder XML dann klicken Sie auf Button Speichern oder OK.

Speichern unter: Um die bearbeitete Datei zu speichern, es ist ähnlich wie Speichern. Speichern unter wird verwendet, wenn man die Datei erneut mit anderem Namen zu speichern möchte. (Abbildung 4, Schritt 5).

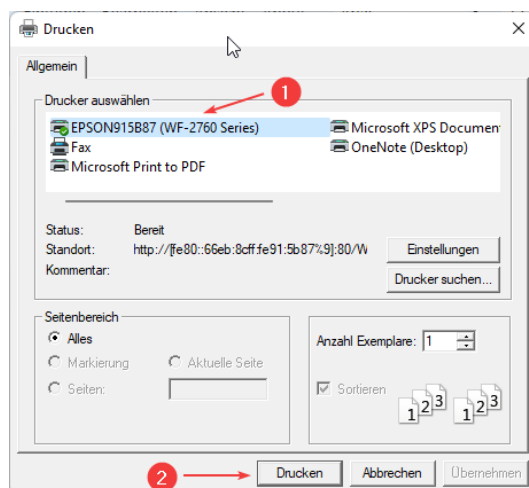
Speichern als PDF: Die Datei als PDF speichern. Klicken Sie auf Menu Datei dann auf Speichern als PDF, es öffnet ein Dialog, wählen Sie ein Speicherort auf Ihrem PC, nennen Sie die Datei um, dann klicken Sie auf Button Speichern oder OK. (Abbildung 4, Schritt 6).

Speichern in DB: Die Dateiinhalte in Datenbank speichern. Klicken Sie auf Menu Datei dann auf Speichern in DB. (Abbildung 4, Schritt 7).

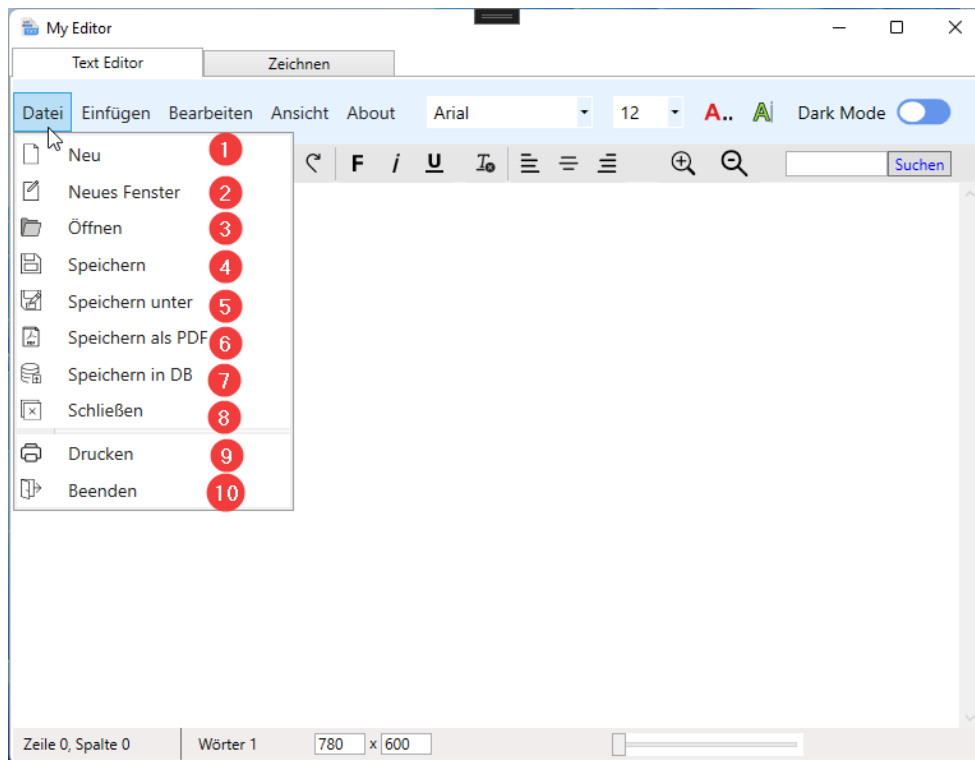
Schließen: Datei schließen. Klicken Sie auf Menu Datei dann auf Schließen, es wird gefragt ob Datei bevor zu schließen, zu speichern. (Abbildung 4, Schritt 8).

Drucken: Datei zu drücken. Klicken Sie auf Menu Datei dann auf Drucken, es öffnet Drucker Dialog, wählen Sie Ihrem Druckernamen dann klicken Sie auf Button Drucken. (Abbildung 4, Schritt 9).

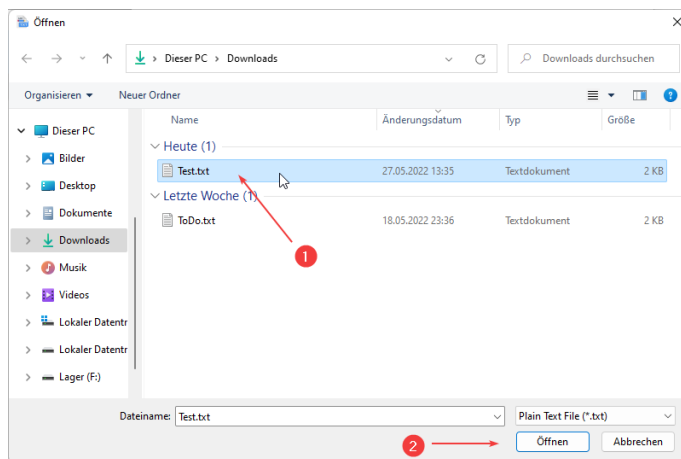
Beenden: Datei sofort beenden ohne was zu speichern. Klicken Sie auf Menu Datei dann auf Beenden.



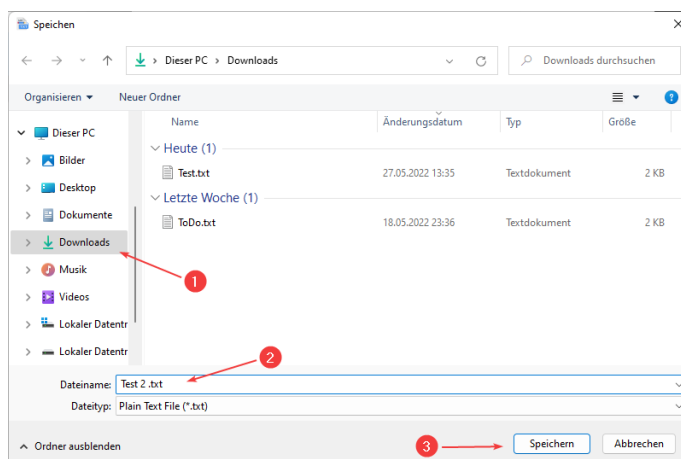
(Abbildung, 7)



(Abbildung 4)



(Abbildung 5)

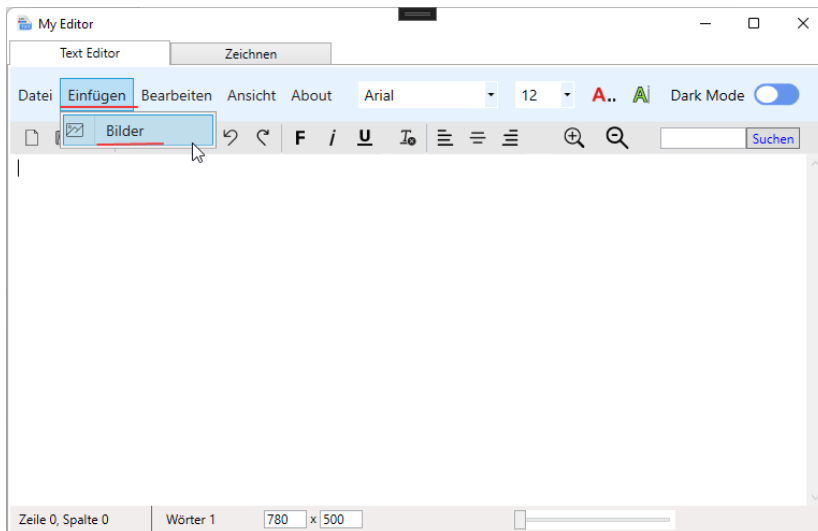


(Abbildung 6)

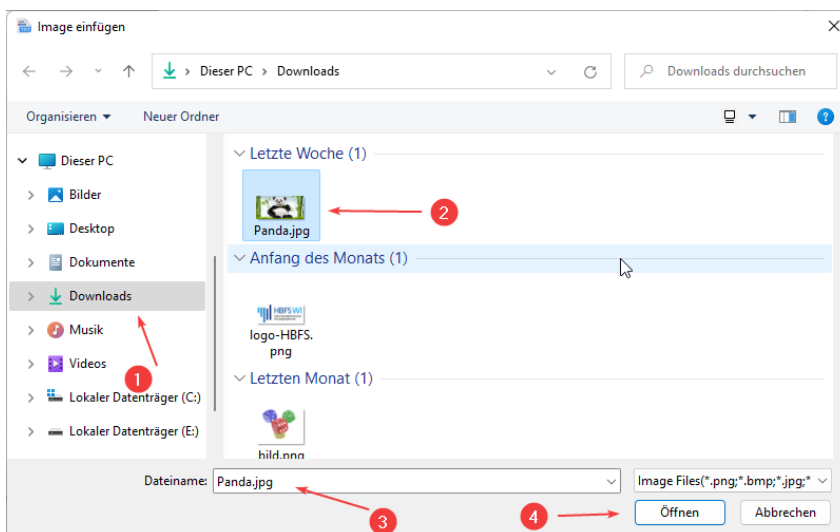
8.1.2 Menu Einfügen

In Menu Einfügen steht die Funktion Bilder einfügen.

