

# Umsetzung eines AR/VR Visualisierungskonzepts für Maschinendaten

## Bachelorarbeit

Abteilung Informatik  
Hochschule für Technik Rapperswil

Frühjahrssemester 2020

Autoren: Janosch Rausch, Michael Näf  
Betreuer: Prof. Dr. Daniel Patrick Politze  
Projektpartner:  
Experte: Ramon Schildknecht  
Gegenleser: Prof. Stefan Richter



## Abstract

### Umsetzen eines AR Designkonzepts

In dieser Arbeit wurde das Designkonzept einer vorhergehenden Studienarbeit umgesetzt. Die grösste Herausforderung hierbei besteht darin, dass die Designs, welche im Konzept erarbeitet wurden, nicht einwandfrei auf die Architektur der Anwendung passen. Somit können gute Ideen im Konzept dazu führen, dass der Fokus nur auf einen Teil der Designs gelegt werden kann.

### Vorgehen

Nach einer Einarbeitungsphase in die schon vorhandene Infrastruktur, auf welcher diese Arbeit aufbaut, wurden die meisten Features erst einmal rudimentär umgesetzt. In den folgenden Sprints wurden die Features dann verfeinert und Fehler ausgebügelt. Gleichzeitig wurde anhand der rudimentären Features ein Usability Test mit berufstätigen Servicetechnikern durchgeführt und das Feedback dieser Tests in eben jene Features eingearbeitet.

### Ergebnis

Die Applikation enthält von Benutzer validierte UI Elemente, welche alle grundlegenden Funktionen erfüllen. Man sieht auf den ersten Blick, welche Daten der Maschine zur Verfügung stehen, welche Teile ersetzt bzw. gewartet werden müssen und wo sich sonst noch potenzielle Fehler befinden könnten.

Trotz einiger Schwierigkeiten mit dem Legacy Code kann das Projekt als Erfolg verbucht werden und der Lerneffekt war riesig.

## Änderungsgeschichte

<b>Datum</b>	<b>Version</b>	<b>Änderung</b>	<b>Autor</b>
<b>01.06.2020</b>	1.00	Initiale Version	Janosch Rausch
<b>02.06.2020</b>	1.01	Situationsanalyse	Janosch Rausch
<b>02.06.2020</b>	1.02	Technologien & Ausblick	Michael Näf
<b>03.06.2020</b>	1.03	Test Szenarien	Janosch Rausch
<b>03.06.2020</b>	1.04	Abstract	Michael Näf
<b>04.06.2020</b>	1.05	Ergebnis Umsetzung - Funktionen	Janosch Rausch
<b>06.06.2020</b>	1.06	Situationsanalyse	Michael Näf
<b>06.06.2020</b>	1.07	Situationsanalyse	Janosch Rausch
<b>07.06.2020</b>	1.08	Situationsanalyse ergänzt, Ergebnis ergänzt	Janosch Rausch
<b>07.06.2020</b>	1.09	Fazit & Architektur Illustration	Michael Näf
<b>08.06.2020</b>	1.10	Initial Nicht-Funktionale Anforderungen	Janosch Rausch
<b>08.06.2020</b>	1.11	Build Anleitung + Fazit	Michael Näf
<b>08.06.2020</b>	2.00	Korrekturen	Janosch Rausch
<b>09.06.2020</b>	2.01	Nicht-Funktionale Anforderungen	Janosch Rausch
<b>09.06.2020</b>	2.02	Grafiken Architektur + Fazit	Michael Näf
<b>10.06.2020</b>	2.03	Korrekturen + kleinere Ergänzungen	Janosch Rausch
<b>10.06.2020</b>	3.00	Konvertierung in Word	Janosch Rausch
<b>10.06.2020</b>	3.01	Korrekturen + Formatierung	Janosch Rausch, Michael Näf

## Inhaltsverzeichnis

1	Abkürzungsverzeichnis.....	1
2	Projektübersicht.....	2
2.1	Ausgangslage.....	2
2.2	Ziel.....	2
2.3	Ausblick.....	2
3	Projektorganisation.....	3
3.1	Organisationsstruktur.....	3
3.2	Projektplan.....	3
4	Situationsanalyse.....	4
4.1	Konzept.....	4
4.2	Funktionen und Anforderungen.....	4
4.2.1	Hauptmenü.....	4
4.2.2	Generische Anzeigen.....	7
4.2.3	Aufgaben Menü.....	8
4.2.4	Maschinenteile Menü.....	10
4.2.5	Dashboard.....	14
4.2.6	Farben.....	16
4.3	Nicht-Funktionale Anforderungen.....	17
4.3.1	Funktionalität.....	17
4.3.2	Benutzbarkeit.....	17
4.3.3	Zuverlässigkeit.....	18
4.3.4	Effizienz.....	18
4.3.5	Änderbarkeit.....	18
5	Technologien.....	20
5.1	Hololens.....	20
5.2	Hololens 2.....	20
5.3	Unity.....	20
5.4	Visual Studio.....	20
5.5	Windows 10.....	20
5.6	Vuforia.....	20
5.7	Mixed Reality Toolkit.....	20
5.8	Build Anleitung.....	21

# Umsetzung eines AR/VR Visualisierungskonzepts für Maschinendaten

6	Architektur .....	22
6.1	Navigation .....	22
6.2	Unity Szenen Struktur .....	24
7	Ergebnis Umsetzung .....	25
7.1	Funktionen .....	25
7.1.1	Hauptmenü .....	25
7.1.2	Generische Anzeigen.....	27
7.1.3	Aufgaben Menü .....	28
7.1.4	Maschinenteile Menü .....	29
7.1.5	Dashboard.....	31
7.1.6	Allgemein .....	33
7.2	Nicht funktionale Anforderungen .....	36
7.2.1	Funktionalität .....	36
7.2.2	Benutzbarkeit.....	36
7.2.3	Zuverlässigkeit.....	37
7.2.4	Effizienz .....	37
7.2.5	Änderbarkeit .....	38
8	Testszenarien .....	39
8.1	Aufbau und Durchführung .....	39
8.2	Feedback und Einfluss.....	40
9	Ausblick .....	41
9.1	Hololens 2 .....	41
9.2	QR Code reading for machine change .....	41
9.3	Pfeil Indikatoren.....	41
9.4	Scrollbare Listen.....	41
9.5	Hervorheben der Bauteile.....	41
10	Fazit.....	42
11	Quellenverzeichnis.....	43
12	Abbildungsverzeichnis .....	44

# 1 Abkürzungsverzeichnis

<b>Abkürzung</b>	<b>Bezeichnung</b>
<b>MRTK</b>	Mixed Reality Toolkit, (ref.)
<b>IDE</b>	Integrierte Entwicklungsumgebung
<b>AR</b>	Artificial Reality, dt. Künstliche Realität
<b>DRY</b>	Don't Repeat Yourself

## 2 Projektübersicht

### 2.1 Ausgangslage

Durch vorhergehende Studien- und Bachelorarbeiten werden Maschinendaten bereits auf der Hololens dargestellt. Aufgaben können angezeigt und erstellt werden. Ebenfalls sind protokollierte Modifikationen der Maschinenteile möglich.

Bisher wurde die Darstellung jedoch sehr dürftig umgesetzt. Das Hauptmenü befindet sich meistens in Bodennähe und ist schnell zu übersehen oder verschwindet manchmal komplett. Alle Anzeigen der Maschinendaten sehen gleich aus und sind von einem Button nicht zu unterscheiden. Die Menüführung ist nicht ersichtlich und es gibt sehr viele grafische Instabilitäten, wie überlappende oder flackernde Menüs.

Vor dieser Bachelorarbeit wurde eine Studienarbeit durchgeführt, in welcher das Konzept zur Verbesserung der Benutzeroberflächen erarbeitet wurde.

### 2.2 Ziel

Das in der vorhergehenden Studienarbeit erstellte Design soll nun umgesetzt werden. Hierfür wird die bereits existierende Anwendung als Ausgangslage verwendet. Am Ende soll eine Anwendung vorhanden sein, welche auch zu Demonstrationszwecken verwendet werden kann.

Aufgaben können betrachtet und als abgeschlossen markiert werden. Durch die Bauteile kann navigiert werden und es ist möglich, die jeweiligen Informationen einzublenden. Das Dashboard zeigt alle Informationen zu den Sensoren, um einen Überblick zu bieten.

Das 3D Modell der Maschine kann betrachtet werden und einzelne Bauteile werden markiert, wenn sie ausgewählt werden. Man kann Bauteile sowohl im Menü als auch im Modell auswählen.

Die Umsetzung wird anhand eines Usability Tests auf die Verständlichkeit überprüft.

### 2.3 Ausblick

Zur Zeit dieser Bachelorarbeit steht nur die Hololens 1 zur Verfügung. In einer nachfolgenden Arbeit kann die Benutzeroberfläche weiter ergänzt und auch auf die Hololens 2 optimiert werden.

Einige experimentelle Funktionalitäten, wie zum Beispiel scrollbare Listen, können zusätzlich integriert werden, wenn diese besser ausgereift sind.

## 3 Projektorganisation

### 3.1 Organisationsstruktur

#### Mitglieder:

- Janosch Rausch
- Michael Näf

#### Betreuer:

Prof. Dr. Daniel Patrick Politze

#### Gegenleser:

Prof. Stefan Richter

#### Experte:

Ramon Schildknecht, SBB AG, Olten 1, SO

#### Andere Kontakte:

Lukas Kretschmar und Mike Schmid:

Sind zuständig für den Server, welcher hinter der Applikation läuft und stehen für technischen Support zu Hilfe.

Pius Anderhub - Ansprechpartner bei Komax

Mit seiner Hilfe wurde der Usability Test durchgeführt.

### 3.2 Projektplan

Die Projektplanung wurde in einem separaten Dokument erstellt. (siehe "Projektplan.pdf")

## 4 Situationsanalyse

Als Grundlage für diese Bachelorarbeit wurde eine Studienarbeit erstellt, in welcher ein neues Konzept für die Benutzeroberflächen entworfen wurde. Das erarbeitete Konzept wird als Anforderung an die Umsetzung betrachtet.

### 4.1 Konzept

Beim Konzept wurde der Schwerpunkt auf eine verständliche und übersichtliche Oberfläche gelegt. Knöpfe sollen gut von Anzeigen zu unterscheiden und wichtige Informationen schnell zugänglich sein.

### 4.2 Funktionen und Anforderungen

#### 4.2.1 Hauptmenü

Das bisherige Hauptmenü muss oft gesucht werden, da es sich nicht immer an der gleichen Position befindet und teilweise gar nicht sichtbar ist. Die Grundlegende Funktionalität, welche ein Hauptmenü zur Verfügung stellt, muss immer zugänglich sein.

Somit ergibt es sich, dass die Position immer gleichbleiben muss und das Hauptmenü immer leicht zu finden sein soll. Für Geräte ohne Eyetracking, wie die Hololens 1, ist dieses Menü stets am unteren Rand des Sichtfeldes zu finden. Neuere Geräte, wie die Hololens 2, verfügen über Eyetracking, welches genutzt wird, um das Hauptmenü anzuzeigen, wenn der Blick an den unteren Rand des Sichtfeldes geführt wird.



Abbildung 1: Layout des Hauptmenüs

#### Funktionen:

- F01: Dashboard anzeigen
- F02: Reset des Modells
- F03: Reset der Menüs
- F04: Farbenblindmodus Ein-/Ausschalten
- F05: Model Transparenz verändern
- F06: Persistenz der Einstellungen
- F07: Fixe Positionierung im Blickfeld

## Umsetzungsmöglichkeiten:

F01: Dashboard anzeigen:

- Bereits in der Szene jedoch inaktiv (nicht sichtbar)
- Aus einem Prefab dynamisch laden und in der Szene platzieren

Das Erzeugen eines GameObjects, vor allem mittels eines komplexen Prefabs, benötigt vergleichsweise viel Rechenleistung und entsprechend viel Zeit.

Aus diesem Grund wurde versucht, soviel wie möglich am Anfang der Applikation zu laden. Unter anderem auch das Dashboard.

