



Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

Pflichtenheft und technische Spezifikation im Programmierprojekt

Potraitgenerator

Mitarbeiter: Maged Yakan

Sefa Batuhan Aykac

Azim Izzudin Ramadhani Mubarak

Inhaltsverzeichnis

Visionen und Ziele	1
Anforderungen an Ihr System	2
Use-Cases	2
Risiken	2
GUI	3
Realisierung	6
Allgemeines	6
Interne Schnittstellen	6
Visual-Studio-Projektsetup	7
Externe Schnittstellen	7
Test und Implementierungsphase	7
Lizenz	9

1 Visionen und Ziele

Heutzutage nimmt die Nutzung von sozialen Medien zu, wie zum Beispiel Facebook, Instagram und viele mehr. Eines der Dinge, denen wir in sozialen Medien häufig finden, sind Fotos. Wie die Phrase *"Picture can paint a thousand words"*, werden Fotos verwendet, um Emotionen auszudrücken, Informationen mitzuteilen und miteinander zu kommunizieren.

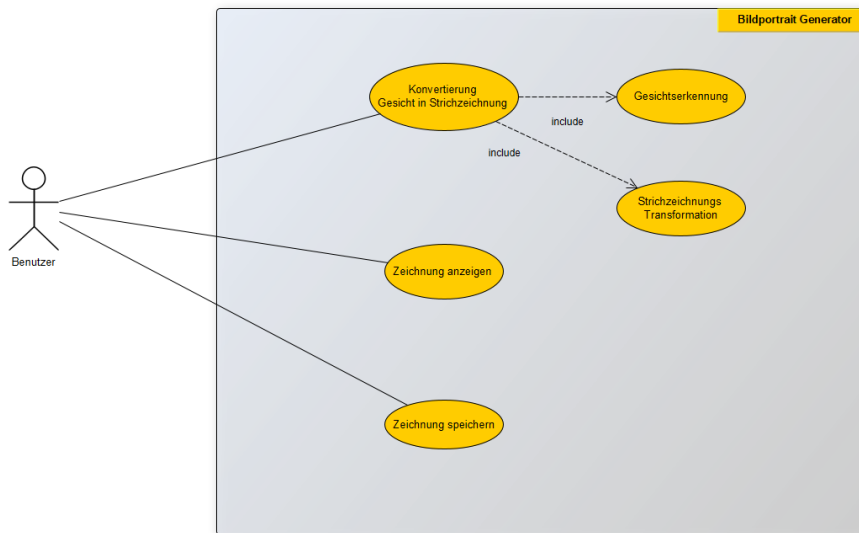
Während sich die Technologie entwickelt, werden viele Möglichkeiten verwendet, um ein Foto attraktiver zu machen. Eine davon ist die Filterfunktion, die in den letzten Zeiten häufig verwendet wird, um nicht nur Fotos zu verschönern, sondern auch um andere künstliche Charakter mit Hilfe deren darzustellen.

Ziel dieses Projekts ist es, ein Programm zu erstellen, mit dem große Fotosammlungen nach Gesichtern durchsucht und alle gefundenen Gesichter mit Hilfe eines neuronalen Netzwerks in Strichzeichnungen umgewandelt werden können. Unterstützt von einer einfachen und übersichtlichen Benutzeroberfläche, damit Benutzer dieses Programm problemlos verwenden können.

Name	System
Azim	Graphic User Interface
Maged	Gesichtserkennung
Sefa	Portrait-Converter

2 Anforderungen an Ihr System

2.1 Use-Cases



2.2 Risiken

- Fehlende Erfahrung mit Python
Recherchen durchführen, Videos anschauen, mini Crashkurs belegen
- Nicht alle Gruppenmitglieder führen das Projekt bis zum Ende durch
Immer aufs neue motivieren, man hat nichts zu verlieren
- Fehlende bzw. schwache Kommunikation
2 mal in der Woche BBB Session
- Algorithmus integrieren bzw. zum laufen bringen
Recherche durchführen, verschiedene Methoden ausprobieren
- Unterschiedliche Erfahrungen und Wissensstände
Die Erfahrungen miteinander teilen bzw. gegenseitig beibringen
- Erstes richtiges Projekt (unerfahren)
Organisiert an das Projekt rangehen, schritt für schritt gemeinsam bearbeiten