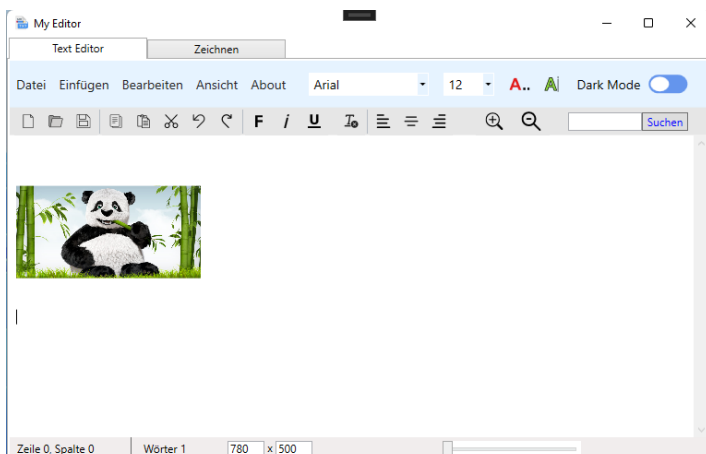
Bilder: fügt Bilder zu Datei ein. Klicken Sie auf Menu Einfügen dann auf Bilder, es öffnet ein Dialog, wählen Sie das Bild auf Ihrem PC dann klicken Sie auf Button Öffnen oder OK.
(Abbildung 8, 9)



(Abbildung 8)



(Abbildung 9)



8.1.3 Menu Bearbeiten

In Menu Bearbeiten steht folgende Funktionen mit Beschreibung der Funktionsweise.
(Abbildung 10)

Kopieren: Kopiert den ausgewählten Text in den Zwischenablage. Wählen Sie den Text aus, Klicken Sie auf den Button Bearbeiten dann auf Kopieren. Alternative Können Sie mit Shortcut Strg + C den Text Kopieren.

Einfügen: fügt den ausgeschnittenen oder kopierten Text aus der Zwischenablage ein. Klicken Sie auf den Button Bearbeiten dann auf Einfügen. Alternative Können Sie mit Shortcut Strg + V den Text einfügen.

Ausschneiden: Kopiert den ausgewählten Text in den Zwischenablage und zusätzlich löscht ihm. Wählen Sie den Text aus, Klicken Sie auf den Button Bearbeiten dann auf Ausschneiden. Alternative Können Sie mit Shortcut Strg + X den Text ausschneiden.

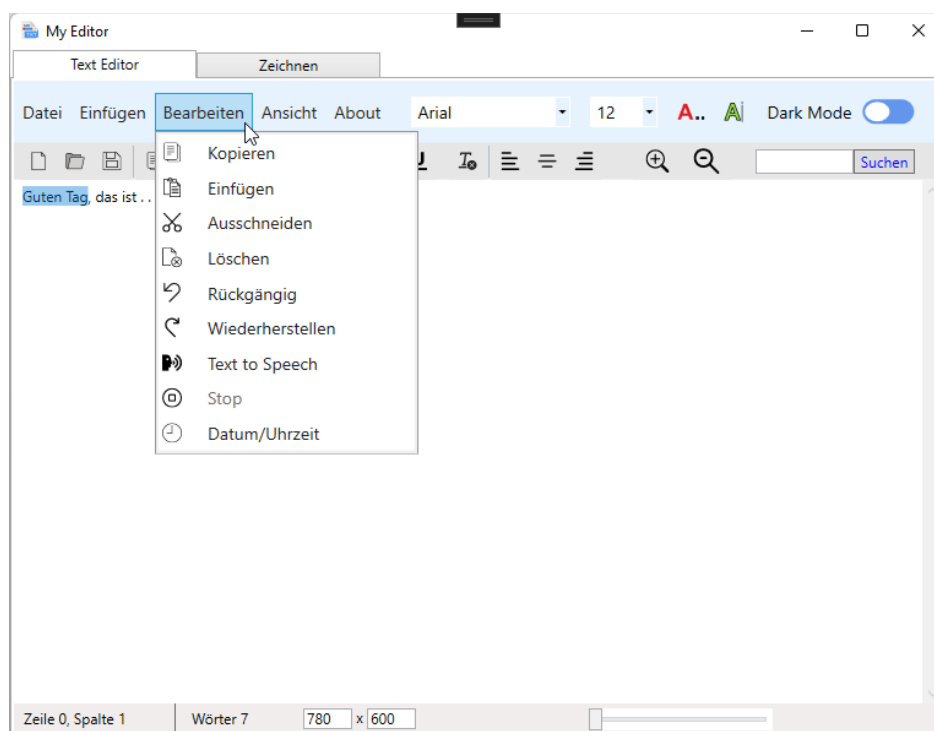
Löschen: Löscht den Text. Wählen Sie den Text aus, Klicken Sie auf den Button Bearbeiten dann auf Löschen. Alternative Können Sie mit Backspace-Taste den Text löschen.

Rückgängig: (Undo) Macht die Änderungen rückgängig. Klicken Sie auf den Button Bearbeiten dann auf Rückgängig. Alternative Können Sie es mit Shortcut Strg + Z machen.

Wiederherstellen: (Redo) Stellt alle zuvor per Undo rückgängig gemachten Aktionen wieder her. Klicken Sie auf den Button Bearbeiten dann auf Wiederherstellen. Alternative Können Sie es mit Shortcut Strg + Y machen.

Text to Speech: Text-to-Speech (TTS) ist eine Art Hilfstechnologie, die digitalen Text laut vorliest. TTS kann Wörter aufnehmen und in Audio umwandeln. Wählen Sie den Text aus, Klicken Sie auf den Button Bearbeiten dann auf Text to Speech.

Datum / Uhrzeit: Fügt aktuelles Datum und Uhrzeit ein. Klicken Sie auf den Button Bearbeiten dann auf Datum / Uhrzeit.



(Abbildung 10)

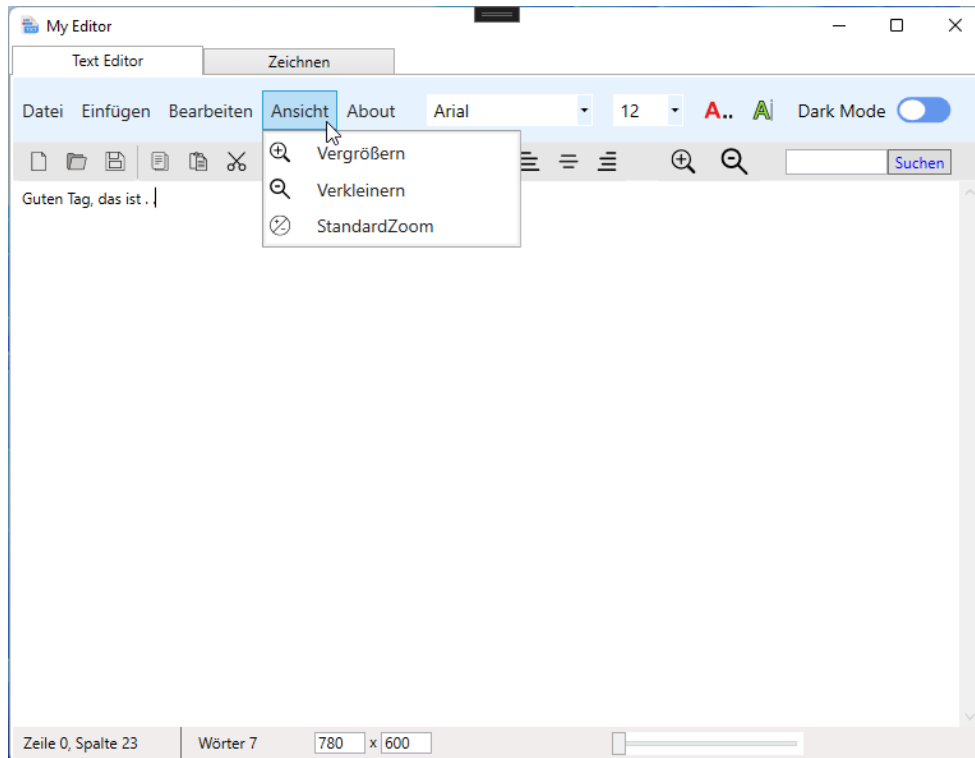
8.1.4 Menu Ansicht

In Menu Ansicht steht folgende Funktionen mit Beschreibung der Funktionsweise.
(Abbildung 11):

Vergrößern: Erhöht die Schriftgröße um 1 Punkt. Klicken Sie auf den Button Ansicht dann auf Vergrößern.

