

# Pflichtenheft und technische Spezifikation im Programmierprojekt

Projekt: Annotationtool für Liniensegmente

Mitarbeiter: Mert Karadeniz, Emirhan Naroglu, William Eppel

## Versionshistorie

Datum	Bearbeiter	Durchgeführte Änderung
20.04.2021	Alle	<ul style="list-style-type: none"> <li>Aufsetzung des Pflichtenheftes</li> </ul>
23.04.2021	Alle	<ul style="list-style-type: none"> <li>Erarbeitung der Ziele des Projektes</li> <li>Erstellung der GUI Mockup</li> <li>Arbeiten am Pflichtenheft</li> </ul>
26.04.2021	Alle	<ul style="list-style-type: none"> <li>Überarbeitung der Visionen und Ziele</li> </ul>
01.05.2021	Alle	<ul style="list-style-type: none"> <li>Überarbeitung des Pflichtenhefts</li> <li>Überarbeitung der GUI</li> </ul>
02.05.2021	Alle	<ul style="list-style-type: none"> <li>Erarbeitung Aktivitätsdiagramm</li> <li>Use-Case-Diagramm erarbeitet</li> <li>Klassendiagramm erarbeitet</li> <li>Komponentendiagramm</li> </ul>
05.05.2021	Alle	<ul style="list-style-type: none"> <li>Überarbeitung Komponentendiagramm</li> <li>Überarbeitung GUI</li> <li>Überarbeitung Klassendiagramm</li> <li>Fertigstellung Use-Case-Diagramm</li> </ul>
08.05.2021	Alle	<ul style="list-style-type: none"> <li>Erstellung der GUI</li> <li>Erstellung der Klassen</li> <li>Gedanken zur Testphase</li> </ul>
12.05.2021	Alle	<ul style="list-style-type: none"> <li>Fertigung des Projektsetups</li> <li>Fertigung des Pflichtenhefts</li> </ul>

## INHALTSVERZEICHNIS

1. Versionen und Ziele
2. Anforderungen an Ihr System
  - 2.1. Use-Case-Diagramm
  - 2.2. Risiken
  - 2.3. User-Test
  - 2.4. Aktivitätsdiagramm
  - 2.5. GUI
3. Realisierung
  - 3.1. Allgemeines
  - 3.2. Komponentendiagramm
  - 3.3. Projektsetup
  - 3.4. Externe Schnittstellen
  - 3.5. Interne Schnittstellen
4. Test und Implementierungsphase
5. Lizenz

## Visionen und Ziele

Die zu entwickelnde Software wird ein Annotationstool zum Bearbeiten von Bildern für gegebene Liniensegmente.

