

LASTENHEFT

Phail-IT – Pflichtenheft

Airport Managementsystem

001

Änderungshistorie

Version	Datum	Änderung	Name
0.1	14.05.2025	Initiale Erstellung	Philipp Becker
0.2	22.05.2025	Änderung Design	Philipp Becker
0.3	28.05.2025	Änderung Inhalt	Philipp Becker
0.4	04.06.2025	Änderung Inhalt	Philipp Becker
1.0	16.06.2025	Finale Version	Philipp Becker

Rechte, auch die des auszugsweisen Nachdrucks, der fotomechanischen Wiedergabe (einschließlich Mikrokopie) sowie der Auswertung durch Datenbanken oder ähnliche Einrichtungen, vorbehalten.

Herausgeber

Phail-IT

LASTENHEFT

Dateiname	Airport_Minden_Pflichtenheft		
Dokumentennummer	2		
Dokumentenbezeichnung	Pflichtenheft		
Version	1.0		
Stand	16.06.2025		
Status	Finale Version		
Autor	Philipp Becker		
Inhaltlich geprüft von	Philipp Becker		
Freigegeben von	Philipp Becker		
Ansprechpartner	Philipp Becker	“Telefonnummer”	“Email-Adresse”
Kurzinfo	„Airport Minden GmbH“ Flughafenmanagementsystem Projekt		

LASTENHEFT

Inhaltsverzeichnis

1	Ausgangssituation und Zielsetzung.....	4
1.1	Ausgangssituation und Hintergründe.....	4
1.2	Beteiligte.....	4
1.2.1	Kunde/Auftraggeber/Interne Beteiligte.....	4
1.2.2	Anwender.....	4
2	Produktübersicht.....	5
2.1	Kundenanwendung.....	5
2.1.1	Adminanwendung.....	5
3	Systemarchitektur.....	6
3.1.1	Techstack.....	6
3.2	Architekturübersicht.....	6
4	Funktionale Anforderungen.....	7
4.1.1	Kundenanwendung.....	7
4.1.2	Adminanwendung.....	7
5	Nicht-funktionale Anforderungen.....	8
6	Schnittstellen.....	9
7	Benutzeroberflächen.....	10
7.1.1	Allgemeine UI.....	10
7.1.2	Kunden UI.....	12
7.1.3	Admin UI.....	14
8	Lieferumfang.....	16

Tabellenverzeichnis

Tabelle 1: Kunde/Auftraggeber/Interne Beteiligte.....	4
Tabelle 2: Anwender.....	4
Tabelle 3: Techstack.....	6
Tabelle 4: Kundenanwendungen.....	7
Tabelle 5: Adminanwendungen.....	7
Tabelle 6: Nicht-funktionale Anforderungen.....	8
Tabelle 7: Schnittstellen.....	9

LASTENHEFT

1 Ausgangssituation und Zielsetzung

1.1 Ausgangssituation und Hintergründe

Basierend auf dem Lastenheft soll eine Softwarelösung entwickelt werden, die die Buchung und Verwaltung von Flügen, Tickets und Gepäck ermöglicht. Die Lösung besteht aus zwei Hauptkomponenten:

- Eine dynamische Webanwendung für Endnutzer (Fluggäste)
- Eine Desktop-Applikation für das Flughafenpersonal

Diese Anwendungen werden mit einem modernen Techstack umgesetzt (Go, React, TypeScript, Tailwind, Electron, Docker).

1.2 Beteiligte

1.2.1 Kunde/Auftraggeber/Interne Beteiligte

Ansprechpartner	Name	Organisation	Telefon
Projektleiter	Thore Schwier	Minden Airport GmbH	Siehe Adressbuch
Projektmanagement	Philipp Becker	Campus Minden IT	Siehe Adressbuch
Informationssicherheit	Giulien Chow	Campus Minden IT	Siehe Adressbuch
Datenschutz	Michael Kolle	Minden Airport GmbH	Siehe Adressbuch
Betriebsverantwortlicher	Leonardo Smigielski	Minden Airport GmbH	Siehe Adressbuch

Tabelle 1: Kunde/Auftraggeber/Interne Beteiligte

1.2.2 Anwender

Nutzer	Geschätzte Anzahl an Personen	Nutzungsart
Fluggäste	-	Webapp
Bodenpersonal	20	Desktop
IT-Admins	1	Webapp & Desktop