Verkleinern: Verringert die Schriftgröße um 1 Punkt. Klicken Sie auf den Button Ansicht dann auf Verkleinern.

Standard Zoom: Setzt die Schriftgröße als Standardgröße. Klicken Sie auf den Button Ansicht dann auf Standard Zoom.



(Abbildung 11)

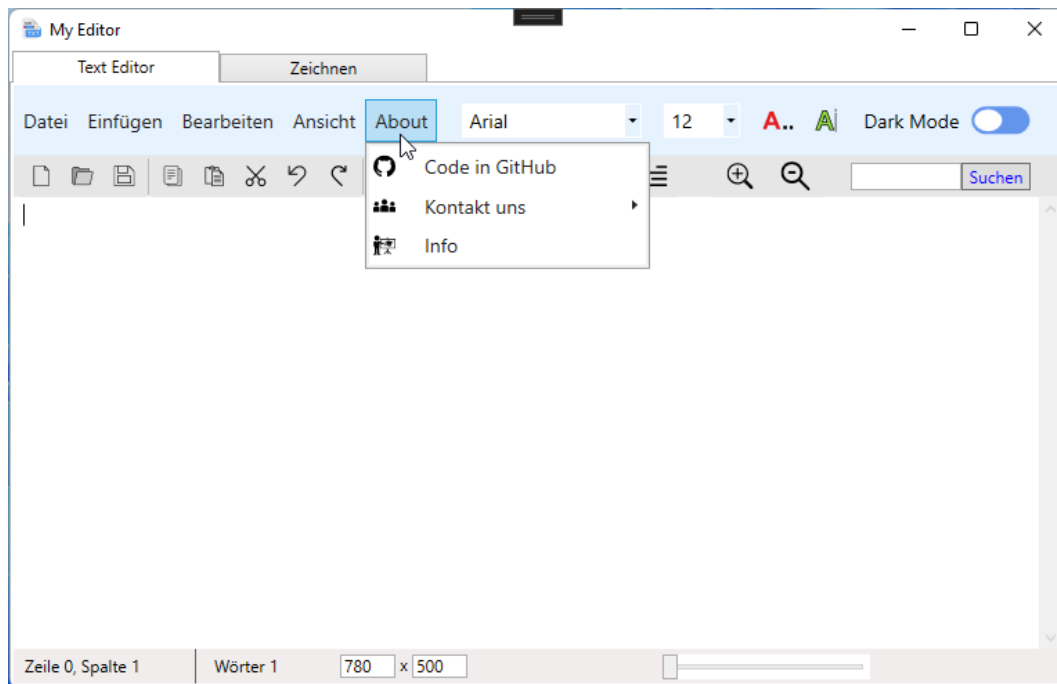
8.1.5 Menu About

In Menu About steht folgende Funktionen mit Beschreibung der Funktionsweise.
(Abbildung 8):

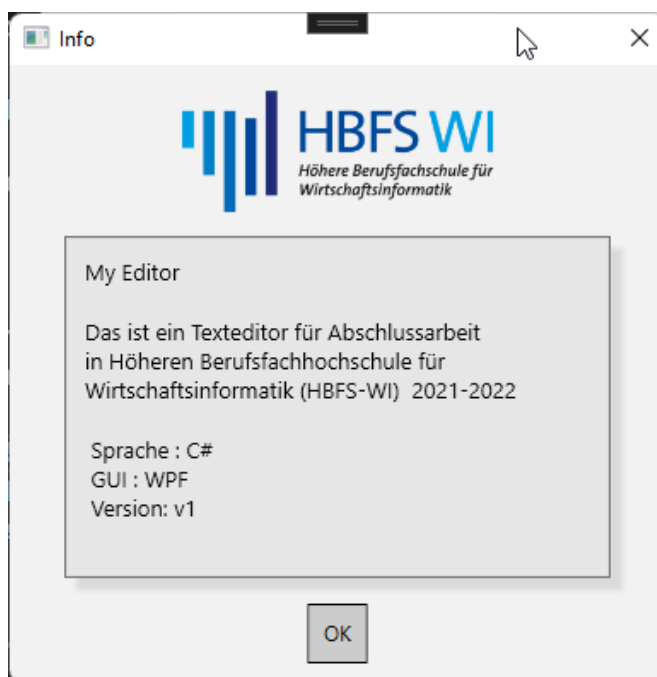
Code in GitHub: Navigiert zu GitHub Webseite und zeigt den Quellcode des Projekts. Klicken Sie auf den Button About dann auf Code in GitHub.

Kontakt uns: Zeigt die E-Mail-Adresse, um mit uns zu Kontakt zu kommen. Klicken Sie auf den Button About dann auf Kontakt, dann Sehen Sie die E-Mail-Adresse.

Info: Hier wird eine Kurze Information durch eine Nachrichtenbox Über das Projekt angezeigt. Klicken Sie auf den Button About dann auf Info. (Abbildung 12)



(Abbildung 12)

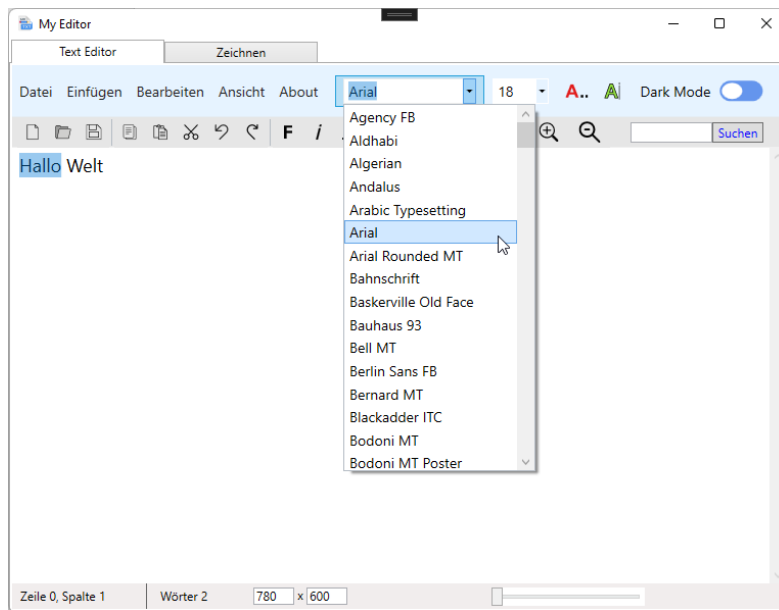


(Abbildung 13)

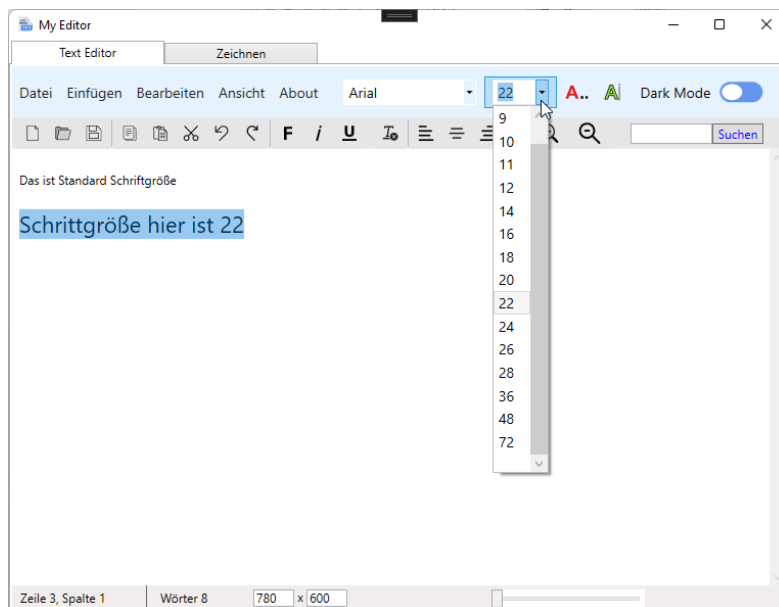
8.1.6 Formatierung

Schriftart Dialog: Ändert die Schriftart des Textes, die Standard Schriftart ist auf Arial festgelegt. Wählen Sie den Text aus, klicken Sie auf den Dialog Schriftart dann auf die anpassende Schriftart. (Abbildung 14)