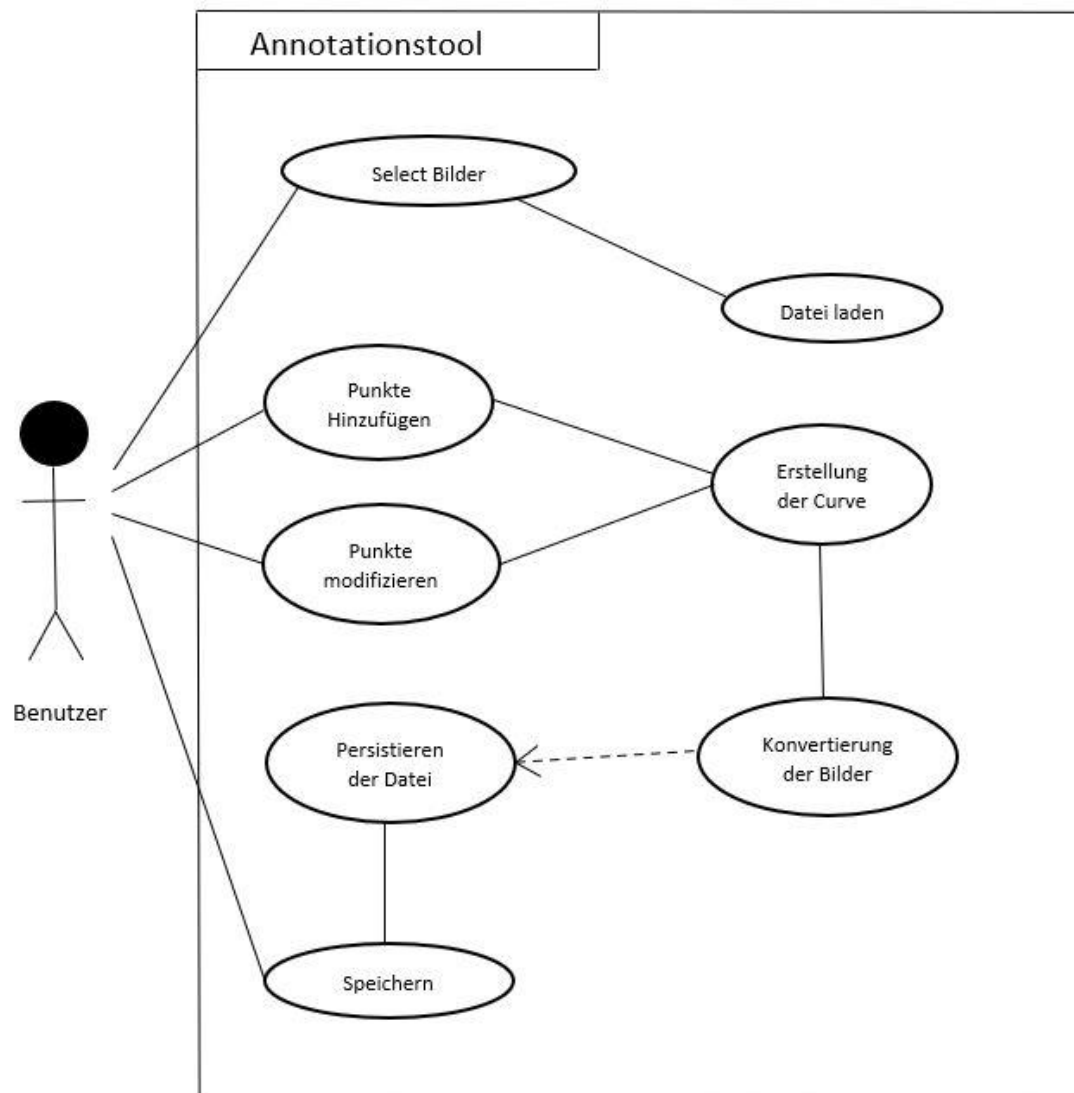
Der Nutzer soll die zu bearbeitenden Bilder innerhalb der Software hochladen und diese direkt bearbeiten können. Nachdem sie bearbeitet wurden soll man diese im selben Datentyp in einem anderen Speicherort speichern können.

Feature	Priorität (1 – 3)	Beschreibung	Aufwand In Tagen
1	1	Erstellung der GUI	4
2	1	Einlesen und Ablesen der Bilddatei	3
3	1	Wiedergabe des Bildes auf der GUI	3
4	1	Bearbeitung der Bilddatei	4
5	2	Einlesen und Auslesen von ganzen Ordnern mit Bilddateien	1
6	2	Verschiedene Farben generieren für verschiedene Schienen	1
7	2	Statusmeldungen ausgeben während der Bearbeitung was als nächstes zutun ist	2
8	3	Fehlerbehandlung: nicht importierbarer Dateityp wurde ausgewählt	2
9	3	Allgemeine Wiederherstellung des Tools: alles zurücksetzen, wie als man das Programm gestartet hatte	1

# Anforderungen an Ihr System

## Use-Case-Diagramm

Use-Case	Beschreibung
Select Bilder	Zuweisung des Dateipfads, des zu bearbeitenden Modells
Datei laden	Ausgewähltes Image verwenden
Punkte Hinzufügen	Annotationstool zur Kurven Generierung
Punkte modifizieren	Möglichkeit zur Bearbeitung bereits erstellter Punkte
Erstellung der Curve	Curve wird aus gesetzten Punkten berechnet und gesetzt
Konvertierung der Bilder	Überführung in ein gültiges Bildformat
Persistieren der Datei	Sicherung des fertiggestellten Images
Speichern	Endgültiges Persistieren der Datei



