

Pflichtenheft und technische Spezifikation im Programmierprojekt

## Datenprüfung STL

...

<i>Gruppenteilnehmer</i>		
<i>Name</i>	<i>Matr.-Nr.</i>	<i>E-Mail</i>
Michael Kaip	567264	michael.kaip@student.htw-berlin.de
Adib Ghassani Waluya	567271	adib.waluya@student.htw-berlin.de
Michael Reno	565907	s0565907@htw-berlin.de
Maximilian Mews	564319	maximilian.mews@student.htw-berlin.de

**Hochschule fuer Technik und Wirtschaft Berlin**  
FB 2 - Ingenieurwissenschaften Technik und Leben  
Studiengang: Ingenieurinformatik

**Betreut durch:**  
Prof. Dr. Jörg Schlingheider  
Tel.: +49 30 5019-4354  
joerg.schlingheider@htw-berlin.de

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>I</b>
<b>Tabellenverzeichnis</b>	<b>II</b>
<b>1 Vision und Projektziele</b>	<b>1</b>
<b>2 Systemanforderungen</b>	<b>2</b>
2.1 Anwendungsfälle . . . . .	2
2.1.1 Beschreibung der Anwendungsfälle . . . . .	3
2.2 Aktivitätsdiagramme . . . . .	10
2.2.1 Aktivitätsdiagramm Datenmodell erzeugen . . . . .	10
2.2.2 Aktivitätsdiagramm Fehlersuche . . . . .	10
2.2.3 Aktivitätsdiagramm ViewModel (OpenGL) . . . . .	11
2.2.4 Aktivitätsdiagramm STL-Export . . . . .	11
2.3 Grafische Benutzeroberfläche . . . . .	12
<b>3 Überblick - Komponentendiagramm</b>	<b>14</b>
<b>4 Realisierung</b>	<b>15</b>
4.1 Allgemeines . . . . .	15
4.2 Interne Schnittstellen . . . . .	15
4.3 Externe Schnittstellen . . . . .	15
<b>5 Projektkalkulation</b>	<b>16</b>
<b>6 Abnahmekriterien</b>	<b>16</b>
<b>7 Lizenz</b>	<b>17</b>

# Abbildungsverzeichnis

1	Mögliche Fehler im STL-Dateiformat (Friedrich, 2012) . . . . .	1
2	Anwendungsfall-Diagramm . . . . .	2
3	Aktivitätsdiagramm (Datenmodell erzeugen) . . . . .	10
4	Aktivitätsdiagramm (Fehlersuche) . . . . .	10
5	Aktivitätsdiagramm (View Model - Datenmodell anzeigen) . . . . .	11
6	Aktivitätsdiagramm (STL-Export) . . . . .	11
7	Zeichenerklärungen (GUI) . . . . .	12
8	Startseite (GUI) . . . . .	12
9	Toleranzeingabe bei Fehlersuche (GUI) . . . . .	13
10	Grundeinstellungen Allgemein (GUI) . . . . .	13
11	Grundeinstellungen Fehlersuche (GUI) . . . . .	14
12	Komponenten-Diagramm . . . . .	14
13	Interne Schnittstellen des Systems . . . . .	15
14	Projektkalkulation . . . . .	16

## Tabellenverzeichnis

1	Use-Case 01 – STL-Datei öffnen . . . . .	3
2	Use-Case 02 – Drehen, Zoomen, Verschieben . . . . .	3
3	Use-Case 03 – STL-Datei speichern . . . . .	4
4	Use-Case 04 – Grundeinstellungen vornehmen . . . . .	4
5	Use-Case 05 – Geometrieelemente auswählen . . . . .	5
6	Use-Case 06 – Geometrie analysieren / Fehler finden . . . . .	5
7	Use-Case 07 – Fehlertoleranz / Anzeigeoptionen definieren . . . . .	6
8	Use-Case 08 – Messen . . . . .	6
9	Use-Case 09 – Skalieren . . . . .	7
10	Use-Case 10 – Modell anzeigen . . . . .	7
11	Use-Case 11 – Datenmodell erzeugen / aktualisieren . . . . .	8
12	Use-Case 12 – Geometrie heilen . . . . .	9
13	Use-Case 13 – STL-Datei erzeugen . . . . .	9
14	Aufgabenverteilung . . . . .	15

# 1 Vision und Projektziele

Es soll eine Software entwickelt werden mit der es möglich ist, in STL-Dateien kodierte 3D-Volumenmodelle, die als Grundlage für den 3D-Druck dienen, zu verarbeiten.

Dabei soll es möglich sein, eine solche Datei zu öffnen und das darin enthaltene 3D-Modell auf der Benutzeroberfläche anzuzeigen. Darüber hinaus soll der Nutzer auch in der Lage sein, die dargestellte Geometrie auf Fehler zu untersuchen, diese zu korrigieren und das entsprechend korrigierte Modell dann wieder im STL-Format abzuspeichern.

Die Arten von Fehlern, die bei der Kodierung von 3D-Modellen in ein STL-Format entstehen können und somit bei der Geometrieanalyse erkannt werden sollen, sind die folgenden:

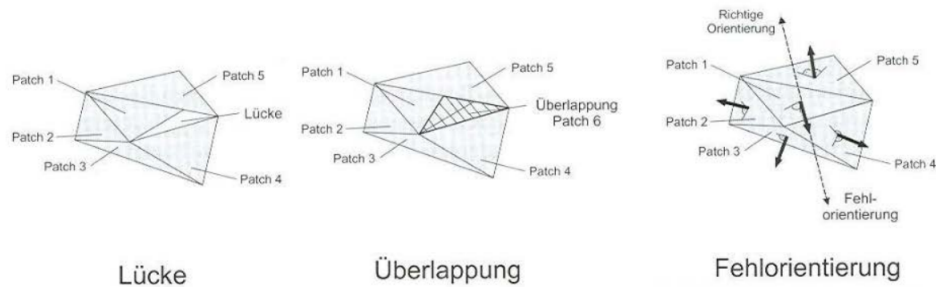


Abbildung 1: Mögliche Fehler im STL-Dateiformat (Friedrich, 2012)

Eine weitere wesentliche Funktionalität der Software soll es dem Nutzer ermöglichen, die reale Größe der dargestellten Bauteile zu messen, um die Größenverhältnisse der Geometrie zu überprüfen. Werden hierbei Fehler hinsichtlich der Größe erkannt soll es auch hier die Möglichkeit geben, diese durch entsprechende Skalierung der Bauteile zu verändern bzw. neu zu berechnen.

Bezüglich der allgemeinen Performance des Systems soll gelten, dass das Laden und Prüfen einer Datei mit einer Größe von etwa 3 MB nicht länger als 20 Sekunden dauert.

