

Punkte:

- Pflichtenheft: 17,0/20
- Projektplan: 5,0/5

Pflichtenheft

HTW Berlin
3D-Scanner mit einer Intel RealSense

Autor: Vinh Thong Trinh, Mert Karadeniz, Habib Ben Khedher, William Eppel
Letzte Änderung: 24. Mai 2022
Dateiname: 02 Muster_Pflichtenheft.docx

Version: 1.0

Copyright

© Mohammad Abuosba

Die Weitergabe, Vervielfältigung oder anderweitige Nutzung dieses Dokumentes oder Teile davon ist unabhängig vom Zweck oder in welcher Form untersagt, es sei denn, die Rechteinhaber/In hat ihre ausdrückliche schriftliche Genehmigung erteilt.

Version Historie

Version	Datum	Verantwortlich	Änderung
0.1	05.05.2022	Vinh Thong Trinh, Mert Karadeniz, Habib Ben Khedher, William Eppel	Einleitung, Überblick
0.2	18.05.2022	Vinh Thong Trinh, Mert Karadeniz, Habib Ben Khedher, William Eppel	Funktionalitäten, Use-Case, Workflow
0.3	23.05.2022	Vinh Thong Trinh, Mert Karadeniz, Habib Ben Khedher, William Eppel	Optimiert, überarbeitet, Rechtschreibung und Grammatik
1.0	25.05.2022	Vinh Thong Trinh, Mert Karadeniz, Habib Ben Khedher, William Eppel	Abgabe

Inhaltsverzeichnis

Abbildungsverzeichnis	II
Tabellenverzeichnis.....	III
Verzeichnis vorhandener Dokumente	IV
1 Überblick	1
2 Hauptziele.....	2
3 Annahmen und Abgrenzungen	3
4 Workflow.....	4
5 Funktionalität	6
5.1 Überblick.....	6
5.2 Kamera initialisieren	11
5.3 Drehplattform durch GUI steuern	12
5.4 Kameraaufnahme starten	13
5.5 Konvertierung der Daten	14
5.6 Gescanntes Objekt Exportieren	16
5.7 Im Format .stl speichern	17
5.8 Einstellungen anpassen	19
5.9 Standardeinstellungen zurücksetzen	21
Modulabhängigkeiten.....	21
6 Wer hat was gemacht.....	22

Zeilenab-
stand

Abbildungsverzeichnis

Abbildung 1: Workflow Diagramm 3D-Scanner und Nutzer	4
Abbildung 2: Workflow beim Fehlerauftreten	5
Abbildung 3: Use-Case Diagramm 3D-Scanner.....	6
Abbildung 4: GUI Mockup mit UI Beschriftung	7
Abbildung 5: 3D-Sicht des Scanners.....	8
Abbildung 6: Draufsicht des Scanners	8
Abbildung 7: Seitenansicht des Scanners	9
Abbildung 8: Vorderansicht des Scanners	9
Abbildung 9: GUI Mockup – Nach Kameraaufnahme + Konvertierung der Daten.....	15
Abbildung 10: GUI Mockup – Export Button Ansicht.....	18
Abbildung 11: GUI Mockup – Einstellung Button Ansicht.....	20

Tabellenverzeichnis

Soll keine Tabelle sein!

Tabelle 1: Verzeichnis vorhandener Dokumente	IV
Tabelle 2: Hauptziele mit Beschreibung	2
Tabelle 3: fachliche und technische Annahmen	3
Tabelle 4: Abgrenzung (Was ist in dieser Lösung nicht enthalten bzw. abgedeckt)	3
Tabelle 5: UI-Elemente mit Erklärungen	7
Tabelle 6: Funktionalität Objekt Scan	10
Tabelle 7: Funktionalität Kamera initialisieren	11
Tabelle 8: Funktionalität Drehplattform durch GUI steuern	12
Tabelle 9: Funktionalität Kameraaufnahme starten	13
Tabelle 10: Funktionalität Konvertierung der Daten	14
Tabelle 11: Funktionalität Gescanntes Objekt exportieren	16
Tabelle 12: Im Format .stl speichern	17
Tabelle 13: Funktionalität Einstellung anpassen	19
Tabelle 14: Standardeinstellung zurücksetzen	21
Tabelle 15: Modulabhängigkeiten	21
Tabelle 16: Wer hat was gemacht	22

Verzeichnis vorhandener Dokumente

Alle für die vorliegende Spezifikation ergänzenden Unterlagen müssen hier aufgeführt werden.

Dokument	Autor	Datum
Lastenheft	HTW-Berlin	26.04.2022

Tabelle 1: Verzeichnis vorhandener Dokumente

Soll keine Tabelle sein

Falls doch: Tabellenüberschrift!

1 Überblick

Im Projekt soll ein 3D-Scanner entwickelt werden, der diverse Objekte, die bestimmte Voraussetzungen erfüllen, digitalisieren kann, sodass die erstellten digitalen Kopien weiterverwendet werden können. Die Schwierigkeit besteht darin, ein Produkt zu schaffen, das einen niedrigen Einkaufspreis sowie niedrige Instandhaltungskosten aufweist und zudem Kompakt und leicht zu transportieren ist. Die Schnittstelle der Hardware und Software besteht aus einem Arduino Mikroprozessor. Die Daten, die mithilfe der Software zur Digitalisierung des Objektes führen, werden durch die Intel Realsense Kamera Model D415 aufgenommen. Die aufgenommenen Daten liegen in Form von Tiefenbildern vor.

Das Produkt soll aus 3D - gedruckten Einzelteilen (Material) bestehen, um ein bestimmtes Gewicht nicht zu überschreiten. Durch intelligente Konzipierung der Konstruktion, soll geboten werden, dass der Nutzer allein die Instandhaltung gewährleisten kann. Der niedrige Einkaufspreis zeichnet sich durch die Nutzung einer einzigen Kamera aus.

2 Hauptziele

In der folgenden Tabelle sind die Hauptziele des Projektes mit den jeweiligen Implementationen aufgeführt.

#	Ziel	Beschreibung der Implementation
1	Erkennung eines Objektes	Intel Realsense D415(Tiefenkamera) Python Package PyRealSense2
2	Anzeigen des 3D-Objektes	Python Package Open3D und PIL
3	Ansteuerung der Hardware	Python Package PySerial
5	Ansteuerung Schrittmotor	Arduino Uno
6	Export des 3D-Objektes (.stl)	Python Package Open3D

Tabelle 2: Hauptziele mit Beschreibung

Tabellenüberschrift

- Inkonsistent zum Lastenheft:

beispielsweise Lastenheft, S. 1: "Ziel des Projektes ist es ein Produkt zu schaffen mit dem jeder Interessierte die Option hat diverse Objekte zu digitalisieren [-> Jedermann-Gerät]. ..." Wo wird dieses Ziel hier genannt?

- Ziele werden immer noch nicht SMART dargestellt

3 Annahmen und Abgrenzungen

In der folgenden Tabelle sind unsere Annahmen nummeriert beschrieben worden.

#	Annahmen
1	Besitz eines Rechners mit USB-Anschluss
2	Die Software ist auf dem Rechner installiert
3	Bilderkennung über Intel Realsense
4	Objekt darf keine spiegelnde Oberfläche besitzen
5	Objekt ist max. groß wie eine Tasse, 8cm Durchmesser, 8,5cm Höhe
6	Das Gewicht des Objektes überschreitet nicht 350 Gramm

Tabelle 3: fachliche und technische Annahmen

In der folgenden Tabelle sind unsere Abgrenzungen nummeriert beschrieben worden.