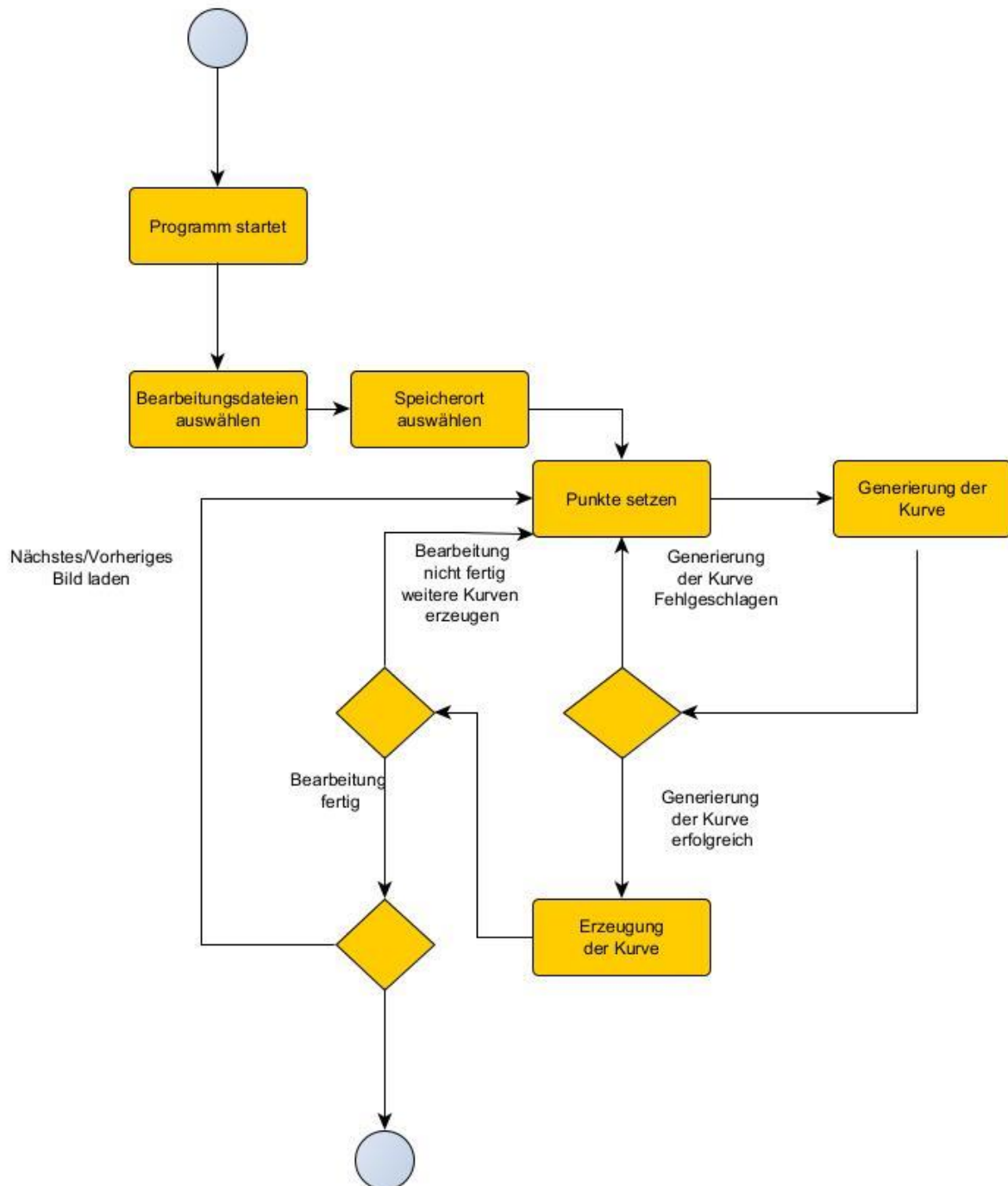
## Risiken

RISIKO	RISIKOBEHANDLUNG
Toolnutzung ist nicht einfach	User-Tests
Kurven werden nicht generiert	Es werden von Anfang an Dummy-tests durchgeführt
Gesetzte Punkte werden nicht angenommen	Das ist der erste Bestandteil unseres Programmes, daher wird sich als erstes durch Dummy-tests darum gekümmert