Tabelle 2: Anwender

LASTENHEFT

2 Produktübersicht

Die angestrebte Lösung ist es eine Webanwendung zu implementieren und diese dann mit Hilfe eines Electron-Wrappers auch in eine Desktop-Applikation zu überführen. Dadurch haben beide Applikationen die gleichen Funktionen, falls wir Desktopterminals in den Flughafen integrieren wollen, auf die Kunden zugriff haben oder auch ein Admin aus dem Homeoffice auf alle Funktionen zugreifen möchte. Im folgenden trennen wir die Anforderungen in „Kundenanwendung“ und „Adminanwendung“.

2.1 Kundenanwendung

Die Kundenanwendung ermöglicht:

- Flugbuchung
- Verwaltung von Tickets
- Gepäcktracking
- Benutzerkonto mit persönlichem Profil

2.1.1 Adminanwendung

Die Adminanwendung ermöglicht:

- Verwaltung von Flugplänen
- Erfassung und Nachverfolgung von Gepäck
- Übersicht über aktuelle Flüge und Check-ins
- Rollen-basierte Rechteverwaltung

LASTENHEFT

3 Systemarchitektur

3.1.1 Techstack

Tabelle 3: Techstack

Komponente	Technologie
Backend	Go
Frontend Web	React + TypeScript + Tailwind CSS
Desktop App	Electron mit React + TypeScript
Datenhaltung	Oracle SQL Database
Containerisierung	Docker
Authentifizierung	JWT + OAuth2
Versionskontrolle	Git (GitHub)

3.2 Architekturübersicht

- Client-Server-Modell
- Single-Page-Application (SPA) im Web
- Electron-Wrapper für Desktop-App
- Gemeinsames API-Backend für beide Clients
- Containerisierte Dienste

LASTENHEFT

4 Funktionale Anforderungen

4.1.1 Kundenanwendung

Tabelle 4: Kundenanwendungen

Funktion	Beschreibung
Benutzerregistrierung / Login	Account anlegen, anmelden, Passwort vergessen
Flug suchen und buchen	Auswahl nach Abflug, Ziel, Datum
Ticketanzeige	Ticketübersicht im Dashboard
Online-Gepäcktracking	Gepäckdaten eingeben und tracken
Profilstatistiken	Persönliche Daten in einem Dashboard angezeigt

4.1.2 Adminanwendung

Tabelle 5: Adminanwendungen

Funktion	Beschreibung
Login mit Rollenrechten	Admin-, Kundenaccounts
Flugplanverwaltung	Flüge erstellen, ändern, archivieren
Statistiken und Übersicht	Zahl der gebuchten Passagiere, Flüge, etc.
Accountverwaltung	Für Admins (Account aktivieren/deaktivieren)

LASTENHEFT

5 Nicht-funktionale Anforderungen

Tabelle 6: Nicht-funktionale Anforderungen

Kategorie	Anforderung
Sicherheit	Authentifizierung mit JWT
Datenschutz	DSGVO-Konformität, Datenverschlüsselung
Performance	Antwortzeit < 1 Sekunde bei API-Requests
Skalierbarkeit	Skalierbares Backend mit Load-Balancing-Fähigkeit
Wartbarkeit	Modularer Aufbau, Clean Code (TypeScript/Go), Dokumentation
Responsiveness	Web-App läuft auf Desktop, Tablet, Smartphone
Plattformunabhängigkeit	Desktop-App: Windows, macOS, Linux via Electron

LASTENHEFT

6 Schnittstellen

Tabelle 7: Schnittstellen

Schnittstelle	Beschreibung
REST API	Kommunikation zwischen Frontend / Desktop und Backend
Authentifizierung	Login mit Token (JWT)

LASTENHEFT

7 Benutzeroberflächen

Hier werden Mockups des Applikationsdesigns verschiedener Seiten festgehalten.
Blau unterlegter Text weist auf die momentane Auswahl der Seitenansicht hin.