#	Abgrenzungen (Was ist in dieser Lösung nicht enthalten bzw. abgedeckt)
1	Es wird nicht gewährleistet, dass das erstellte Model vollkommen identisch mit dem gescannten Objekt übereinstimmt. Es wird damit gerechnet, dass es geringfügige Abweichungen der Maße zwischen Objekt und Modell gibt. Das sollte über Toleranz-Anforderungen definiert werden.

Tabelle 4: Abgrenzung (Was ist in dieser Lösung nicht enthalten bzw. abgedeckt)

Fehlt hier nicht Punkte, wie beispielsweise:

- Keine Realisierung einer Schnittstelle zu 3D-Drucker
- ...

4 Workflow

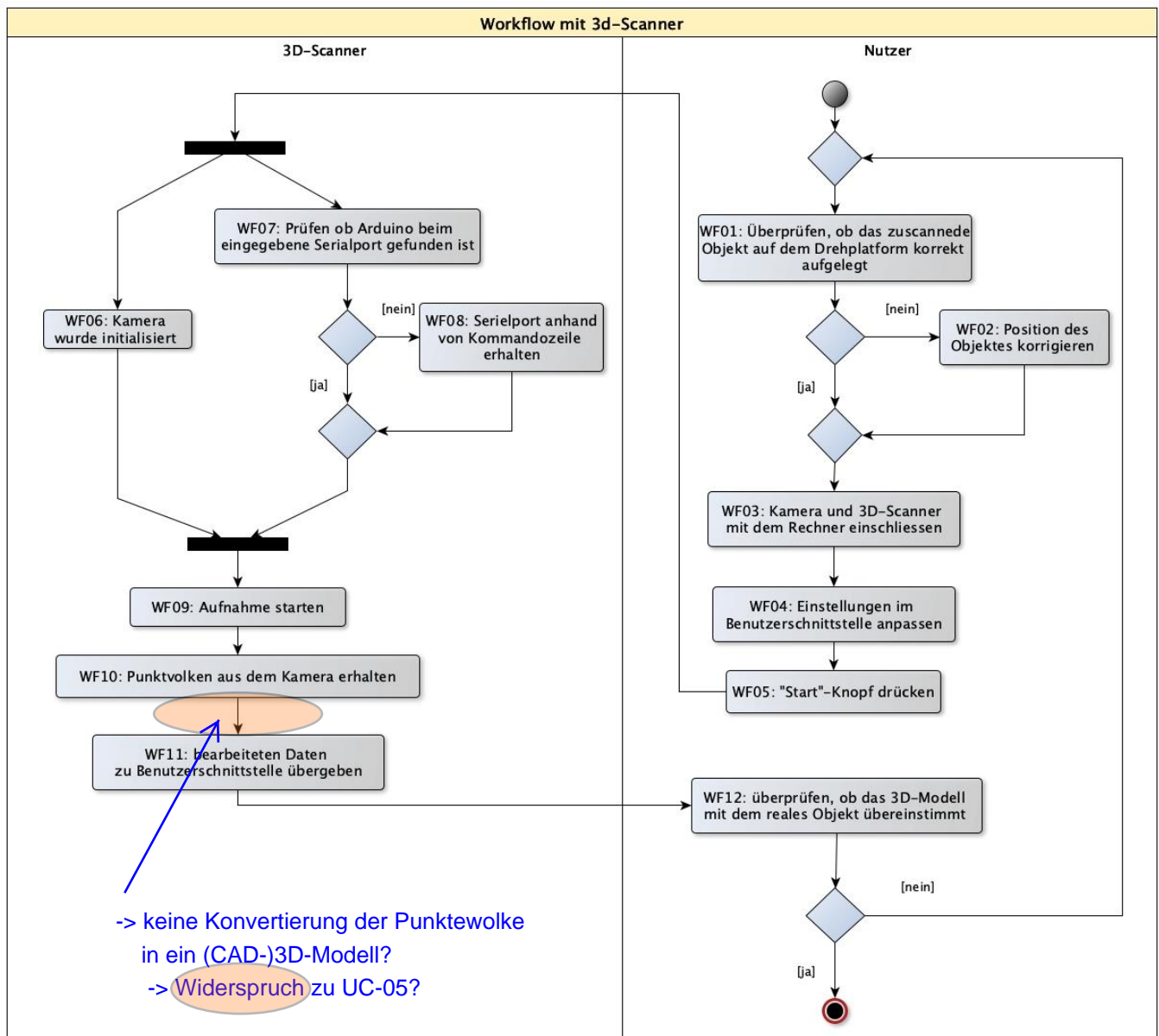


Abbildung 1: Workflow Diagramm 3D-Scanner und Nutzer

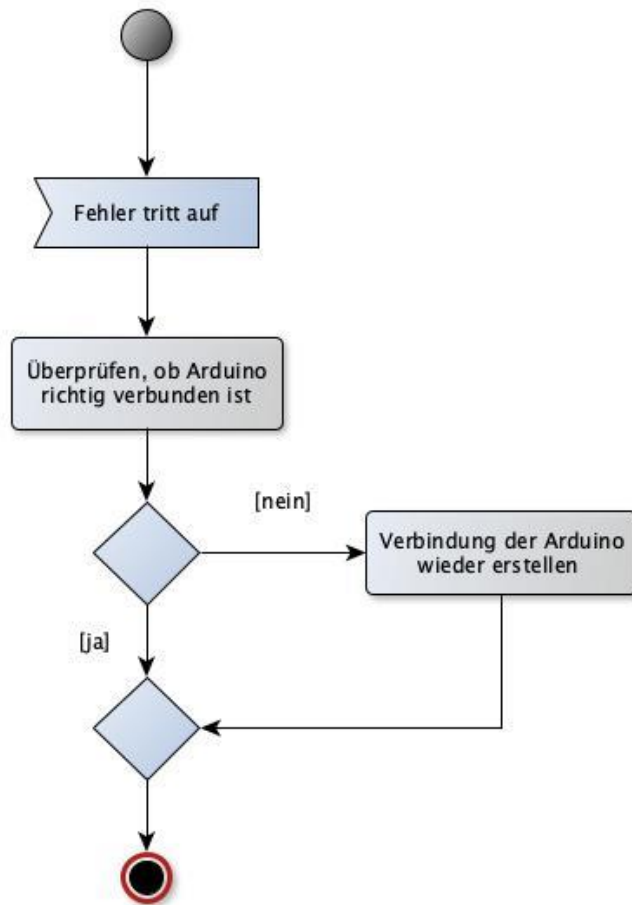


Abbildung 2: Workflow beim Fehlerauftreten

5 Funktionalität

5.1 Überblick

Die folgende Abbildung beschreibt das Use-Case Diagramm der Software.