Schriftgrad Dialog: Ändert die Schriftgröße des Textes, die Standard Schriftgröße ist auf 12 festgelegt. Wählen Sie den Text aus, klicken Sie auf den Dialog Schriftgröße dann auf die anpassende Schriftgröße. (Abbildung 15)



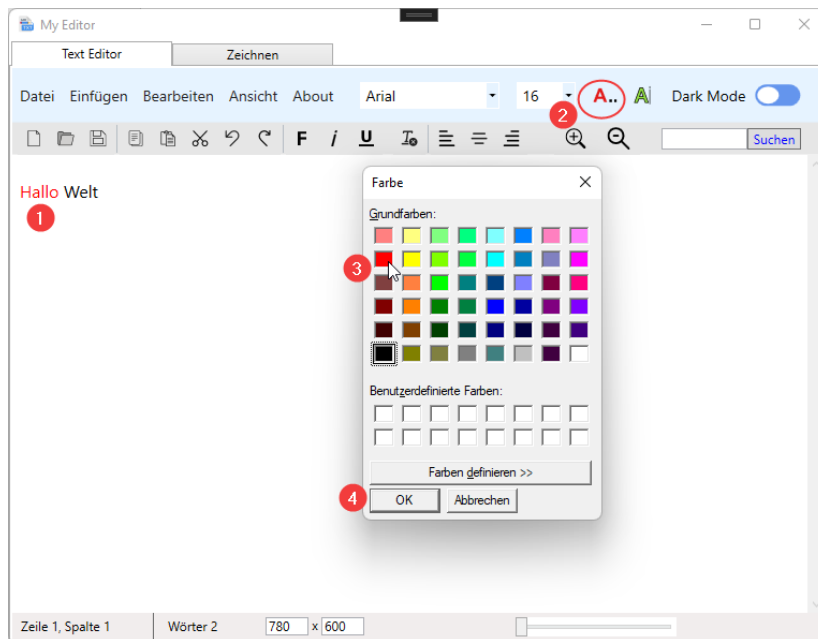
(Abbildung 14)



(Abbildung 15)

Schriftfarbe Dialog: Ändert die Schriftfarbe des Textes, die Standard Schriftart ist auf Schwarz festgelegt. Wählen Sie den Text aus, klicken Sie auf den Dialog Schriftfarbe dann auf die anpassende Farbe. (Abbildung 16, 17)

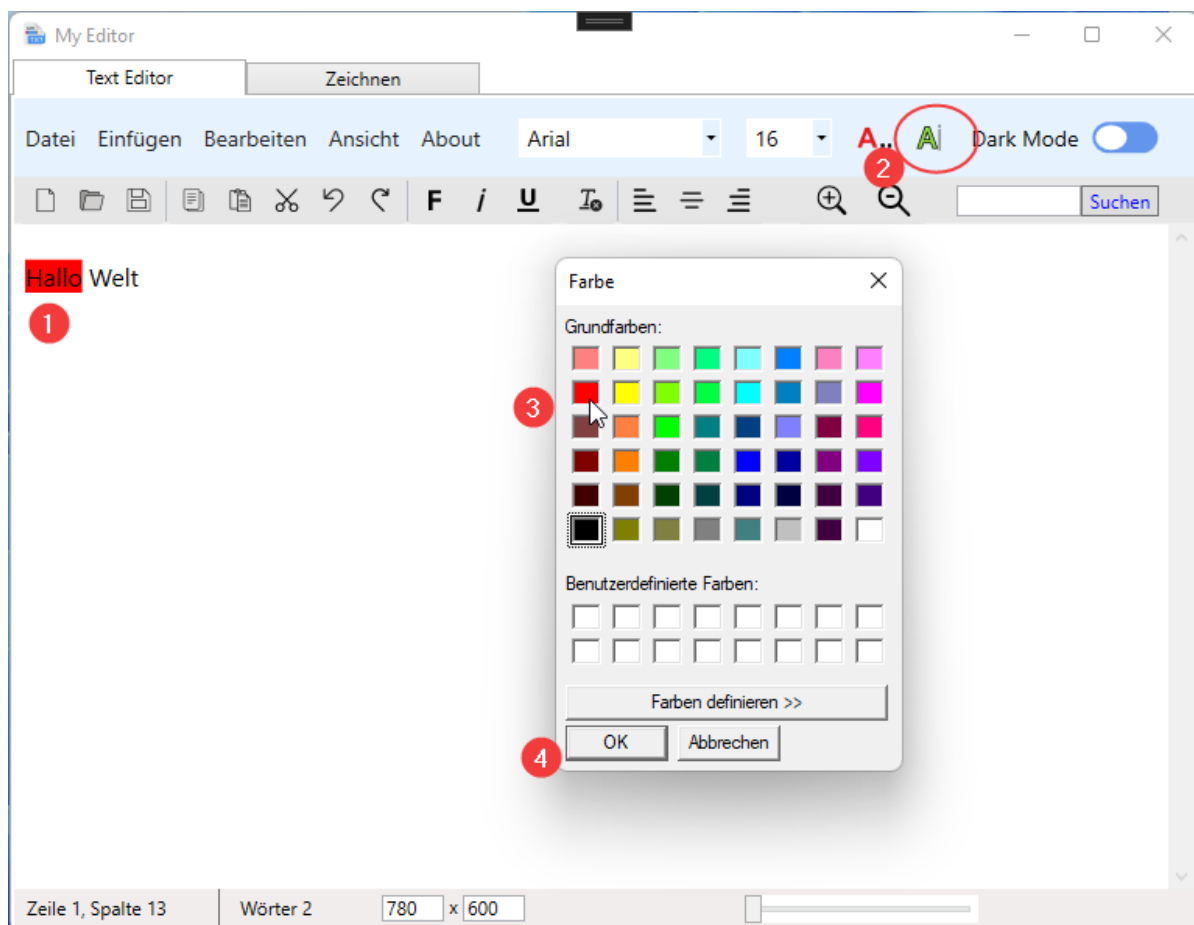
Hintergrundfarbe Dialog: Ändert die Hintergrundfarbe des Textes, die Standard Farbe ist auf Weis festgelegt. Wählen Sie den Text aus, klicken Sie auf den Dialog Hintergrundfarbe dann auf die anpassende Farbe. (Abbildung 18)



(Abbildung 16)

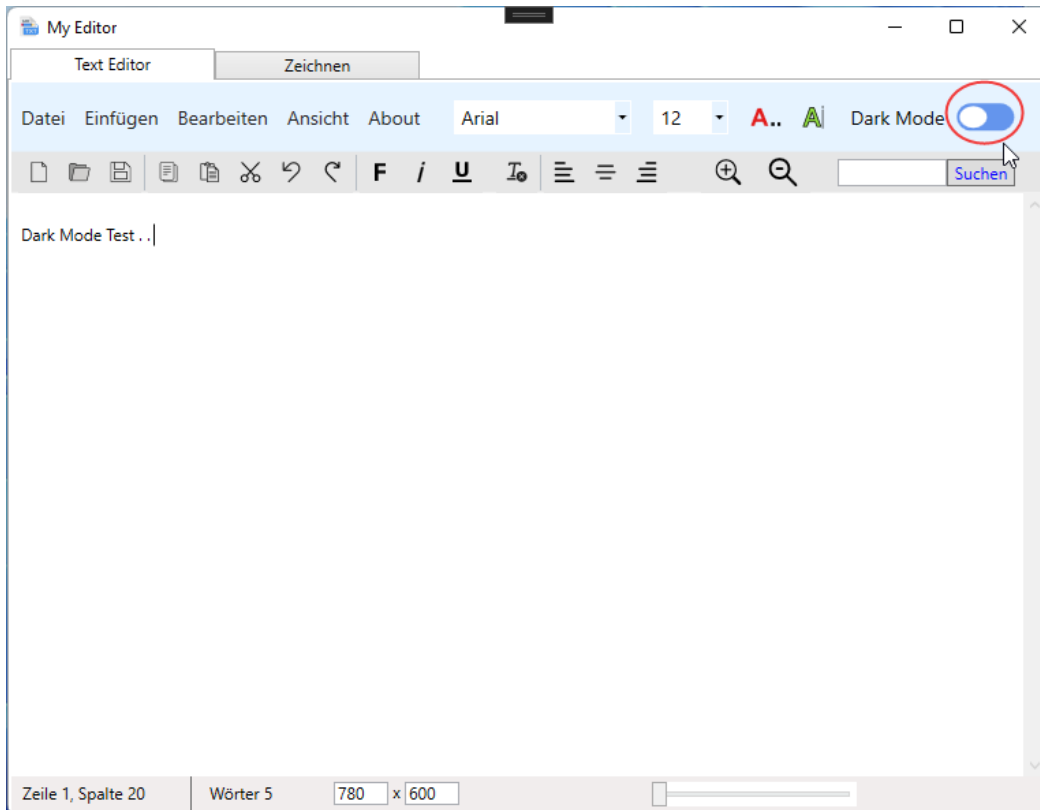


(Abbildung 17)