## 2 Systemanforderungen

### 2.1 Anwendungsfälle

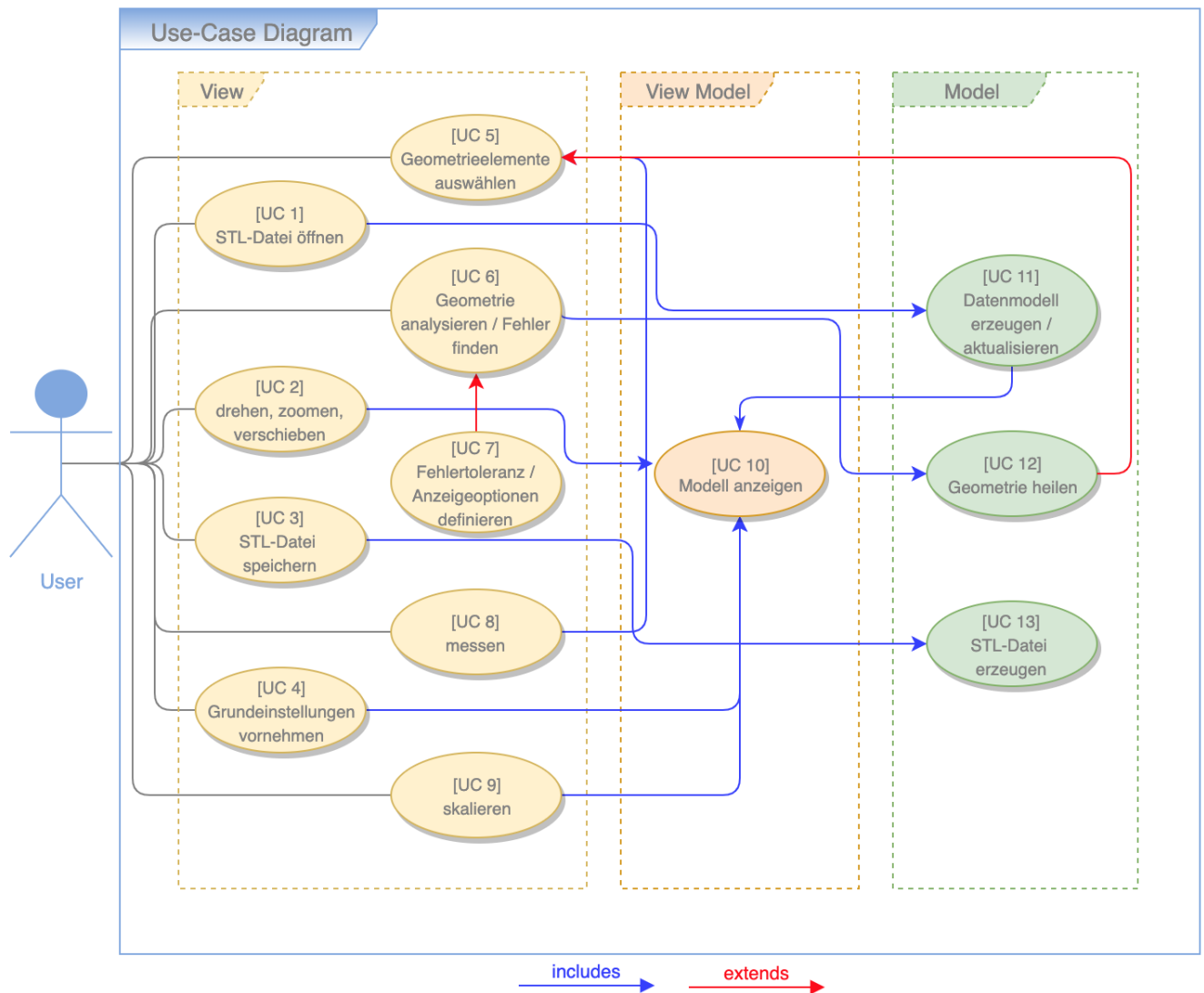


Abbildung 2: Anwendungsfall-Diagramm

### 2.1.1 Beschreibung der Anwendungsfälle

- [UC 1] STL-Datei öffnen

<i>Informationsart</i>	<i>Beschreibung</i>
<i>Identifikation</i>	UC 1
<i>Beschreibung</i>	STL-Datei öffnen
<i>Akteure</i>	User, System
<i>Vorbedingungen</i>	Gültige STL-Datei wurde vom User ausgewählt.
<i>Standardablauf</i>	1) Funktion wird über die GUI aufgerufen. 2) User wählt eine STL-Datei ueber einen angebotenen Dialog. 3) Pfad zur Datei wird übergeben. 4) Übergabe-String und Datei werden auf Gültigkeit hin überprüft. 5) Prüfung, in welchem Format die STL-Datei vorliegt (ASCII oder Binär). 6) STL-Datei wird vollständig eingelesen und temporär gespeichert.
<i>Alternative Abläufe</i>	Fehlermeldung bei nicht vorliegen einer gültigen STL-Datei.
<i>Nachbedingungen</i>	STL-Datei im vorliegenden Format wurde vollständig importiert.
<i>Bemerkung</i>	Use-Case ist Voraussetzung für Datenmodell erzeugen [UC 11].

Tabelle 1: Use-Case 01 – STL-Datei öffnen

- [UC 2] Drehen, Zoomen, Verschieben

<i>Informationsart</i>	<i>Beschreibung</i>
<i>Identifikation</i>	UC 2
<i>Beschreibung</i>	Drehen, Zoomen, Verschieben
<i>Akteure</i>	User, System
<i>Vorbedingungen</i>	3D-Modell vollständig geladen und auf der Benutzeroberfläche angezeigt.
<i>Standardablauf</i>	1) Nutzer greift das dargestellte Objekte mit der Maus. 2) System berechnet die Veränderung der Darstellung. 3) Objekt dreht sich entsprechend der vorgegebenen Bewegungsrichtung.
<i>Alternative Abläufe</i>	
<i>Nachbedingungen</i>	Das Modell wird in der vom Nutzer gewählten Perspektive angezeigt.
<i>Bemerkung</i>	

Tabelle 2: Use-Case 02 – Drehen, Zoomen, Verschieben

- [UC 3] STL-Datei speichern

<i>Informationsart</i>	<i>Beschreibung</i>
<i>Identifikation</i>	UC 3
<i>Beschreibung</i>	STL-Datei speichern
<i>Akteure</i>	User, System
<i>Vorbedingungen</i>	3D-Modell vollständig geladen - entsprechendes Datenmodell (in Boundary Representation) liegt vor.
<i>Standardablauf</i>	1) Nutzer wählt die Funktion über die GUI aus. 2) Nutzer benennt die Datei und wählt den gewünschten Speicherort und Datentyp. 3) System erzeugt aus dem Datenmodell eine valide STL-Datei und speichert diese am gewünschten Ort.
<i>Alternative Abläufe</i>	Sollte kein Datenmodell vorliegen, wird eine Fehlermeldung angezeigt.
<i>Nachbedingungen</i>	STL-Datei wurde erzeugt und am gewünschten Ort gespeichert.
<i>Bemerkung</i>	

