



Abschlussprüfung Winter 2021/22

Fachinformatiker für Anwendungsentwicklung  
Dokumentation zur betrieblichen Projektarbeit

## **Flight Operation Analyser**

Moderne Webapplikation zur Echtzeit Flugsteuerung

**Zeitraum: 24.09.2021 - 24.11.2021**

**Prüfungsbewerber:**

Tobias Jung  
Zeppelinstraße 4  
65428 Rüsselsheim



**Lufthansa Systems**

**Ausbildungsbetrieb:**

Lufthansa Systems GmbH & Co. KG  
Am Messeplatz 1  
65479 Raunheim

Martin Trzaska  
+49 151 58922204

# Inhaltsverzeichnis

<b>Inhaltsverzeichnis</b>	<b>I</b>
<b>Abkürzungsverzeichnis</b>	<b>III</b>
<b>1 Vorwort</b>	<b>1</b>
1.1 Lufthansa Passage Airline . . . . .	1
1.2 Lufthansa Systems GmbH . . . . .	1
<b>2 Projektdefinition</b>	<b>2</b>
2.1 Projektumfeld . . . . .	2
2.1.1 Technisches Umfeld . . . . .	2
2.2 Ist-Analyse . . . . .	2
<b>3 Projektplanung</b>	<b>4</b>
3.1 Bewertung des Ist-Zustandes . . . . .	4
3.2 Nutzwertanalyse . . . . .	4
3.2.1 Gewichtungen . . . . .	4
3.2.2 Auswertung . . . . .	5
3.3 Definition von Zielen . . . . .	5
3.4 Zeit- und Ressourcenplanung . . . . .	5
3.4.1 Zeitplanung . . . . .	5
3.4.2 Personalplanung . . . . .	6
3.5 Kostenplan . . . . .	6
3.5.1 Personalkosten . . . . .	6
3.5.2 Sonstige Kosten . . . . .	6
3.5.3 Gesamtkosten . . . . .	7
<b>4 Projektdurchführung</b>	<b>8</b>
4.1 Frontend . . . . .	8
4.1.1 Komponentenaufbau . . . . .	8
4.2 Backend . . . . .	8
4.2.1 Obelisk . . . . .	9
4.3 DNS . . . . .	9
4.4 Netzwerkfreigaben . . . . .	10
<b>5 Fazit</b>	<b>11</b>
<b>Quellenverzeichnis</b>	<b>i</b>
<b>Appendices</b>	<b>ii</b>
<b>A Detaillierte Zeitplanung</b>	<b>ii</b>

<b>B</b>	<b>Detaillierte Zeitplanung - Soll zu Ist</b>	<b>iii</b>
<b>C</b>	<b>Obelisk HCC-Reporer</b>	<b>iv</b>
<b>D</b>	<b>Obelisk Core - Explorer</b>	<b>v</b>
<b>E</b>	<b>Obelisk Core - Selektionstool</b>	<b>vi</b>
<b>F</b>	<b>Tableau - DeepDives</b>	<b>vii</b>
<b>G</b>	<b>Tableau - DeepDive Deboarding</b>	<b>viii</b>
<b>H</b>	<b>Flight Operation Analyser - Filter</b>	<b>ix</b>
<b>I</b>	<b>Flight Operation Analyser - Filter - Selektion</b>	<b>x</b>
<b>J</b>	<b>Flight Operation Analyser - Filter - Vorgefertigte Sets</b>	<b>xi</b>
<b>K</b>	<b>Nutzwertanalyse</b>	<b>xii</b>
<b>L</b>	<b>Angular Example Component - TypeScript</b>	<b>xiii</b>
<b>M</b>	<b>Angular Example Component - HTML</b>	<b>xiv</b>
<b>N</b>	<b>Angular Example Component - Style Sheet</b>	<b>xv</b>
<b>O</b>	<b>MongoDB Model</b>	<b>xvi</b>

## Abkürzungsverzeichnis

**DOM** Document Object Model 8

**HCC** Hub Control Center 2

**HT** Hyper-Threading 2

**HTML** Hypertext Markup Language 8

**IDE** Integrated Development Environment 2

**IHK** Industrie- und Handelskammern 5

**OBELISK** Operationelle Betriebsdatenerfassung Lufthansa - Information, Statistik, Kosten [Lufthansa Datawarehouse] I, 2, 5, 9

**RAM** Random-Access Memory 2

**SCSS** Sassy CSS 8

**SODIMM** Small Outline Dual Inline Memory Module 2

**SQL** Structured Query Language 2

# 1 Vorwort

Die betriebliche Projektarbeit wird im Rahmen der Abschlussprüfung der Ausbildung zum Fachinformatiker Anwendungsentwicklung im Winter 2021 bei der Lufthansa Systems GmbH durchgeführt. Das Projekt beinhaltet die Erstellung einer begleitenden Dokumentation und einer Präsentation mit anschließendem Fachgespräch. Aus Vereinfachungsgründen wird im weiteren Verlauf dieser Dokumentation auf eine geschlechterspezifische Differenzierung verzichtet.

## 1.1 Lufthansa Passage Airline

Die Lufthansa Passage Airline fortfolgend als "Lufthansa" bezeichnet, ist Deutschlands größte Fluggesellschaft. Die Kranich-Airline wird gemeinhin als Flagcarrier Deutschlands wahrgenommen; sie wurde 1953 neu gegründet nach ihrer Auflösung in 1951. Lufthansa hat zwei Drehkreuze (Hubs) in Deutschland, Frankfurt und München.