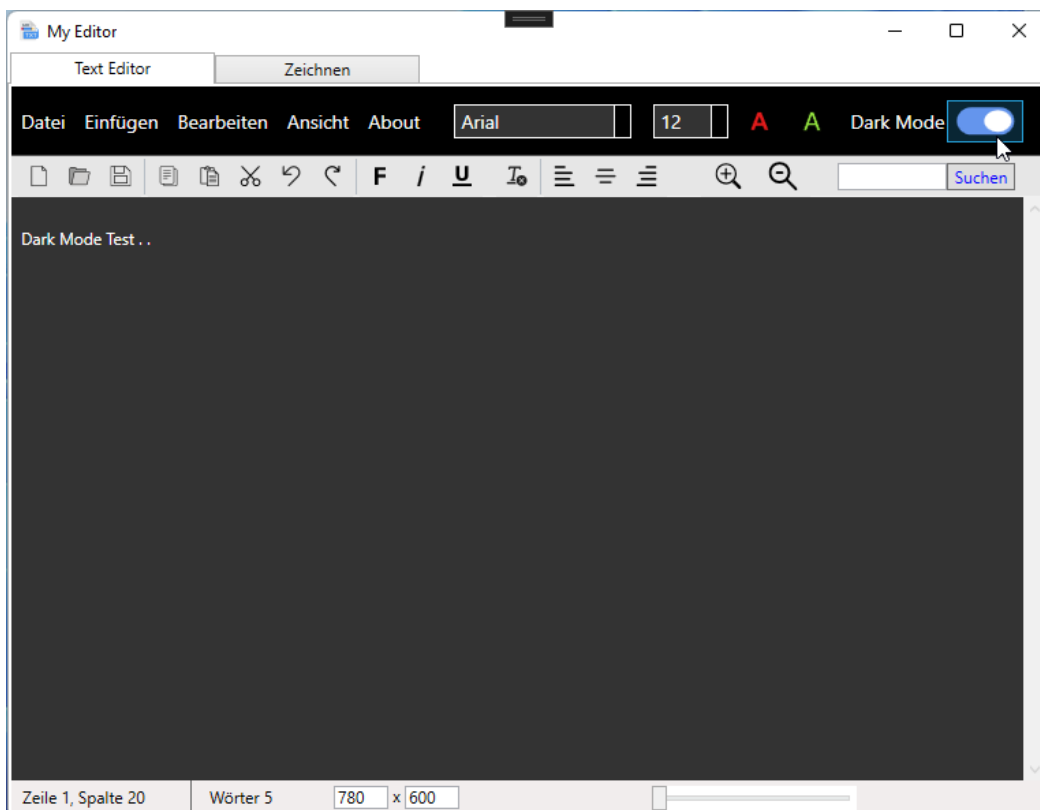


(Abbildung 18)

Dark Mode: Ändern die Hintergrundfarbe und die Farbe des Textes der gesamten App. Klicken Sie auf die rechte Seite des Umschaltknopfs (toggle-button). Klicken Sie auf die Linke Seite des Umschaltknopfs, um zu Light Mode zu wechseln. (Abbildung 19, 20)



(Abbildung 19)



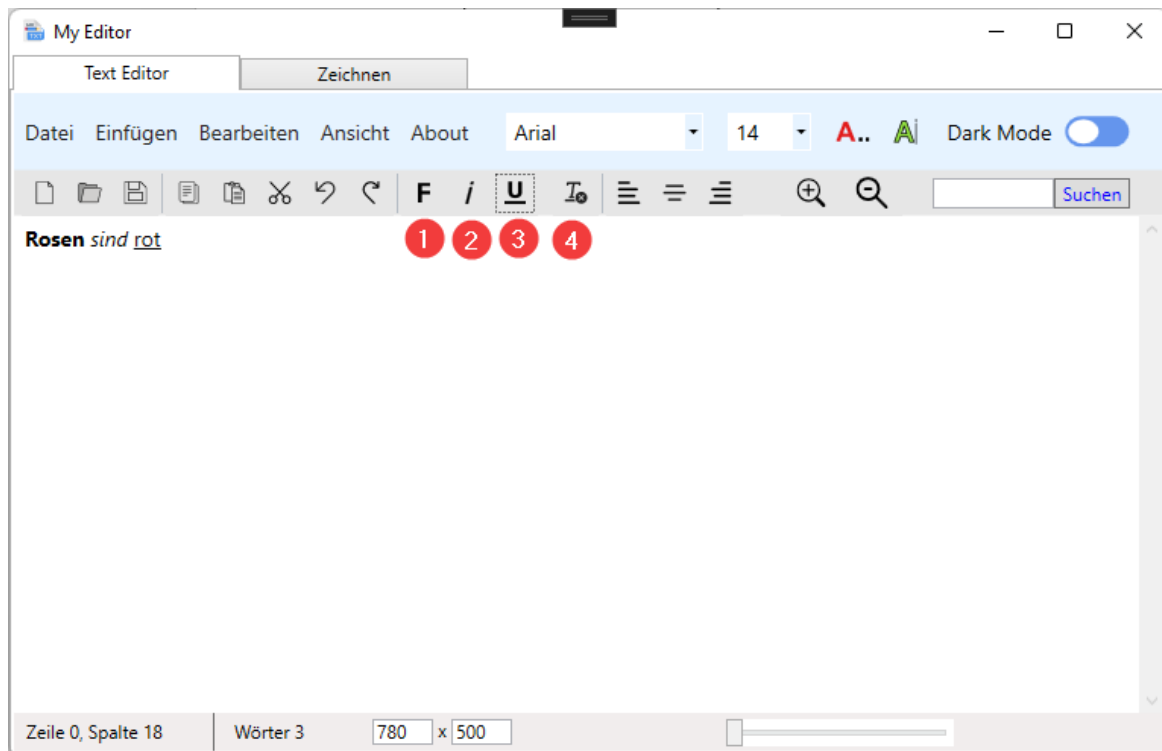
(Abbildung 20, Dark Mode)

Fett: Formatiert den markierten Text fett. Markieren Sie den Text und Klicken Sie auf den Button **F**. (Abbildung 21, Schritt 1)

Kursiv: Formatiert den markierten Text kursiv. Markieren Sie den Text und Klicken Sie auf den Button **i**. (Abbildung 21 Schritt 2)

Text Dekorationen: Unterstreichen den markierten Text. Markieren Sie den Text und Klicken Sie auf den Button **U**. (Abbildung 21, Schritt 3)

Formatierung zurücksetzen: löscht die gesetzte Formatierung. Markieren Sie den Text mit der Formatierung, die Sie löschen möchten. Dann Klicken Sie auf den Button Formatierung zurücksetzen. (Abbildung 21, Schritt 4)



(Abbildung 21)

Textausrichtung: Richtet die Inhalte des Texteditors an der linken, zentralen oder rechten Seite aus. Legen Sie die Mauszeiger vor, auf oder nach dem Text und Klicken Sie auf den Button Textausrichtung. (Abbildung 22)

Suchen: Sucht nach bestimmten Worten, wenn findet es, hebt das Ergebnis mit gelber Farbe hervor. Also ändert die Hintergrundfarbe von gefundenen Worten zu gelb. Klicken Sie auf den Kästchen neben Suchen dann geben Sie das Suchwort und klicken Sie auf Suchen. (Abbildung 23)

8.1.7 Statusleiste

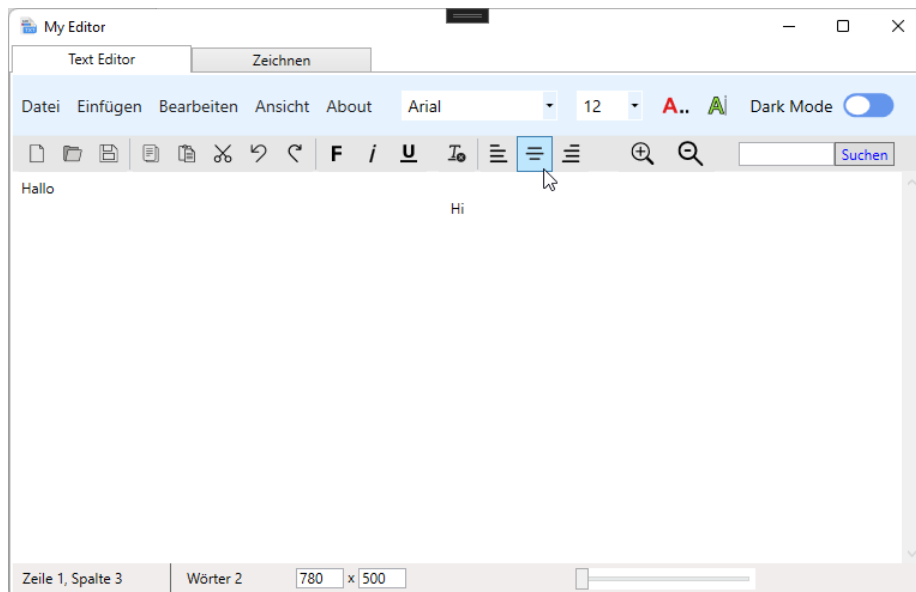
In Statusleiste (Statusbar) steht Informationen über die den Text und Fenster. (Abbildung 24)

Zeile und Spalte: Zeigt die Anzahl der Zeilen und Spalten in der Datei an.