Tabelle 3: Use-Case 03 – STL-Datei speichern

- [UC 4] Grundeinstellungen vornehmen

<i>Informationsart</i>	<i>Beschreibung</i>
<i>Identifikation</i>	UC 4
<i>Beschreibung</i>	Grundeinstellungen vornehmen
<i>Akteure</i>	User
<i>Vorbedingungen</i>	
<i>Standardablauf</i>	1) User ruft Einstellungen auf. 2) Er wählt die Rubrik Grundeinstellungen. 3) User gibt die gewünschte Fehlertoleranz sowie die Anzeigefarbe der Fehler sowie des Datenmodells an. 4) User speichert seine gewählten Einstellungen. 5) Falls bereits ein geladenes Datenmodell vorliegt, werden diese Einstellungen direkt übernommen und das 3D-Modell entsprechend angezeigt.
<i>Alternative Abläufe</i>	
<i>Nachbedingungen</i>	3D-Modelle werden entsprechend den Einstellungen angezeigt.
<i>Bemerkung</i>	

Tabelle 4: Use-Case 04 – Grundeinstellungen vornehmen



- [UC 5] Geometrieelemente auswählen

<i>Informationsart</i>	<i>Beschreibung</i>
<i>Identifikation</i>	UC 5
<i>Beschreibung</i>	Geometrieelemente auswählen
<i>Akteure</i>	User, System
<i>Vorbedingungen</i>	3D-Modell vollständig geladen und auf der Benutzeroberfläche angezeigt.
<i>Standardablauf</i>	1) Geometrieelemente wie Kanten, Punkte und Flächen können mit der Maus gefangen und durch Mausklick ausgewählt werden. 2) Möglichkeit Geometrieelemente farblich zu verändern oder Abstände zu messen
<i>Alternative Abläufe</i>	
<i>Nachbedingungen</i>	Das Modell wird entsprechend der Veränderungen angezeigt.
<i>Bemerkung</i>	Use-Case ist Voraussetzung für Messen [UC 8] und Heilen [UC 12].

Tabelle 5: Use-Case 05 – Geometrieelemente auswählen

- [UC 6] Geometrie analysieren / Fehler finden

<i>Informationsart</i>	<i>Beschreibung</i>
<i>Identifikation</i>	UC 6
<i>Beschreibung</i>	Geometrie analysieren / Fehler finden
<i>Akteure</i>	User, System
<i>Vorbedingungen</i>	3D-Modell vollständig geladen und auf der Benutzeroberfläche angezeigt.
<i>Standardablauf</i>	1) User ruft die Funktion auf. 2) Funktion ruft eingestellt Fehlertoleranz ab. 3) Datenmodell wird aktualisiert.
<i>Alternative Abläufe</i>	1) User ruft die Funktion auf. 2) User stellt Fehlertoleranz ein. 3) Datenmodell wird aktualisiert.
<i>Nachbedingungen</i>	Sollte ein Fehler vorliegen, so werden die entsprechenden Punkte und Kanten eingefärbt und angezeigt.
<i>Bemerkung</i>	Use-Case ist Voraussetzung für Heilen [UC 12].

Tabelle 6: Use-Case 06 – Geometrie analysieren / Fehler finden

- [UC 7] Fehlertoleranz/Anzeigeoptionen definieren

<i>Informationsart</i>	<i>Beschreibung</i>
<i>Identifikation</i>	UC 7
<i>Beschreibung</i>	Fehlertoleranz / Anzeigeoptionen definieren
<i>Akteure</i>	User, System
<i>Vorbedingungen</i>	
<i>Standardablauf</i>	1) User ruft die Funktion auf. 2) Mögliche Aktionen: - User veraendert die Fehlertoleranz - User wählt ein Objekt und ändert die Farbe - User verändert die Hintergrundfarbe
<i>Alternative Abläufe</i>	
<i>Nachbedingungen</i>	Moegliche Ergebnisse: - Neue Fehlertoleranz wurde eingestellt - Ein Objekt wird in einer anderen Farbe dargestellt - Hintergrund wird in neuer Farbe dargestellt
<i>Bemerkung</i>	Use-Case ist Voraussetzung fuer Geometrie analysieren [UC 6].

Tabelle 7: Use-Case 07 – Fehlertoleranz / Anzeigeoptionen definieren

- [UC 8] Messen

<i>Informationsart</i>	<i>Beschreibung</i>
<i>Identifikation</i>	UC 8
<i>Beschreibung</i>	Messen
<i>Akteure</i>	User, System
<i>Vorbedingungen</i>	3D-Modell vollständig geladen und auf der Benutzeroberfläche angezeigt.
<i>Standardablauf</i>	1) Nutzer wählt mit durch Mausklick einen beliebigen Punkt auf der Oberfläche des Modells aus. 2) Punkt wird gefangen und fixiert. 3) Bei Bewegung des Mauszeigers wird der fixierte Punkt kontinuierlich mit einer sichtbaren Linie verbunden. 4) Entfernung zur aktuellen Position wird vom System berechnet und der Wert über dem Mauszeiger angezeigt und ständig aktualisiert.
<i>Alternative Abläufe</i>	
<i>Nachbedingungen</i>	Geladenes 3D-Modell bleibt unverändert.
<i>Bemerkung</i>	Use-Case ist Voraussetzung für Skalieren [UC 9]

Tabelle 8: Use-Case 08 – Messen

- [UC 9] Skalieren

<i>Informationsart</i>	<i>Beschreibung</i>
<i>Identifikation</i>	UC 9
<i>Beschreibung</i>	Skalieren
<i>Akteure</i>	User, System
<i>Vorbedingungen</i>	3D-Modell vollständig geladen und auf der Benutzeroberfläche angezeigt. Bauteil wurde bereits gemessen und Messergebnisse liegen vor.
<i>Standardablauf</i>	1) User wählt das Bauteil und ruft durch Rechtsklick die Funktion auf. 2) Ein Fenster öffnet sich, in dem die aktuellen Abmessungen des Modells angezeigt werden. 3) User macht Eingaben zu den gewünschten Abmessungen. 4) 3D-Modell wird entsprechend skaliert.
<i>Alternative Abläufe</i>	Wenn das Bauteil noch nicht gemessen wurde, bekommt der User im Fenster einen entsprechenden Hinweis und die Option, die Funktion "Messen" von dort direkt aufzurufen.
<i>Nachbedingungen</i>	3D-Modell wurde auf die gewünschte Größe skaliert und das Datenmodell entsprechend verändert.
<i>Bemerkung</i>	

Tabelle 9: Use-Case 09 – Skalieren

- [UC 10] Modell anzeigen

<i>Informationsart</i>	<i>Beschreibung</i>
<i>Identifikation</i>	UC 10
<i>Beschreibung</i>	Modell anzeigen
<i>Akteure</i>	System
<i>Vorbedingungen</i>	STL-Datei fehlerfrei geladen und Datenmodell erzeugt.
<i>Standardablauf</i>	Sobald die Erzeugung des Datenmodells abgeschlossen bzw. das Datenmodell durch den User verändert wurde, wird die Funktion automatisch aufgerufen.
<i>Alternative Abläufe</i>	
<i>Nachbedingungen</i>	Das jeweils aktuellste Datenmodell wird auf der Benutzeroberfläche angezeigt.
<i>Bemerkung</i>	Voraussetzung für [UC 2], [UC 5], [UC 6], [UC 8], [UC 9], [UC 12].