Jedoch ist das Dashboard nicht inaktiv. Der Techniker soll sofort einen Blick auf die Datenübersicht haben.

F02: Reset des Modells:

- Ganze Szene neu laden und Menüs persistieren
- Ganzes Modell löschen und neu laden
- Undo aller Operationen auf dem Modell

In dieser Arbeit wurden keine Funktionen implementiert, welche das Modell verändern, weshalb diese Funktionalität nicht implementiert wurde. Das Laden der CAD Daten zur Erstellung des Modells benötigt viel Zeit, weswegen es zu empfehlen ist, alle Operationen in einer Liste zu speichern und diese Operationen rückwärts auszuführen.

F03: Reset der Menüs:

- Kaskadierend jedes Menü für sich selbst
- Löschen und Neuerstellen aller Menüs im Standardzustand

Aufgrund der einfachen Implementation eines kaskadierenden Zurücksetzens aller Menüs wurde diese Lösung gewählt. Hierbei stellt jedes Fenster für sich selbst seinen Ursprungszustand wieder her und ruft anschliessend die Funktion für diejenigen Fenster auf, deren Referenz es hält. Somit muss dem Button nur das erste Fenster bekannt sein.

Beispiel: Das Fenster mit der Liste der Service Cases öffnet das Fenster mit den Details, sobald ein Service Case ausgewählt wird. Das Aufgabenmenü hält eine Referenz auf das Detailfenster und kann dieses wieder ausblenden und den Reset starten.

F04: Farbenblindmodus Ein-/Ausschalten:

- Option mittels normalem Button<sup>1</sup>
- Option mittels ToggleButton

Da das MRTK einen vorgefertigten ToggleButton zur Verfügung stellt und dieser genau die Funktionalität bietet, die benötigt wird, wurde der ToggleButton genutzt.

---

<sup>1</sup> [https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/README\\_Button.html](https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/README_Button.html)

## Umsetzung eines AR/VR Visualisierungskonzepts für Maschinendaten

F05: Model Transparenz verändern:

- Vorgabe von mehreren Transparenzstufen die zyklisch durchlaufen werden
- Schieberegler zur Feinjustierung der Transparenz

Das MRTK bietet einen vorgefertigten Schieberegler welcher stabil funktioniert. Präzisere Transparenzeinstellungen sind ebenfalls wünschenswert.

Deshalb wurde ein Schieberegler benutzt.

F06: Persistenz der Einstellungen:

- PlayerPrefs<sup>2</sup> von Unity nutzen
- Eigene Persistierung von Einstellungen

Unity bietet bereits eine Möglichkeit an, wie Einstellungen persistiert werden können. Es wurde entschieden, diese zu verwenden. Der Aufwand eine eigene Persistierung für die Daten zu schreiben überwiegt bei weitem.

F07: Fixe Positionierung im Blickfeld:

- Vorhandenen Solver<sup>3</sup> konfigurieren
- Eigenen Solver implementieren

Eine konstante Positionierung im Blickfeld kann mit den vom MRTK angebotenen Solver erreicht werden. Aus diesem Grund wurde entschieden, dass der Orbital Solver verwendet wird, welcher ein Objekt in einem spezifischen Winkel zur Hololens positioniert.

---

<sup>2</sup> <https://docs.unity3d.com/ScriptReference/PlayerPrefs.html>

<sup>3</sup> [https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/README\\_Solver.html](https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/README_Solver.html)

## 4.2.2 Generische Anzeigen

Die Anzeigen der Vorarbeit heben sich nicht von anderen Bestandteilen der Benutzeroberfläche ab. Messwerte werden zwar angezeigt, jedoch ohne Relation zu einem Maximal- oder Minimalwert. Ebenfalls lässt sich kein Verlauf der gemessenen Werte erkennen, sondern nur der letzte Wert, welcher gemessen wurde.

Es wurde eine neue Anzeige entworfen, mit welcher schnell erkennbar ist, ob sich ein Messwert in einem normalen Rahmen befindet. Zusätzlich wurde für die farbliche Unterscheidung ein Farbenblindmodus erarbeitet und miteinbezogen.

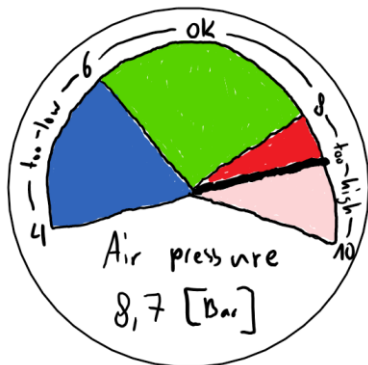


Abbildung 2: Veranschaulichung einer generischen Anzeige

Ergänzend zu den aktuellen Messwerten wurde ein Dashboard konzipiert, welches einer besseren Übersicht dient. Mehr dazu unter 4.2.5 Dashboard.

### Funktionen:

- F08: Wertanzeige
- F09: Dynamische Einheiten
- F10: Dynamische Min. und Max.
- F11: Farbwechsel über und unter dem Messwert
- F12: Dynamische Anzeige des aktuellen Messwertes
- F13: Ändert Farbpalette nach Ein-/Ausstellung des Farbenblindmodus

### Umsetzungsmöglichkeiten:

Auf ein Umsetzen der generischen Anzeigen wird verzichtet, da der zusätzliche Informationsgehalt als zu gering empfunden wurde. Ein Verlauf der Messwerte, wie auf dem Dashboard zu sehen, ist viel aussagekräftiger als die jeweils aktuellen Messwerte. Zudem läuft die Maschine meistens nicht, da sie ja repariert werden soll und die aktuellen Messwerte sind aus diesem Grund nicht aussagekräftig.

### 4.2.3 Aufgaben Menü

Die Aufgaben können betrachtet werden, jedoch sind sie nicht direkt ersichtlich und es gibt keine genaueren Informationen dazu. Sie müssen zuerst über das Hauptmenü angezeigt werden, welches teilweise verschwindet, womit auch der Zugriff auf die Aufgaben wegfällt.

Die Aufgaben werden direkt nach dem Start der Applikation angezeigt und müssen nicht erst geladen werden. Das Fenster befindet sich immer auf der rechten Seite des Benutzers, auch wenn sich dieser um die Maschine bewegt. Somit sind diese immer leicht zu finden.

Eine Aufgabe kann ausgewählt werden, dabei öffnet sich ein weiteres Fenster, welches die Details anzeigt. Ebenfalls können bereits abgeschlossene Aufgaben betrachtet werden, was, im Falle eines ähnlichen Fehlers, hilfreich sein kann.

In den Details kann, durch einen Klick auf den Status einer offenen Aufgabe, diese auf "Done" gesetzt werden, damit später an einem Computer schnell zu erkennen ist, welche Aufgabe erledigt wurde.

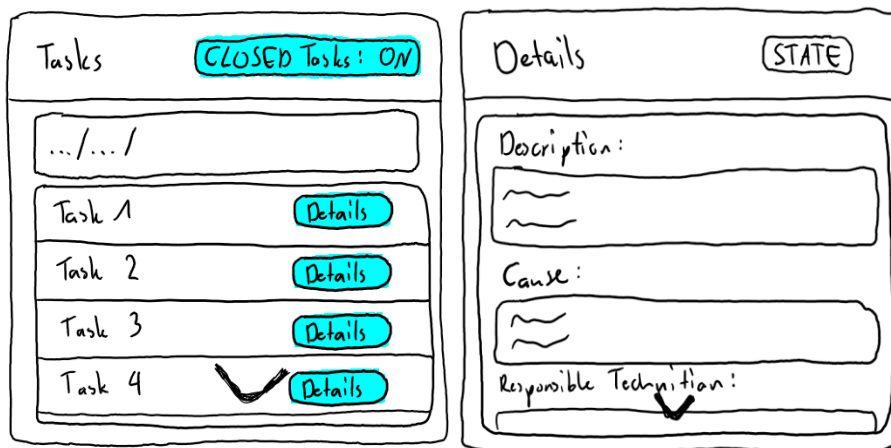


Abbildung 3: Fenster des Aufgabenmenüs

#### Funktionen:

- F14: Positionierung der Fenster
- F15: Abgeschlossene Aufgaben ein- und ausblenden
- F16: Scrollbare Liste
- F17: Pfeilindikator, welcher andeutet, dass die Liste scrollbar ist
- F18: Anzeige der Details für einen Auftrag
- F19: Status veränderbar auf "Done"

## Umsetzungsmöglichkeiten:

F14: Positionierung der Fenster:

- Vorhandenen Solver<sup>3</sup> konfigurieren.
- Eigenen Solver implementieren.

Für die Positionierung der Fenster wurde im Konzept eine Strategie entworfen. Es wird ein Kreis um die Maschine gezogen, welcher durch die Position der Hololens verläuft. Zusätzlich wird ein Kreis um die Hololens gezogen, welcher den Abstand der Fenster zum Benutzer darstellt. Diese Kreise schneiden sich in zwei Punkten, welche als Ausgangspunkt für die Positionierung der Fenster verwendet werden.

Für diese Strategie gibt es keinen passenden Solver, welcher konfiguriert werden kann, weshalb entschieden wurde, dass ein eigener Solver implementiert wird.

F15: Abgeschlossene Aufgaben ein- und ausblenden:

- Button<sup>1</sup> mit einer Funktion, welche einen Zustand hält
- Togglebutton mit zwei Funktionen für eine Aktivierung und Deaktivierung

Ein Togglebutton mit zwei Funktionen trifft genau auf die benötigte Funktionalität zu. Zudem kann somit ein eigenes Verwalten über den Zustand vermieden werden. Dies führte zu dem Entscheid, einen Togglebutton zu verwenden.

F16: Scrollbare Liste:

- Die scrollbare Liste<sup>4</sup> aus den MRTK experimentellen Funktionen
- Als Alternative eine seitenbasierte Liste

Die scrollbare Liste wurde auf der Hololens getestet und es musste leider festgestellt werden, dass diese auf der Hololens 1 nicht gut funktioniert und auch unpraktisch zu bedienen ist.

Stabilität ist eine wichtige Eigenschaft, deswegen wurde entschieden eine seitenbasierte Liste umzusetzen. Solche Listen sind sehr verbreitet und dem Benutzer vertraut. Was für einen intuitiven Umgang sorgt.

Für eine Umsetzung auf der Hololens 2 wird empfohlen, die scrollbaren Listen erneut zu beurteilen, da auf der Hololens 2 unabhängig von der Blickrichtung mit Objekten interagiert werden kann. Dies macht eine Bedienung von scrollbaren Listen wesentlich angenehmer.

F17: Pfeilindikator, welcher andeutet, dass die Liste scrollbar ist:

Nachdem entschieden wurde, dass die scrollbaren Listen nicht umgesetzt werden, wurde der Fokus nicht mehr auf diese Indikatoren gesetzt.

---

<sup>4</sup> <https://microsoft.github.io/MixedRealityToolkit-Unity/Assets/MRTK/SDK/Experimental/ScrollingObjectCollection/README.html>

## Umsetzung eines AR/VR Visualisierungskonzepts für Maschinendaten

F18: Anzeige der Details für einen Auftrag:

- Erstellen/Anzeigen eines weiteren Fensters, welches die Details enthält
- Erweitern des bestehenden Fensters, um die Details anzuzeigen.

Es wurde getestet, wie sich sehr breite Fenster verhalten, wenn man die Hololens trägt und es stellte sich heraus, dass Inhalte weit links oder rechts verzerrt werden, da diese eher seitlich, als von vorne betrachtet werden. Aus diesem Grund wurde beschlossen, die Details in einem separaten Fenster anzuzeigen, welches unabhängig ausgerichtet werden kann.

F19: Status veränderbar auf "Done":

- Anstatt "Done" die vorhandene Funktionalität für "Closed" verwenden
- Das Backend erweitern, damit dieser zusätzliche Status unterstützt wird

Da der Fokus auf die Benutzeroberfläche gelegt wurde, ist ein weitreichender Eingriff in das Backend nicht das Ziel. Es wurde deshalb entschieden, dass kein zusätzlicher Zustand hinzugefügt wird, sondern der Auftrag auf "Closed" gesetzt werden kann.