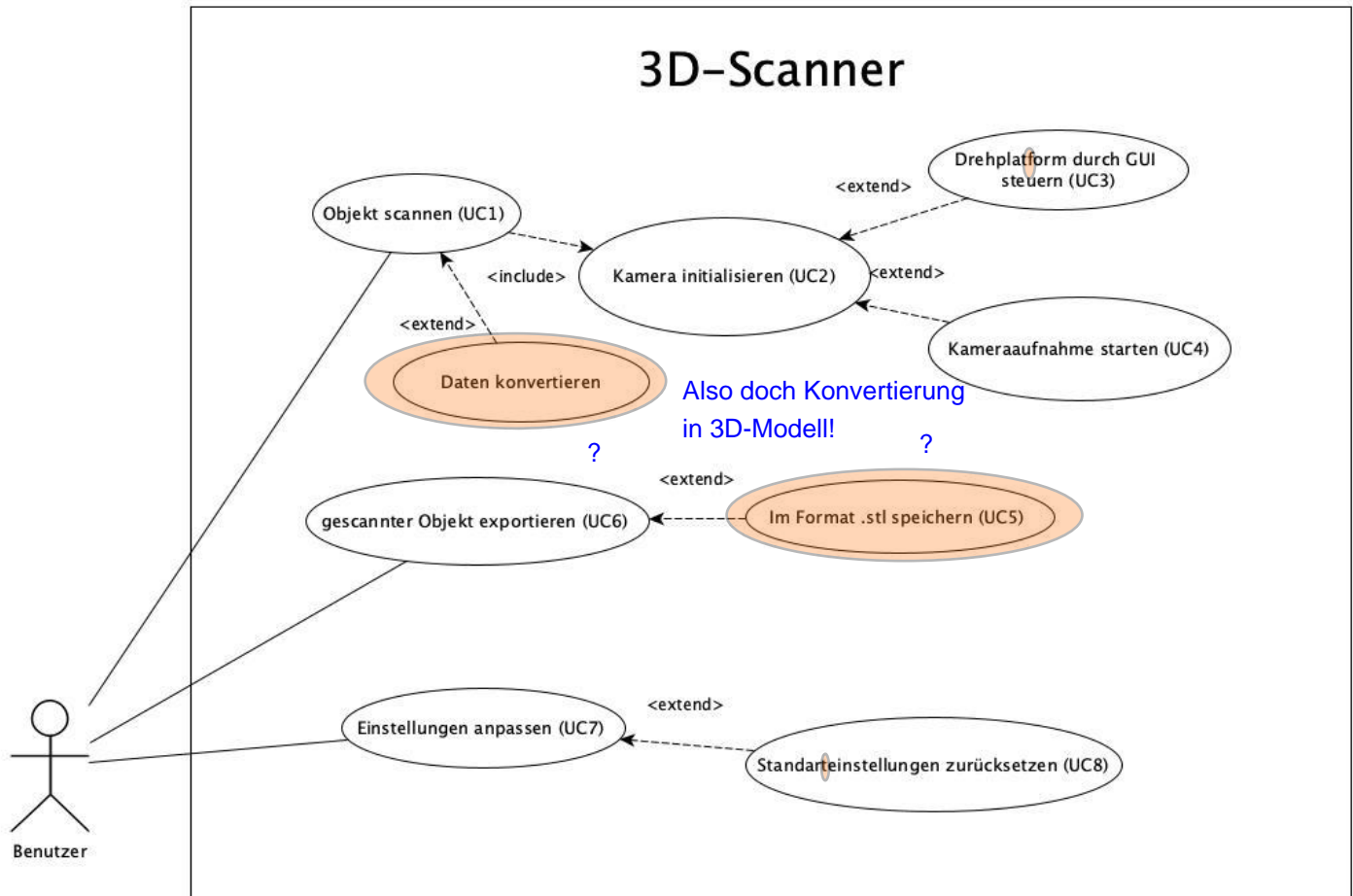


Abbildung 3: Use-Case Diagramm 3D-Scanner

Die folgende Abbildung beinhaltet die GUI der Software.

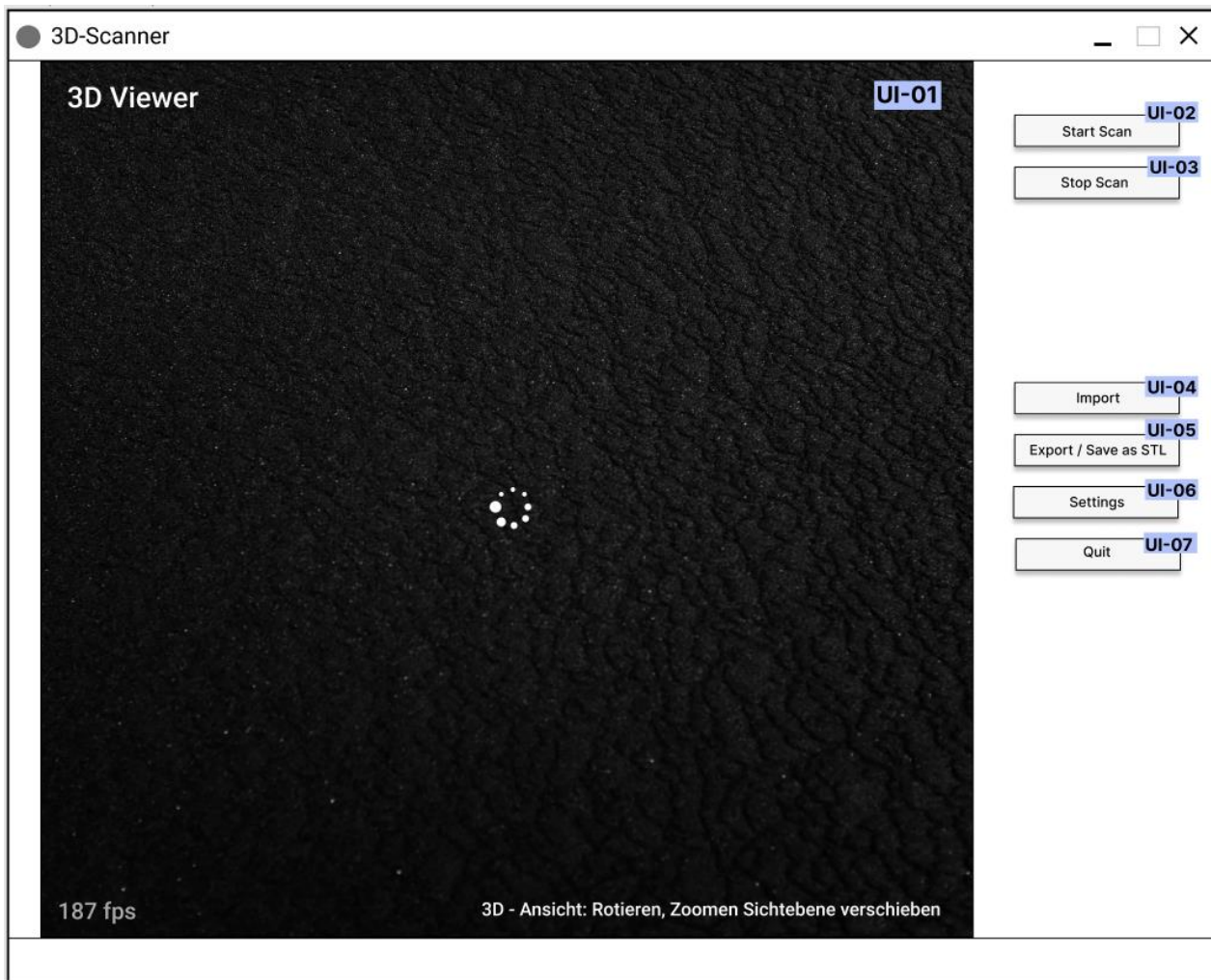


Abbildung 4: GUI Mockup mit UI Beschriftung

Die folgende Tabelle beschreibt die UI-Elemente unserer GUI.