## 1.2 Lufthansa Systems GmbH

Die Lufthansa Systems GmbH & Co. KG ist mit etwa 2.400 Mitarbeitern in 17 Ländern primärer IT-Dienstleister und hundertprozentige Tochtergesellschaft der Deutschen Lufthansa AG. Die Lufthansa Systems bietet 350 Kunden ein großes Portfolio an IT-Produkten und Dienstleistungen, die Produkte zielen darauf ab die Prozesseffizienz zu steigern und das Reiseerlebniss der Passagiere zu verbessern.<sup>1</sup>

---

<sup>1</sup>Lufthansa Systems. *Über uns*. URL: <https://www.lhsystems.de/about-us/uber-uns>.

## 2 Projektdefinition

In der Projektdefinition werden zunächst alle projektrelevanten Rahmenbedingungen erfasst. Die Definition beinhaltet eine Analyse des Ist-Zustands und des Weiteren eine Beschreibung des Projektumfelds.

### 2.1 Projektumfeld

Die Projektdurchführung erfolgt bei nach Ausleihung durch die Lufthansa Systems GmbH an die Lufthansa Passage in der Flugsteuerung [FRA L/GS - Operational Steering & HCC (HUB Control Center Frankfurt)].

Die Verantwortung für den weltweiten Verkehrssteuerungsprozess der Lufthansa wird von der Abteilung FRA L/GS wahrgenommen. Dazu gehören insbesondere die Koordination, das Monitoring und die Steuerung aller stationsrelevanten Bodenprozesse für Lufthansa und deren Handling Partner. Der Fokus liegt hierbei sowohl auf der Pünktlichkeit als auch auf der Wirtschaftlichkeit.

#### 2.1.1 Technisches Umfeld

Für die Umsetzung der Projektarbeit steht ein Lenovo ThinkPad T490 mit einem Intel i7-8665U (4x 2,11 GHz HT) und 16 GB SODIMM RAM zur Verfügung. Für die Entwicklung des Backends steht NodeJS mit einer Datenbankansbindung an MongoDB zur Verfügung. Als IDE wird Visual Studio Code verwendet.

Als Server steht eine Lenovo ThinkStation zur Verfügung.

.....

### 2.2 Ist-Analyse

Die nachfolgende Ist-Analyse zeigt den groben Aufbau der Reportingtools der Lufthansa.

Es gibt derzeit zwei große Systeme in der Lufthansa, das erste ist Obelisk, welches nicht nur das Datawarehouse von Lufthansa ist, sondern ebenfalls noch eine umfassende Weboberfläche bietet. Obelisk stammt etwa aus dem Jahre 2010. Diese Oberfläche hat dutzende Views, auf denen man unterschiedlichste Flugwerte sich anzeigen lassen kann. Die wichtigste View für die Projektabteilung ist, der HCC-Reporter. Dieser ist dafür da Flugvolumenzahlen, wie die Passagieranzahl und Flüge anzuzeigen. (Appendix: C) Obelisk bietet die möglich mehrere vordefinierte SQL Skripte an, welche im Browser ausgeführt werden können (Appendix: D). Zusätzlich gibt es mehrere Selektionstools, in denen man über eine grafische Oberfläche SQL Statements nachstellen kann (Appendix: E).

Das zweite große System ist, Tableau. Tableau ist eine Software für Datenvisualisierung und Reporting. Tableau wird größtenteils für die Prozessanalyse

verwendet. Prozesse sind die einzelnen Abläufe für die Flugzeugabfertigung (Deboarding, Cleaning, Catering, Fueling, Loading, Boarding, Pushback).

Übersicht über die DeepDives (Appendix: F)  
Beispiel DeepDive - Deboarding (Appendix: G)

## 3 Projektplanung

### 3.1 Bewertung des Ist-Zustandes

Während der Analyse des Ist-Zustandes wurde festgestellt, dass die bereits bestehenden Systeme für eine aktive Flugsteuerung nicht ausreichend sind. Mit den bestehenden Systemen ist es unmöglich einen Einblick in die Performance des aktuellen Tages zu bekommen. Diese Systeme sind nur auf Vortagswerte ausgelegt. Für eine aktive Flugsteuerung ist es nötig, die Prozesse in Echtzeit abzubilden. Zusätzlich sind beide Systeme, aufgrund ihres hohen Alters, inperformant, man muss mehrere Minuten auf eine simple Auswertung warten. Beide Systeme stellen keine mobile Ansicht zur Verfügung.

### 3.2 Nutzwertanalyse

Für die Bewertung werden Punkte von **1-5** vergeben, je höher die Zahl, desto positiver wird das Kriterium von dem Produkt erfüllt.

#### 3.2.1 Gewichtungen

##### Echtzeitfunktion

Das Thema Echtzeitfunktion ist eines der Hauptgründe für das Projekt. Für eine aktive Flugsteuerung ist es notwendig Livedaten über Flüge und aggregierte Werte zu bekommen, ohne aufwändige manuelle Analysen machen zu müssen. Aus diesem Grund wird die Gewichtung auf **25 %** gesetzt.

##### Mobile Verfügbarkeit

Die Mobile Verfügbarkeit ist der zweite Hauptgrund warum dieses Projekt ins Leben gerufen wurde. Es ist notwendig, dass auch Mitarbeiter auf dem Vorfeld ohne Laptop von ihrem Handy aus auf das Reporting- und Echtzeittool zugreifen können (KPI in der Hosentasche). Aus diesem Grund wird die Gewichtung ebenfalls auf **25 %** gesetzt.

##### Performance

Das Thema Performance ist in der heutigen Zeit ein wichtiges Thema, da die Technik und die Rechenleistung immer besser geworden ist, so möchte man nicht allzu lange auf seine Auswertung warten. Zusätzlich muss die Performance für ein Echtzeitsystem sehr hoch sein, um aktuelle Zahlen zeitgerecht zu liefern. Aus diesem Grund wird die Gewichtung auf **10 %** gesetzt.