### 4.2.4 Maschinenteile Menü

Bisher konnten Maschinenteile ausgewählt werden, jedoch wurden keine genauen Informationen dazu angezeigt. In die Unterbauteile zu navigieren war sehr komplex und unübersichtlich. Ein Pfad, welcher die Position im Baum ersichtlich macht, war auch nicht vorhanden. Die Felder waren zudem sehr instabil und flackerten oder verschoben sich oft.

Die einzelnen Teile wurden in einem Fenster zusammengefasst. Nach dem Start sind alle Bauteile der obersten Gruppe sichtbar. Die Auflistung ist nach Seriennummer, Datum der letzten Wartung und Datum der letzten Modifikation sortierbar. Auch wird oben am Fenster, ähnlich einer Ordnerstruktur, der Pfad im Baum angezeigt.

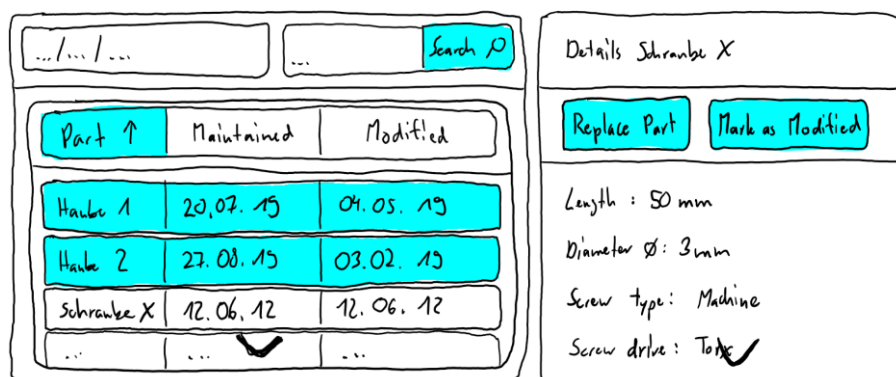


Abbildung 4: Fenster des Maschinenteilemenüs

Die Anzahl der Bauteile einer Maschine kann sehr umfangreich sein, weshalb ein Suchfeld oben rechts geplant ist, über welches nach spezifischen Bauteilen gesucht werden kann. Die Bauteile werden in einer scrollbaren Liste angezeigt und können angeklickt werden, um weitere Informationen zu den Bauteilen zu erhalten.

## Umsetzung eines AR/VR Visualisierungskonzepts für Maschinendaten

Diese Informationen sind unterschiedlich für Baugruppen und Bauteile. Baugruppen bieten die Möglichkeit Schemas oder Anleitungen zu laden, wohingegen Bauteile gewartet oder ersetzt werden können. Ebenso werden für Bauteile, falls vorhanden, zusätzliche Informationen angezeigt.

### Funktionen:

- F20: Maschinenteile anzeigen
- F21: Sortierung der Bauteile
- F22: Farbliche Hervorhebung
- F23: Suche von Bauteilen
- F24: Angabe des Pfades
- F25: Anzeige von Details für Baugruppen und Bauteilen
- F26: Bauteile können ersetzt und gewartet werden
- F27: Auswahl eines Bauteils markiert dieses im 3D Modell
- F28: Auswahl eines Teiles im 3D Modell wählt dieses im Menü aus
- F29: Bestätigungsdialog, um Fehleingaben zu verhindern

Zusätzlich betreffen dieses Menü mehrere, schon aufgezählte, Funktionen, wie zum Beispiel die Positionierung des Fensters. Für die Bewertung werden diese Funktionen unter der Gruppe Allgemein zusammengefasst.

### Umsetzungsmöglichkeiten:

F20: Maschinenteile anzeigen:

- In einer GridObjectCollection<sup>5</sup>
- In einer LayoutGroup<sup>6</sup>
- In GameObjects, welche korrekt positioniert sind

Die GridObjectCollection wurde bereits von den Vorgängern verwendet und sollte auch weiterhin verwendet werden. Leider war dies nicht möglich, da die Anzahl der Einträge in der Liste nicht konstant ist. Im Gegensatz zu den Vorgängern, welche die Listen frei in der Luft schwebend hatten, wurden in dieser Arbeit fixe Fenster verwendet, welche die Daten anzeigen sollen. Die Elemente in der GridObjectCollection sind in der verwendeten Version vom MRTK immer zentral positioniert. Wenn nur zwei anstatt vier Objekte angezeigt werden, führt dies dazu, dass somit das oberste und unterste verschwinden und die mittleren zwei übrig bleiben.

Aus diesem Grund wurde die GridObjectCollection wieder verworfen und ein Versuch mit der LayoutGroup gestartet. Diese funktioniert jedoch nicht korrekt mit dynamisch erstellten Objekten, wenn sich das Fenster im Raum bewegt.

Somit wurde die letzte Variante umgesetzt. Es gibt also immer vier Einträge, welche mit den Daten gespeist werden, die angezeigt werden sollen. Sind nur zwei von vier Einträgen notwendig, so werden die letzten beiden ausgeblendet. Dies hat den Vorteil, dass die Position immer korrekt ist.

---

<sup>5</sup> [https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/README\\_ObjectCollection.html](https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/README_ObjectCollection.html)

<sup>6</sup> <https://docs.unity3d.com/2019.1/Documentation/ScriptReference/UI.LayoutGroup.html>

### F21: Sortierung der Bauteile:

- Sortierung mit dem CollectionService
- Sortierung in einer eigenen Klasse
- Sortierung durch Extension Methoden

Jeder Typ von Auflistungen hat einen eigenen CollectionService. Es wäre also möglich, dass der CollectionService für die Maschinenteile die Sortierung selbst macht. Aus dem Prinzip für Single-Responsibility<sup>7</sup> geht jedoch hervor, dass diese Funktionalität getrennt vom Service sein soll. Schliesslich wurde beschlossen, dass diese Funktionalität als Extension Methods zur Liste hinzugefügt werden. Da diese typenabhängig sind, können die Extension Methods nur auf Listen von Maschinenteilen angewendet werden.

### F22: Farbliche Hervorhebung:

- Hintergrundfarbe ändern

Diese Methode wurde auch von den Vorgängern genutzt, um farbliche Veränderungen zu tätigen. Diese Art wird weitergeführt.

### F23: Suche von Bauteilen:

- MRTK System Keyboard<sup>8</sup> mit TouchScreenKeyboard
- Eingabe mittels Spracherkennung

Weil die Spracheingabe, laut Interview in der Studienarbeit, nicht verwendet werden kann, ist die einzige andere Möglichkeit eine Texteingabe. Hierfür existiert das System Keyboard vom MRTK, welches das TouchScreenKeyboard nutzt, jedoch ist dieses nicht mit einem D3D Projekt kompatibel. Um das Keyboard zu nutzen, muss die App während der Eingabe von 3D auf 2D<sup>9</sup> wechseln, während dem die Maschine und alle Menüs nicht mehr sichtbar wären. Es wurde entschieden, dass das System Keyboard für den Anwendungsfall nicht zu gebrauchen ist.

### F24: Angabe des Pfades:

- Stack für Aufrufverlauf verwenden
- Pfad ausgehend vom gewählten Bauteil berechnen

Da jedes Bauteil oder jede Baugruppe die übergeordnete Baugruppe kennt, kann der Pfad berechnet werden. Es muss also kein Aufrufverlauf angelegt werden.

---

<sup>7</sup> <https://de.wikipedia.org/wiki/Single-Responsibility-Prinzip>

<sup>8</sup> [https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/README\\_SystemKeyboard.html](https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/README_SystemKeyboard.html)

<sup>9</sup> <https://docs.microsoft.com/en-us/windows/mixed-reality/keyboard-input-in-unity>

## Umsetzung eines AR/VR Visualisierungskonzepts für Maschinendaten

F25: Anzeige von Details für Baugruppen und Bauteilen:

- Dynamische Liste für Details
- Neues Fenster mit fixen Feldern für bestimmte Details

Die verschiedenen Bauteile besitzen sehr unterschiedliche Informationen über ihre physischen Eigenschaften, weshalb alle möglichen Informationen angezeigt werden sollten.

Es wird eine Liste von Tupeln mit der Bezeichnung und dem Wert verwendet. So kann eine dynamische Anzahl an Informationen im Fenster dargestellt werden. Standardmässig werden die Serien- und Artikelnummer angezeigt. Zusätzliche Informationen sollten vom Backend in Form einer passenden Liste von Tupeln kommen und könnten dann direkt angehängt werden.

F26: Bauteile können ersetzt und gewartet werden:

- Vorhandene Funktionalität nutzen

In der Vorgängerversion existiert diese Funktionalität bereits und muss mit der neuen Benutzeroberfläche nur korrekt aufgerufen werden.

F27: Auswahl eines Bauteils markiert dieses im 3D Modell:

- Vorhandene Funktionalität anpassen
- Unabhängig von vorhandener Funktionalität neu implementieren

Eine Art der Markierung ist in der Vorgängerversion vorhanden und könnte verwendet werden, damit diese Markierung umgesetzt wird. In dieser Arbeit wurde diese Funktion jedoch zu Gunsten der Qualität anderer Funktionen ausgelassen.

F28: Auswahl eines Teiles im 3D Modell wählt dieses im Menü aus:

- Modifikation von F26 um eine Art "Two-Way-Binding" zu erreichen
- Separate neu Implementierung

Diese Funktion wurde aus denselben Gründen wie F26 in dieser Arbeit nicht weiterverfolgt. Jedoch ist eine Art "Two-Way-Binding" einer Neuimplementation vorzuziehen, da ansonsten auf Grund der Ähnlichkeit zu F26 wahrscheinlich DRY verletzt wird.

F29: Bestätigungsdialog, um Fehleingaben zu verhindern:

- Callback der zu Bestätigenden Aktion mitgeben
- Rückgabewert des Bestätigungsfensters zur Unterscheidung (Boolean)

Callback Funktionen sind sehr vielfältig und verhindern starke Kopplung. Aus diesem Grund wurde beschlossen, dass eine Bestätigung eine Callback Funktion auslöst. Im Gegenteil dazu schliesst ein Ablehnen lediglich das Fenster.

## 4.2.5 Dashboard

Auf dem Dashboard werden die gemessenen Werte als Graphen dargestellt. Diese zeigen den Verlauf über einen gewählten Zeitraum an. Somit lassen sich Abweichungen sehr schnell erkennen.

Pro Sensortyp gibt es einen Tab. Jeder dieser Tabs enthält alle Graphen der Sensoren, welche diesem Typ entsprechen. Wenn ein Graph ausgewählt wird öffnet sich eine Detailansicht mit genaueren Informationen dazu.

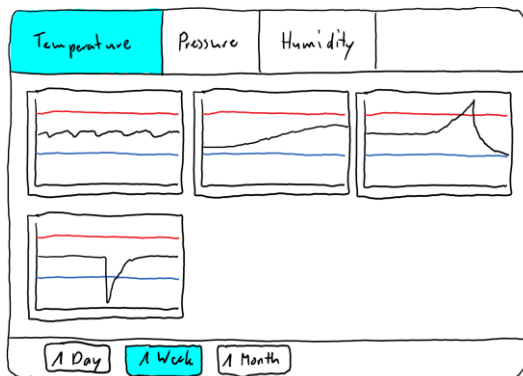


Abbildung 5: Ansicht des Dashboards

### Funktionen:

- F30: Dynamische Tabs nach Sensortypen
- F31: Graphen mit den Messwerten
- F32: Zeitraum wählbar
- F33: Details zu einem Sensor
- F34: Klick auf dem Graphen eines Sensors markiert das Bauteil im 3D Modell und Menü
- F35: Berechnungen von statistischen Werten

### Umsetzungsmöglichkeiten:

F30: Dynamische Tabs nach Sensortypen:

- Erstellen und Löschen eines Tabs sobald entsprechende Sensordaten empfangen werden

Da die Tabs dynamisch erstellt werden sollen, gibt es nur die Möglichkeit, diese nach Bedarf, das heißt nach Anzahl anzuzeigender Sensortypen, zu erstellen und zu löschen. Hierbei übernimmt der TabContainer jegliche Logik wie das Verwalten der Tabs und das Einsortieren der Graphen in die Tabs.