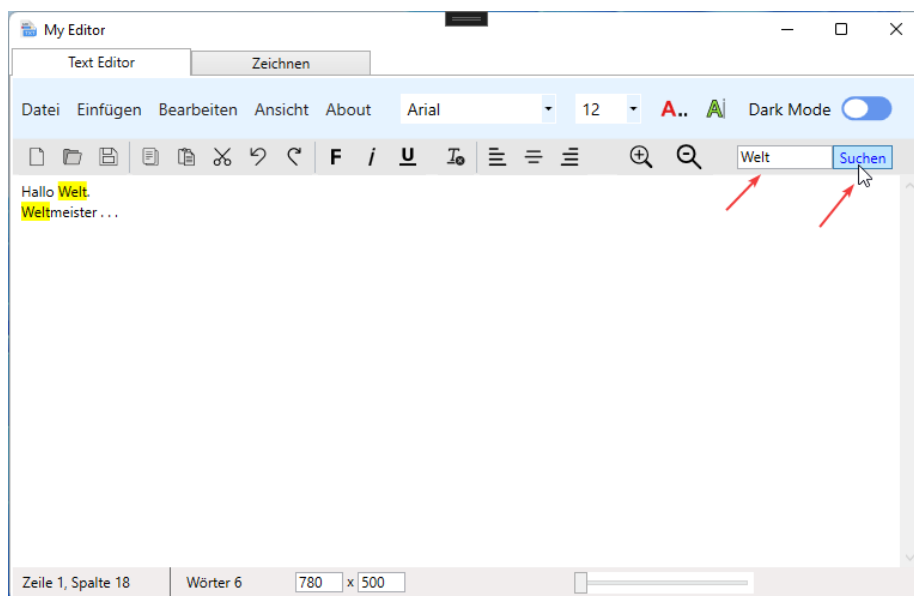
Wörter: Zeigt die ganzen Wörter in der Datei an.

Fensterdimension: Zeigt die Höhe und Breite des Texteditors an.

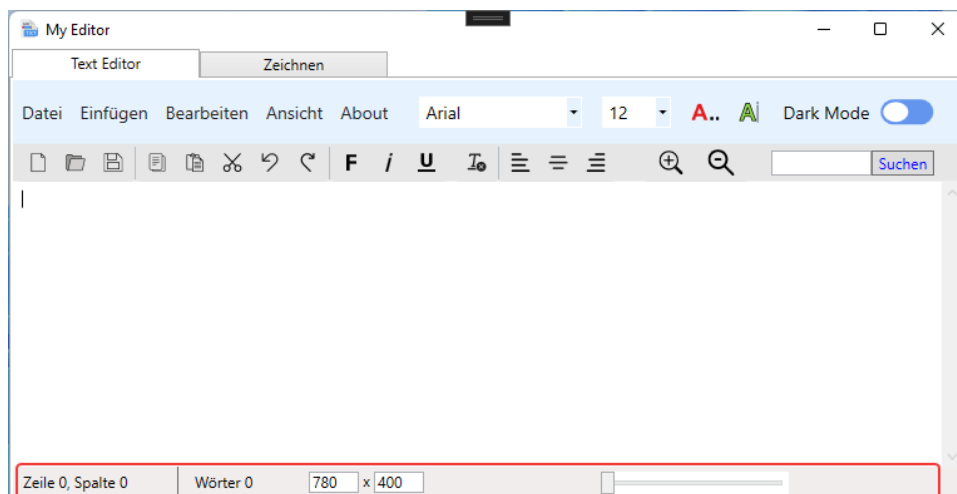
Zoom: Vergrößert und verkleinert das ganze Fensterinhalte.



(Abbildung 22)



(Abbildung 23)



(Abbildung 24)

8.2 Zeichnen Beschreibung

Zeichnen befindet sich in UserControl_Zeichnen.xaml. Im Folgenden wird Die Funktionen von Zeichnen beschrieben.

Speichern: Speichert die Inhalte als Bild mit einer der folgenden Dateierweiterung (PNG, JPG). (Abbildung 24, Schritt 1)

Löschen: Löscht alle Inhalte des Zeichnens. Abbildung 18, Schritt 2)

Auswählen: Wählt die gezeichnete Sache und kann es in andere Position verschieben (Abbildung 24, Schritt 3)

Pen, Stift: Zeichnen ist Standard auf Stift festgelegt. (Abbildung 24, Schritt 4)

Radiergummi: Löscht markierte Sache. (Abbildung 24, Schritt 5)

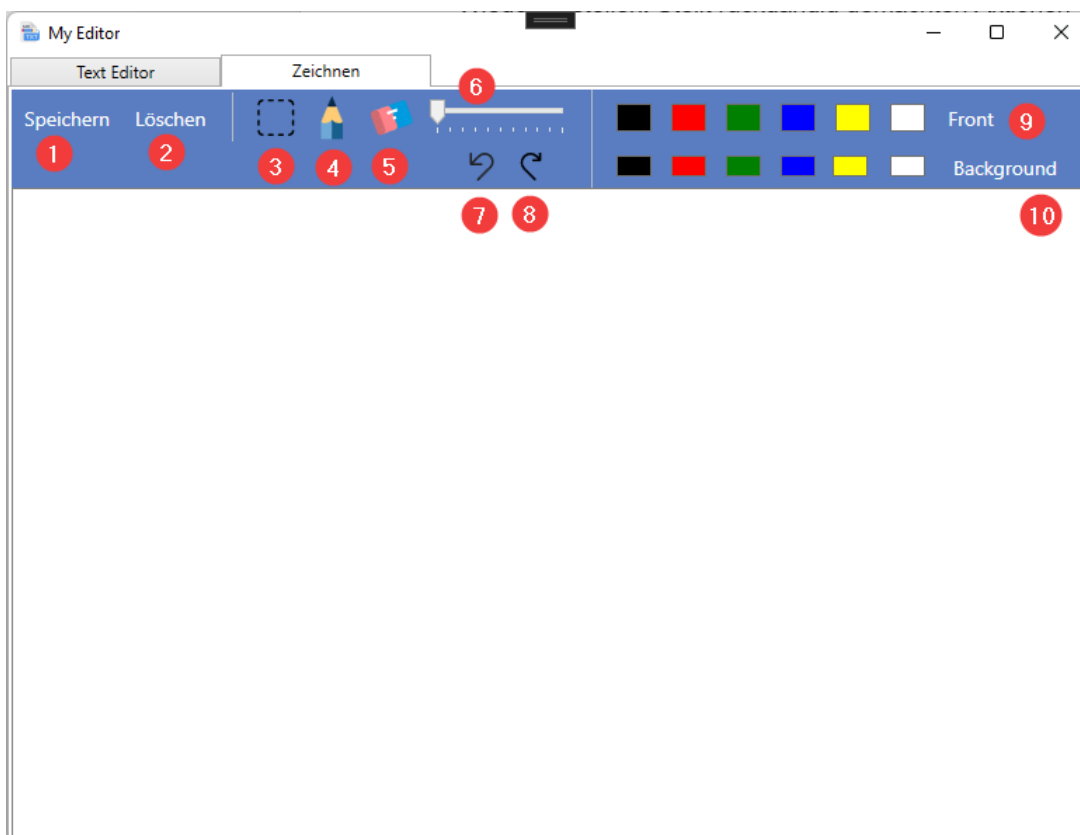
Stift Größe: Ändert die Dicke von Stift. (Abbildung 24, Schritt 6)

Rückgängig: Macht die Änderungen rückgängig. (Abbildung 24, Schritt 7)

Wiederherstellen: Stellt rückgängig gemachten Aktionen wieder her. (Abbildung 18, Schritt 8)

Vordergrundfarbe: Ändert die Vordergrundfarbe. (Abbildung 24, Schritt 9)

Hintergrundfarbe: Ändert die Hintergrundfarbe. (Abbildung 24, Schritt 10)



(Abbildung 24)

9 Implementierung

9.1 Abweichungen vom Pflichtenheft.

Während der Arbeit an dem Projekt, wurden dem Projekt einige neue Funktionen hinzugefügt und neue WPF-Steuerelemente verwendet:

- Bilder einfügen: Als ich mit der Anwendung eingefangen habe, habe ich für den Textbereich das TextBox-Steuerelement verwendet. Später bei Textformatierung habe ich festgestellt, dass es RichTextBox-Steuerelement besser geeignet als TextBox. RichTextBox-Steuerelement ist umfangreicher und bietet viele Funktion für Formatierung von Text, Bilder oder Tabellen. RichTextBox ist neue Steuerelement und es wird für Formatierung empfohlen RichTextBox statt TextBox zu verwenden. Für diese Gründen und um sicherzustellen, dass, den Texteditor in Zukunft die Wünsche von Benutzer erfüllen kann, wurde die Funktion Bilder Einfügen zum Projekt hinzugefügt. Die folgende Tabelle von Dokumentation Webseite von Microsoft fasst die Hauptfunktionen von TextBox und RichTextBox zusammen.