7.1.1 Allgemeine UI

Startseite

Dashboard Airport Minden Mockup

Time/Date

Role/Login

My Tickets

Flight Data

On-Time Data

Delay Data

Depature Data

Refresh Button

All Depature Arrival Status Filter: Dropdown Menu

Showing x of x Flight

Flight 1 Data
Start -> Destination
Depature Info
Arrival Info
Flight Status

Flight 2 Data
Start -> Destination
Depature Info
Arrival Info
Flight Status

Design der Allgemeinen Startseite der Applikation

LASTENHEFT

Login-Seite

Login Symbol
for Context

Sign in Description

Already have an account reminder with hyperlinks

Email

Password

☐ Remember me? [forgot password
hyperlink](#)

Login/Sign in

Mockup der Login-Seite von hier geht es je nach Accounttype in den Bereich des Kunden- oder Admin UIs

LASTENHEFT

7.1.2 Kunden UI

User Dashboard (Past Flights)

The mockup shows a user dashboard interface. At the top, there is a header bar with a user profile icon (labeled 'User profile Pic'), a welcome message 'Welcome User', and links for 'Date/Time' and 'Logout'. Below the header, there are five main dashboard cards: 'Total Tickets', 'Upcoming Flights', 'Past Flights' (which is the active tab), 'Confirmed Flights', and 'Baggage Items'. Under these cards is a row of four sub-sections: 'Upcoming Flights (counter)', 'Past Flights (counter)' (highlighted in blue), 'All Tickets (counter)', and 'Baggage (counter)'. The main content area is divided into a light blue header section labeled 'Flight Start -> Flight Destination' with a 'Booking Status' button, and a white body section. The body section contains two large boxes for 'Departure Data' and 'Arrival Data'. Below these are four smaller boxes for 'Seat Data', 'Seat Data', 'Gate Data', and 'Terminal Data'. At the bottom of the body section, there is a row with 'Booking Date' and 'Price' labels, and a long horizontal box labeled 'Flight Status'.

Mockup User Dashboard. Boxen „Total Tickets“, etc. sollen User Statistiken anzeigen. „Upcoming Flights (counter)“, etc. wird der Counter durch Kundendaten ausgetauscht, damit dieser sieht wieviele Tickets/Gepäck sich hinter welchen Reiter befinden. Seitenansicht bei aktiven Reiter „Past Flights“.

LASTENHEFT

User Dashboard (Baggage)