## User-Tests

Version	Beschreibung	User
Version 1	Das Programm sollte seinen Zweck erfüllen Die Grundfunktionen sollten funktionieren	Testen der Funktionen Testen der Handhabung Feedback an das Team
Version 2	User Feedback wird bearbeitet und integriert ins Programm Weitere geplante Features werden implementiert	Testen der neuen Handhabung Testen der Features Feedback an das Team
Version 3 (Final)	Feedback wird bearbeitet und integriert Feinschliffe am Programm	

## Aktivitätsdiagramm





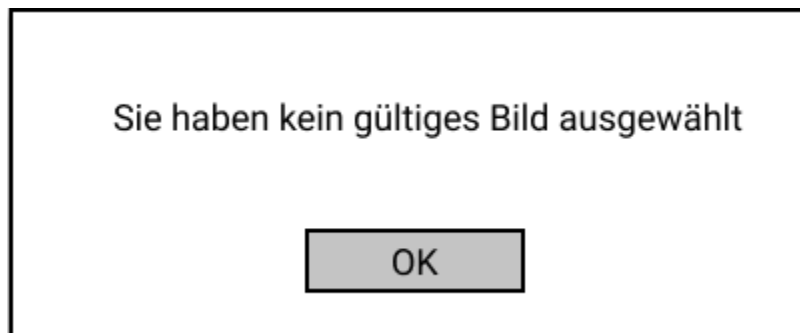
## GUI

Der Benutzer soll über die Taskleiste seinen Ordner auswählen können indem er die zu bearbeitenden Bilder abgespeichert hat, somit werden alle Bilddateien hochgeladen und sind bereit um bearbeitet zu werden. Bei der Bearbeitung setzen wir darauf, dass der Nutzer eine große Fläche für das Bild hat, somit sollten alle Bildformate reinpassen und es sollten keine Probleme entstehen. Da das Bild auch größer wär könnte der Nutzer optimalere Punkte setzen. Die Buttons werden in einem Dropdown-Menü wiedergegeben, in dieser befinden sich die Buttons, Punkte(p), Neue Schiene(n), nächstes Bild(Pfeil nach rechts), vorheriges Bild(Pfeil nach links). Um die Tool-Nutzung optimal für jedermann zu machen gibt es für jeden Button noch einen Shortcut, sodass man auch nicht jedes Mal in das Drop-Menü muss



So würde das Dropdown-Menü aussehen. Es werden da nochmal die Shortcuts gezeigt und Nutzer die diese nicht nutzen wollen, können sich hier ihre Funktionen auswählen

Punkte	(p)
Neue Schiene	(n)
nächstes Bild	(->)
vorheriges Bild	(<-)