2.3 GUI

Abbildung 1: Home Screen

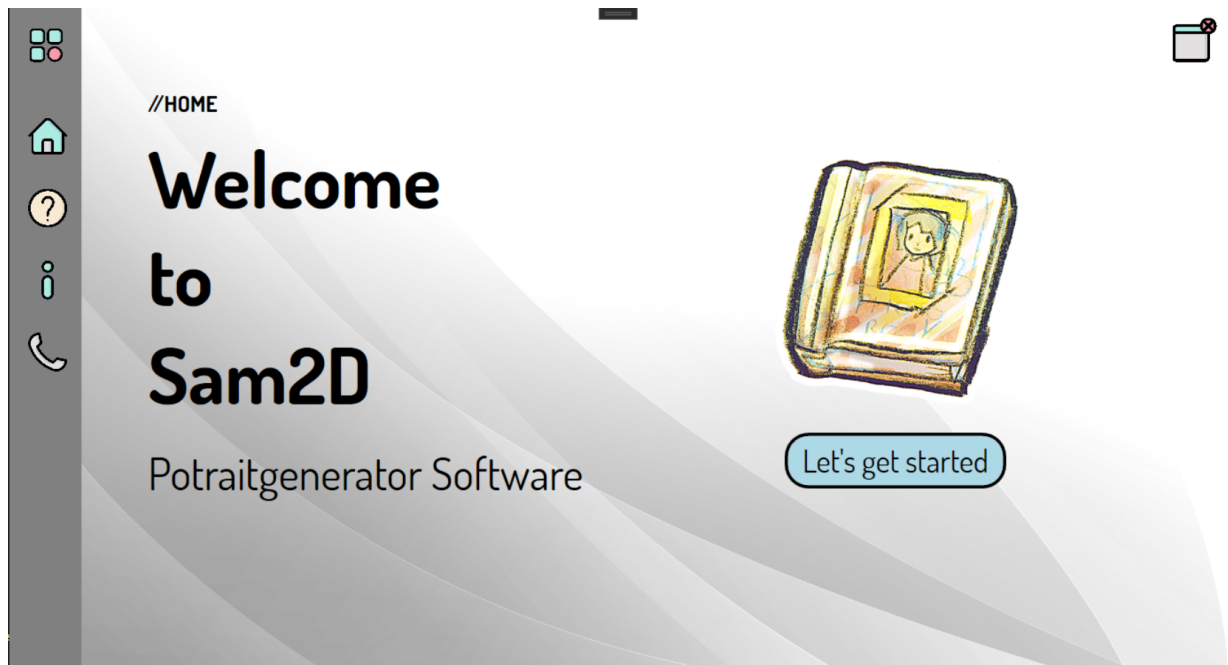


Abbildung 2: Menu



Abbildung 3: Datei hochladen

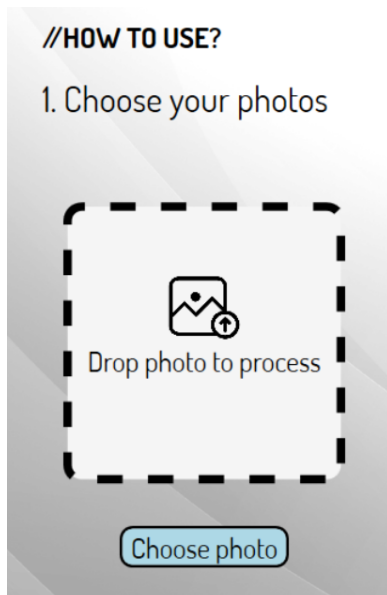


Abbildung 4: Ladevorgang

2. Are you sure? Check it!

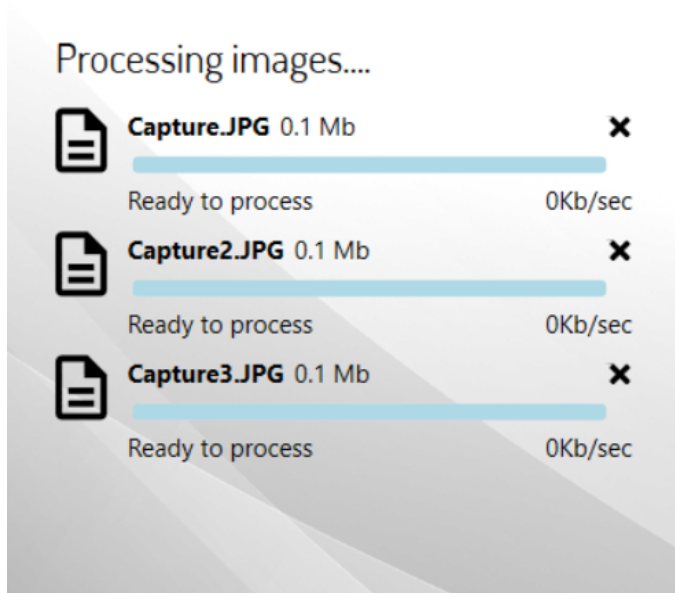


Abbildung 5.1: Button zum Ausführen des Programms wenn keine oder falsche Bilddatei hochgeladen wird