## Umsetzung eines AR/VR Visualisierungskonzepts für Maschinendaten

F31: Graphen mit den Messwerten:

- Verwenden einer Datendarstellungsbibliothek
- Neuimplementation einer Datendarstellung
- Darstellung der Daten auf einer Website und reines Anzeigen der Website in der Applikation

Bedauerlicherweise gibt es keine stabile oder simple Möglichkeit eine Webview, das heisst die Ansicht einer Website, in Unity zu erstellen.

Ebenfalls gibt es keine Datendarstellungsbibliotheken, welche in Unity und mit dem MRTK funktionieren.

Aus diesen Gründen mussten die Graphen neu implementiert werden. Hierbei handelt es sich um einen simplen Graphen, welcher mittels Kreis- und LinienSprites auf einen Canvas gemalt wird.

F32: Zeitraum wählbar:

- Eingabefelder, um jeglichen Zeitraum filterbar zu machen
- Buttons welche vorgegebene Zeiträume filtern

Wie schon in F22 erwähnt, sind die Eingabemöglichkeiten zum Zeitpunkt dieser Arbeit ziemlich beschränkt. Vor allem wenn man den Arbeitsablauf eines Mechanikers bedenkt, fallen klobige Eingabefelder weg.

Deshalb hat die Applikation 3 Knöpfe welche die Daten im Zeitraum eines Jahres, eines Monats und einer Woche entsprechend filtern.

F33: Details zu einem Sensor:

- Darstellung der Details in einem neuen Fenster
- Darstellung der Details im gleichen Fenster

In der aktuellen Form benötigt die Darstellung der Sensoren keine zusätzlichen Details, weshalb diese Funktion nicht umgesetzt wurde.

Sollte eine Detailansicht eines Sensors notwendig werden, ist die Darstellung in einem neuen Fenster zu bevorzugen. Da somit schneller zwischen den verschiedenen Sensoren gewechselt werden kann und der Überblick nicht verloren geht.

F34: Klick auf dem Graphen eines Sensors markiert das Bauteil im 3D Modell und Menü:

- Aufbau auf F26
- Neuimplementation

Es wird empfohlen die Lösung aus F26 entweder zu erweitern oder anzupassen, da dies praktisch der gleiche Anwendungsfall ist.

Wie in F26 aufgeführt, konnte diese Funktion in dieser Arbeit jedoch nicht implementiert werden.

F35: Berechnungen von statistischen Werten:

- Nutzung einer Statistikbibliothek
- Eigene Implementierung verschiedener Statistikfunktionen

Die momentan berechneten Werte beschränken sich auf die Mittel-, Minimal- und Maximalwerte der Sensordaten, weshalb auf eine spezielle Bibliothek verzichtet wurde.

Sollten komplexere Werte berechnet werden, wird empfohlen die Math.NET<sup>10</sup> Bibliotheken zu verwenden.

### 4.2.6 Farben

Bisher werden, bis auf den Hintergrund, keine Farben verwendet. Es gibt keine farblichen Hervorhebungen.

Für ein einheitliches Erscheinungsbild wurden mehrere Farbpaletten ausgearbeitet. Diese kommen vor allem bei den Anzeigen von Sensordaten zum Zug. Für die Hintergrund- und die Schriftfarbe wurde eine Kombination von einem dunklen und einem sehr hellen Grauton gewählt. Ein dunkler Hintergrund mit heller Schrift eignet sich am besten für eine AR Anwendung, da damit der Text vor dunklen und hellen Gegenständen gut erkennbar ist.

Für den Farbenblindmodus wurden schraffierte Farbpaletten erstellt. Dies ist eine der gängigsten Methoden eine Unterscheidung für Farbenblinde zu ermöglichen.

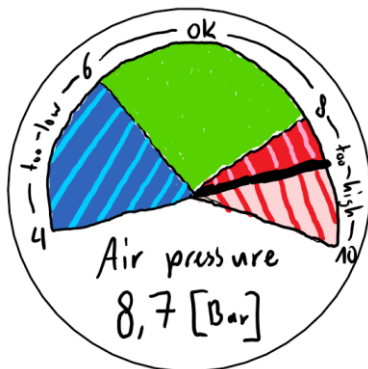


Abbildung 6: Beispiel für den Farbenblindmodus

<sup>10</sup> <https://www.mathdotnet.com/>

## 4.3 Nicht-Funktionale Anforderungen

Die Nicht-Funktionalen Anforderungen lassen sich sehr gut aus einem Katalog ableiten. Ein solcher Katalog ist zum Beispiel FURPS<sup>11</sup>. Wo sinnvoll wird mittels Landing Zones bewertet. Der linke Wert gilt dabei als ausserordentlich gut, der mittlere Wert als gut und der rechte Wert als genügend.

linker Wert	mittlerer Wert	rechter Wert

Für dieses Projekt werden folgende Punkte betrachtet:

### 4.3.1 Funktionalität

#### Umfang

Für den Umfang wird bewertet, wie viele der geplanten Funktionen umgesetzt werden konnten.

#### Wiederverwendbarkeit

Um die Wiederverwendbarkeit zu beurteilen, wird überprüft, wie unabhängig die einzelnen Teile voneinander sind. Dafür wird die Metrik der Cyclomatic Complexity betrachtet. Bei mehr Codezeilen wird auch die Cyclomatic Complexity steigen. Jedoch sollte diese proportional weniger stark zunehmen als die Codezeilen.

#### Sicherheit

Da es in der Arbeit um die Konzeptumsetzung einer Benutzeroberfläche geht, werden für die Sicherheit lediglich benutzeroberflächenspezifische Punkte, wie zum Beispiel Eingaben, berücksichtigt.

### 4.3.2 Benutzbarkeit

#### Menschliche Faktoren

Die menschlichen Faktoren wurden bereits in der vorhergehenden Studienarbeit bei der Erstellung des Konzepts bedacht, weshalb in dieser Arbeit nicht weiter darauf eingegangen wird.

#### Ansprechbarkeit

Für die Ansprechbarkeit wird bewertet, wie schnell die Applikation auf einen Klick reagiert. Um einen genauen Vergleich herzustellen, wurden Landing Zones erstellt.

weniger als 1 Sekunde	weniger als 2 Sekunde	weniger als 5 Sekunde

<sup>11</sup> <https://en.wikipedia.org/wiki/FURPS>

### 4.3.3 Zuverlässigkeit

#### Berechenbarkeit

Zur Bewertung der Berechenbarkeit werden die Ergebnisse des Usability Tests betrachtet. Anhand dessen lässt sich ablesen, wie intuitiv die Bedienung ist.

### 4.3.4 Effizienz

#### Leistung

Die Leistung wird anhand der gemessenen Bilder pro Sekunde bewertet. Die Hololens läuft mit maximal 60 Bildern pro Sekunde. Auch hier bieten sich Landing Zones für einen Vergleich an.

ungefähr 60 FPS	ungefähr 50 FPS	ungefähr 40 FPS

#### Ressourcennutzung

Als Ressource wird in diesem Fall der Speicher betrachtet, welcher während der Laufzeit verwendet wird. Dafür wurden folgende Landing Zones definiert.

weniger als 200 MB	200MB bis 400 MB	400MB bis 600 MB

### 4.3.5 Änderbarkeit

#### Wartbarkeit

Um die Wartbarkeit abschätzen zu können, wird auf den Maintainability Index geschaut. Der Maintainability Index der Vorarbeit wird als Referenz verwendet. Das Ziel ist es, den Wert nicht zu verschlechtern. Für einen Vergleich wurden wiederum Landing Zones erstellt.

mindestens 85	80 - 85	75 - 80

#### Testbarkeit

Laut der vorhergehenden Bachelorarbeit lässt sich Testing nicht umsetzen, weshalb dies auch nicht angegangen wurde. (Bachelorarbeit "Visualisierung von Maschinendaten einer Industriemaschine in der erweiterten Realität", Abschnitt 6.2.9 Testing)

**Flexibilität**

Um die Flexibilität zu bewerten, wird das Class Coupling erfasst. Dieser Wert sagt aus, wie stark die Klassen miteinander verknüpft sind. Ist der Wert klein, so können diese unabhängig voneinander verwendet werden. Das Class Coupling lässt sich gut mit Landing Zones vergleichen.

450 - 500	500 - 550	mehr als 550

## 5 Technologien

### 5.1 Hololens

Die Hololens ist Teil der ersten Generation von Mixed Reality Brillen. Mit ihr lassen sich die echte Welt und computergenerierte Objekte kombinieren.

In dieser Arbeit wird dies für die Darstellung von Informationen und Modelle von Maschinen genutzt.

### 5.2 Hololens 2

Die Hololens 2 ist die Folgegeneration der Hololens mit zusätzlicher Funktionalität. Diese umfasst unter anderem Eye-Tracking und direktes Interagieren mit Objekten durch verbesserte Handerkennung.

Zum Zeitpunkt dieser Arbeit ist die Hololens 2 noch nicht durch die HSR verfügbar. Es wurde jedoch Wert daraufgelegt, dass die ganze App einfach zu portieren ist.

### 5.3 Unity

Unity wird von Microsoft als Entwicklungsumgebung für Mixed Reality Anwendungen empfohlen. Vorwiegend wird es für das Programmieren von 2D und 3D Spielen genutzt. Man kann damit auch sehr gut Virtual, Augmented und Mixed Reality Applikationen erstellen.

### 5.4 Visual Studio

Unity nutzt primär C#, welches eine Programmiersprache ist, die von Microsoft entwickelt wird. Visual Studio ist eine Entwicklungsumgebung für C# und wird ebenfalls von Microsoft entwickelt. Ausserdem hat Visual Studio offizielle Plugins zur Integrierung von Unity.

### 5.5 Windows 10

Windows 10 ist eine Voraussetzung für die Entwicklung von Hololens Apps, da die Windows SDK von Windows 10 benötigt wird.

### 5.6 Vuforia

Vuforia<sup>12</sup> ist eine weit verbreitete, gut unterstützte Bild- und Formerkennungssoftware, welche für das Platzieren von Maschinenmodellen auf die richtige Maschine verwendet wird.

### 5.7 Mixed Reality Toolkit

MRTK<sup>13</sup> ist das offiziell von Microsoft entwickelte Framework zur Entwicklung von Mixed Reality Applikationen. Es ist ebenfalls auf Unity ausgelegt.

---

<sup>12</sup> <https://www.ptc.com/de/products/augmented-reality/vuforia>

<sup>13</sup> <https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/GettingStartedWithTheMRTK.html>

## 5.8 Build Anleitung

Das Buiden der Applikation ist komplex. Es gibt mehrere versteckte Optionen und man muss die App zweimal in unterschiedlichen IDEs builden. Der erste Build wird in Unity durchgeführt und der Output dieses Builds wird dann verwendet um die App in Visual Studio erneut zu builden.

Aus diesem Grund existiert eine Anleitung, welche in dem Dokument "Erweiterte\_Dokumentation\_Machine\_Visualization.pdf" zu finden ist. Die Anleitung wurde von Mike Schmid verfasst.

## 6 Architektur

### 6.1 Navigation

Die Fenster sind so angeordnet, dass man, wenn sie angezeigt werden, lediglich den Kopf drehen muss, um von einem Fenster zum nächsten zu navigieren. Um ein ausgeblendetes Fenster oder die Details eines anderen Objektes anzuzeigen genügt es, eben jenes Objekt auszuwählen und dann den Kopf zu drehen.