##### Benutzerfreundlichkeit

Für moderne Webseiten wird die Benutzerfreundlichkeit Groß geschrieben. Eine gut bedienbare und übersichtliche Weboberfläche wirkt sich positiv auf die Nutzer und deren Produktivität aus. Aus diesem Grund wird die Gewichtung auf **15 %** gesetzt.

##### Anpassbarkeit

Um ein bereits bestehendes Tool zu erweitern, ist es von Vorteil wenn dieses



Modular aufgebaut ist. Für eine aktive Steuerung ist es ebenfalls notwendig das es möglich ist kurzfristige Änderungen schnell umzusetzen um die operativen Mitarbeiter mit den benötigten Reportingtools auszustatten. Aus diesem Grund wird die Gewichtung ebenfalls auf **15 %** gesetzt.

#### Kosten

Die Kosten spielen ebenfalls eine entscheidende Rolle, vorallem da das Projekt während der Coronazeit stattfand und dort die Ressourcen noch knapper als unter Normalbedingungen sind. Die Gewichtung wird auf **10 %** gesetzt.

### **3.2.2 Auswertung**

Nach einer ausführlichen Analyse der bestehenden Produkte und des geplanten Projektes wurde folgendes Ergebnis erzielt.

- Platz 1: Flight Operation Analyser (400 Punkte)
- Platz 2: Tableau (165 Punkte)
- Platz 3: Obelisk (135 Punkte)

**Für eine ausführliche Nutzwertanalyse: (Appendix: K)**

## **3.3 Definition von Zielen**

Nach der Bewertung des Ist-Zustandes wurden folgende Ziele festgelegt.

- Historische Auswertungen
- Realtime Funktion
- Mobile Ansicht
- Erreichbarkeit außerhalb des Lufthansa Netzwerkes

## **3.4 Zeit- und Ressourcenplanung**

### **3.4.1 Zeitplanung**

Für die Umsetzung des Projektes standen 70 Stunden zur Verfügung, wie es die IHK Darmstadt vorschreibt.<sup>2</sup> Bevor mit dem Projekt gestartet wurde, fand eine Aufteilung auf verschiedene Phasen statt, die den kompletten Prozess der Softwareentwicklung abdecken.

---

<sup>2</sup>IHK Darmstadt. *Die Projektarbeit in den IT-Berufen*. URL: <https://www.darmstadt.ihk.de/produktmarken/aus-und-weiterbildung-channel/pruefungen/downloads/muther/it-hinweise-projektarbeit-2551050>.

Phase	Geplant
Analysephase	4 h
Entwurfsphase	16 h
Implementierungsphase	27 h
Abnahme- und Deploymentphase	12 h
Dokumentationsphase	10 h
Summe	69 h

Siehe Appendix: A für eine ausführliche Gliederung

### 3.4.2 Personalplanung

Das Projekt wurde mit der Hilfe von Matthias Partzsch geplant und realisiert. Die unten aufgeführten Personenkreise haben regelmäßig das Interface auf ihre Bedienbarkeit und Funktionen getestet.

Beteiligte Person / Personenkreis	Rolle
Tobias Jung	Projektmitarbeiter
Matthias Partzsch	Projektleiter
Hub Duty Manager	Nutzer (UX Tester)
Hub Duty Officer	Nutzer (UX Tester)

## 3.5 Kostenplan

### 3.5.1 Personalkosten

In die Personalkostenplanung fließen nur meine tatsächlichen 69.5 Arbeitsstunden ein, da ich der einzige war, der an dem Projekt aktiv gearbeitet hat.

Position	Wert
Kosten pro Stunde	9.42 Euro
Umgesetzte Arbeitsstunden	69.5h

$$69.5h * 9.42 \frac{Euro}{h} = 654.69 Euro$$

### 3.5.2 Sonstige Kosten

In die Sonstigen Kosten fließen die Server (Hardware / Betriebssystem) und Lizenzkosten (MongoDB) ein. Gerechnet wird mit einer Nutzungsdauer von 24

Monaten (2 Jahre).

Position	Kosten pro Monat
ThinkStation	75 Euro
MongoDB Cloud Database	60 Euro

$$(75Euro + 60Euro) * 24 = 3240Euro$$

### 3.5.3 Gesamtkosten

Gerechnet für eine 24 monatige Laufzeit

$$(Personalkosten + SonstigeKosten) = Gesamtkosten$$

$$(654.69Euro + 3240Euro) = 3694.64Euro$$

## 4 Projektdurchführung

### 4.1 Frontend

Für die Entwicklung des Frontends entschied ich mich für Angular in der Version 12 mit Angular Material als Designbibliothek. Angular ist ein Frontend Webapplikationsframework welches auf TypeScript basiert. Es wird ständig durch eine Community bestehend aus tausenden von Einzelpersonen, Unternehmen und Organisationen angeführt von Google weiterentwickelt.<sup>3</sup>

Angular ist Komponenten basiert, somit setzt sich die Applikation aus einer Hauptkomponente zusammen welche sich wiederum dann aus mehreren Unterkomponenten zusammensetzt. Derzeit sind es 31 Module, welche die Prozesse, den Liveticker und verschiedene Steuermodule enthalten. Das gesamte Projekt ist modular aufgebaut, sodass man alle Module/Bausteine wiederverwenden kann und somit die Effizienz, Entwickelbarkeit und Wiedererkennung zu verbessern.