UI-Element	UI-01 3D-Viewer	UI-02 Start Scan	UI-03 Stop Scan	UI-04 Import	UI-05 Export	UI-06 Settings	UI-07 Quit
Erklärung	Visuelle 3D Anzeige um das Objekt anzeigen zu lassen. In dem Viewer kann man das gescannte Objekt drehen, rotieren und zoomen.	Button der den Scan Prozess startet.	Button der den Scan Prozess stoppt.	Button der .stl Dateien zur Visualisierung öffnet	Button der gescannten Objekte exportiert	Button um die Einstellungen zu öffnen.	Button zum Beenden der Software

Tabelle 5: UI-Elemente mit Erklärungen

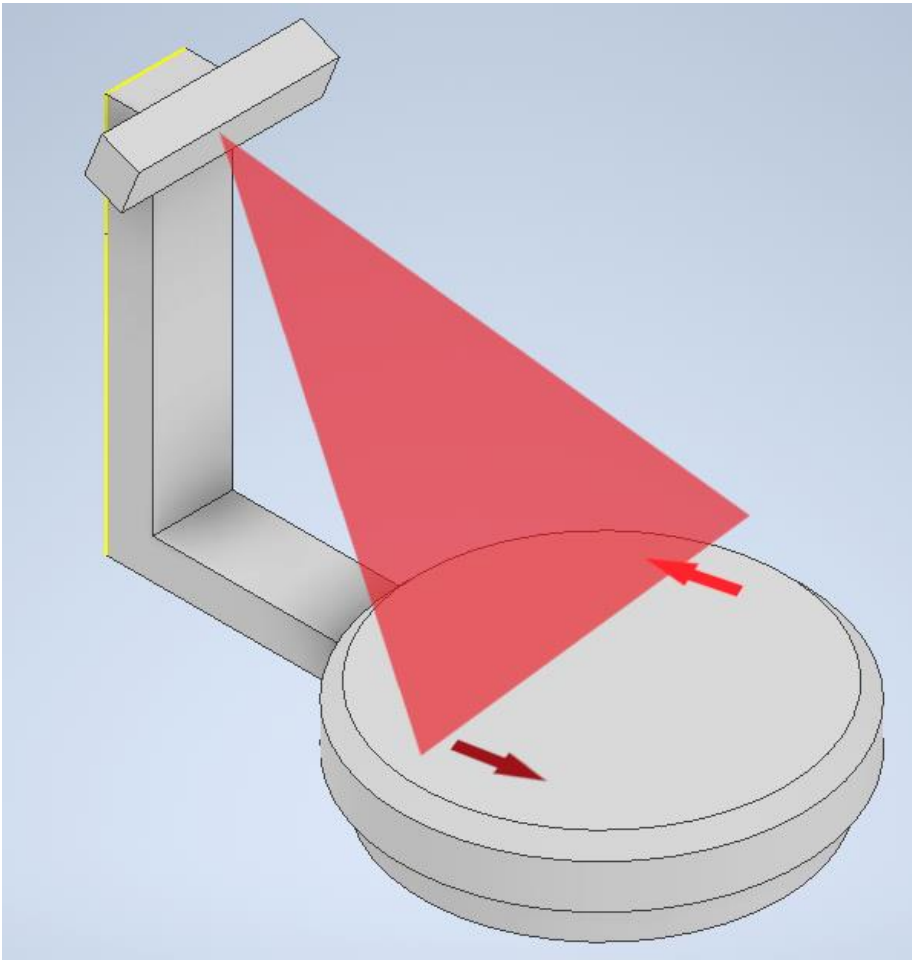


Abbildung 5: 3D-Sicht des Scanners

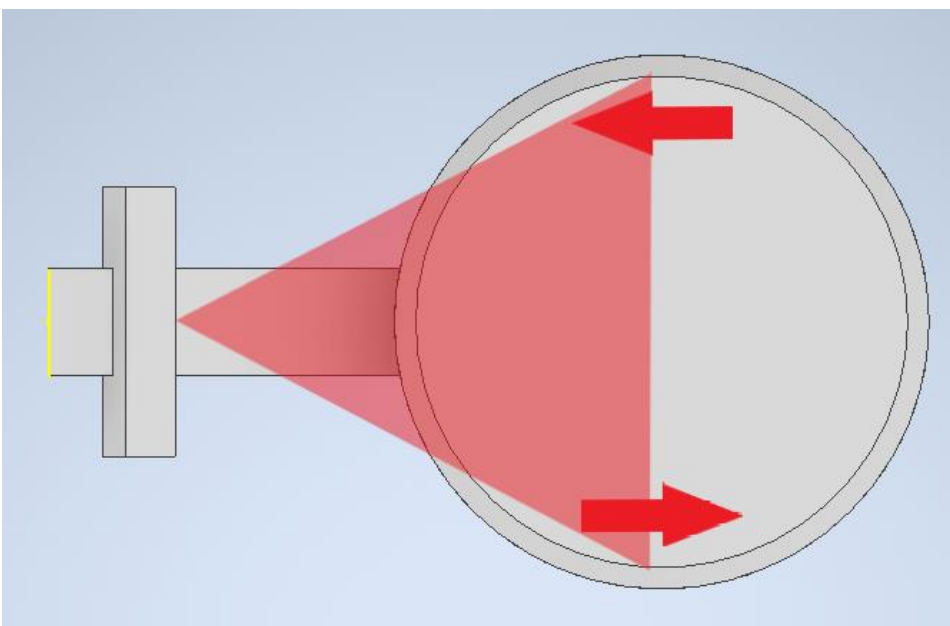


Abbildung 6: Draufsicht des Scanners

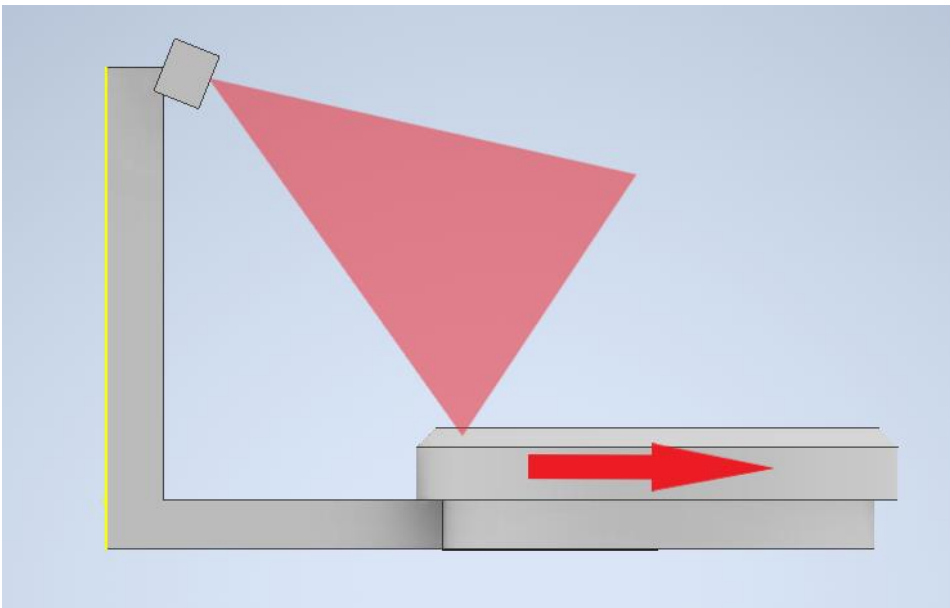


Abbildung 7: Seitenansicht des Scanners

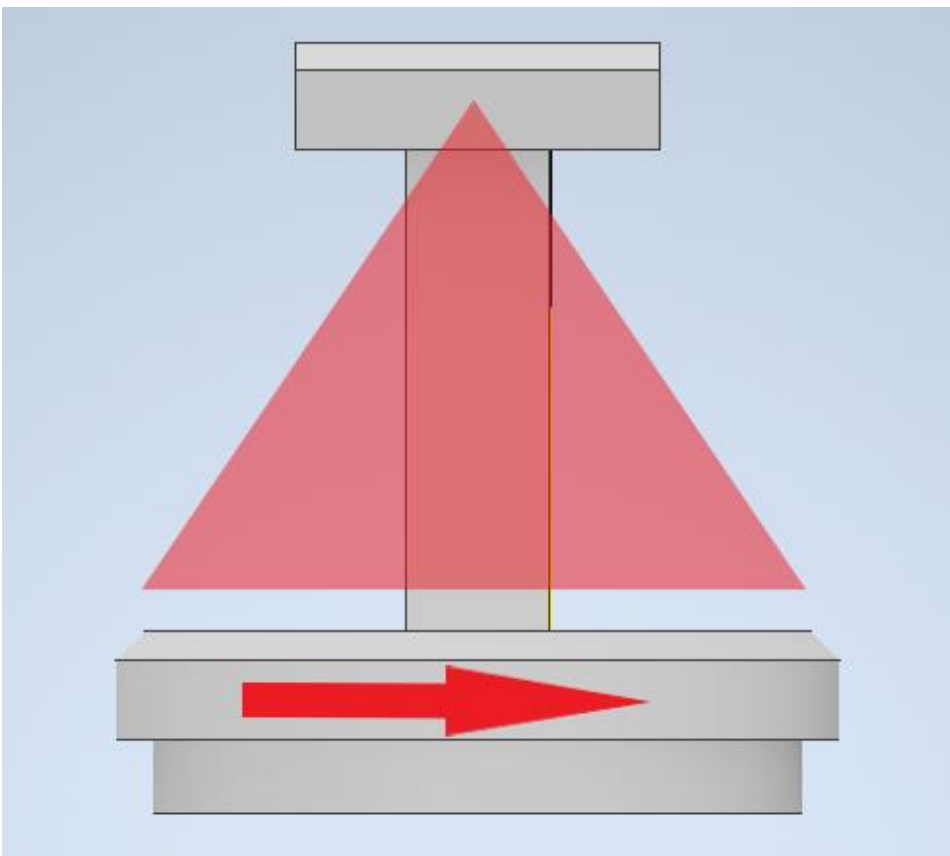


Abbildung 8: Vorderansicht des Scanners

Objekt Scan

5.2 Objekt scannen

Überschrift fehlt

Die folgende Tabelle erläutert die Funktion "Objekt scannen".

Zweck/Ziel	Mit dieser Funktion wird das Objekt welches sich auf der Plattform befindet, gescannt.
Akteur/Auslöser	User und System
Berechtigung	Keine spezifischen Rechte notwendig
WF-Rerferenz	UC-01 WF ??
Vorbedingung	<ul style="list-style-type: none"> • Benutzer hat das System erfolgreich mit allen Komponenten gestartet und befindet sich in der GUI • Kamera ist angeschlossen und an der entsprechenden Position platziert • Objekt befindet sich auf der Plattform
Daten-Input	<ul style="list-style-type: none"> • Keine Daten notwendig
Verarbeitungsschritte	<ol style="list-style-type: none"> 1. Funktion startet mittels eines Buttons auf der GUI 2. Gibt Information weiter, dass der User einen Scann gestartet hat 3. Empfängt die gewonnen Daten
Ergebnis	Das Objekt wurde gescannt, die aufgenommenen Daten sind bereit zur Weiterverarbeitung (z.B. Export)
Plausibilitäten	Auflistung mögliche Plausibilitätsprüfungen, Attribut-Status**
Fehlerhandling	<ul style="list-style-type: none"> • In Der Benutzeroberfläche wird eine Fehlermeldung angezeigt • Aufforderung zum Überprüfen der Verbindungen von Kamera und Arduino