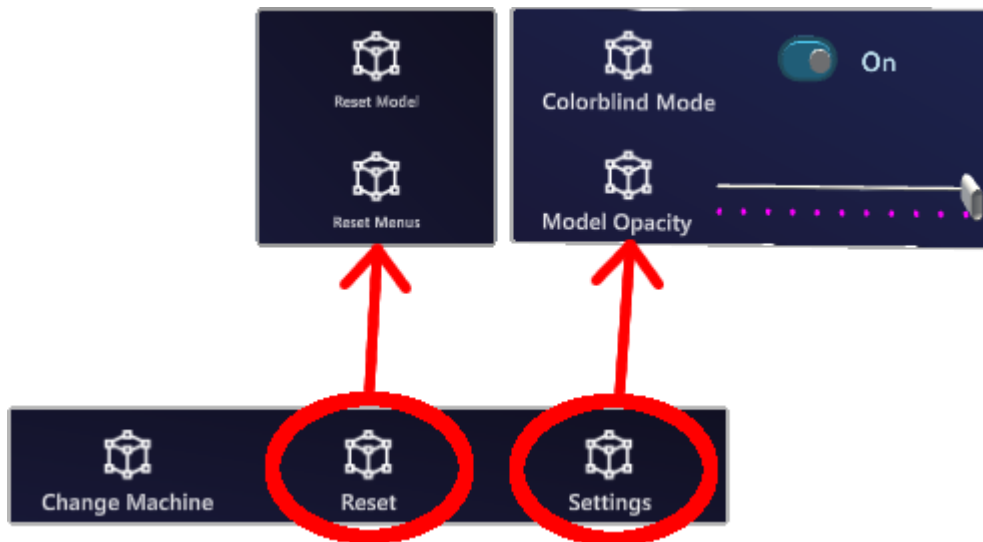


Abbildung 7: Navigationsmöglichkeiten des Hauptmenüs

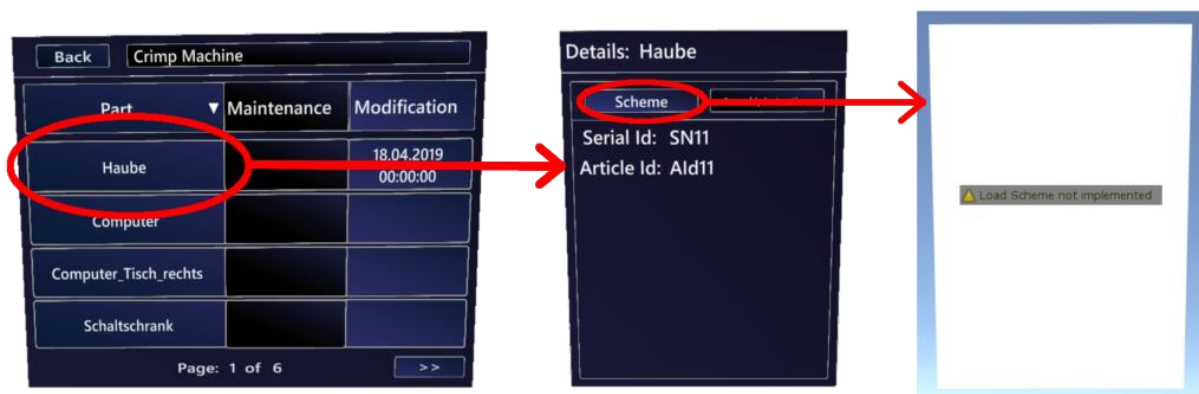


Abbildung 8: Zusammenhänge der Fenster des Maschinenteilemenüs

# Umsetzung eines AR/VR Visualisierungskonzepts für Maschinendaten

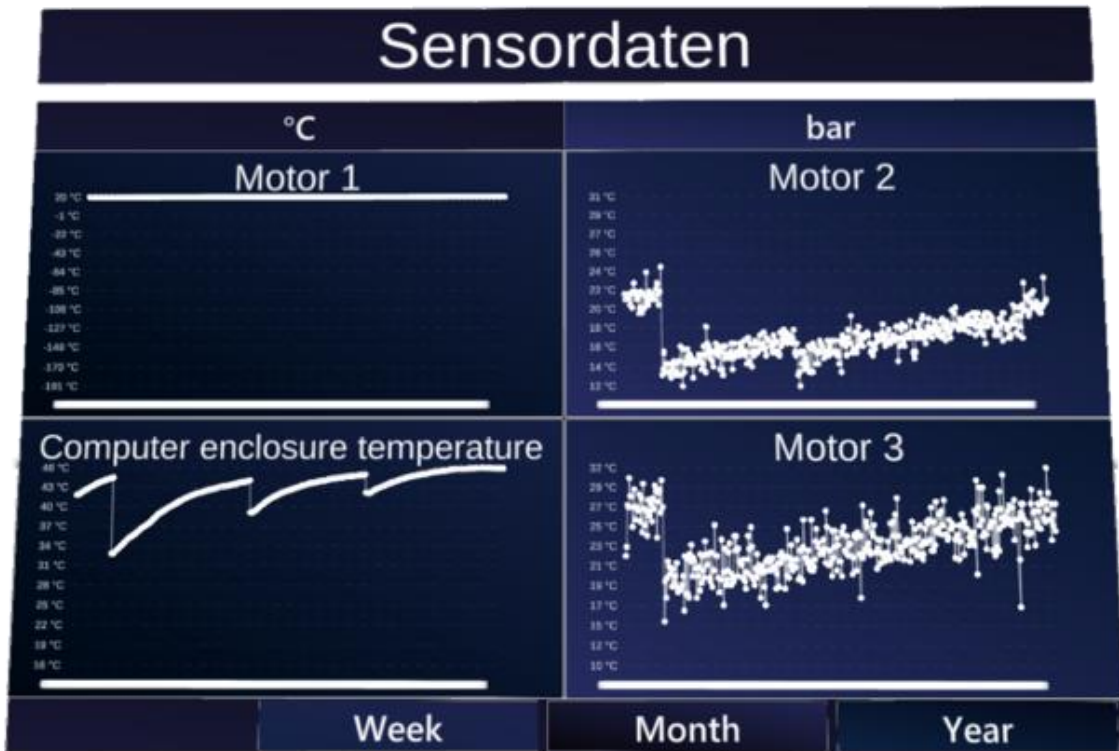


Abbildung 9: Das Dashboard interagiert mit keinem anderen Fenster

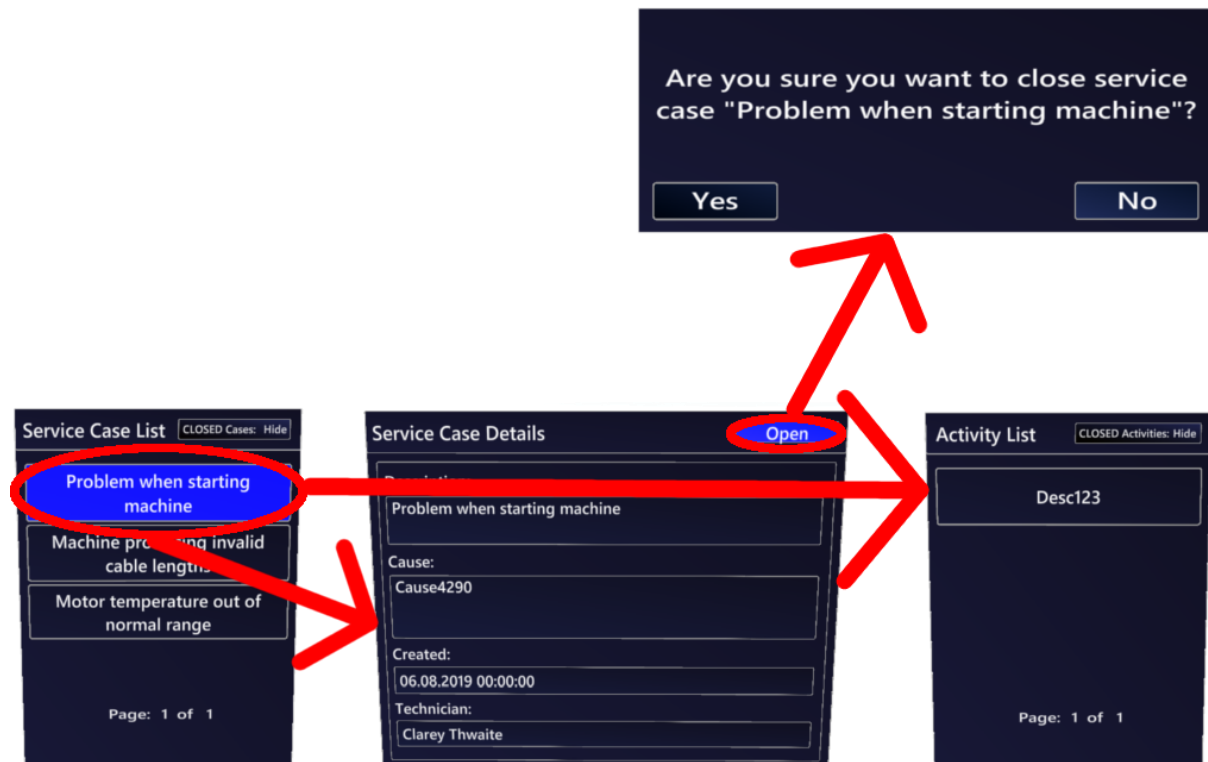


Abbildung 10: Zusammenhänge der Fenster des Aufgabenmenüs

Alle Fenster werden beim Start der Applikation geladen. Die Detailfenster sind noch deaktiviert, das bedeutet, dass sie unsichtbar sind und mit ihnen nicht interagiert werden kann. Jedes Detailfenster wird dann aktiviert, wenn das zu den Details gehörende Objekt angewählt wird.

Ausserdem gibt es eine klare Trennung zwischen den Fenstern auf der linken und der rechten Seite des Benutzers, wenn dieser die Maschine betrachtet. Links des Benutzers befindet sich das Dashboard mit allen Graphen der Sensordaten und mögliche assoziierte Fenster. Rechts des Benutzers befinden sich die Listen der Aufgaben und Bauteile sowie die dazugehörigen Detailfenster.

## 6.2 Unity Szenen Struktur

Unity Szenen sind nur schwer mit Version Control vereinbar. Szenen sind riesige XML-Dateien, welche sich bei jeder Änderung an der Szene im Unity Editor ebenfalls ändern. Jedoch können die meisten Merge Tools diese Änderungen nicht nachvollziehen, was zu sehr vielen Mergekonflikten führt. Dies hat wiederum zur Folge, dass es praktisch unmöglich ist, mehrere Programmierer an der gleichen Szene arbeiten zu lassen.

Es gibt jedoch einen Workaround. Werden Prefabs, welche in der Szene existieren, verändert, dann bleibt die Szene unberührt.

Deshalb sind die Funktionen dieser Arbeit als Container in die Szene eingepflegt. Diese Container sind Prefabs leerer Gameobjekte, an welchen die eigentlichen Gameobjekte der Funktionen angehängt wurden.

Somit konnte Version Control ohne Bedenken genutzt werden. Als Nebeneffekt führt dies ausserdem zu mehr Modularität.

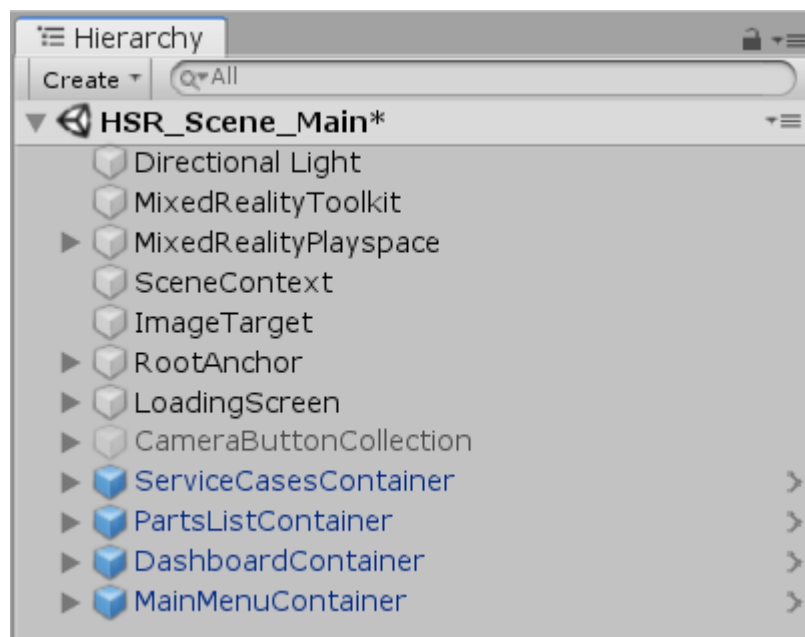


Abbildung 11: Aufbau der Szene in Unity

## 7 Ergebnis Umsetzung

Für die Umsetzungsbewertung werden Landing Zones verwendet. Hierfür sind drei Stufen (Bronze, Silber und Gold) definiert und jede Funktion wird danach bewertet. Abschliessend wird für jede Gruppe an Funktionen, zum Beispiel das Hauptmenü, eine Tabelle erstellt, welche die Bewertungen zusammenfasst. Somit lässt sich für jedes Menü erkennen, wie ausgereift dieses ist.