Die Filter Komponente, welche eine zentrale Funktion inne hat, ist dafür da, den Datenfluss zwischen Frontend und Backend aufgrund der Nutzereinstellungen zu steuern. (Appendix: H) Jeder einzelne Filter bietet die Möglichkeit exakte Selektionen für die Auswahl der Flüge zu treffen. (Appendix: I) Zusätzlich zu der Selektion, gibt es "Vordefinierte Sets (Predefined Sets) womit es möglich ist, Filteroptionen zu gruppieren. Zum Beispiel bei dem Filter "Subfleet" bei dem man den Flugzeugtyp auswählen kann, gibt es die Sets: Narrow Body (Schmale Flugzeuge) und Wide Body (Breite Flugzeuge), das sorgt für eine bessere User Experience. J)

#### 4.1.1 Komponentenaufbau

Jede Komponente ist gleich aufgebaut, im Projektverzeichnis gibt es einen Unterordner mit dem Namen, in diesem Ordner sind drei Dateien.

Die **NAME.component.ts** ist die zentrale Datei der Komponente welche die Logik beinhaltet. (Appendix: L)

Die **NAME.component.html** beinhaltet das HTML Template welches dann in die Webapplikation eingebunden wird. (Appendix: M).

Die **NAME.component.scss** beinhaltet das SCSS Stylesheet welches exklusiv für die HTML Datei der Komponente zuständig ist und nicht das restliche DOM beeinflusst. (Appendix: N).

### 4.2 Backend

Das Backend wurde in NodeJS umgesetzt. NodeJS ist eine Open-Source-JavaScript-Laufzeitumgebung, welche auf der Google V8 Engine basiert und für hochskalier-

---

<sup>3</sup>Google. *Angular*. URL: <https://angular.io/>.

bare Netzwerkanwendungen entworfen wurde. Node.js-Anwendungen sind ereignisbasiert und laufen asynchron ab, diese sind lediglich Single Thread fähig, man kann eine Applikation aber "forken" um diese zu replizieren um sie mit mehreren Threads laufen zu lassen.<sup>4</sup>

Das Backend hat mehrere Funktionen, einmal eine bestehende OracleDB Datenbankverbindung zum Lufthansa Datawarehouse (Obelisk) aufrecht zuhalten und verschiedene Tabellen für Flugerreignisse und Prozesszeitstempel in der eigenen lokalen MongoDB Datenbank zu replizieren und nachzuberechnen. Dies geschieht alle 30 Sekunden um eine "Nahezu Echtzeit" Auswertung möglich zu machen. (Appendix: O)

#### 4.2.1 Obelisk

Obelisk ist das Data Warehouse der Lufthansa mit Schwerpunkt auf Daten aus dem operativen Umfeld. Obelisk wird teilweise auch konzernübergreifend genutzt.

Das Archiv des Data Warehouses reicht zum Teil über mehrere Jahrzehnte zurück, dessen Verarbeitung aber im Wesentlichen echtzeitnah erfolgt.

Obelisk basiert auf einem Oracle Enterprise RDBMS, neben dem ein Apache Webserver aufgesetzt wurde, um eine webbasierte Plattform anzubieten. Abfragen über Business Intelligence Tools wie IBM Cognos sind ebenfalls im Programm enthalten.

**Aus Sicherheitsgründen werden die interne Struktur und die verwendeten Tabellen nicht offengelegt.**

### 4.3 DNS

Um die Webapplikation bequem von etwaigen Geräten erreichbar zu machen, wurden 4 Subdomains auf den Lufthansa DNS Servern im Intranet angelegt. Die Interne Domain lautet "dlh.de", somit sind alle Subdomains dieser unterzuordnen.

IP Adressen werden aus Sicherheitsgründen anonymisiert.

#### A-Record (IPv4)

hcc.dlh.de → xxx.xxx.xxx.xxx [hcc.dlh.de. 86400 IN A xxx.xxx.xxx.xxx]

#### CNAME-Record (Canonical Name)

foa.dlh.de → hcc.dlh.de [foa.dlh.de. 86400 IN CNAME hcc.dlh.de]

occ.dlh.de → hcc.dlh.de [occ.dlh.de. 86400 IN CNAME hcc.dlh.de]

iocc.dlh.de → hcc.dlh.de [iocc.dlh.de. 86400 IN CNAME hcc.dlh.de]

---

<sup>4</sup>NodeJS. *About*. URL: <https://nodejs.org/en/about/>.

## 4.4 Netzwerkfreigaben

Damit man aus dem Netzwerk auf den Webserver zugreifen kann, ist es zwingend erforderlich in der Lufthansa Internen Firewall die Ports für bestimmte Nutzergruppen zu öffnen. Die Anforderung war, dass jeder aus dem Lufthansa Netz und über VPN auf den Webserver zugreifen kann, glücklicherweise geht jeder Traffic über den Lufthansa Proxy, weshalb lediglich die Portfreigabe zwischen dem Proxy Server und dem Webserver bestehen muss.

Folgende Ports werden aufgrund des Webservers geöffnet:

- 80/TCP - HTTP Server (Leitet auf HTTPS weiter)
- 443/TCP - HTTPS Server

## 5 Fazit

Das Projekt "Flight Operation Analyser" war erfolgreich, damit wurde ein Tool für operative und administrative Mitarbeiter geschaffen welches von Ihren Workstations und Mobilgeräten erreichbar ist. Die Bedienbarkeit und Performance hebt sich von den bereits bestehenden Systemen erheblich ab und wird von den Mitarbeitern gut angenommen. Zusätzlich bietet das Tool gewisse Views, welche an verschiedenen Monitoren aufgespielt werden um zentral an Eingängen die Echtzeitdaten sichtbar zu machen. Bis auf eine kleine Abweichung in der Zeitplanung von 0,5h (absolut) und 23,5h (intern) wurden alle Bedingungen akribisch eingehalten.