Folgeprozess	Verarbeitung der gewonnen Daten in die GUI
Out of Scope	Diese Funktion dient dazu den Scan zu starten und die gewonnen Daten weiterzuleiten
Anforderung	
Release	
Test Cases	

Tabelle 6: Funktionalität Objekt Scan

5.2 Kamera initialisieren

Die folgende Tabelle erläutert die Funktion "Initialisierung der Kamera".

Zweck/Ziel	Durch diese Funktion werden die Nebenprozesse die für den Scan nötig sind gesteuert
Akteur/Auslöser	System
Berechtigung	Die Software benötigt administrative Rechte um auf die USB-Ports zugreifen zu können
WF-Rererenz	UC-02 WF ??
Vorbedingung	<ul style="list-style-type: none"> • Kamera und Arduino sind angeschlossen • Software ist aktiv • Scan wurde durch User gestartet
Daten-Input	<ul style="list-style-type: none"> • Es sind keine Daten notwendig für diese Funktion • Funktion wird durch vorherige Funktion gestartet (UC-01)
Verarbeitungsschritte	<ol style="list-style-type: none"> 1. Arduino wird Aufgefordert zum drehen der Plattform 2. Intel Realsense Kamera nimmt nötige Daten auf 3. Daten werden zur Weiterverarbeitung weitergeleitet
Ergebnis	Die Kamera und Plattform haben die nötigen Daten aufgenommen und sind bereit für den nächsten Scan
Plausibilitäten	Auflistung mögliche Plausibilitätsprüfungen, Attribut-Status**
Fehlerhandling	<ul style="list-style-type: none"> • In Der Benutzeroberfläche wird eine Fehlermeldung angezeigt • Aufforderung zum Überprüfen der Verbindungen von Kamera und Arduino
Folgeprozess	Die Funktion wird durch die Prozesse bei der Kamera und dem Arduino weitergeführt und endet damit die gewonnen Daten der Hauptfunktion UC-01 zurückzugeben
Out of Scope	Die Daten werden in dieser Funktion nicht verarbeitet, die Funktion dient lediglich zur Steuerung der nötigen Prozesse
Anforderung	
Release	
Test Cases	<ul style="list-style-type: none"> • Ansprechen der Kamera • Ansprechen des Arduinos

Tabelle 7: Funktionalität Kamera initialisieren

5.3 Drehplattform durch GUI steuern

Die folgende Tabelle erläutert die Funktion "Rotation der Plattform".