Control	Rechtschreibprüfung in Echtzeit	Kontextmenü	Formatierungsbefehle wie ToggleBold (STRG+B)	FlowDocument -Inhalte wie Bilder, Absätze, Tabellen usw.
TextBox	Ja	Ja	Nein	Nein.
RichTextBox	Ja	Ja	Ja	Ja

- Zeichnen: es wurde zusätzlich ein Zeichnen (Draw) mit einfachen Funktionen zu Projekt hinzugefügt. Für weitere Versionen von dieser App auf andere Geräte wie Z. B. Tablette ist ein Zeichnen Funktion unverzichtbar.
- Es wurde weiter Funktion, die nicht spezifisch in pflichtenheften erwähnt und heutzutage fast in jeden Texteditor verfügbar sein müssen, hinzugefügt. Z. B.
 - Undo und Redo.
 - Datei als PDF-Datei speichern.
 - Wörter Zähler.
 - Zählung von Zeilen und Spalten.
 - Höhe und Breite von Windows anzeigen.
 - Button für Information und Kontakt.

9.2 Beschreibung wichtiger Quellcode Ausschnitte.

Hier werden das Design und einige Funktionen des Texteditors mithilfe von Screenshots des Quellcodes beschrieben.

9.2.1 MainWindow:

besteht aus einem Tab Control mit zwei Item (User Control) Texteditor und Zeichnen, hier wird zwischen beide Ansichten navigiert. Die erste Ansicht nach Starten der App ist Texteditoransicht. Main Windows enthält feste Eigenschaften für die ganze App wie das Symbol, die Höhe, die Breite, Mindesthöhe, Mindestbreite und Fensterstartposition des Hauptfensters. (Abb. 25)

Für Übersichtlichkeit und bessere Struktur ist Quellcode des Texteditors und Zeichnen in zwei separaten Dateien geschrieben. Jede Datei befindet sich in einem Verzeichnis.

```
<Window x:Class="Texteditor.MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:Texteditor"
    mc:Ignorable="d"
    xmlns:Text="clr-namespace:Texteditor.Texteditor_"
    xmlns:Zeichnen="clr-namespace:Texteditor.Zeichnen_"
    Title="My Editor"
    Height="800"
    Width="790"
    MinWidth="338"
    MinHeight="400"
    WindowStartupLocation="CenterScreen"
    Icon="Images/icon.ico">

    <TabControl>
        <TabItem Header="Text Editor" Width="150">
            <Text:UserControlTexteditor/>
        </TabItem>
        <TabItem Header="Zeichnen" Width="150">
            <Zeichnen:UserControlZeichnen/>
        </TabItem>
    </TabControl>
</Window>
```

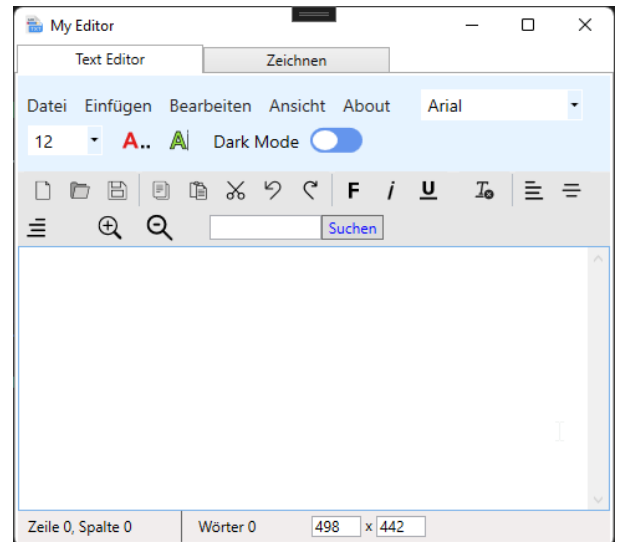
(Abb. 25, MainWindow)

9.2.2 UserControl_Texteditor

9.2.3 SizeChanged Ereignis

Durch dieses Ereignis wird das Konzept von Responsive Design realisiert. Responsive Webdesign oder Responsive Design ist ein Ansatz für Webdesign, der darauf abzielt, Webseiten auf einer Vielzahl von Geräten und Fenster- oder Bildschirmgrößen von minimaler bis maximaler Anzeigegröße gut darzustellen, um Benutzerfreundlichkeit und Zufriedenheit zu gewährleisten. (Wikipedia)

Es wird ein Objekt von GridLength initialisiert, (Abb. 25) GridLength stellt die Länge von Elementen dar, Elemente wie ColumnDefinition und RowDefinition verwenden diesen Typ, um Breite und Höhe zu beschreiben. Dieses Objekt enthält der Parameter GridUnitType mit Auto Eigenschaft, GridUnitType Beschreibt die Art von Wert, den ein GridLength-Objekt enthält, Auto Die Größe wird durch die Größeneigenschaften des Inhaltsobjekts bestimmt. Wie in (Abb. 26) zu sehen die Inhalte von Zeile 1 und Zeile 2 (Menu und Icons) passen sich bei Verkleinerung der Fenstergröße an Fenstergröße an.



(Abb. 26)

```
1 Verweis
private void Window_SizeChanged(object sender, SizeChangedEventArgs e)
{
    GridLength length = new GridLength(0, GridUnitType.Auto);
    rtbRow1.Height = length;
    rtbRow2.Height = length;
}
```

(Abb. 25)

9.2.4 Windows-Resources:

Ressourcen wird benutzt, um Daten zu packen und diese zentral zu manipulieren. Dieses Konzept nutzt Styles, in Styles wird verschiedene Eigenschaften je nach Steuerelement benutzt. In Code wird Background, Foreground, BorderBrush, Padding, Margin und FontSize Properties benutzt. Statt alle diese Eigenschaften auf jede Steuerelemente einsetzen, wird diese in Ressource eingepackt und nur Style in entspreche Steuerelemente verwendet. (Abb. 27, 28)


```

<UserControl.Resources>
  <Style TargetType="MenuItem" x:Key="menuItemProperties">
    <Setter Property="Background" Value="#ffffff"/>
    <Setter Property="Foreground" Value="White"/>
    <Setter Property="BorderBrush" Value="#ffffff"/>
    <Setter Property="Padding" Value="5"/>
    <Setter Property="Margin" Value="8 0"/>
    <Setter Property="FontSize" Value="14"/>
  </Style>
</UserControl.Resources>

```

(Abb. 27)

```

<MenuItem Name="menuItemNew" Header="Neu" Click="new_Click"
  Style="{StaticResource menuItemProperties}">
  <MenuItem.Icon>
    <svgc:SvgViewbox Source="/Images/new.svg"/>
  </MenuItem.Icon>
</MenuItem>

```

(Abb. 28)

9.2.5 Kopieren und Einfügen

Die Klasse Clipboard bietet Funktionen zum Platzieren von Daten in der Zwischenablage und zum Abrufen von Daten aus der Zwischenablage. Die Methode Clear löscht alle Daten aus der Zwischenablage des Systems. Die Methode Copy kopiert die aktuelle Auswahl des Texts Steuerelement in Clipboard.

Die Methode Paste fügt den Inhalt der Zwischenablage über der aktuellen Auswahl im Steuerelement ein. Die Methode SetText speichert den Text in der Zwischenablage. (Abb. 29)

```

1 Verweis
private void Copy_Click(object sender, RoutedEventArgs e)
{
    Clipboard.Clear();
    richtxtbox.Copy();
}

1 Verweis
private void Paste_Click(object sender, RoutedEventArgs e)
{
    richtxtbox.Paste();
    richtxtbox.Selection.Text = "";
}

1 Verweis
private void Cut_Click(object sender, RoutedEventArgs e)
{
    Clipboard.Clear();
    Clipboard.SetText(richtxtbox.Selection.Text);
    richtxtbox.Selection.Text = "";
}

```

(Abbildung 29)