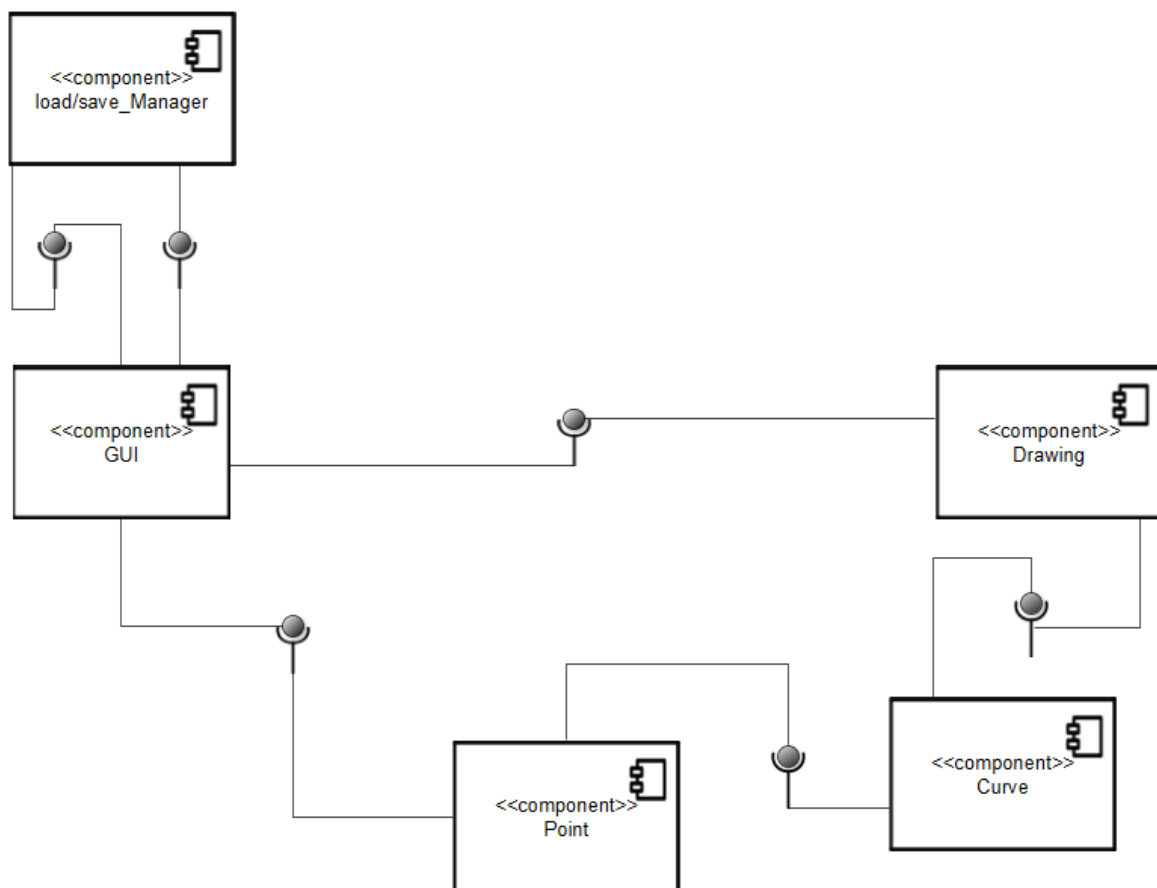
So in etwa könnte eine Fehlermeldung aussehen, diese würde erscheinen mit verschiedener Beschreibung zum Error, wenn zum Beispiel: Kein Bild ausgewählt wurde und Punkte gesetzt werden,

oder ein Datei mit falschem Format ausgewählt wird.