## References

- [1] IHK Darmstadt. *Die Projektarbeit in den IT-Berufen*. URL: <https://www.darmstadt.ihk.de/produktmarken/aus-und-weiterbildung-channel/pruefungen/downloads/muther/it-hinweise-projektarbeit-2551050>.
- [2] Google. *Angular*. URL: <https://angular.io/>.
- [3] NodeJS. *About*. URL: <https://nodejs.org/en/about/>.
- [4] Lufthansa Systems. *Über uns*. URL: <https://www.lhsystems.de/about-us/uber-uns>.



# Appendices

## Appendix A Detaillierte Zeitplanung

Phase	Geplant
Analyse der bestehenden Systeme	3 h
Bewertung des Ist-Zustandes	1 h
Definition von Zielen	3 h
Zeit- und Ressourcenplanung	2 h
Auseinandersetzung mit Ops Kollegen der Flugsteuerung	3 h
Aufstellen von Style Guidelines	1 h
Skizzierung eines ersten Entwurfes	1 h
Planung der Backendstruktur (mit Technologien)	3 h
Planung der Frontendstruktur (mit Technologien)	3 h
Programmierung eines Prototypen (Frontend)	10 h
Programmierung eines Prototypen (Backend)	10 h
Vorstellung des Prototypen	2 h
Anpassung der Bedienung und Fehlerbehebung	5 h
Anbindung an die Datenbank	2 h
Testen der Datenbankverbindung	2 h
Code Sichtung und Cleanup + Fehlerbehebung	5 h
Live Schaltung der Applikation und Monitoring	3 h
Dokumentation des Projektes	10 h
Summe	69 h

## Appendix B Detaillierte Zeitplanung - Soll zu Ist

Phase	Geplant	Tatsächlich	Differenz
Analyse der bestehenden Systeme	3 h	2 h	-1 h
Bewertung des Ist-Zustandes	1 h	0.5 h	-0.5 h
Definition von Zielen	3 h	2 h	-1 h
Zeit- und Ressourcenplanung	2 h	2 h	0 h
Auseinandersetzung mit Ops Kollegen der Flugsteuerung	3 h	3 h	0 h
Aufstellen von Style Guidelines	1 h	1 h	0 h
Skizzierung eines ersten Entwurfes	1 h	1 h	0 h
Planung der Backendstruktur (mit Technologien)	3 h	2 h	-1 h
Planung der Frontendstruktur (mit Technologien)	3 h	2 h	-1 h
Programmierung eines Prototypen (Frontend)	10 h	15 h	5 h
Programmierung eines Prototypen (Backend)	10 h	15 h	5 h
Vorstellung des Prototypen	2 h	1 h	-1 h
Anpassung der Bedienung und Fehlerbehebung	5 h	3 h	-2 h
Anbindung an die Datenbank	2 h	1 h	-1 h
Testen der Datenbankverbindung	2 h	1 h	-1 h
Code Sichtung und Cleanup + Fehlerbehebung	5 h	5 h	0 h
Live Schaltung der Applikation und Monitoring	3 h	1 h	-2 h
Dokumentation des Projektes	10 h	12 h	2 h
Summe	69 h	69.5 h	0.5 h

## Appendix C Obelisk HCC-Reporer



## Appendix D Obelisk Core - Explorer

**OBELISK Explorer**

- Flug
- Cargo
- Passagier
- Boden
- Flugzeug
- Flughafen
- Fluglinie
- Services
  - SQL Skripte
- Administration
- Info

### SQL Scripts

Browse through all available sql scripts

Scripts

Display  
Browser

Item

Current Directory: /

Title	Description
Allegro	-
BIT	-
Betriebsplanung-DE-Stationen	-
Betriebsplanung-FRA	-
Bundesbahn	-
Cargo	-
CityLine	-
Condor	-
DELVAG	-
ED-F	-
EP-O	-
EQ-O	-
ES-E	-
ES-MP	-
Fuel_Efficiency	-
GLC	-
GlobeGround-LHR	-
HAM-MT	-
HAM-ST	-
HCC	-
HCC-MUC	-
HN-F	-
IDG	-
KI-L	-
L-GM	-
LEOS	-

## Appendix E Obelisk Core - Selektionstool

### Query Tool (Information Leg)

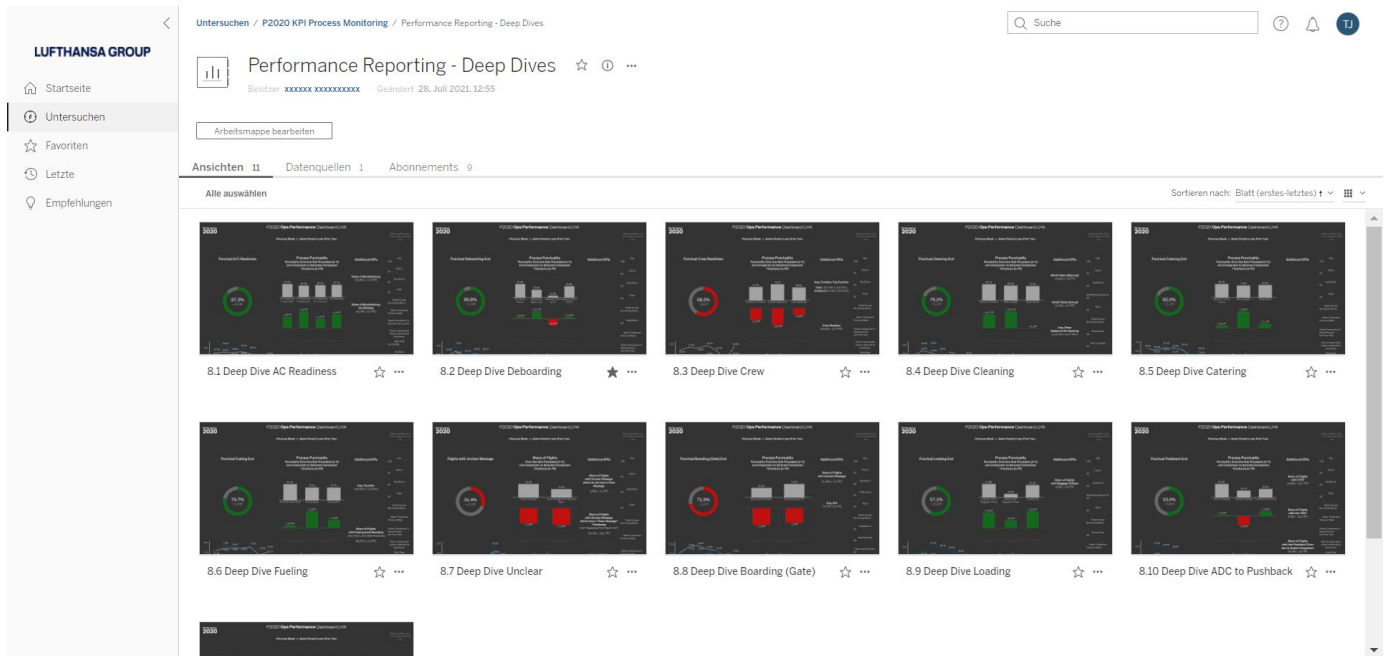
### Definition of own selections and download of the result