Zweck/Ziel	Die Funktion ist dazu da die Plattform sich rotieren zu lassen
Akteur/Auslöser	System
Berechtigung	Keine Rechte notwendig
WF-Rererenz	UC-03
Vorbedingung	<ul style="list-style-type: none">• Arduino ist am Computer angeschlossen
Daten-Input	<ul style="list-style-type: none">• Es sind keine Daten notwendig
Verarbeitungs-schritte	<ol style="list-style-type: none">1. Stepper Motor dreht sich2. Durch Stepper Motor dreht sich die Plattform
Ergebnis	Durch die Drehung (Rotation) der Plattform gelingt es der Kamera aus jeder Perspektive zum Bild Daten aufzunehmen
Plausibilitäten	Auflistung mögliche Plausibilitätsprüfungen, Attribut-Status
Fehlerhandling	<ul style="list-style-type: none">• In Der Benutzeroberfläche wird eine Fehlermeldung angezeigt• Aufforderung zum Überprüfen der Verbindungen von Kamera und Arduino
Folgeprozess	Während die Plattform rotiert werden Daten aufgenommen, diese Daten werden zur Weiterverarbeitung weitergeleitet durch UC-01
Out of Scope	Die Funktion dient nur zur Rotation der Kamera
Anforderung	
Release	
Test Cases	Rotation der Plattform außerhalb der Software

Tabelle 8: Funktionalität Drehplattform durch GUI steuern

5.4 Kameraaufnahme starten

Die folgende Tabelle erläutert die Funktion "Aufnahme durch die Kamera".

Zweck/Ziel	In dieser Funktion werden die Daten durch die Kamera aufgenommen
Akteur/Auslöser	System
Berechtigung	Es sind keine Rechte notwendig
WF-Rerferenz	UC-04
Vorbedingung	<ul style="list-style-type: none">• Kamera ist angeschlossen
Daten-Input	<ul style="list-style-type: none">• Es sind keine Daten notwendig
Verarbeitungs-schritte	<ol style="list-style-type: none">1. Kamera schaltet in den Tiefenbilder Modus2. Kamera schießt Bilder in bestimmten Zeitabständen
Ergebnis	Durch die geschossenen Bilder ist ein Datensatz entstanden
Plausibilitäten	
Fehlerhandling	<ul style="list-style-type: none">• In Der Benutzeroberfläche wird eine Fehlermeldung angezeigt• Aufforderung zum Überprüfen der Verbindungen von Kamera und Arduino
Folgeprozess	Der aufgenommene Datensatz wird weitergeleitet an UC-01
Out of Scope	Diese Funktion ist lediglich dazu verantwortet die Bilder aufzunehmen
Anforderung	
Release	
Test Cases	Die Kamera wird vorher auf Funktionalität getestet

Tabelle 9: Funktionalität Kameraaufnahme starten

5.5 Konvertierung der Daten

Begriff zutreffend?

Die folgende Tabelle erläutert die Funktion "Konvertierung der Daten".

Zweck/Ziel	In dieser Funktion werden die Daten Verarbeitet
Akteur/Auslöser	System
Berechtigung	Es sind keine Rechte notwendig
WF-Rerferenz	UC-09
Vorbedingung	<ul style="list-style-type: none"> Die Plattform hat eine 360° Drehung hinter sich Die Kamera hat Tiefenbilder geschossen Der Datensatz ist vollständig gefüllt
Daten-Input	<ul style="list-style-type: none"> Mehrere Tiefenbilder vom Objekt
Verarbeitungs-schritte	<ol style="list-style-type: none"> Datensatz wird verarbeitet Objekt wird in der GUI visualisiert
Ergebnis	Das Objekt ist digitalisiert und wird visuell in der GUI angezeigt
Plausibilitäten	
Fehlerhandling	<ul style="list-style-type: none"> In Der Benutzeroberfläche wird eine Fehlermeldung angezeigt Aufforderung zum Wiederholen des Scans Gegebenenfalls genaue Beschreibung des Fehlers
Folgeprozess	Der Object Scan ist somit beendet, der Nutzer kann nun die nächsten Prozesse durch die GUI einleiten
Out of Scope	Diese Funktion nimmt ist nicht für die Aufnahme der Daten verantwortlich, sie kriegt die Daten und verarbeitet sie
Anforderung	
Release	
Test Cases	

Tabelle 10: Funktionalität Konvertierung der Daten

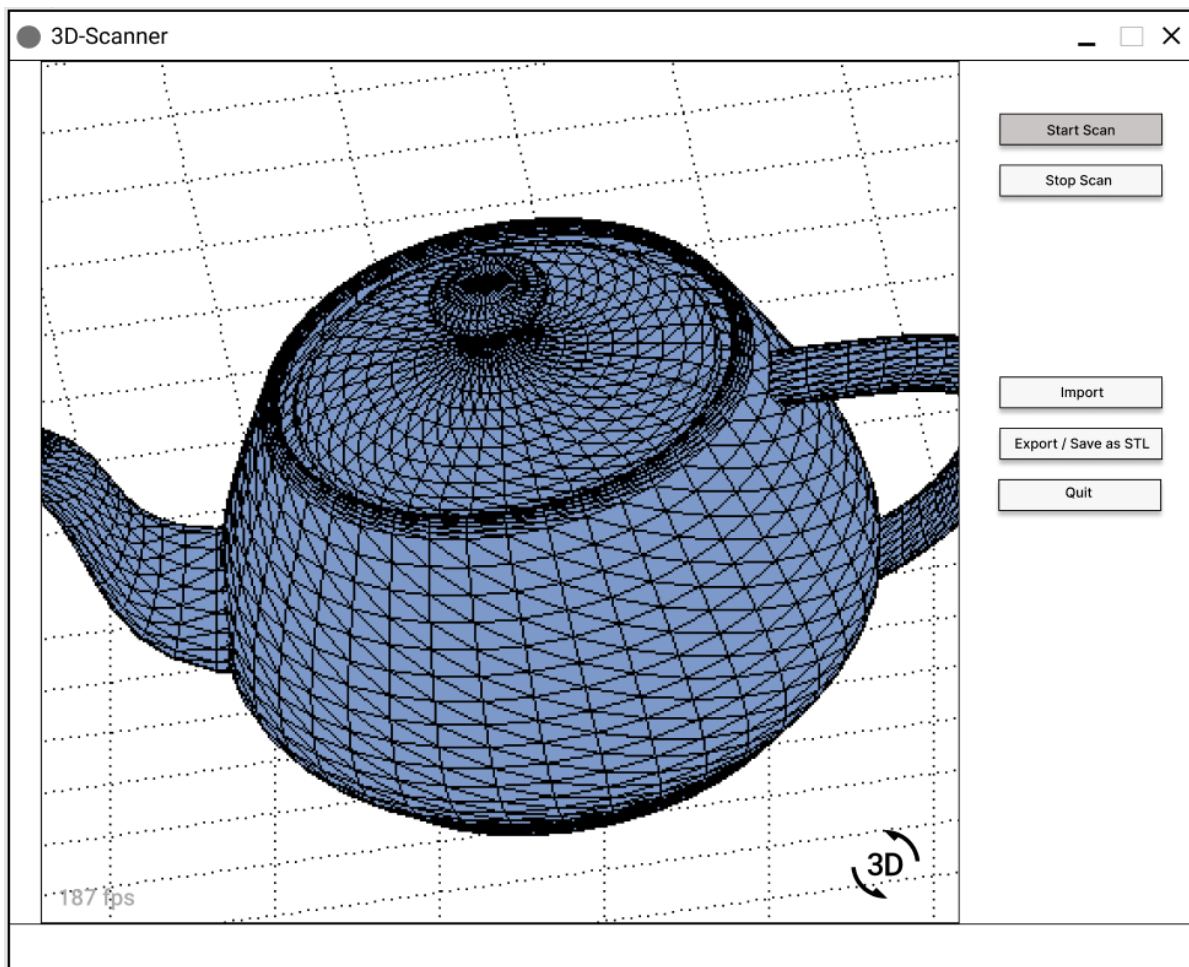


Abbildung 9: GUI Mockup – Nach Kameraaufnahme + Konvertierung der Daten

5.6 Gescanntes Objekt Exportieren

Die folgende Tabelle erläutert die Funktion "Exportierung des gescannten Objekts".