3. If you sure, then...

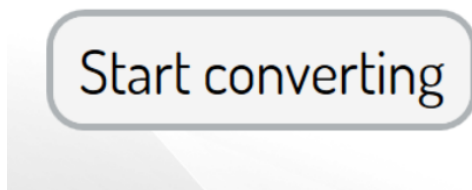
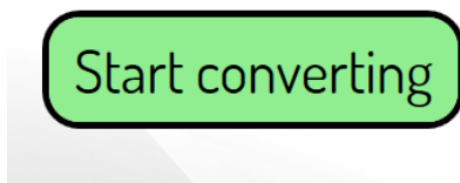


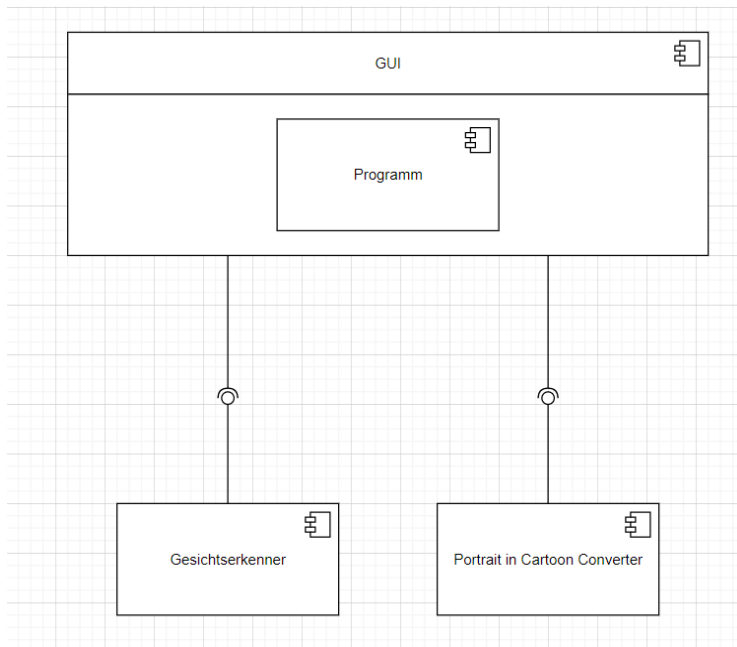
Abbildung 5.2: Button zum Ausführen des Programms wenn eine passende Bilddatei hochgeladen wird

3. If you sure, then...

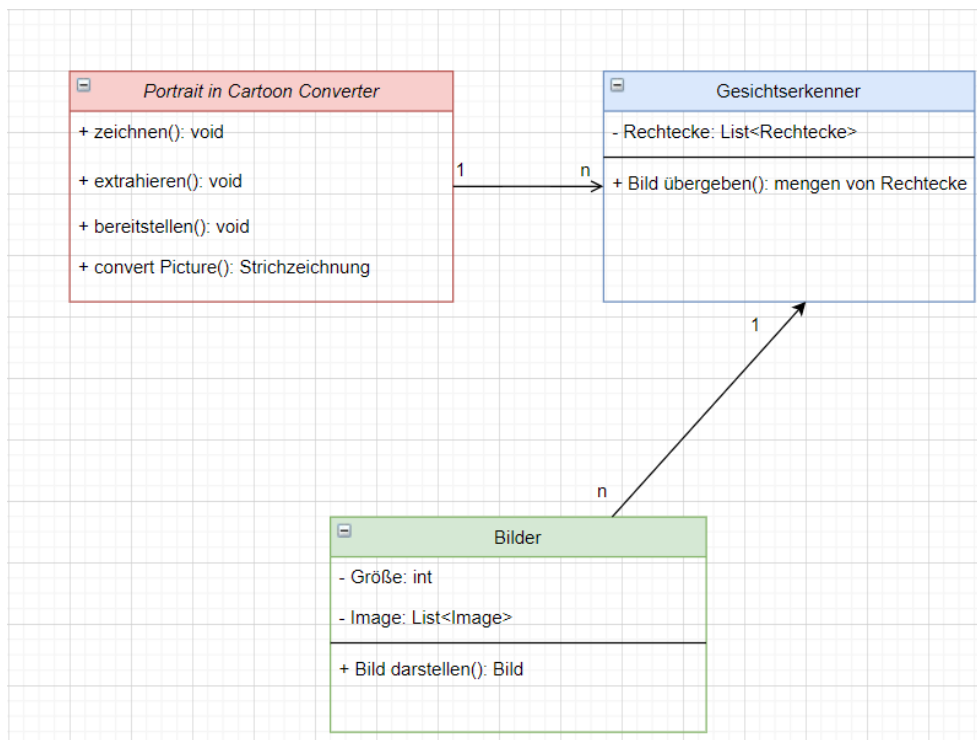


3 Realisierung

3.1 Allgemeines



3.2 Interne Schnittstellen



3.3 Visual-Studio-Projektsetup

Sofern Sie UI-Elemente in Ihrem Prototyp haben, erstellen Sie das in 2.3 vorgeschlagene Mockup (noch ohne Funktion) für Ihren Prototyp.