Die Landing Zones sind folgendermassen definiert:

- Bronze: Minimalanforderungen, damit die Funktion keine Fehler verursacht
- Silber: Stand der von dem Team erreicht werden will
- Gold: Funktion ist ausserordentlich umgesetzt und ausgereift

### 7.1 Funktionen

#### 7.1.1 Hauptmenü

Funktion	Beschreibung	Status
F01	Dashboard anzeigen	Ersetzt/Bronze
Kommentar		
<p>Es wurde entschieden, dass das Dashboard immer angezeigt wird. Die Funktionalität des Buttons wurde geändert, damit die Maschine gewechselt werden kann. Hierfür soll ein QR Code gelesen werden, welcher auf dem Cockpit angezeigt werden könnte. Somit ist die Instanz ID einfach auf die Brille übertragbar.</p> <p>Für einen QR Reader, welcher mit der Hololens funktioniert, wurde dieses Beispiel<sup>14</sup> gefunden: Aufbauend auf diesem wurde ein Testprojekt erstellt, welches erfolgreich getestet wurde. Bisher war die Instanz ID im Code fix und konnte nicht geändert werden. Dies wurde angepasst, so dass die Instance ID in den PlayerPrefs persistiert wird und beim Start der Applikation geladen wird.</p>		

Funktion	Beschreibung	Status
F02	Reset des Modells	Nicht umgesetzt
Kommentar		
<p>Das Modell wird im aktuellen Stand nicht interaktiv verändert, womit dieser Menüpunkt nicht im Fokus stand und nicht umgesetzt wurde.</p>		

<sup>14</sup> <https://unitylist.com/p/jk6/Holo-Lens-QR-Reader>

## Umsetzung eines AR/VR Visualisierungskonzepts für Maschinendaten

Funktion	Beschreibung	Status
F04	Farbenblindmodus Ein-/Ausschalten	nicht benötigt
Kommentar		
Der gesetzte Wert wird nicht verwendet, da die Anzeigen nicht umgesetzt wurden. (ref.)		

Funktion	Beschreibung	Status
F05	Modell Transparenz verändern	nicht umgesetzt
Kommentar		
Ähnlich wie der Farbenblindmodus wird auch dieser Wert nicht verwendet. Weil eine Modifikation des Modells nicht im Fokus stand, wurde dies nicht weiterverfolgt.		

Funktion	Beschreibung	Status
F06	Persistenz der Einstellungen	Gold
Kommentar		
Die Einstellungen werden in den PlayerPrefs persistiert. Beim Start der Applikation werden die Werte ausgelesen und in der Benutzeroberfläche entsprechend angezeigt.		

Funktion	Beschreibung	Status
F07	Fixe Positionierung im Blickfeld	Gold
Kommentar		
<p>Das neue Hauptmenü ist immer direkt im Blickfeld positioniert und verschwindet nicht. Es hat sich jedoch herausgestellt, dass dies nicht mit der Hololens 1 kompatibel ist, da bei dieser nur mit Objekten in der Mitte vom Bildschirm interagiert werden kann.</p> <p>Die Position wurde so angepasst, dass das Menü immer vor dem Benutzer liegt, jedoch nicht dem Blick nach oben und unten folgt.</p>		

Zusammenfassung für das Hauptmenü:

Funktion	Bronze	Silber	Gold
F01: Dashboard anzeigen	X		
F02: Reset des Modells			
F04: Farbenblindmodus Ein-/Ausschalten			
F05: Modell Transparenz verändern			
F06: Persistenz der Einstellungen			X
F07: Fixe Positionierung im Blickfeld			X

Beim Hauptmenü wurden nur 50% der Funktionen umgesetzt. Die Qualität der Umsetzungen ist jedoch gut gelungen und die weggefallenen Funktionen sind Folgen davon, dass die generischen Anzeigen weggelassen wurden oder die Priorität gering war.

### 7.1.2 Generische Anzeigen

- F08: Wertanzeige
- F09: Dynamische Einheiten
- F10: Dynamische Min. und Max.
- F11: Farbwechsel zwischen über und unter dem Messwert
- F12: Dynamische Anzeige des aktuellen Messwertes
- F13: Ändert Farbpalette nach Ein-/Ausstellung des Farbenblindmodus

Funktion	Beschreibung	Status
F08-F13	Diverse bezüglich generischer Anzeige	Nicht umgesetzt
Kommentar		
<p>Nicht umgesetzt aus folgenden Gründen:</p> <ul style="list-style-type: none"> <li>- Wenn für jeden Sensor eine so detaillierte Anzeige im Raum steht, welche recht gross sein muss, damit sie lesbar ist, wird es schnell unübersichtlich.</li> <li>- Der zusätzliche Informationsgehalt ist sehr niedrig, da nur der aktuelle Messwert abzulesen ist. Im Vergleich dazu sind auf dem Dashboard viel mehr Informationen auffindbar.</li> </ul>		

### 7.1.3 Aufgaben Menü

Funktion	Beschreibung	Status
F15	Abgeschlossene Aufgaben ein- und ausblenden	Gold
Kommentar		
Die Liste der Aufgaben kann gefiltert werden, womit ein Umstellen einfach möglich ist.		

Funktion	Beschreibung	Status
F18	Anzeige der Details für einen Auftrag	Silber
Kommentar		
Sobald eine Aufgabe gewählt wird, öffnet sich ein Detailfenster mit zusätzlichen Informationen zu dieser Aufgabe.		

Funktion	Beschreibung	Status
F19	Status veränderbar auf "Done"	Silber
Kommentar		
Da diese Funktionalität nicht im Fokus stand und vom Backend nicht unterstützt wird, wurde entschieden, dass diese abgeändert wird und der Status auf "Closed" geändert werden kann.		

Zusammenfassung für das Aufgaben Menü:

Funktion	Bronze	Silber	Gold
F15: Abgeschlossene Aufgaben ein- und ausblenden			X
F18: Anzeige der Details für einen Auftrag		X	
F19: Status veränderbar auf "Done"		X	

Alle Funktionen des Aufgaben Menüs wurden umgesetzt. F19 wurde etwas angepasst und die Umsetzung ist gut gelungen.

### 7.1.4 Maschinenteile Menü

Funktion	Beschreibung	Status
F20	Maschinenteile anzeigen	Silber
Kommentar		
Die Baugruppen und Bauteile der Maschine werden angezeigt und sind einfach navigierbar. In der Liste ist der Name des Bauteils und die letzten Daten für die Wartung und Modifikation enthalten.		

Funktion	Beschreibung	Status
F21	Sortierung der Bauteile	Gold
Kommentar		
Es ist möglich die Bauteile nach Seriennummer, Datum der letzten Wartung und Datum der letzten Modifikation zu sortieren. Falls eine Sortierung ein zweites Mal angewählt wird, dann wird die Liste invertiert. Die aktuelle Sortierung ist mit einem Pfeil markiert.		

Funktion	Beschreibung	Status
F23	Suche von Bauteilen	nicht umgesetzt
Kommentar		
Die Hololens Tastatur ist nicht mit dem Projekt kompatibel und Spracheingabe ist, basierend auf der Studienarbeit, für diese Anwendung untauglich. Aus diesen Gründen wurde beschlossen, dass auf die Suche verzichtet wird.		

Funktion	Beschreibung	Status
F24	Angabe des Pfades	Gold
Kommentar		
Der Pfad wird ausgehend vom ausgewählten Bauteil berechnet und angezeigt. Aus Platzgründen werden maximal 2 obere Baugruppen angezeigt. Falls es diese nicht gibt, wird der Name der Instanz (die Maschine) verwendet.		

## Umsetzung eines AR/VR Visualisierungskonzepts für Maschinendaten

Funktion	Beschreibung	Status
F25	Anzeige von Details für Baugruppen und Bauteilen	Silber
Kommentar		
<p>Details für Baugruppen und Bauteile werden angezeigt, sobald diese ausgewählt werden. Die zusätzlichen Details sind beschränkt auf die Serien- und Artikelnummern. Falls noch weitere Informationen, zum Beispiel zu physischen Eigenschaften, vom Backend geladen werden, ist eine Erweiterung nötig, um diese dynamisch hinzuzufügen.</p>		

Funktion	Beschreibung	Status
F26	Bauteile können ersetzt und gewartet werden	Silber
Kommentar		
<p>Sowohl Bauteil als auch Baugruppen können ersetzt und gewartet werden. Bei einem Klick auf den entsprechenden Knopf wird ein Informationsfenster geöffnet falls noch kein Auftrag gewählt wurde. Ist bereits ein Auftrag ausgewählt, so wird ein Bestätigungsfenster angezeigt, um Falscheingaben zu verhindern.</p> <p>Das Backend liefert vor allem bei Modifikationen an Baugruppen fehlerhafte Rückmeldungen an die Hololens. Auf dem dazugehörigen Cockpit werden die Daten korrekt aktualisiert. Sobald die Antworten bzw. das Backend überarbeitet worden sind, wird auch die Anzeige auf der Hololens die richtigen Informationen anzeigen.</p> <p>Der Fokus dieser Arbeit lag auf der Benutzeroberfläche, weshalb auf einen Eingriff in das Backend verzichtet wurde.</p>		

Funktion	Beschreibung	Status
F27	Auswahl eines Bauteils markiert dieses im 3D Modell	Bronze
Kommentar		
<p>Funktioniert im Unity Editor, jedoch nicht auf der Hololens. Auf der Hololens ist stets das ganze Modell markiert und das markierte Bauteil kann nicht ausgemacht werden.</p>		

## Umsetzung eines AR/VR Visualisierungskonzepts für Maschinendaten

Funktion	Beschreibung	Status
F28	Auswahl eines Teiles im 3D Modell wählt dieses im Menü aus	nicht umgesetzt
Kommentar		
Die Interaktion mit dem 3D Modell ist sehr komplex und der Fokus lag mehr auf den Fenstern, als auf dem Modell. Aus diesem Grund wurde diese Funktion im Rahmen dieser Arbeit nicht umgesetzt.		

### Zusammenfassung Maschinenteile Menü:

Funktion	Bronze	Silber	Gold
F20: Maschinenteile anzeigen		X	
F21: Sortierung der Bauteile			X
F23: Suche von Bauteilen			
F24: Angabe des Pfades			X
F25: Anzeige von Details für Baugruppen und Bauteilen		X	
F26: Bauteile können ersetzt und gewartet werden		X	
F27: Auswahl eines Bauteils markiert dieses im 3D Modell	X		
F28: Auswahl eines Teiles im 3D Modell wählt dieses im Menü aus			

Für das Maschinenteile Menü wurden die meisten Funktionen umgesetzt. Auf eine Suche musste verzichtet werden, da diese nicht mit dem Projekt kompatibel ist. F27 und F28 wurden weniger gewichtet, als die anderen und hatten aus diesem Grund auch am wenigsten Fortschritt.