Zweck/Ziel	In dieser Funktion kann der User das gescannte Objekt exportieren, um es für jegliches zu Benutzen
Akteur/Auslöser	User, System
Berechtigung	Es sind Festplatten Rechte nötig, um die Datei abzuspeichern
WF-Rerferenz	UC-06
Vorbedingung	<ul style="list-style-type: none"> Erfolgreicher Scan ohne Fehler Gescanntes Objekt ist in GUI visualisiert
Daten-Input	<ul style="list-style-type: none"> Output vom Objekt Scan (UC-01) wird hier weiterverwendet
Verarbeitungs-schritte	<ol style="list-style-type: none"> Daten werden in eine Datei konvertiert abgelegt/abgespeicher? User wählt Ort aus um die Datei abzuspeichern Datei wird abgespeichert
Ergebnis	Das gescannte Objekt ist in verschiedene andere Software weiterverwendbar
Plausibilitäten	
Fehlerhandling	<ul style="list-style-type: none"> In Der Benutzeroberfläche wird eine Fehlermeldung angezeigt Aufforderung zur Überprüfung des Speichers
Folgeprozess	Der User hat erfolgreich die Hauptfunktion der Software beendet, und könnte nun mit einem neuen Objekt fortfahren
Out of Scope	
Anforderung	
Release	
Test Cases	Es werden mehrere vorgegebene Objekte abgespeichert

Tabelle 11: Funktionalität Gescanntes Objekt exportieren

5.7 Im Format .stl speichern

Die folgende Tabelle erläutert die Funktion "Konvertierung der Daten in das .stl Dateiformat".

Zweck/Ziel	In dieser Funktion wird die zu abspeichernde Datei in eine .stl Datei konvertiert
Akteur/Auslöser	System
Berechtigung	Es sind keine Rechte notwendig
WF-Rerferenz	UC-05
Vorbedingung	<ul style="list-style-type: none">• Es besteht eine Datei
Daten-Input	<ul style="list-style-type: none">• Output vom Objekt Scan (UC-01) wird hier weiterverwendet
Verarbeitungs-schritte	1. Datei wird in .stl konvertiert
Ergebnis	Es besteht eine .stl Datei, die das gescannte Objekt beinhaltet
Plausibilitäten	
Fehlerhandling	<ul style="list-style-type: none">• In Der Benutzeroberfläche wird eine Fehlermeldung angezeigt• Aufforderung zur Überprüfung des Speichers
Folgeprozess	Der User hat erfolgreich die Hauptfunktion der Software beendet, und könnte nun mit einem neuen Objekt fortfahren
Out of Scope	
Anforderung	
Release	
Test Cases	Die Kamera wird vorher auf Funktionalität getestet

Tabelle 12: Im Format .stl speichern

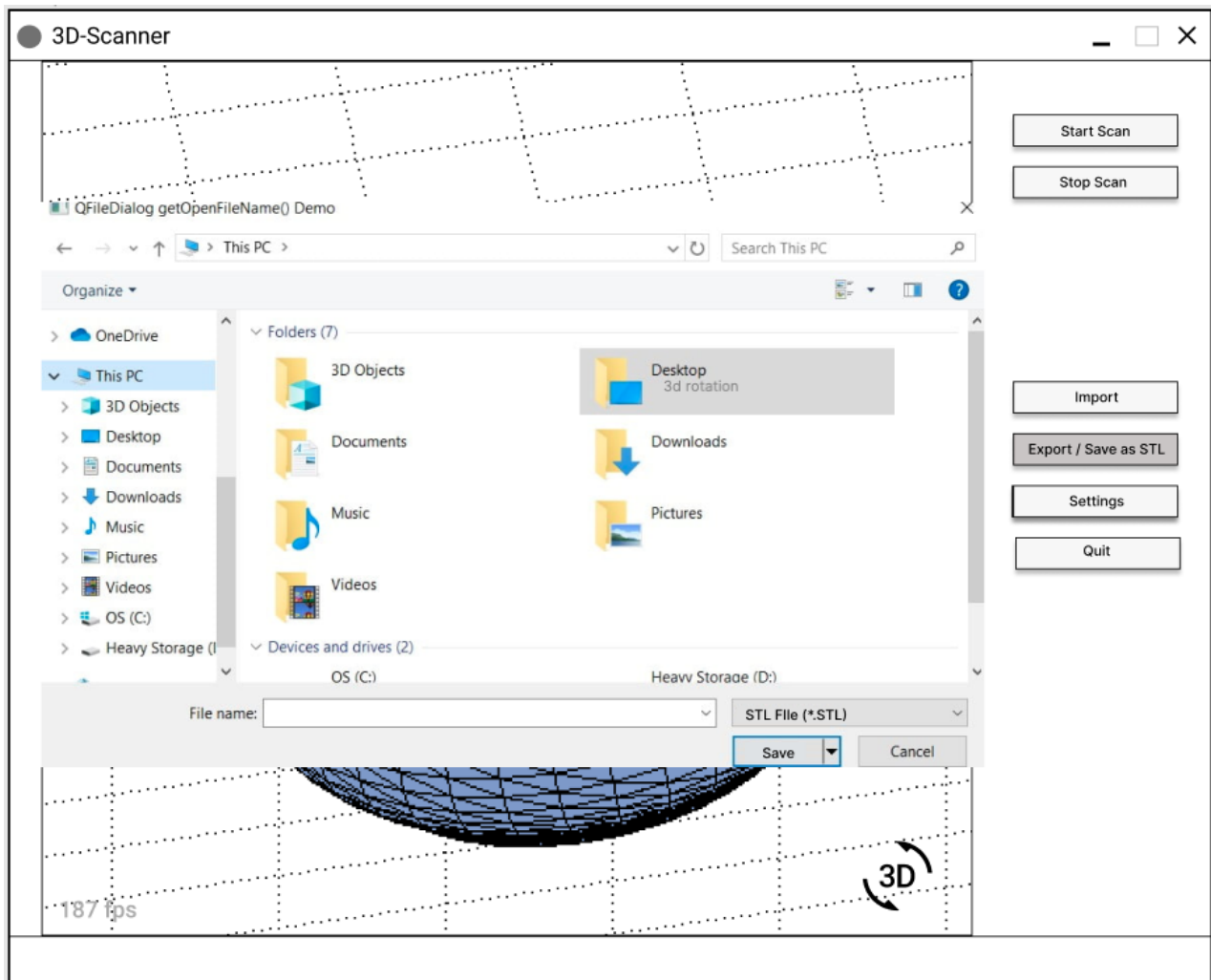


Abbildung 10: GUI Mockup – Export Button Ansicht

5.8 Einstellungen anpassen

Die folgende Tabelle erläutert die Funktion "Einstellungen".