Tabelle 10: Use-Case 10 – Modell anzeigen

- [UC 11] Datenmodell erzeugen / aktualisieren

<i>Informationsart</i>	<i>Beschreibung</i>
<i>Identifikation</i>	UC 11
<i>Beschreibung</i>	Datenmodell erzeugen / aktualisieren
<i>Akteure</i>	System
<i>Vorbedingungen</i>	STL-Datei (ASCII oder Binär) geöffnet und eingelesen.
<i>Standardablauf</i>	1) System wandelt die geöffnete STL-Datei in eine Datenstruktur (Boundary Representation) um. 2) Die Datenstruktur beinhaltet sämtliche Punkte, Kanten und Flächen. 3) Es wird intern eine Verweisstruktur aufgebaut, bei der jede Kante Verweise auf zugehörige Punkte, sowie jede Fläche Verweise auf ihre zugehörigen Kanten enthält.
<i>Alternative Abläufe</i>	
<i>Nachbedingungen</i>	Datenmodell wurde vollständig erzeugt.
<i>Bemerkung</i>	Use-Case ist Voraussetzung für Modell anzeigen [UC 10].

Tabelle 11: Use-Case 11 – Datenmodell erzeugen / aktualisieren

- [UC 12] Geometrie heilen

<i>Informationsart</i>	<i>Beschreibung</i>
<i>Identifikation</i>	UC 12
<i>Beschreibung</i>	Geometrie heilen
<i>Akteure</i>	User, System
<i>Vorbedingungen</i>	3D-Modell vollständig geladen und auf der Benutzeroberfläche angezeigt. Geometrieanalyse (UC 6) abgeschlossen, Fehler gefunden.
<i>Standardablauf</i>	1) User ruft die Funktion auf. 2) User wählt Art der Heilung. 3) User wählt Punkte zur Heilung und ggf. weitere Optionen. 4) User bestätigt seine Eingabe. 5) Datenmodell wird aktualisiert.
<i>Alternative Abläufe</i>	1) User wählt Punkte zur Heilung. 2) User ruft die Funktion auf. 3) User wählt Art der Heilung. 4) User wählt ggf. weitere Optionen. 5) User bestätigt seine Eingabe. 6) Datenmodell wird aktualisiert.
<i>Nachbedingungen</i>	Je nach Art der Heilung entsteht entweder eine neue Fläche, oder es werden Punkte von bereits existierenden Flächen zusammengeführt.
<i>Bemerkung</i>	

Tabelle 12: Use-Case 12 – Geometrie heilen

- [UC 13] STL-Datei erzeugen

<i>Informationsart</i>	<i>Beschreibung</i>
<i>Identifikation</i>	UC 13
<i>Beschreibung</i>	STL-Datei erzeugen
<i>Akteure</i>	User, System
<i>Vorbedingungen</i>	3D-Modell vollständig geladen und Datenmodell erzeugt.
<i>Standardablauf</i>	1) User ruft Funktion auf. 2) User wählt den Speicherort aus. 3) User bestätigt den Datenexport. 4) BRep-Datenmodell wird in STL-Datei geschrieben und gespeichert.
<i>Alternative Abläufe</i>	
<i>Nachbedingungen</i>	Datenmodell wurde als valide STL-Datei gespeichert.
<i>Bemerkung</i>	

Tabelle 13: Use-Case 13 – STL-Datei erzeugen

## 2.2 Aktivitätsdiagramme

### 2.2.1 Aktivitätsdiagramm Datenmodell erzeugen

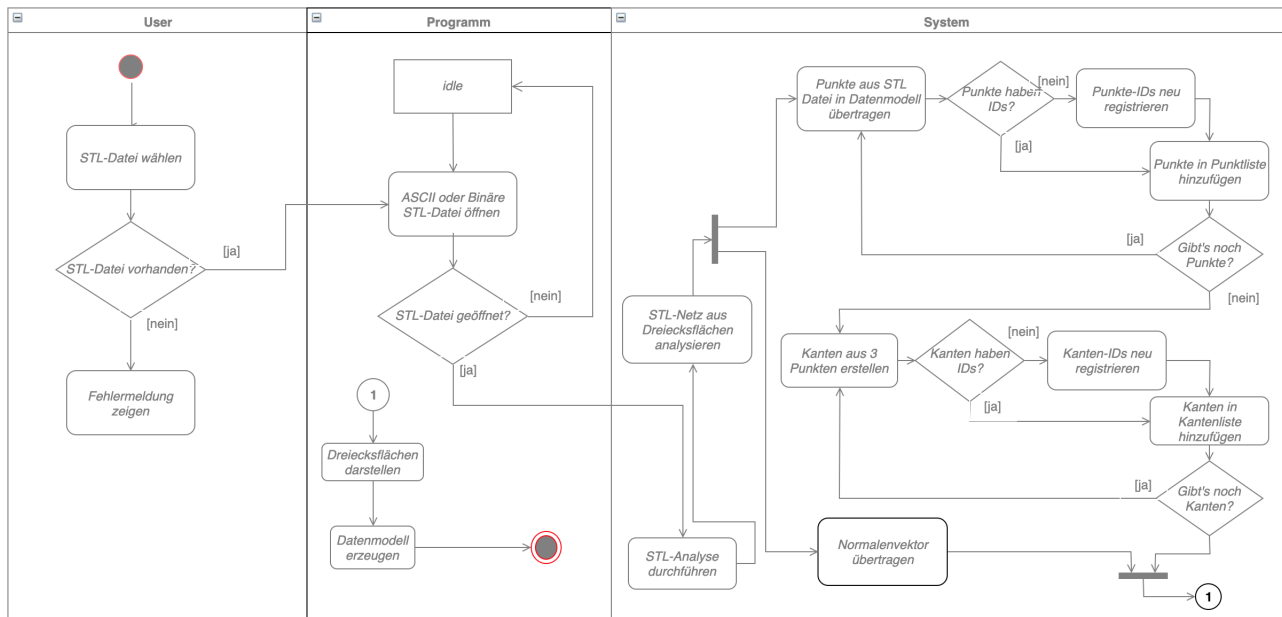


Abbildung 3: Aktivitätsdiagramm (Datenmodell erzeugen)