### 7.1.5 Dashboard

Funktion	Beschreibung	Status
F30	Dynamische Tabs nach Sensortypen	Silber
Kommentar		
Graphen unterschiedlicher Sensortypen werden in separaten Tabs dargestellt. Die Darstellung kann potenziell verbessert werden.		

## Umsetzung eines AR/VR Visualisierungskonzepts für Maschinendaten

Funktion	Beschreibung	Status
F31	Graphen mit den Messwerten	Silber
Kommentar		
<p>Die Messwerte werden in Graphen dynamisch dargestellt. Der Ausschnitt des Graphen wird immer optimal gewählt, so dass alle Datenpunkte sichtbar sind. Jedoch kann die Interpolation der Datenpunkte erheblich verbessert werden. Besonders da es sein kann, dass ein Sensor mehrere Datenpunkte pro Minute sammelt und speichert.</p>		

Funktion	Beschreibung	Status
F32	Zeitraum wählbar	Silber
Kommentar		
<p>Der Zeitraum der angezeigten Datenpunkte ist wählbar. Zur Auswahl stehen: Diese Woche, dieser Monat und dieses Jahr. Ob eine exaktere Auswahl sinnvoll wäre, müsste bei der Weiterführung dieser Arbeit abgewägt werden.</p>		

Funktion	Beschreibung	Status
F33	Details zu einem Sensor	Nicht umgesetzt
Kommentar		
<p>Die Sensorübersicht bietet genügend Platz für alle wichtigen Sensormetriken, deshalb wurde entschieden diese Funktion nicht umzusetzen.</p>		

Funktion	Beschreibung	Status
F34	Klick auf dem Graphen eines Sensors markiert das Bauteil im 3D Modell und Menü	Nicht umgesetzt
Kommentar		
<p>Die Verlinkung eines Bauteils mit einem anderen Gameobjekt ist schwieriger als zunächst angenommen. Der Fokus lag auf der Benutzeroberfläche, weshalb diese Funktion nicht umgesetzt wurde.</p>		

## Umsetzung eines AR/VR Visualisierungskonzepts für Maschinendaten

Funktion	Beschreibung	Status
F35	Berechnungen von statistischen Werten	Bronze
Kommentar		
Statistische Werte, wie die der durchschnittliche Sensorwert oder der maximale und minimale Sensorwert dieses Intervalls, werden in der Sensorübersicht dargestellt.		

### Zusammenfassung Dashboard:

Funktion	Bronze	Silber	Gold
F30: Dynamische Tabs nach Sensortypen		X	
F31: Graphen mit den Messwerten		X	
F32: Zeitraum wählbar		X	
F33: Details zu einem Sensor			
F34: Klick auf dem Graphen eines Sensors markiert das Bauteil im 3D Modell und Menü			
F35: Berechnungen von statistischen Werten	X		

Das Dashboard wurde grösstenteils nach Plan umgesetzt. Ein Detailfenster wurde mangels Notwendigkeit nicht umgesetzt. Ebenfalls wurde eine Verbindung zum 3D Modell nicht umgesetzt, da der Fokus auf der Benutzeroberfläche lag.

### 7.1.6 Allgemein

Funktion	Beschreibung	Status
F03	Reset der Menüs	Silber
Kommentar		
Die Menüs können zurückgesetzt werden. Damit verschwinden die Detailfenster und ausgewählte Aufgaben oder Bauteile werden abgewählt. Die Bauteile der obersten Stufe werden wieder angezeigt. Ebenfalls werden die abgeschlossenen Aufgaben wieder ausgeblendet und die Sortierung der Bauteile auf den Standard gesetzt, welcher nach der Seriennummer sortiert.		

## Umsetzung eines AR/VR Visualisierungskonzepts für Maschinendaten

Funktion	Beschreibung	Status
F14	Positionierung der Fenster	Gold
Kommentar		
<p>Für die Positionierung der Fenster wurde ein eigener Solver geschrieben. Es können mehrere Einstellungen vorgenommen werden, um die Position zu bestimmen. Die Positionen werden berechnet ausgehend von den Schnittpunkten zwischen einem Kreis um die Maschine, welcher durch den Benutzer geht, und einem Kreis um den Benutzer mit einem modifizierbaren Radius. Diese Anpassungen können unabhängig voneinander in Unity angepasst werden. Die Parameter sind gut beschriftet und mit einem Tooltip versehen, in dem genau beschrieben wird, was dieser Parameter verändert.</p> <p>Es wäre auch denkbar, dass diese Parameter vom Benutzer während der Laufzeit verändert werden können.</p>		

Funktion	Beschreibung	Status
F16	Scrollbare Liste	Ersetzt/Silber
Kommentar		
<p>Die scrollbare Liste ist auf der Hololens experimentell und nicht genügend ausgereift. Es wurde beschlossen, dass Sie durch eine seitenbasierte Auflistung ersetzt wird, welche vollständig umgesetzt wurde.</p>		

Funktion	Beschreibung	Status
F17	Pfeilindikator, welcher andeutet, dass die Liste scrollbar ist	Nicht nötig
Kommentar		
<p>Da die scrollbaren Listen durch eine seitenbasierte Liste ersetzt wurden, sind diese nicht mehr notwendig.</p>		

Funktion	Beschreibung	Status
F22	Farbliche Hervorhebung	Silber
Kommentar		
<p>Bei den Baugruppen gibt es leichte Abweichungen vom Konzept. Der Hintergrund von Baugruppen wird in der Liste nicht hervorgehoben, damit zwischen einem gewählten Bauteil und einer Baugruppe unterschieden werden kann.</p>		

## Umsetzung eines AR/VR Visualisierungskonzepts für Maschinendaten

Funktion	Beschreibung	Status
F29	Bestätigungsdialog, um Fehleingaben zu verhindern	Silber
Kommentar		
Es gibt ein Prefab, welches als Bestätigungsfenster verwendet wird. Ein Aufruf mit einer Callback Funktion führt dazu, dass bei einem Ablehnen nur das Fenster geschlossen wird und bei einer Bestätigung die Callback Funktion ausgeführt wird.		

### Zusammenfassung Allgemein:

Funktion	Bronze	Silber	Gold
F03: Reset der Menüs		X	
F14: Positionierung der Fenster			X
F16: Scrollbare Liste		X	
F17: Pfeilindikator, welcher andeutet, dass die Liste scrollbar ist			
F22: Farbliche Hervorhebung		X	
F29: Bestätigungsdialog, um Fehleingaben zu verhindern		X	

Die menüübergreifenden Funktionen wurden alle umgesetzt. Da die scrollbaren Listen ersetzt wurden, ist F17 weggefallen.

## 7.2 Nicht funktionale Anforderungen

### 7.2.1 Funktionalität

#### Umfang

Total Anzahl Funktionen: 35

Nach Abzug von begründet nicht umgesetzten Funktionen: 24

Umgesetzte Funktionen: 21

Quotient:  $21 / 24 = 0.875$

Anhand der berechneten Zahlen lässt sich erkennen, dass 87.5% der geplanten Funktionen umgesetzt wurden. Die verbleibenden Funktionen beziehen sich mehrheitlich auf die Interaktion mit dem 3D Modell. Diese wurden als weniger wichtig eingestuft im Vergleich zu den diversen Menüs, und konnten im Rahmen des Projektes leider nicht umgesetzt werden.

#### Wiederverwendbarkeit

Analyse der Cyclomatic Complexity in Visual Studio:

Referenzwert der Vorarbeit: 1280

Neuer Wert nach dieser Arbeit: 1649

Verhältnis der Cyclomatic Complexity:  $1649 / 1280 = 1.29$

Verhältnis der Codezeilen:  $9202 / 6737 = 1.37$

Verhältnis der ausführbaren Codezeilen:  $2706 / 1929 = 1.40$

Das Verhältnis der Cyclomatic Complexity ist kleiner als das Verhältnis der Codezeilen. Somit wurde mehr Funktionalität als Komplexität hinzugefügt. Dieses Ergebnis wird als positiv betrachtet.

#### Sicherheit

Es können nur Gesteneingaben gemacht werden, welche von der Hololens erkannt werden, um die Applikation zu steuern. Somit ist keine Manipulation der App ausserhalb dieser Grenzen möglich. Da diese Erkennung von den Herstellern der Hololens umgesetzt wurde, liegt die Verantwortung für Sicherheitslücken bei Microsoft.

### 7.2.2 Benutzbarkeit

#### Ansprechbarkeit

weniger als 1 Sekunde	weniger als 2 Sekunde	weniger als 5 Sekunde
X		

Die Reaktionszeit ist sehr schnell und selbst neue Fenster erscheinen ohne wahrnehmbare Verzögerung.

### 7.2.3 Zuverlässigkeit

#### Berechenbarkeit

Während dem Usability Test sind 11 Entscheidungen zu treffen. Diese sind zum Beispiel ein Klick auf ein Menüpunkt oder ein Umschauen, damit ein Fenster in das Blickfeld rückt. (siehe auch ref. Testszenerien)

Für eine Abschätzung der Berechenbarkeit wird bewertet, wie viele der Entscheidungen korrekt getroffen wurden.

Feedback 1:

Umschauen = 0 von 2

Menüführung = 8 von 9

Feedback 2 (nicht vollständig ausgefüllt):

Umschauen = 0 von 1

Menüführung = 1 von 2

Von den Entscheidungen, welche mit Umschauen im Zusammenhang stehen, wurde keine korrekt beantwortet. Der Hauptgrund dafür ist, dass der Test auf Papier durchgeführt wurde und nicht mithilfe einer HoloLens. (siehe auch ref. Testszenerien)

Die Menüführung wurde im Feedback 1 zu 89% korrekt ausgefüllt. Dieses Ergebnis wird als sehr positiv empfunden, weil es eine unbekannte Benutzeroberfläche ist, welche dennoch von Servicetechniker ohne Einführung intuitiv bedient werden kann.

### 7.2.4 Effizienz

#### Leistung

Für eine Messung der Bilder pro Sekunde wurde der Visual Profiler<sup>15</sup> des MRTK verwendet.

ungefähr 60 FPS	ungefähr 50 FPS	ungefähr 40 FPS
		X

Die Applikation hat nur etwa 40 Frames pro Sekunde. Dies ist zwar weit unter den gewünschten 60 FPS, jedoch hat sich gezeigt, dass sich das Benutzererlebnis nicht merklich verschlechtert. Dies liegt primär an den fehlenden Animationen. Sollten solche eingefügt werden wird empfohlen den UI Code nochmals zu optimieren.

<sup>15</sup> <https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/Diagnostics/UsingVisualProfiler.html>

### Ressourcennutzung

Um die Speichernutzung zu messen wurde ebenfalls der Visual Profiler verwendet.

weniger als 300 MB	300 MB - 400 MB	400 MB - 500 MB
		X

Die Arbeitsspeicherauslastung ist höher als erwartet. Der Grund für dieses Verhalten konnte jedoch nicht ermittelt werden.

### 7.2.5 Änderbarkeit

#### Wartbarkeit

Analyse des Maintainability Indexes in Visual Studio:

Ausgangswert der Vorarbeit: 85

Neuer Wert nach dieser Arbeit: 84

mindestens 85	80 - 85	75 - 80
	X	

Der Index wurde minimal verringert, was ein gutes Ergebnis ist.

#### Flexibilität

Analyse des Class Couplings in Visual Studio:

Ausgangswert der Vorarbeit: 454

Neuer Wert nach dieser Arbeit: 545

450 - 500	500 - 550	mehr als 550
	X	

Das Class Coupling wurde etwas verschlechtert, welches jedoch noch im Rahmen liegt.

## 8 Testszenarios

Es wurde ein Testplan erstellt, mit welchem eine Gruppe von Service Technikern mehrere Szenarien durchgegangen sind und jeweils entscheiden mussten, was sie als nächstes unternehmen.

Eine Simulation vor Ort auf der Hololens war nicht möglich. Deshalb ist der Testplan als PDF-Dokument mit vielen Bildern von der Anwendung aufgebaut.