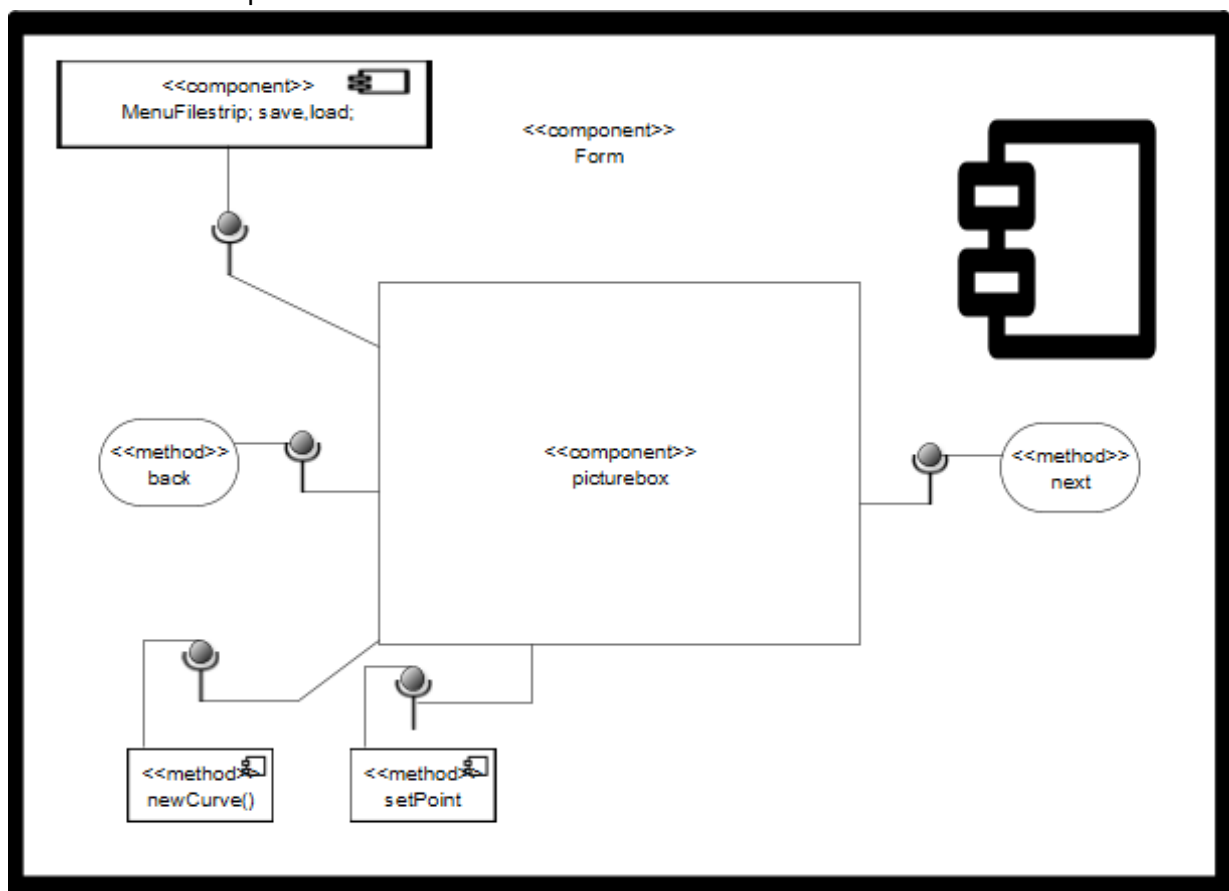
## Realisierung

### Allgemeines

So sieht das Komponentendiagramm des Programms aus.



So sehen die Komponenten der GUI aus.



Komponente	Beschreibung	Team-Mitglied
GUI	<ul style="list-style-type: none"> <li>• Auswahl der Bearbeitungsdatei</li> <li>• Auswahl des Speicherorts</li> <li>• Gibt ausgewählte Bilddatei aus</li> <li>• Spielraum für die Bearbeitung</li> <li>• Gibt Fehler aus, wenn welche entstehen</li> </ul>	Emirhan
load/save_manager	<ul style="list-style-type: none"> <li>• Nimmt Speicherinformationen von der GUI und führt diese aus</li> <li>• Speichert und Lädt die in der GUI ausgewählten Dateien</li> </ul>	William
StatusManager	<ul style="list-style-type: none"> <li>• Empfängt Informationen von der GUI was als nächstes zutun ist</li> <li>• Gibt dem Nutzer Informationen was als nächstes zutun ist</li> </ul>	William
Point	<ul style="list-style-type: none"> <li>• Bietet den Konstruktor für die gewählten Punkte in der GUI</li> </ul>	William
Curve	<ul style="list-style-type: none"> <li>• Nimmt mehrere Punkte als Liste auf und generiert dadurch eine Kurve</li> </ul>	Mert
Drawing	<ul style="list-style-type: none"> <li>• Erzeugt die Kurve die von Curve generiert wurde auf der GUI</li> <li>• Gibt die Farbe der Kurve vor</li> <li>• Unterscheidet zwischen linker oder rechter Schiene</li> <li>• Unterscheidet zwischen neuer oder gegebener Schiene</li> </ul>	Mert

## GUI

Die GUI bietet den kompletten Spielplatz des Nutzers beim benutzen unseres Tools. Er lädt das zu bearbeitende Bild, und speichert es auch wenn der Nutzer es verlangt dafür nutzt er die Load/Save Komponente. Der Nutzer kann auf der GUI Punkte generieren indem er frei auf der GUI klicken kann, diese werden dann der Komponente Point gegeben.