Profiles	Selection	Attributes	SQL	
Attribute		Rule	Operator Selection	
Flight Date	▼	----- ▼	= ▼	
-----	▼	----- ▼	-- ▼	
-----	▼	----- ▼	-- ▼	
-----	▼	----- ▼	-- ▼	
-----	▼	----- ▼	-- ▼	
-----	▼	----- ▼	-- ▼	
-----	▼	----- ▼	-- ▼	
-----	▼	----- ▼	-- ▼	
-----	▼	----- ▼	-- ▼	
-----	▼	----- ▼	-- ▼	

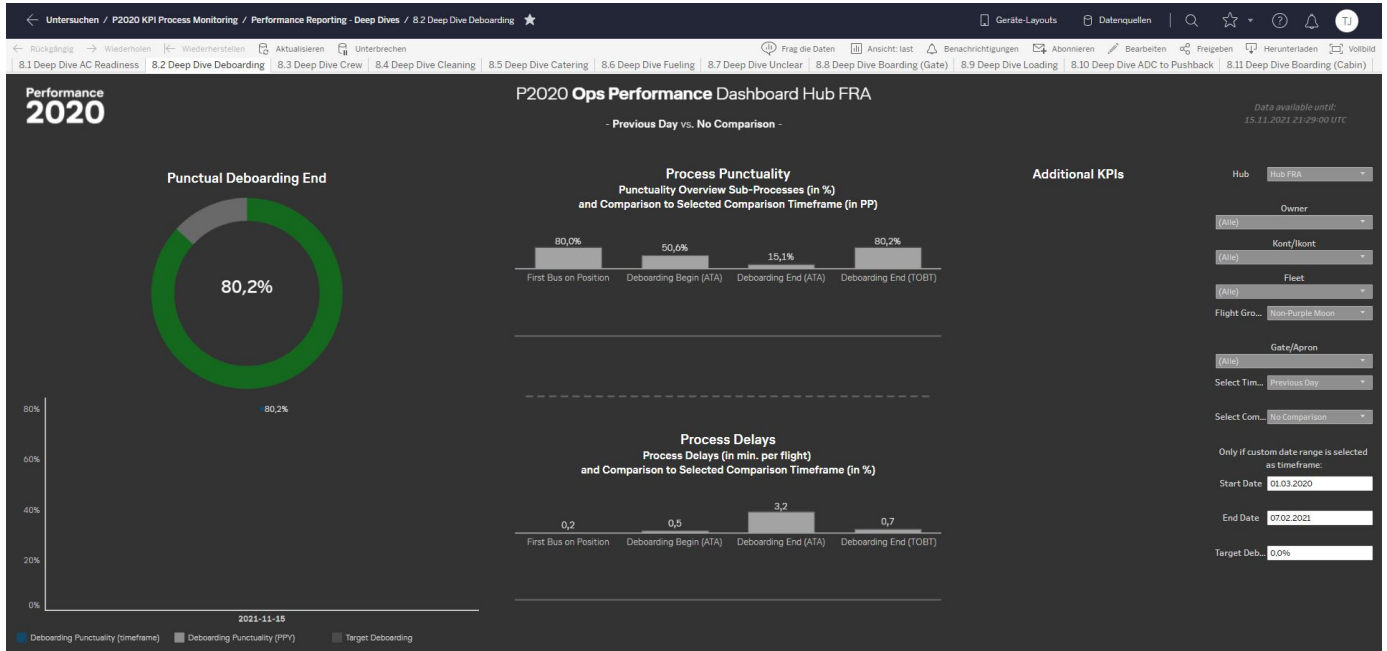
Sorting  
Flight Date + STD ▼

Reset Report Clear

## Appendix F Tableau - DeepDives



## Appendix G Tableau - DeepDive Deboarding



## Appendix H Flight Operation Analyser - Filter

Filter

Groundcase Indicator

Bereitstellung, Turnaround

Traffic Area Codes

North Atlantic, South Atlantic, Asia / ...