Bevor mit der Ausarbeitung eines Usability Test angefangen wird, ist es empfehlenswert festzuhalten, was überprüft werden soll. Zu diesem Zweck wurden folgende Punkte definiert:

- Finden die Testkandidaten...
  - ...ihre Aufgabe?
  - ...das Dashboard?
  - ...die Einstellungen?
  - ...Informationen über ein Bauteil/eine Baugruppe?
  - ...eine Anleitung für den Einbau eines Bauteiles?
- Können die Testkandidaten ein Bauteil...
  - ...ersetzen?
  - ...warten?

Aufgrund dieser Fragen wurden die Szenarien für den Usability Test entworfen. Um die Anweisungen für die Kandidaten nicht zu eindeutig zu machen, sollen nicht die gleichen Begriffe der Anwendung für die Szenarien verwendet werden. (siehe externes Dokument "Testszenarios.pdf")

### 8.1 Aufbau und Durchführung

Da nur wenige Personen mit einer Hololens vertraut sind, wurden die Möglichkeiten und Funktionen kurz erläutert. Nach dieser Einführung wird der Kontext für diesen Test erfasst. Der Servicetechniker hat bereits alle Daten auf die Hololens geladen und startet diese.

Es folgt eine Reihe von verschiedenen Szenarien, welche einen realistischen Ablauf aufzeigen. Zunächst soll eine Einstellung verändert werden. Danach muss der Techniker seine zugewiesene Aufgabe finden. Daraufhin soll der Techniker Informationen zu den von den Sensoren gemessenen Werte anschauen, damit er eine Idee kriegt, wo der Fehler liegen könnte. Anschliessend muss das betroffene Bauteil gesucht werden, welches dann ersetzt werden soll. Hierfür soll noch eine Hilfe auffindbar sein, wie zum Beispiel eine Montageanleitung. Zum Schluss soll noch ein weiteres Bauteil gewartet werden, da dieses von der Störung tangiert wurde.

Die Durchführung wurde von unserem Kontakt bei Komax in die Wege geleitet. Jeder Service Techniker muss das Dokument ohne Hilfestellungen oder Erklärungen durcharbeiten und für jeden Schritt eine Entscheidung treffen. Bei einem Verfehlen der richtigen Wahl, konnte der Testplan weitergeführt werden, da die korrekte Wahl anschliessend erläutert wird und die Folgen davon als Ausgangslage für die nächste Entscheidung gilt.

## 8.2 Feedback und Einfluss

Aufgrund des geringen Rücklaufs ist nur bedingt Feedback vorhanden. Dies ist auf die ausserordentliche Lage zur Zeit dieser Arbeit zurückzuführen.

Die Entscheidungen wurden bewertet, ob sie mit dem erwarteten Verhalten übereinstimmen. So wurde schnell erkannt, ob die Benutzung für einen Servicetechniker intuitiv ist.

Der Versuch eine AR Applikation per Usability Test auf Papier zu überprüfen, hatte einige Schwachstellen. Zum Beispiel hat sich niemand im Raum umgeschaut, obwohl bei einem Test mit einer Hololens dies die meisten ausprobieren würden.

Die Bedienung der Menüs war für alle verständlich. Hingegen haben nur wenige die Fenster von selbst gefunden. Dies ist zurückzuführen auf die bereits angesprochene Eingeschränktheit auf Papier.

Eine Lösung für dieses Problem wäre, dass am Rand des Bildschirms kleine Pfeile angezeigt werden, welche in die Richtungen der verschiedenen Menüs deuten, solange diese ausserhalb des Blickfeldes liegen. Diese Funktion<sup>16</sup> wird auch vom MRTK angeboten, jedoch ist es noch ein Experimental Feature.

Ein anderer Punkt ist, dass alle auf den "Dashboard" Knopf gedrückt haben, als sie die Aufgaben suchen sollten. Das Dashboard beinhaltet jedoch nur die Sensordaten. Eine treffende Bezeichnung, wie zum Beispiel "Sensor Data", würde hier Abhilfe schaffen.

Viele wollten Bauteile auf dem 3D Modell der Maschine anklicken. Dies soll auch möglich sein, wurde jedoch im Testplan nicht angesprochen, weil es nicht im Fokus stand.

---

<sup>16</sup> [https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/README\\_Solver.html#directional-indicator](https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/README_Solver.html#directional-indicator)

## 9 Ausblick

Einige Funktionen konnten in dieser Arbeit nicht umgesetzt werden bzw. waren von Anfang an ausserhalb des Scopes.

### 9.1 Hololens 2

Die Portierung zur Hololens 2 sollte relativ einfach gelingen. Für eine zukünftige Arbeit sollte diese dann auch zur Verfügung stehen.

Mit einer Hololens 2 würden zusätzliche Funktionen<sup>17</sup>, wie zum Beispiel Eye-Tracking oder direktes Interagieren mit virtuellen Objekten, ermöglicht werden.

### 9.2 QR Code reading for machine change

Mittels eines QR Codes kann die GUID der Maschine auf dem Web Frontend dargestellt werden und direkt von der Brille gespeichert werden.

Dies würde die Notwendigkeit eliminieren, die GUID manuell einzugeben, um die Maschinendaten laden zu können.

### 9.3 Pfeil Indikatoren

Manche Elemente der Benutzeroberfläche befinden sich ausserhalb des Sichtfeldes. Durch Indikatorpfeile kann auf die Position dieser Elemente aufmerksam gemacht werden, ohne die Sicht zu beeinflussen oder von der Arbeit abzulenken.

Ein Experimental Feature<sup>16</sup> existiert bereits für Pfeil Indikatoren:

### 9.4 Scrollbare Listen

Die momentane Implementierung einer scrollbaren Listen<sup>4</sup> des Mixed Reality Toolkits befindet sich noch in der experimentellen Phase, weshalb auf eine eben solche verzichtet wurde.

Sobald diese Phase beendet ist, wird die Implementation einer solchen Liste anstatt einer seitenbasierten Ansicht empfohlen. Durch eine scrollbare Liste sind die Listenelemente schneller erreichbar was den Arbeitsablauf verbessert.

### 9.5 Hervorheben der Bauteile

Zum jetzigen Zeitpunkt werden die Bauteile nur farblich markiert. Ideal wäre es, wenn nach der Auswahl eines Teiles, der Rest der Maschine in den Hintergrund tritt. Damit wäre der Fokus auch visuell ganz auf dem ausgewählten Teil.

Zusätzlich könnte man ein ausgewähltes Bauteil visuell ansatzweise in seine Einzelteile zerlegen. Somit wäre klar welche Teile als nächstes ausgewählt werden können und wo die Grenzen zwischen den Einzelteilen liegt, was das Auswählen erleichtert.

Diese Funktion ist voraussichtlich mit einem sehr hohen Zeitaufwand verbunden.

---

<sup>17</sup> <https://www.microsoft.com/de-de/hololens/hardware>

## 10 Fazit

Da diese Arbeit zu einem grossen Teil auf dem Konzept der direkten Vorarbeit, sowie der Infrastruktur weiterer Vorarbeiten aufbaut, wurde viel Wert auf die Umsetzung eben jenes Konzeptes gelegt.

Dank einer guten Projektplanung zu Beginn waren die Arbeitspakete für jeden Sprint schnell und klar definiert. Somit konnte zielorientiert an den Sprints gearbeitet werden.

Ebenfalls konnten die meisten Meetings, insbesondere Sprintmeetings, online zur Zufriedenheit aller Beteiligten durchgeführt werden.

Obwohl mehrere ungeplante Hindernisse überwunden werden mussten, wie zum Beispiel ein Backend welches mehr schlecht wie recht funktioniert bzw. verfügbar war, konnte die Arbeit zu einem zufriedenstellenden Ergebnis gebracht werden.

Ein Grossteil der Funktionalität konnte schon früh im Entwicklungsprozess rudimentär implementiert werden.

Dadurch hätten idealerweise schon zeitig User Tests durchgeführt werden können. Diese wurden jedoch durch die Coronapandemie fast verunmöglicht. Es konnten zwar ein paar allgemeine Fakten und Informationen in Erfahrung gebracht werden, aber die räumliche Darstellung in der Hololens und die Interaktionen mit eben jener sind schlecht bis gar nicht auf Papier zu verdeutlichen.

Die gesteckten Ziele konnten erreicht werden und führten zu lehrreichen Erkenntnissen bezüglich der Arbeit mit Unity und AR. Es konnten ebenfalls viele Erfahrungen bezüglich des Umgangs mit bestehenden Projekten gesammelt werden.

## 11 Quellenverzeichnis

### Quelle

- [1] MRTK Button, [https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/README\\_Button.html](https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/README_Button.html), letzter Zugriff am 10.06.2020
- [2] PlayerPrefs in Unity, <https://docs.unity3d.com/ScriptReference/PlayerPrefs.html>, letzter Zugriff am 10.06.2020
- [3] MRTK Solvers, [https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/README\\_Solver.html](https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/README_Solver.html), letzter Zugriff am 10.06.2020
- [4] Experimental Feature Scrolling Object Collection, <https://microsoft.github.io/MixedRealityToolkit-Unity/Assets/MRTK/SDK/Experimental/ScrollingObjectCollection/README.html>, letzter Zugriff am 10.06.2020
- [5] MRTK ObjectCollection, [https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/README\\_ObjectCollection.html](https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/README_ObjectCollection.html), letzter Zugriff am 10.06.2020
- [6] Unity LayoutGroup, <https://docs.unity3d.com/2019.1/Documentation/ScriptReference/UI.LayoutGroup.html>, letzter Zugriff am 10.06.2020
- [7] Single-Responsibility-Prinzip auf Wikipedia, <https://de.wikipedia.org/wiki/Single-Responsibility-Prinzip>, letzter Zugriff am 10.06.2020
- [8] MRTK SystemKeyboard, [https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/README\\_SystemKeyboard.html](https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/README_SystemKeyboard.html), letzter Zugriff am 10.06.2020
- [9] Hololens SystemKeyboard Verhalten in Unity, <https://docs.microsoft.com/en-us/windows/mixed-reality/keyboard-input-in-unity>, letzter Zugriff am 10.06.2020
- [10] :NET Mathematik Bibliothek , <https://www.mathdotnet.com/>, letzter Zugriff am 10.06.2020
- [11] FURPS Definition auf Wikipedia, <https://en.wikipedia.org/wiki/FURPS>, letzter Zugriff am 10.06.2020
- [12] Vuforia Startseite, <https://www.ptc.com/de/products/augmented-reality/vuforia>, letzter Zugriff am 10.06.2020
- [13] MRTK Dokumentation, <https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/GettingStartedWithTheMRTK.html>, letzter Zugriff am 10.06.2020
- [14] Beispiel für einen Hololens QR Reader, <https://unitylist.com/p/jk6/Holo-Lens-QR-Reader>, letzter Zugriff am 10.06.2020
- [15] Visual Profiler des MRTK, <https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/Diagnostics/UsingVisualProfiler.html>, letzter Zugriff am 10.06.2020
- [16] MRTK Directional Indicator Solver, [https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/README\\_Solver.html#directional-indicator](https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/README_Solver.html#directional-indicator), letzter Zugriff am 10.06.2020
- [17] Hololens 2 Features und technische Daten <https://www.microsoft.com/de-de/hololens/hardware>, letzter Zugriff am 10.06.2020

## 12 Abbildungsverzeichnis

Abbildung 1: Layout des Hauptmenüs .....	4
Abbildung 2: Veranschaulichung einer generischen Anzeige .....	7
Abbildung 3: Fenster des Aufgabenmenüs .....	8
Abbildung 4: Fenster des Maschinenteilemenüs.....	10
Abbildung 5: Ansicht des Dashboards .....	14
Abbildung 6: Beispiel für den Farbenblindmodus.....	16
Abbildung 7: Navigationsmöglichkeiten des Hauptmenüs .....	22
Abbildung 8: Zusammenhänge der Fenster des Maschinenteilemenüs.....	22
Abbildung 9: Das Dashboard interagiert mit keinem anderen Fenster .....	23
Abbildung 10: Zusammenhänge der Fenster des Aufgabenmenüs .....	23
Abbildung 11: Aufbau der Szene in Unity .....	24