## load/save\_manager

Der load/save\_manager speichert und lädt die Dateien wie der Name schon sagt. Er kriegt Informationen, wie ob er speichern soll oder laden soll von der GUI, und führt diese danach aus. Außerdem kriegt er den Dateispeicherort von der GUI.

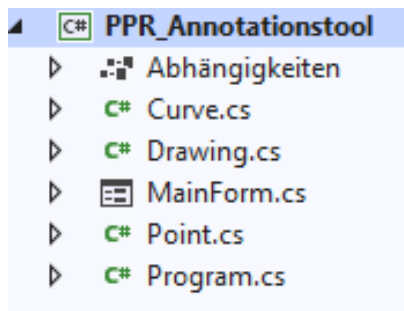
## Curve

Die Curve nimmt die erstellten Punkte als Liste, und modifiziert sie wenn gewollt (z.B Löschen eines Punktes, oder Hinzufügen eines Punktes). Außerdem generiert die Curve durch die Punkte auch die gewollte Kurve die, die Schiene hat.

## Drawing

Die Drawing Komponente, ist die Komponente die sozusagen alles auf Papier bringt. Diese bringt mehrere bestimmte Farben mit sich. Es kann die generierte Curve erzeugen, aber auch wieder löschen.

## Projektsetup

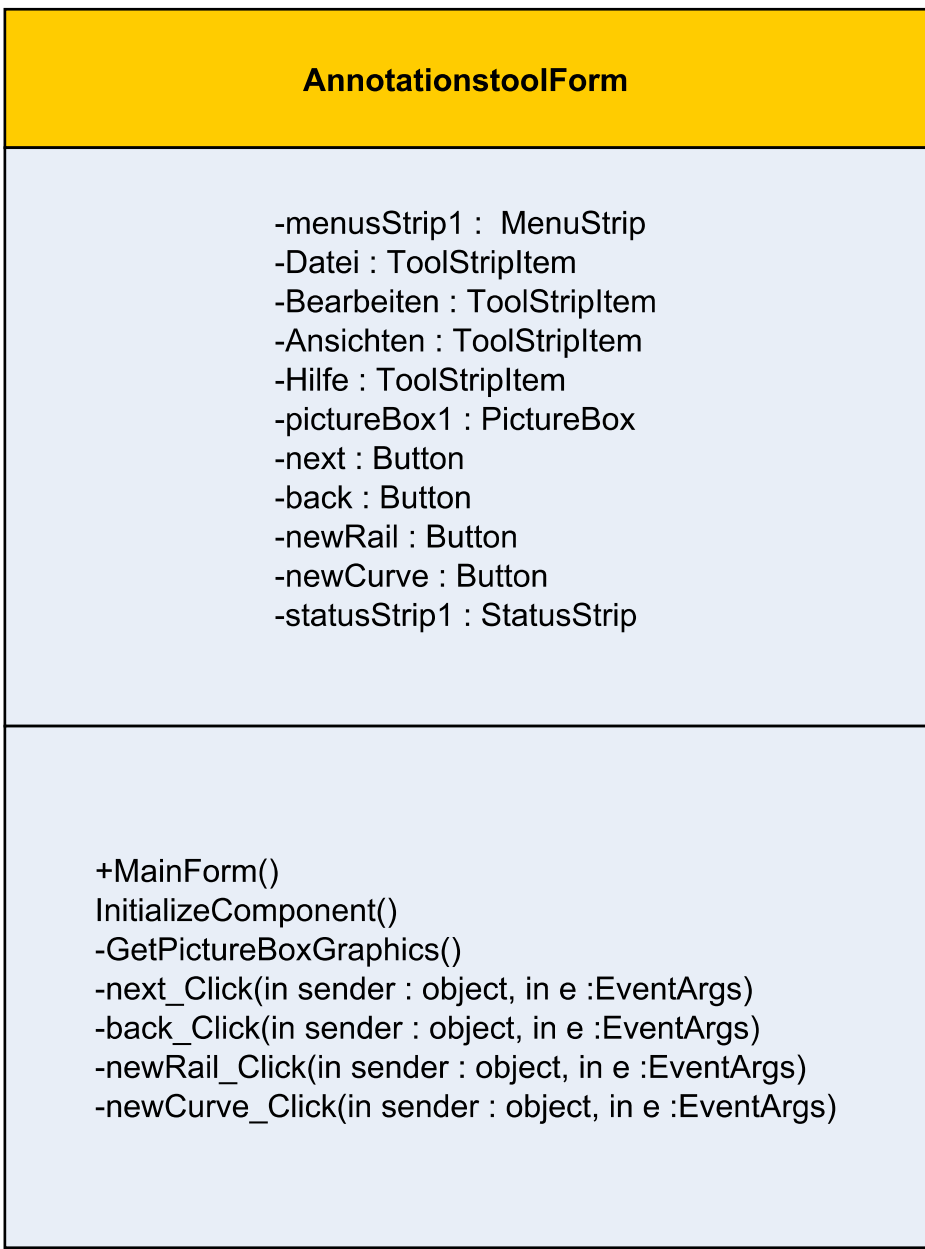


Das Projektsetup besteht aus 4 Hauptkomponenten. Die GUI wird nachdem gezeigten Mockup erstellt. Und die Klassen nachdem Klassendiagramm. Diese können dann auch in Git besichtigt werden.

## Externe Schnittstellen

Für das Zeichnen unserer Kurve benutzen wir die Bibliothek System.Drawing;, diese enthält Methoden und zum Beispiel auch eine Klasse "Pen". Außerdem arbeiten wir mit lokal geladenen Dateien, demnach brauchen wir einen externen Zugriff auf Dateien. Die Bilder die vom Nutzer ausgewählt werden können, können vom Typ .jpg .png sein.

## Interne Schnittstellen



## Test und Implementierungsphase

Branch	Beschreibung	Test