### 2.2.2 Aktivitätsdiagramm Fehlersuche

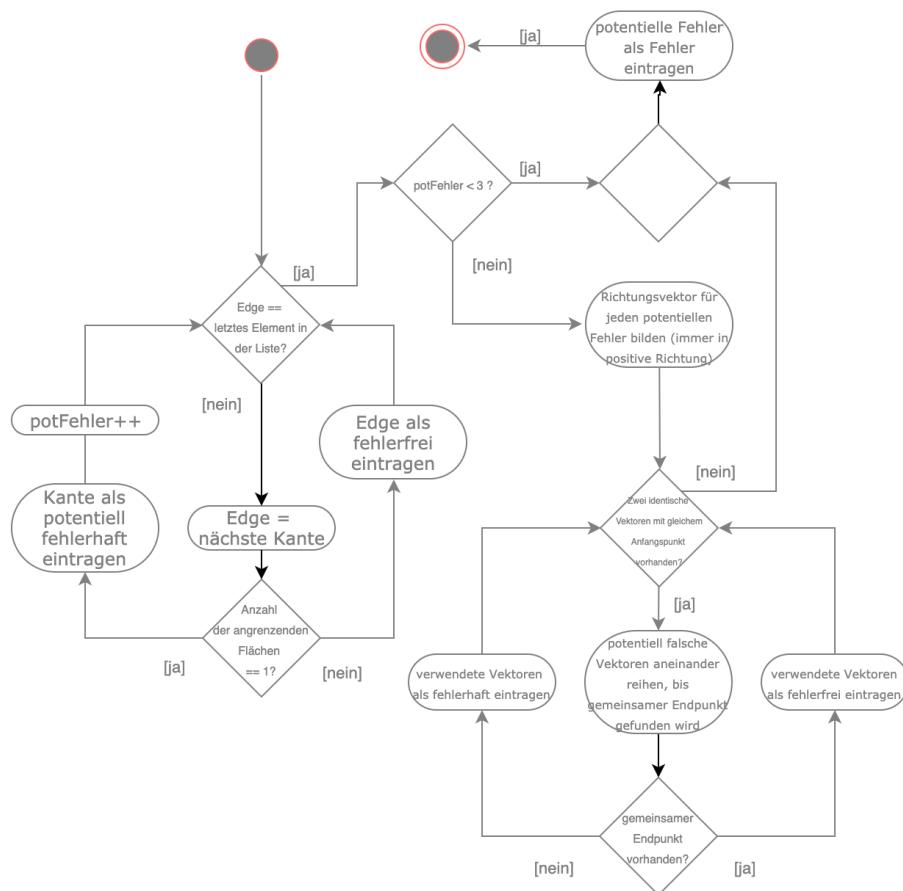


Abbildung 4: Aktivitätsdiagramm (Fehlersuche)

### 2.2.3 Aktivitätsdiagramm ViewModel (OpenGL)

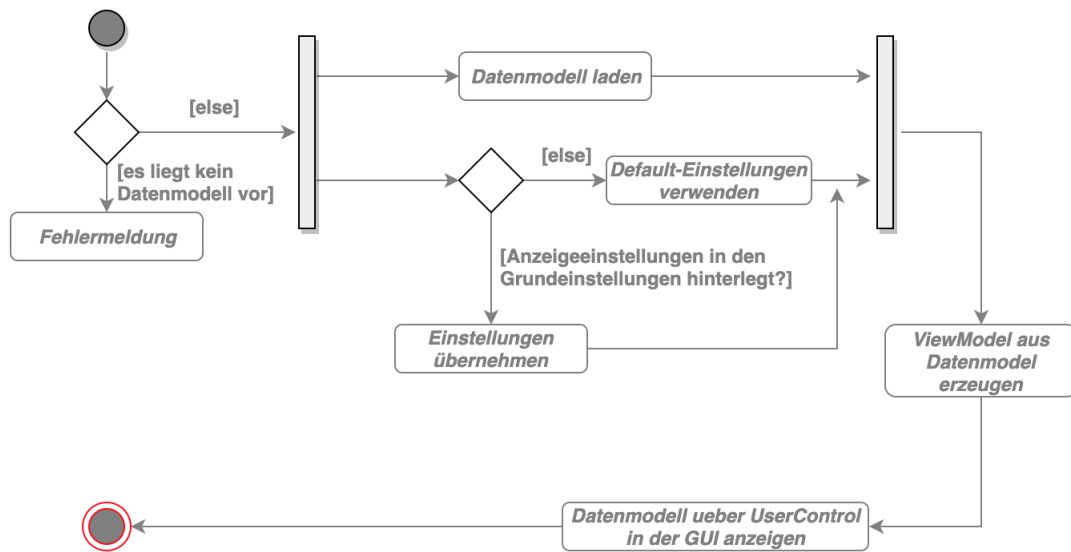


Abbildung 5: Aktivitätsdiagramm (View Model - Datenmodell anzeigen)

### 2.2.4 Aktivitätsdiagramm STL-Export

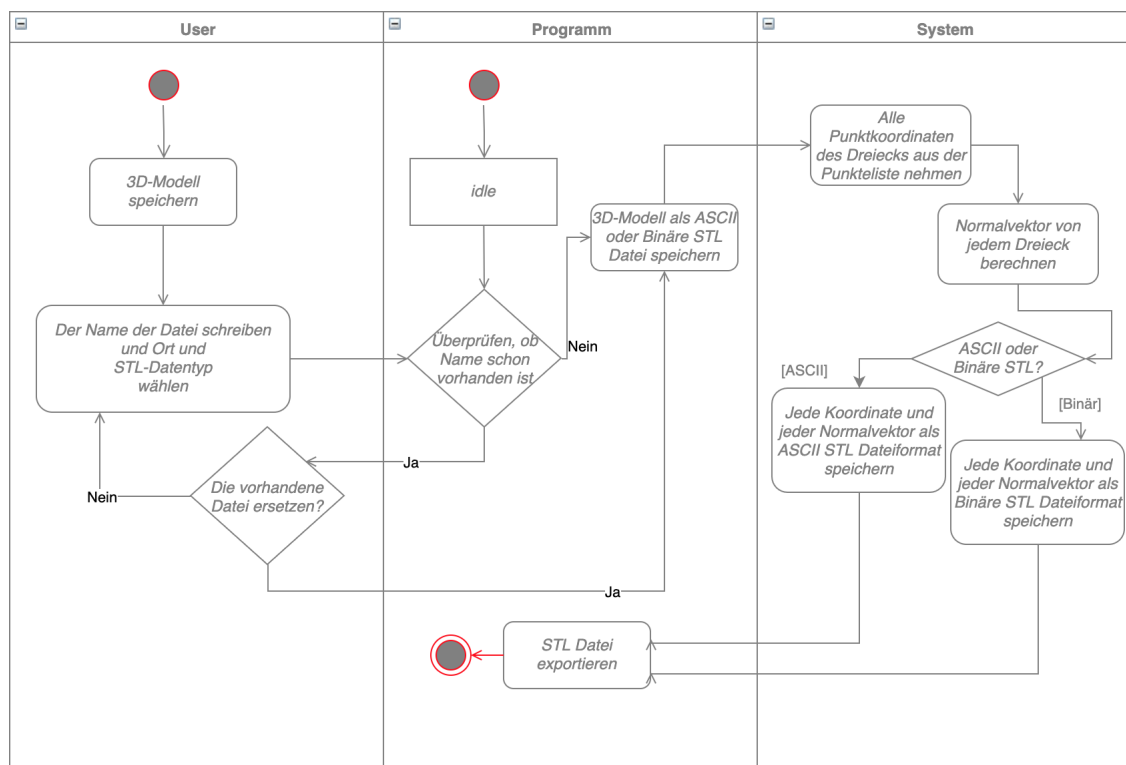


Abbildung 6: Aktivitätsdiagramm (STL-Export)