Zweck/Ziel	In dieser Funktionen können interne Einstellungen angepasst werden
Akteur/Auslöser	User, System
Berechtigung	Es sind keine Rechte notwendig
WF-Rererenz	UC-07
Vorbedingung	<ul style="list-style-type: none"> Die Einstellungen sind nur über die GUI erreichbar
Daten-Input	<ul style="list-style-type: none"> Es sind keine Daten notwendig
Verarbeitungs-schritte	<ol style="list-style-type: none"> GUI schaltet in Einstellungsfenster Verschiede Dinge können angepasst werden(z.B Port für Arduino)
Ergebnis	Software benutzt nun richtigen Port für Arduino
Plausibilitäten	Darkmode
Fehlerhandling	<ul style="list-style-type: none"> In Der Benutzeroberfläche wird eine Fehlermeldung angezeigt Aufforderung zum Starten der Software mit Admin Rechten
Folgeprozess	Der User kann neuen Scan starten
Out of Scope	Durch die Einstellungen kann man nicht erhebliche Änderungen an der Software ausführen, wie Anpassung der Toleranzen, etc.
Anforderung	
Release	
Test Cases	

Tabelle 13: Funktionalität Einstellung anpassen

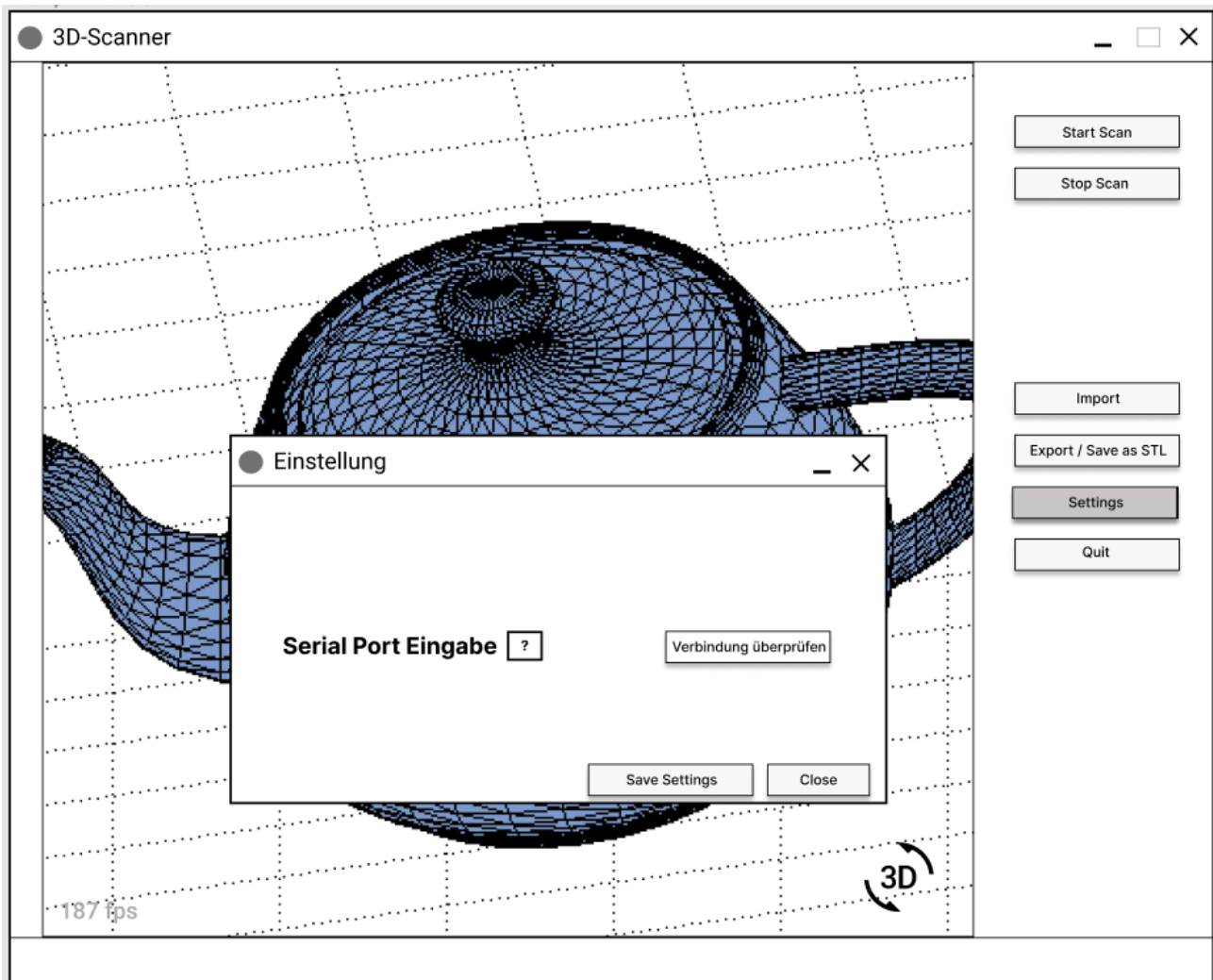


Abbildung 11: GUI Mockup – Einstellung Button Ansicht

5.9 Standarteinstellungen zurücksetzen

Die folgende Tabelle erläutert die Funktion "Standarteinstellungen zurücksetzen".

Zweck/Ziel	In dieser Funktionen werden die Einstellungen auf Werkseinstellungen zurückgesetzt
Akteur/Auslöser	User, System
Berechtigung	Es sind keine Rechte notwendig
WF-Rererenz	UC-08
Vorbedingung	<ul style="list-style-type: none"> Die Einstellungen sind nur über die GUI erreichbar
Daten-Input	<ul style="list-style-type: none"> Die Werkseinstellungen sind innerhalb der Software gespeichert
Verarbeitungs-schritte	<ol style="list-style-type: none"> Die Werkseinstellungen werden übernommen Die aktuellen Einstellungen werde mit den Werkseinstellungen überschrieben
Ergebnis	Die Software ist auf dem Stand wie sie runtergeladen wurde
Plausibilitäten	
Fehlerhandling	<ul style="list-style-type: none"> In Der Benutzeroberfläche wird eine Fehlermeldung angezeigt Aufforderung zum Starten der Software mit Admin Rechten
Folgeprozess	Der User kann neuen Scan starten
Out of Scope	
Anforderung	
Release	
Test Cases	

Tabelle 14: Standardeinstellung zurücksetzen

Modulabhängigkeiten

Modul	Beschreibung
Programmierung	Die Software wird programmiert
Konstruktion	Es wird eine Plattform konstruiert um das Objekt draufzustellen
Mechatronik	Es wird ein Mikroprozessor zur Steuerung des Stepper Motors verwendet

Tabelle 15: Modulabhängigkeiten

6 Wer hat was gemacht

Autor	Aufgabe/Kapitel	Anteil
Vinh Thong Trinh	Alles	25%
Habib Ben-Khedher	Alles	25%
Mert Karadeniz	Alles	25%
<i>William Eppel</i>	<i>Alles</i>	25%

Tabelle 16: Wer hat was gemacht