Master	Zusammenführung fertiger Teile	<ul style="list-style-type: none"> <li>User-test (Risikobehandlung)</li> </ul>
Merts	<ul style="list-style-type: none"> <li>Implementierung von Generierung und Erzeugung der Linien auf der Picture box</li> </ul>	<ul style="list-style-type: none"> <li>benötigt Dummy klasse von Point</li> <li>testet auf einer beliebigen Picture box von Windowsforms</li> </ul>
Williams	<ul style="list-style-type: none"> <li>Implementierung der Klasse Point</li> <li>Implementierung des StatusManagers</li> <li>Implementierung des load/Save_managers</li> </ul>	<ul style="list-style-type: none"> <li>Generierung diverser Punkte</li> <li>Abgleich der Statusmeldungen mit gewünschten Werten</li> <li>Speichern / Laden von zu bearbeitenden bzw. fertig bearbeiteten Dateien</li> </ul>
Emirhans	<ul style="list-style-type: none"> <li>Beinhaltet die Benutzeroberfläche des Programms</li> <li>Visualisierung der ausgewählten Bilder</li> <li>Visualisierung der Drawing Komponente</li> </ul>	<ul style="list-style-type: none"> <li>benötigt Dummy klasse von Point und Drawing</li> <li>Prüfung der Funktionalität der Steuerungsmechanismen der GUI durch User - Tests</li> </ul>





**Hochschule für Technik  
und Wirtschaft Berlin**

University of Applied Sciences

## Lizenz

Wir verwenden die übliche Lizenz für Hochschulprojekte, die MIT-Lizenz.

[Copyright](#) (c) 2021 HTW

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.