## 2.3 Grafische Benutzeroberfläche











-  : Linksklick
-  : Rechtsklick
-  : Mouseover
-  : Öffnen
-  : Speichern
-  : Rückgängig machen
-  : Nochmal machen
-  : Einstellungen
-  : Hilfe
-  : Fehler finden

Abbildung 7: Zeichenerklärungen (GUI)

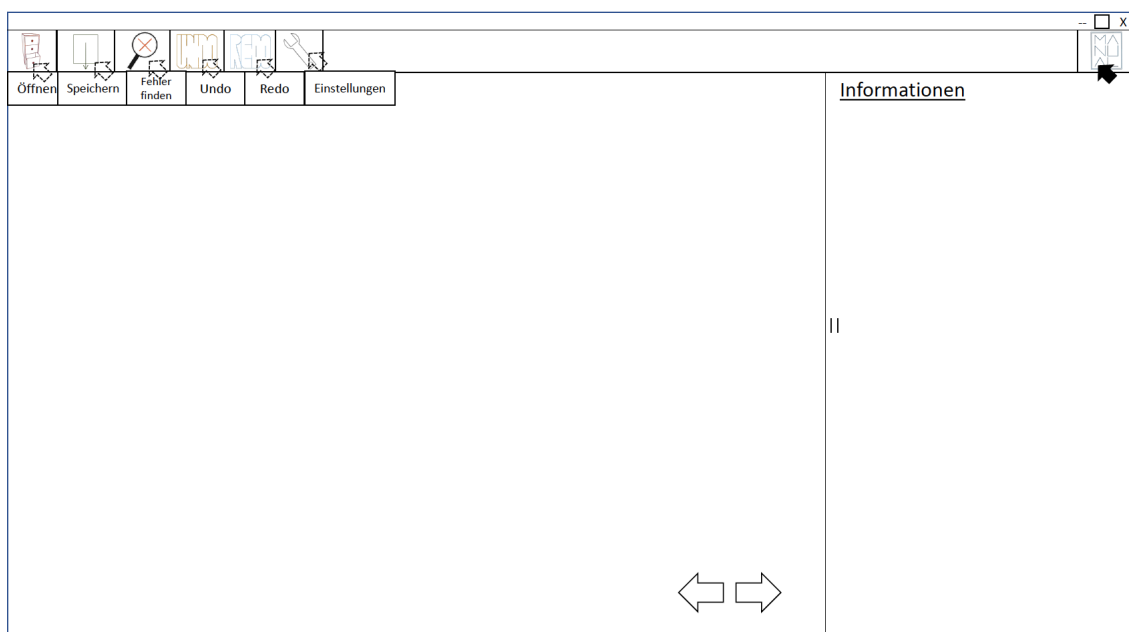


Abbildung 8: Startseite (GUI)



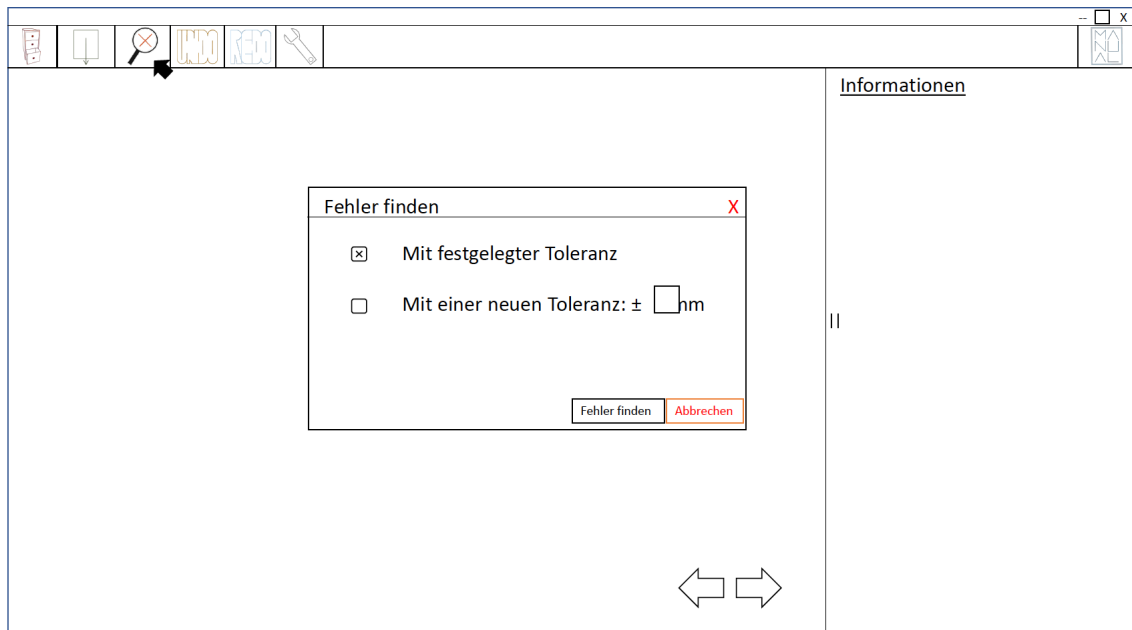


Abbildung 9: Toleranzeingabe bei Fehlersuche (GUI)

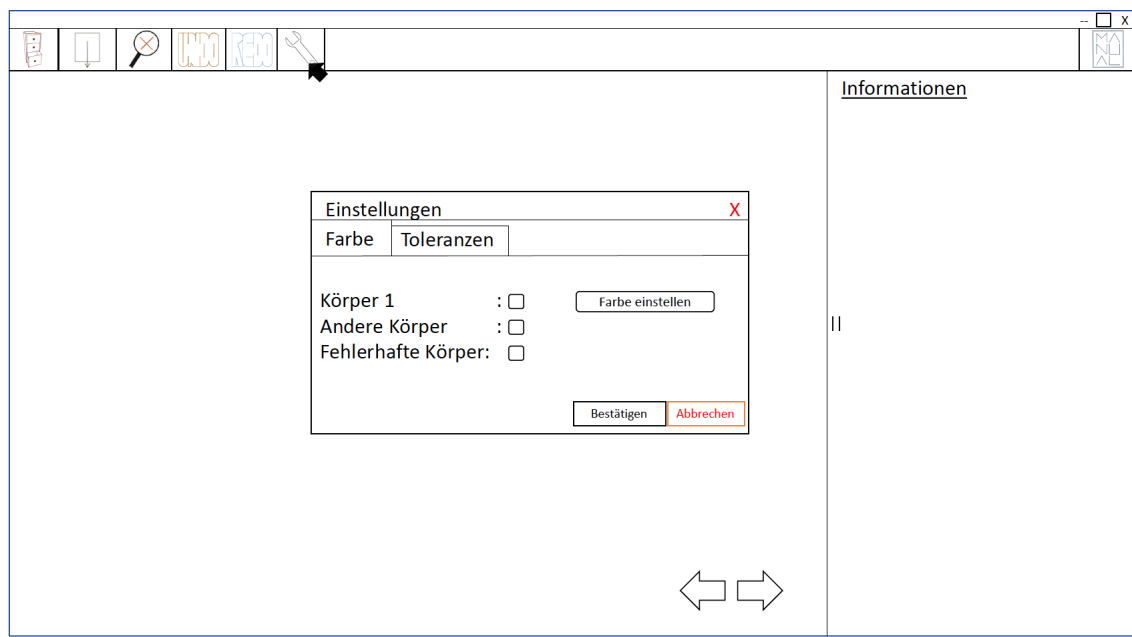


Abbildung 10: Grundeinstellungen Allgemein (GUI)