9.2.6 Open File

Eine der wichtigsten Funktionen, die in Jedem Texteditor verfügbar ist, ist Datei zu öffnen. Durch Klicken auf den Button Öffnen, öffnet Open File Dialog. Zuerst wird ein Objekt von Klasse OpenFileDialog initialisiert und dieses mit Eigenschaften Title und Filter belegt. In Filter kann man verschieden Dateitypen festlegen um diese angezeigt werden. Die Datatypen hier sind TXT, RTF und XAML. Wenn Dialog geöffnet ist, wählt man eine Datei auf PC dann wird überprüft welche Dateitype ist das. Um einen Stream für die Datei bereitzustellen, wird ein Objekt von Klasse FileStream initialisiert und dies zwei Argumente eingeben, Dateiname durch FileName Eigenschaft von Objekt Openfile und FileMode.Open. Dann wird ein Bereich festgelegt und die neue Datei beladen zu werden. Dann wird die Methode Load benutzt, um aktuelle ausgewählt Datei aus Stream in angegebenes Dateiformat zu laden. (Abbildung 30)

```
2 Verweise
private void openFile_Click(object sender, RoutedEventArgs e)
{
    OpenFileDialog openFile = new OpenFileDialog();
    openFile.Title = "Öffnen";
    openFile.Filter = "Plain Text File (*.txt)|*.txt|Rich Text File (*.rtf)|*.rtf| " +
        "XAML File (*.xaml)|*.xaml|All files (*.*)|*.*";
    if (openFile.ShowDialog() == true)
    {
        if (openFile.FileName.EndsWith(".txt"))
        {
            FileStream fileStream = new FileStream(openFile.FileName, FileMode.Open);
            TextRange range = new TextRange(richtxtbox.Document.ContentStart, richtxtbox.Document.ContentEnd);
            range.Load(fileStream, DataFormats.Text);
            fileStream.Close();
        }
        else if (openFile.FileName.EndsWith(".rtf"))
        {
            FileStream fileStream = new FileStream(openFile.FileName, FileMode.Open);
            TextRange range = new TextRange(richtxtbox.Document.ContentStart, richtxtbox.Document.ContentEnd);
            range.Load(fileStream, DataFormats.Rtf);
            fileStream.Close();
        }
        else if (openFile.FileName.EndsWith(".xaml"))
        {
            FileStream fileStream = new FileStream(openFile.FileName, FileMode.Open);
            TextRange range = new TextRange(richtxtbox.Document.ContentStart, richtxtbox.Document.ContentEnd);
            range.Load(fileStream, DataFormats.Xaml);
            fileStream.Close();
        }
    }
}
```

(Abbildung 30)

9.2.7 Save File

Die Funktion Datei Speichern funktioniert ähnlich wie Funktion Datei Öffnen, folgende sind drei Unterschiede in Funktionsweise: Statt Open File Dialog wird Save File Dialog angerufen, statt in Stream angegebene Argument FileMode.Open wird FileMode.Save eingeben und der dritte Unterschied ist Methode Save statt Method Load. (Abbildung 31)

```

3 Verweise
public void save()
{
    SaveFileDialog saveFile = new SaveFileDialog();
    saveFile.DefaultExt = "*.txt";
    saveFile.Filter = "Plain Text File (*.txt)|*.txt|All files (*.*)|*.*";

    if (saveFile.ShowDialog() == true)
    {
        FileStream fileStream = new FileStream(saveFile.FileName, FileMode.Create);
        TextRange range = new TextRange(richtxtbox.Document.ContentStart, richtxtbox.Document.ContentEnd);
        range.Save(fileStream, DataFormats.Text);
    }
}

```

(Abbildung 31)

9.2.8 Speech Funktion

Wandelt Text zu Audio um, also wandelt geschriebenen Text in gesprochenen Text. Es wird ein SpeechSynthesizer-Objekt initialisiert, das bietet Zugriff auf die Funktionalität einer installierten Sprachsynthese-Engine. Dann wird die Methode SpeakAsync mit Text als Parameter eingeben, SpeakAsync Spricht synchron den Inhalt eines -Objekts. Die Methode Dispose verwirft das SpeechSynthesizer-Objekt und gibt Ressourcen frei, die während der Sitzung verwendet werden, ohne diese ohne kann den Text gleichzeitig mehrmals gesprochen. (Abb. 32)

```

1 Verweis
private void speech_Click(object sender, RoutedEventArgs e)
{
    reader.Dispose();
    string textSelected = richtxtbox.Selection.Text;
    string allText = new TextRange(richtxtbox.Document.ContentStart, richtxtbox.Document.ContentEnd).Text;

    if (allText == "")
    {
        MessageBox.Show("Bitte Schreiben Sie den Text zuerst", "Speech Text",
            MessageBoxButton.OK, MessageBoxImage.Warning);
    }
    else if (textSelected == "")
    {
        MessageBox.Show("Bitte Markieren Sie den Text zuerst", "Speech Text",
            MessageBoxButton.OK, MessageBoxImage.Information);
    }
    else
    {
        reader = new SpeechSynthesizer();
        reader.SpeakAsync(richtxtbox.Selection.Text);
        menuItemspeechStop.IsEnabled = true;
    }
}

```

(Abbildung 32)

9.2.9 UserControl_Zeichnen

9.2.10 Select Pen Click:

Hier wird durch Click-Ereignis festgelegt, welche Bearbeitungsmodus zu verwenden, Stift Radiergummi oder Select bereich. Durch Switch wird der Name des Buttons geprüft und für jeden Fall die entsprechende Bearbeitungsmodus für Inkcanvas festgelegt. (Abbildung 33)

```
3 Verweise
private void Select_PenModus_Click(object sender, RoutedEventArgs e)
{
    Button btn = (Button)sender;
    switch (btn.Name)
    {
        case "pen":
            myInkcanvas.EditingMode = InkCanvasEditingMode.Ink;

            break;
        case "eraser":

            myInkcanvas.EditingMode = InkCanvasEditingMode.EraseByPoint;
            myInkcanvas.Cursor = Cursors.Cross;

            break;
        case "select":
            myInkcanvas.EditingMode = InkCanvasEditingMode.Select;
            break;
    }
}
```

(Abbildung 33)

9.3 Installationsanleitung

Bei Verbindung mit Datenbank muss Servername (connectionString) angepasst werden.

```
1 Verweis
private void SaveDB_Click(object sender, RoutedEventArgs e)
{
    string richtextbox = new TextRange(richtxtbox.Document.ContentStart, richtxtbox.Document.ContentEnd).Text;
    string dbName = "Editor";
    //string connectionString = $"Server=KALO\\SQLEXPRESS; Database={dbName};
    //                                     Trusted_Connection=True; MultipleActiveResultSets=True";

    string connectionString = $"Server=localhost\\Initial Catalog={dbName};" +
        "User id=sa;" +
        "Password=mssqlserver;" +
        "MultipleActiveResultSets=True";

    using (SqlConnection con = new SqlConnection(connectionString))
    {
        con.Open();
        string sqlSelect = "INSERT INTO Data(Text, Zeitstempel)"
            + $" VALUES('{richtextbox}' , GETDATE());";

        using (SqlCommand cmd = new SqlCommand(sqlSelect, con))
    }
}
```

9.4 Verwendete NuGet-Pakete

SharpVectors: Für Verwendung von SVG.

iTextSharp: Für Verwendung von PDF-Datei.

10 Testplan

Beschreibung: Text schreiben

Vorbedingung: Texteditor ist geöffnet.

Test-Schritte: Text wird in Textbereich geschrieben.

Erweitertes Resultat: den Text kann formatiert werden.

Beschreibung: Datei öffnen

Vorbedingung: Texteditor ist geöffnet.

Test-Schritte: 1. Auf den Button Öffnen wird gedrückt.
2. Vorhandene Datei wird ausgewählt.
3. Neue Datei wird geöffnet

Erweitertes Resultat: Es kann Datei mit verschiedene Dateierweiterung geöffnet werden.

Beschreibung: Datei speichern

Vorbedingung: Texteditor ist geöffnet.

Test-Schritte: 1. Text wird geschrieben.
2. Auf den Button Speichern wird gedrückt.
3. Datei wird umbenannt.
4. Datei wird gespeichert

Erweitertes Resultat: Die Datei kann mit verschiedene Dateierweiterung gespeichert werden.

Beschreibung: Image einfügen

Vorbedingung: Texteditor ist geöffnet.

Test-Schritte: 1. Auf Button Image Einfügen wird gedrückt.
2. Image wird ausgewählt
3. Image wird eingefügt.

Erweitertes Resultat: Es kann Bilder mit verschiedene Dateierweiterung eingefügt werden.

Beschreibung: Nach Wörter Suchen

Vorbedingung: Datei hat Textinhalte

Test-Schritte: 1. Maus auf Suchbereich klicken
2. Text schreiben
3. Auf Button Suchen Klicken
4. Gefundene Wörter werden mit gelb hervorgehoben

Erweitertes Resultat: Es kann nach einzelnen Buchstaben oder nach Worten gesucht werden.

11 Anhang / Ressourcen

11.1 Ressourcen

- [Google.com](https://www.google.com)
- [Stack Overflow](https://stackoverflow.com)
- [Youtube.com](https://www.youtube.com)
- [Wpf-tutorial.com](https://www.wpf-tutorial.com)
- [W3schools.com](https://www.w3schools.com)
- [Svgrepo.com](https://www.svgrepo.com)

11.2 Demo Video

YouTube Link: <https://youtu.be/44l2DmpPg28>