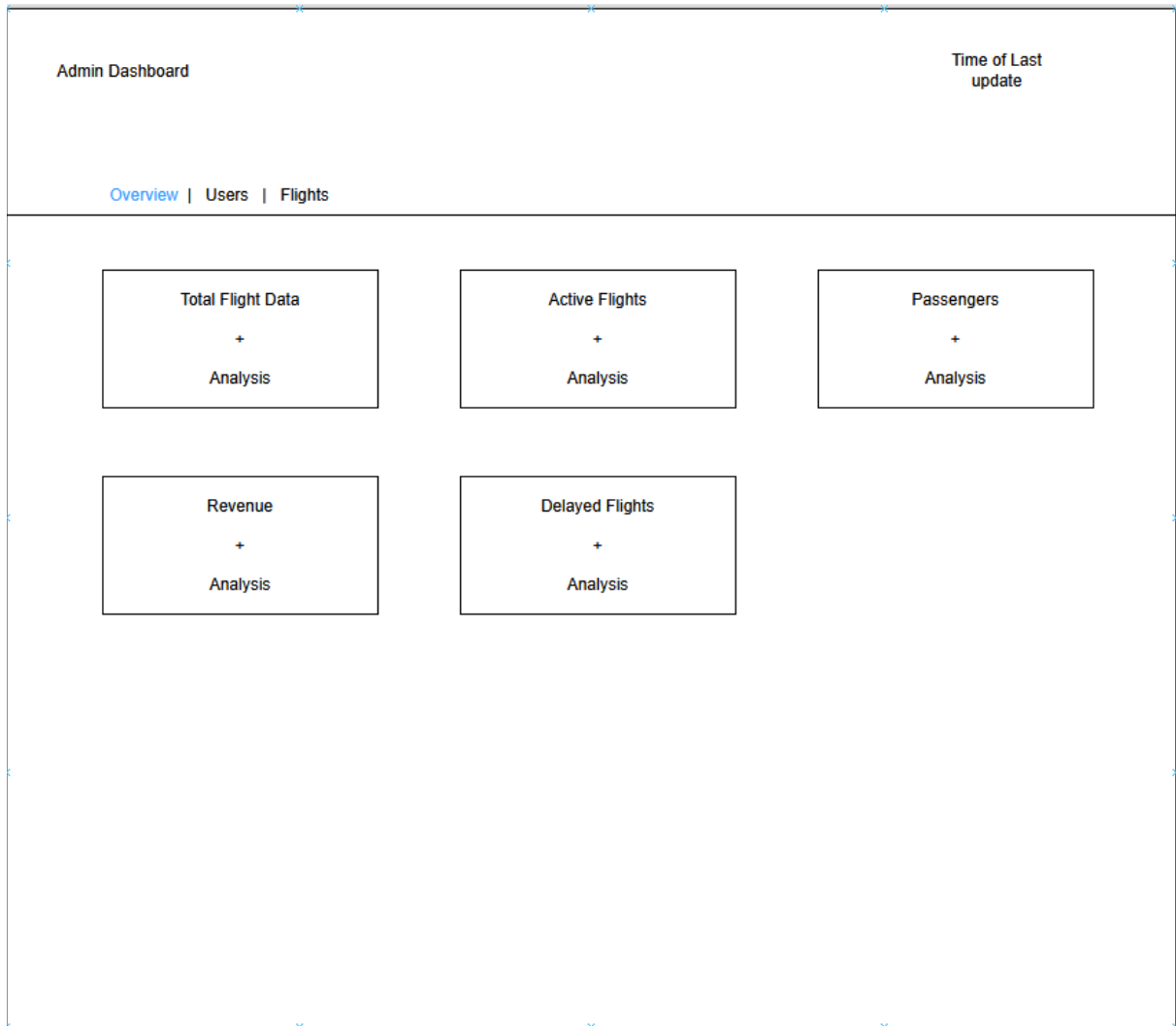
The mockup shows a user dashboard for the 'Baggage' tab. The header includes a user profile icon, a welcome message, the current date and time, and a logout button. The main content area is divided into several sections. At the top, there are five summary cards: 'Total Tickets', 'Upcoming Flights', 'Past Flights', 'Confirmed Flights', and 'Baggage Items'. Below these cards is a row of four counters: 'Upcoming Flights (counter)', 'Past Flights (counter)', 'All Tickets (counter)', and 'Baggage (counter)'. A search bar for baggage is located below the counters, with the label 'Baggage Search (Textfield, Status dropdown, Type dropdown)'. At the bottom of the dashboard, there are two large rounded rectangles, each labeled 'Baggage Data (Name, Status, Weight, Special handling)', intended for displaying detailed baggage information.

Mockup des User Dashboards bei aktiven Reiter „Baggage“. Gepäcksuche besteht aus Suchfeld, welches aus einem Textfeld und zwei Dropdown Menüs besteht mit denen Gepäck gesucht und gefiltert werden kann. Gepäckdaten werden daraufhin angezeigt.

LASTENHEFT

7.1.3 Admin UI

Admin-Dashboard für Monitoring



Mockup des Admin Dashboards. Sehr simpel, aber funktionell gehaltenes Design. Boxen enthalten Statistiken des Flughafens und deren Entwicklung zum Vormonat. Ansicht bei aktivem Reiter „Overview“.

LASTENHEFT

Admin Benutzerverwaltung

Admin Dashboard

Time of Last update

[Overview](#) | [Users](#) | [Flights](#)

User Management

User	Role	Status	Tickets	Joined	Actions
		Userdata 1			Context Menu
		Userdata 2			Context Menu
		Userdata 3			Context Menu
		Userdata 4			Context Menu
		Userdata 5			Context Menu

Mockup der Benutzerverwaltung. User sind durchsuchbar nach Namen mit Hilfe eines Textfelds. User werden aufgeführt mit den Daten „User“ (Name), „Role“, „Status“, „Tickets“ und „Joined“. Am Ende gibt es ein Kontextmenü um Aktionen auf die Nutzerkonten auszuführen. Ansicht mit aktiven Reiter „Users“.

LASTENHEFT

8 Lieferumfang

- Quellcode (Git-Repository)
- Docker-Compose-Konfiguration
- Technische Dokumentation (Markdown / HTML)
- Setup-Skripte für lokale Entwicklung
- Benutzerhandbuch (für Admins und Fluggäste)
- Schulungsmaterial (Slides oder PDF)