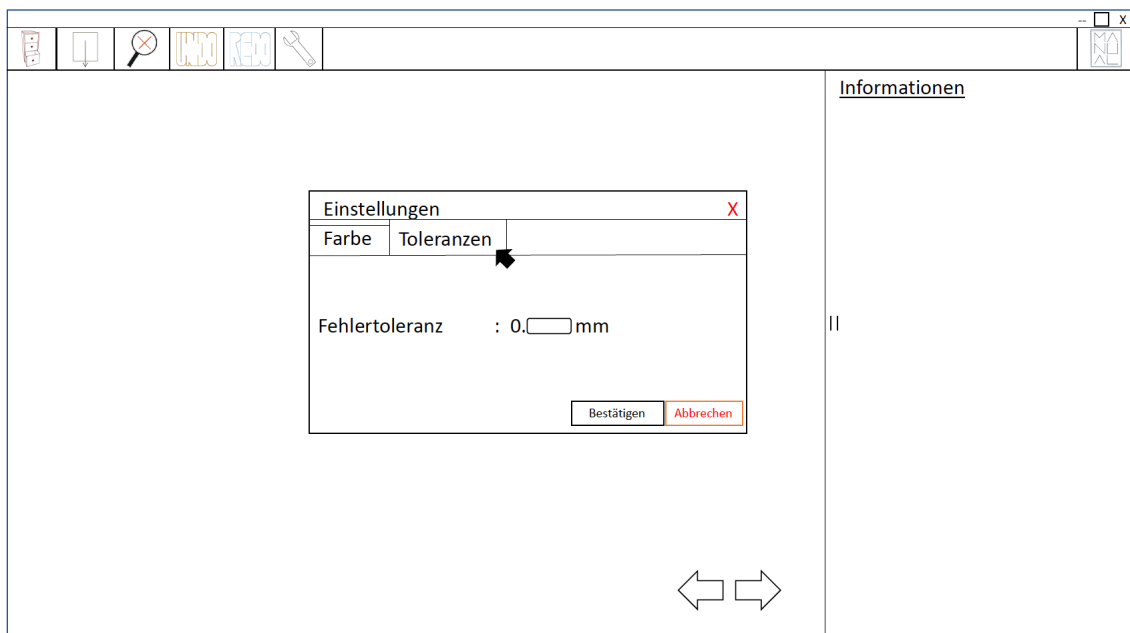


Abbildung 11: Grundeinstellungen Fehlersuche (GUI)

### 3 Überblick - Komponentendiagramm

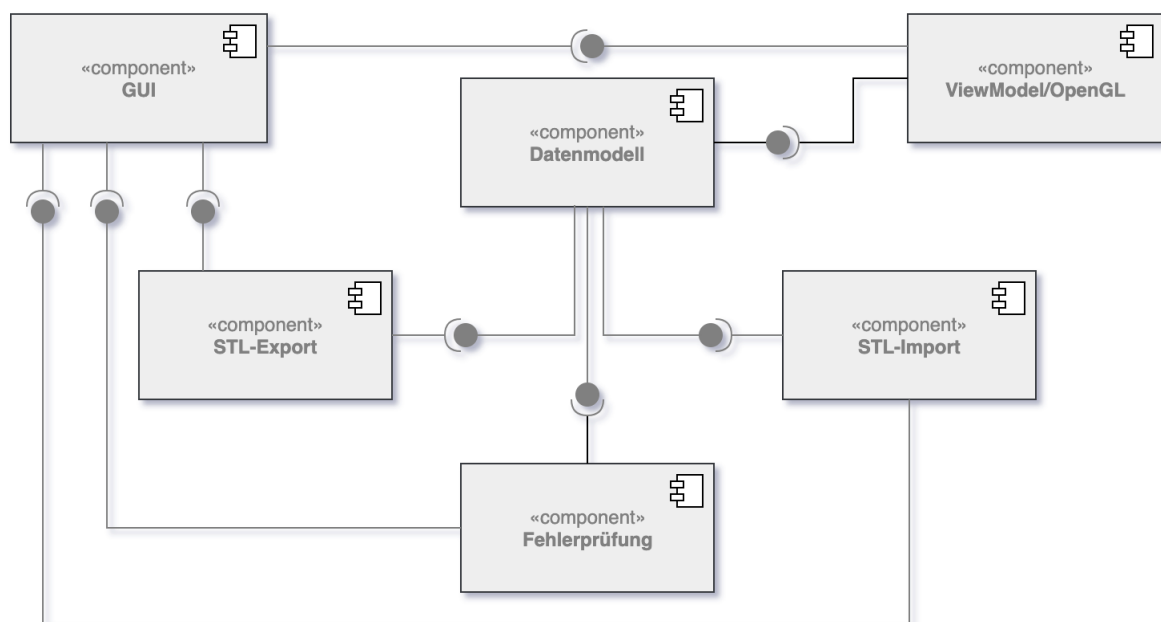


Abbildung 12: Komponenten-Diagramm

## 4 Realisierung

### 4.1 Allgemeines

Die folgende Tabelle bietet einen Überblick über die zu erstellenden Komponenten (siehe auch Komponenten-Diagramm, S. 14), den in den einzelnen Komponenten enthaltenen Use-Cases sowie eine Zuordnung, welches Teammitglied welche Komponenten bearbeitet.