Ergänzen Sie Ihr Visual-Studio-Projekt um Klassen, die die in 3.2 definierte Schnittstelle implementieren bzw. von der abstrakten Klasse erben und geben Sie der Klasse einen zweckmäßigen Namen. Hinweis: alle Klassenbibliotheken (bis auf die CommonInterfaces) müssen eine Klasse beinhalten, die ein Interface implementiert oder ein Interface einer anderen Komponente benutzen.

Checken Sie die Projektmappe in git ein.

3.4 Externe Schnittstellen

Methode	Begründung
Windows Presentation Foundation (WPF)	Moderne grafische Benutzeroberfläche, vielfältiger als Windows Forms
IronPython	Kompatibel mit c#, um Python Codes auszuführen
OpenCV	Dient dazu die Bilder im Cartoon Skizzen zu zeichnen

4 Test und Implementierungsphase

Test-Szenario	Überprüfen, ob Benutzer mit der GUI interagieren kann
Test 1	Alle buttons durchklicken und auf Funktionalität testen
Test 2	Fehleranalyse starten - debuggen
Test 3	Mit GUI Testing Software überprüfen

Test-Szenario	Überprüfen, ob Benutzer Bilder auswählen können
Test 1	Abfrage von Bildern mit JPG, JPEG und PNG Datentyp
Test 2	Durch Drag & Drop

Test-Szenario	Überprüfen, ob von den Gesichtern alle Details erkannt werden
Test 1	Prozess durchführen, bis alle wichtigen Details erkannt werden
Test 2	Fehlerhafte Erkennungen beheben

Um während der Implementierungsphase kleine bzw. große Änderungen am Code oder Fehlerbehebungen durchzuführen, werden wir mit git Branches arbeiten.

Des Weiteren werden wir alle Teilergebnisse durch das Mergen in Git zusammenfügen.

5 Lizenz

Copyright (c) <2021> < ML.Students (Sefa Batuhan Aykac, Maged Yakan, Azim Izzudin Ramadhani Mubarak >

Jedem, der eine Kopie dieser Software und der zugehörigen Dokumentationsdateien (die "Software") erhält, wird hiermit kostenlos die Erlaubnis erteilt, ohne Einschränkung mit der Software zu handeln, einschließlich und ohne Einschränkung der Rechte zur Nutzung, zum Kopieren, Ändern, Zusammenführen, Veröffentlichen, Verteilen, Unterlizenzieren und/oder Verkaufen von Kopien der Software, und Personen, denen die Software zur Verfügung gestellt wird, dies unter den folgenden Bedingungen zu gestatten:

Der obige Urheberrechtshinweis und dieser Genehmigungshinweis müssen in allen Kopien oder wesentlichen Teilen der Software enthalten sein.

DIE SOFTWARE WIRD OHNE MÄNGELGEWÄHR UND OHNE JEDLICHE AUSDRÜCKLICHE ODER STILLSCHWEIGENDE GEWÄHRLEISTUNG, EINSCHLIEßLICH, ABER NICHT BESCHRÄNKT AUF DIE GEWÄHRLEISTUNG DER MARKTGÄNGIGKEIT, DER EIGNUNG FÜR EINEN BESTIMMTEN ZWECK UND DER NICHTVERLETZUNG VON RECHTEN DRITTER, ZUR VERFÜGUNG GESTELLT. DIE AUTOREN ODER URHEBERRECHTSINHABER SIND IN KEINEM FALL HAFTBAR FÜR ANSPRÜCHE, SCHÄDEN ODER ANDERE VERPFLICHTUNGEN, OB IN EINER VERTRAGS- ODER HAFTUNGSKLAGE, EINER UNERLAUBTEN HANDLUNG ODER ANDERWEITIG, DIE SICH AUS, AUS ODER IN VERBINDUNG MIT DER SOFTWARE ODER DER NUTZUNG ODER ANDEREN GESCHÄFTEN MIT DER SOFTWARE ERGEBEN.