Subfleets

Origins

FRA

Destinations

Gates

Positions

Runways

Custom Filters

apply filters

reset



## Appendix I Flight Operation Analyser - Filter - Selektion

Filter

Enter a date range

07.11.2021 – 07.11.2021

Groundcase Indicator

Bereitstellung, Turnaround

Traffic Area Codes

North Atlantic, South Atlantic, Asia / Pacific

Subfleets

Origins

RA

Destinations

Gates

Positions

Custom Filters

apply filters

reset

Select All

Deselect All

Predefined Sets

☐ 319

☐ 320

☐ 321

☐ 32A

## Appendix J Flight Operation Analyser - Filter - Vorgefertigte Sets

Filter

Enter a date range  
07.11.2021 – 07.11.2021

Groundcase Indicator  
Bereitstellung, Turnaround

Traffic Area Codes  
North Atlantic, South Atlantic, Asia / Pacific

Subfleets  
Select All  
Deselect All  
Predefined Sets  
Narrow Body  
Wide Body

Origins  
RA

Destinations

Gates

Positions

Custom Filters

apply filters

reset

## Appendix K Nutzwertanalyse

Kriterien	Gewichtung in Prozent	Obelisk		Tableau		Flight Operation Analyser	
		Zielertrag	Wert	Zielwert	Wert	Zielwert	Wert
Echtzeitfunktion	25%	0	0	2	50	4	100
Performance	10%	3	30	2	20	4	40
Mobile Verfügbarkeit	25%	1	25	1	25	4	100
Benutzerfreundlichkeit	15%	2	30	3	45	5	75
Anpassbarkeit	15%	2	30	1	15	3	45
Kosten	10%	2	20	1	10	4	40
Summe	100%	135		165		400	
Rang		Rang 3		Rang 2		Rang 1	

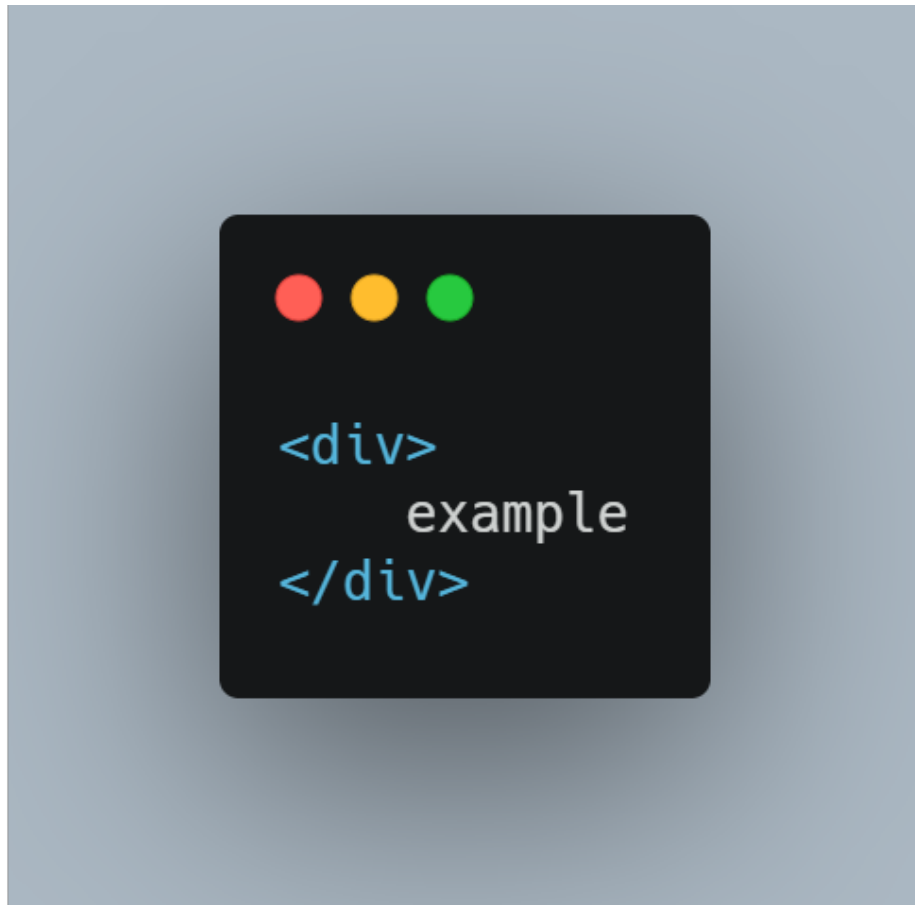
## Appendix L Angular Example Component - TypeScript



```
import { Component, OnInit, ViewChild } from "@angular/core";

@Component({
  selector: "app-example",
  templateUrl: "./example.component.html",
  styleUrls: ["./example.component.scss"],
  providers: []
})
export class ExampleComponent {
  constructor(){}
}
```