<i><b>Komponente</b></i>	<i><b>Use-Cases</b></i>	<i><b>Bearbeiter</b></i>
GUI	UC 4	Michael Reno
STL-Import	UC 1	Adib Ghassani Waluya
Datenmodell	UC 9, UC 11	Adib Ghassani Waluya
ViewModel / OpenGL	UC 2, UC 5, UC 8, UC 10	Michael Kaip
Fehlerprüfung	UC 6, UC 7, UC 12	Maximilian Mews
STL-Export	UC 3, UC 13	Michael Reno

Tabelle 14: Aufgabenverteilung

### 4.2 Interne Schnittstellen

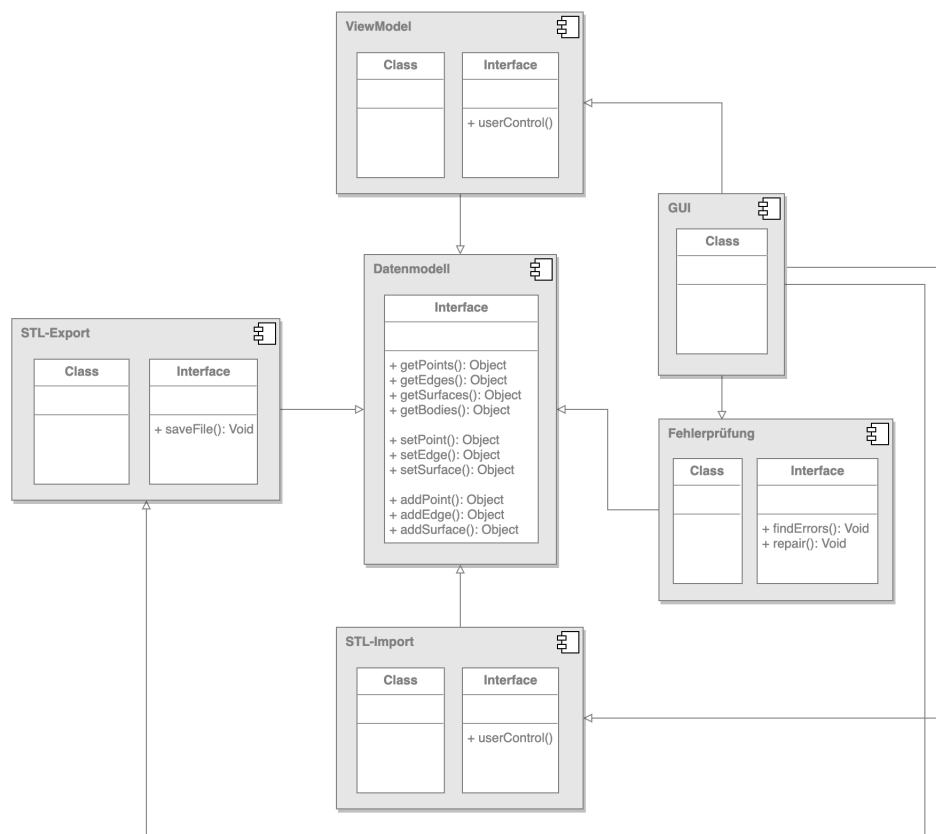


Abbildung 13: Interne Schnittstellen des Systems

### 4.3 Externe Schnittstellen

Das System wird als eine eigenständige Software und nicht als Plug-In für eine andere Software entwickelt, welche bis zu der in diesem Projekt geplanten Entwicklungsstufe nicht die Anbindung oder Ansteuerung anderer externer System (wie z.B. eines 3D-Druckers) vorsieht.

Um bspw. das Datenmodell anzuzeigen, zu messen verwenden wir das OpenGL-Framework OpenTK, welches einen Wrapper für die Verwendung der in OpenGL bereitgestellten Funktionalität in der Programmiersprache C# bereitstellt.

## 5 Projektkalkulation

Komponente / Art	Bezeichnung	Arbeitsaufwand (in Stunden)		Zuständigkeit
		Einzel	Komponente	
GUI	Layout-Entwicklung und -Programmierung	40,0	54,0	Michael R.
	Use-Case 4 (Umsetzung)	8,0		Michael R.
	Testen	6,0		Michael R.
STL-Import	Klassenstruktur erstellen	8,0	14,0	Adib
	Use-Case 1 (Umsetzung)	3,0		Adib
	Testen	3,0		
Datenmodell	Klassenstruktur erstellen	8,0	98,0	Adib
	Use-Case 9 (Umsetzung)	20,0		Adib
	Use-Case 11 (Umsetzung)	40,0		Adib
	Testen	30,0		Adib
ViewModel	Klassenstruktur erstellen	8,0	104,0	Michael K.
	Use-Case 2 (Umsetzung)	16,0		Michael K.
	Use-Case 5 (Umsetzung)	8,0		Michael K.
	Use-Case 8 (Umsetzung)	24,0		Michael K.
	Use-Case 10 (Umsetzung)	24,0		Michael K.
	Testen	24,0		Michael K.
Fehlerprüfung	Klassenstruktur erstellen	8,0	102,0	Maximilian
	Use-Case 6 (Umsetzung)	30,0		Maximilian
	Use-Case 7 (Umsetzung)	4,0		Maximilian
	Use-Case 12 (Umsetzung)	40,0		Maximilian
	Testen	20,0		Maximilian
STL-Export	Klassenstruktur erstellen	8,0	53,0	Michael R.
	Use-Case 3 (Umsetzung)	5,0		Michael R.
	Use-Case 13 (Umsetzung)	20,0		Michael R.
	Testen	20,0		Michael R.
Integration und Test	Einzelkomponenten integrieren	3,0	23,0	
	Integrationstests durchführen	20,0		
		<b><u>Gesamt</u></b>	<b><u>448,0</u></b>	

Abbildung 14: Projektkalkulation

## 6 Abnahmekriterien

1. Die in den Use-Cases beschriebene Funktionalität muss gegeben sein.
2. Das Laden einer Datei mit der Größe von 3 MB und Prüfung auf Fehler darf nicht länger als 20 Sekunden dauern.

## 7 Lizenz

Copyright (c) <2019>

<Michael Kaip, Maximilian Mews, Michael Reno, Adib Ghassani Waluya>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files, 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.

---

Hiermit wird unentgeltlich jeder Person, die eine Kopie der Software und der zugehörigen Dokumentationen erhält, die Erlaubnis erteilt, sie uneingeschränkt zu nutzen, inklusive und ohne Ausnahme mit dem Recht, sie zu verwenden, zu kopieren, zu verändern, zusammenzufügen, zu veröffentlichen, zu verbreiten, zu unterlizenzieren und/oder zu verkaufen, und Personen, denen diese Software überlassen wird, diese Rechte zu verschaffen, unter den folgenden Bedingungen:

DIE SOFTWARE WIRD OHNE JEDE AUSDRÜCKLICHE ODER IMPLIZIERTE GARANTIE BEREITGESTELLT, EINSCHLIESSLICH DER GARANTIE ZUR BENUTZUNG FÜR DEN VORGESEHENEN ODER EINEM BESTIMMTEN ZWECK SOWIE JEDLICHER RECHTSVERLETZUNG, JEDOCH NICHT DARAUFGESCHRÄNKT. IN KEINEM FALL SIND DIE AUTOREN ODER COPYRIGHTINHABER FÜR JEDLICHEN SCHADEN ODER SONSTIGE ANSPRÜCHE HAFTBAR ZU MACHEN, OB INFOLGE DER ERFÜLLUNG EINES VERTRAGES, EINES DELIKTES ODER ANDERS IM ZUSAMMENHANG MIT DER SOFTWARE ODER SONSTIGER VERWENDUNG DER SOFTWARE ENTSTANDEN.