## Appendix M Angular Example Component - HTML



## Appendix N Angular Example Component - Style Sheet



## Appendix O MongoDB Model

```
1 {
2   leg_id: { type: Number, unique: true },
3   inbound_leg_id: { type: Number },
4   flight: {
5     airline: { type: String, trim: true },
6     number: { type: Number },
7     suffix: { type: String, trim: true },
8     date: { type: Date },
9     callsign: { type: String }
10  },
11  flags: {
12    diverted: { type: Boolean },
13    cancelled: { type: Boolean },
14    overbooked: { type: Boolean },
15    isQMCCleg: { type: Boolean },
16  },
17  crew: {
18    onboard: {
19      cockpit: { type: Number },
20      cabin: { type: Number }
21    }
22  },
23  route: {
24    origin: { type: String, trim: true },
25    destination: { type: String, trim: true },
26    scheduled_destination: { type: String, trim: true }
27  },
28  aircraft: {
29    owner: { type: String, trim: true },
30    registration: { type: String, trim: true },
31    subfleet: { type: String, trim: true }
32  },
33  codes: {
34    traffic_area: { type: Number },
35    service_type: { type: String, trim: true },
36    ground_case: { type: String, trim: true }
37  },
38  passengers: {
39    onboard: {
40      wab: {
41        first: { type: Number },
42        business: { type: Number },
43        premium_economy: { type: Number },
44        economy: { type: Number },
45        other: { type: Number },
46        total: { type: Number }
47      },
48      teflos: {
49        first: { type: Number },
50        business: { type: Number },
51        premium_economy: { type: Number },
52        economy: { type: Number },
53        infants: { type: Number },
54        pads: { type: Number },
55        k: { type: Number },
56        u: { type: Number },
57        w: { type: Number }
58      },
59      tango: {
60        total: { type: Number }
61      },
62      special: {
63        wheelchair: { type: Number },
64        information: { type: String, trim: true }
65      }
66    }
67  }
68 }
```

```

66     },
67     booked: {
68       storm: {
69         first: { type: Number },
70         business: { type: Number },
71         premium_economy: { type: Number },
72         economy: { type: Number },
73         total: { type: Number }
74       },
75     },
76     overbooked_kpi: { type: Number },
77     capacity: {
78       actual: {
79         first: { type: Number },
80         business: { type: Number },
81         premium_economy: { type: Number },
82         economy: { type: Number },
83         total: { type: Number }
84       },
85       blocked: {
86         storm: {
87           first: { type: Number },
88           business: { type: Number },
89           premium_economy: { type: Number },
90           economy: { type: Number },
91           total: { type: Number }
92         },
93       },
94     },
95   },
96   processes: {
97     pushback: {
98       position: {
99         actual: { type: Date },
100        scheduled: { type: Date },
101        delay: { type: Number },
102      },
103      request: {
104        actual: { type: Date },
105      },
106      given: {
107        actual: { type: Date },
108      },
109      begin: {
110        actual: { type: Date },
111      },
112      end: {
113        actual: { type: Date },
114      },
115    },
116    catering: {
117      vehicles: { type: Number },
118      begin: {
119        actual: { type: Date },
120        scheduled: { type: Date },
121        delay: { type: Number },
122      },
123      end: {
124        actual: { type: Date },
125        scheduled: { type: Date },
126        delay: { type: Number },
127      },
128      position: {
129        actual: { type: Date },
130        scheduled: { type: Date },
131        delay: { type: Number },

```



```

132     },
133   },
134   fueling: {
135     vehicles: { type: Number },
136     begin: {
137       actual: { type: Date },
138       scheduled: { type: Date },
139       delay: { type: Number },
140     },
141     end: {
142       actual: { type: Date },
143       scheduled: { type: Date },
144       delay: { type: Number },
145     },
146     position: {
147       actual: { type: Date },
148       scheduled: { type: Date },
149       delay: { type: Number },
150     },
151     block: {
152       actual: { type: Date },
153       scheduled: { type: Date },
154       delay: { type: Number },
155     },
156   },
157   boarding: {
158     begin: {
159       actual: { type: Date },
160       scheduled: { type: Date },
161       delay: { type: Number },
162     },
163     end: {
164       actual: { type: Date },
165       scheduled: { type: Date },
166       delay: { type: Number },
167     },
168     door_closed: {
169       actual: { type: Date },
170       scheduled: { type: Date },
171       delay: { type: Number },
172     },
173   },
174   loading: {
175     end: {
176       actual: { type: Date },
177       scheduled: { type: Date },
178       delay: { type: Number },
179     },
180   },
181   cleaning: {
182     type: { type: String, trim: true },
183     vehicles: { type: Number },
184     begin: {
185       actual: { type: Date },
186       scheduled: { type: Date },
187       delay: { type: Number },
188     },
189     end: {
190       actual: { type: Date },
191       scheduled: { type: Date },
192       delay: { type: Number },
193     },
194     position: {
195       actual: { type: Date },
196       scheduled: { type: Date },
197       delay: { type: Number },

```

```

198     },
199     quality: {
200       score: { type: Number },
201       status: { type: String, trim: true },
202       items: [
203         {
204           status: { type: String, trim: true },
205           cleaning_item: { type: String, trim: true },
206           cleaning_item_code: { type: String, trim: true },
207           score: { type: Number },
208           class: { type: String, trim: true },
209           cleaning_class: { type: String, trim: true },
210           class_score: { type: Number },
211           total_score: { type: Number },
212         }
213       ]
214     },
215     baggage: {
216       loaded: {
217         wab: {
218           amount: { type: Number },
219           weight: { type: Number },
220         },
221         loaded_at_departure_airport: { type: Number },
222         mishandled_at_departure_airport: { type: Number },
223         mishandled_at_departure_airport_int: { type: Number },
224         mishandled_at_arrival_airport: { type: Number },
225         mishandled_at_arrival_airport_int: { type: Number },
226       }
227     },
228     times: {
229       general: {
230         std: { type: Date },
231         atd: { type: Date },
232         sta: { type: Date },
233         ata: { type: Date },
234         tobt: { type: Date },
235         tsat: { type: Date },
236         adc: { type: Date },
237       },
238       startup: {
239         given: { type: Date },
240         request: { type: Date },
241       },
242       delta: {
243         atd_std: { type: Number },
244         ata_sta: { type: Number },
245       }
246     },
247     positions: {
248       departure_gate: { type: String, trim: true },
249       departure_position: { type: String, trim: true },
250       departure_runway: { type: String, trim: true },
251       arrival_gate: { type: String, trim: true },
252       arrival_position: { type: String, trim: true },
253       arrival_runway: { type: String, trim: true },
254     },
255     delays: [
256       {
257         delay_code: { type: String, trim: true },
258         sub_delay_code: { type: String, trim: true },
259         comment: { type: String, trim: true },
260         delay: { type: Number },
261       }

```