Pflichtenheft

|  |  |
| --- | --- |
| **Projektbezeichnung** | Digitales Flugplatzmanagement |
| **Projektleiter** | Julian Halbmayr |
| **Erstellt am** | 31.07.2025 |
| **Letzte Änderung am** | 28.08.2025 |
| **Status** | Fertig |
| **Aktuelle Version** | 1.6 |

**Änderungsverlauf**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nr.** | **Datum** | **Version** | **Geänderte Kapitel** | **Art der Änderung** | **Autor** | **Status** |
| 1 | 31.07.2025 | 1.1 | Alle | Überarbeitung | Julian Halbmayr | In Bearbeitung |
| 2 | 02.08.2025 | 1.2 | Alle | Überarbeitung | Julian Halbmayr | In Bearbeitung |
| 3 | 03.08.2025 | 1.3 | Alle | Überarbeitung | Julian Halbmayr | In Bearbeitung |
| 4 | 04.08.2025 | 1.4 | Alle | Fertigstellung | Julian Halbmayr | In Bearbeitung |
| 5 | 28.08.2025 | 1.5 | Alle | Fertigstellung | Julian Halbmayr | Fertig |
| 6 | 24.09.2025 | 1.6 | Funktionsbeschreibung | Überarbeitung | Julian Halbmayr | In Bearbeitung |

Inhalt

[Einleitung 4](#_Toc209602373)

[Allgemeines 4](#_Toc209602374)

[Teams und Schnittstellen 4](#_Toc209602375)

[Ziele 5](#_Toc209602376)

[Zentrale Daten- und Benutzerverwaltung 5](#_Toc209602377)

[Sicherer Web-App Zugriff 5](#_Toc209602378)

[Bereitstellung von Systemdiensten 5](#_Toc209602379)

[Skalierbarkeit und Zuverlässigkeit 5](#_Toc209602380)

[Benutzer- und Datenmanagement 5](#_Toc209602381)

[Test- und Produktivsystem 5](#_Toc209602382)

[Systemsicherheit 5](#_Toc209602383)

[Funktionale Anforderungen 6](#_Toc209602384)

[Anforderung 1: Zugriff auf Systemdienste 6](#_Toc209602385)

[Anforderung 2: Systemverfügbarkeit und Infrastruktur 6](#_Toc209602386)

[Anforderung 3: Sicherheitsmaßnahmen 6](#_Toc209602387)

[Nichtfunktionale Anforderungen 7](#_Toc209602388)

[Allgemeine Anforderungen 7](#_Toc209602389)

[Qualitäts- und Sicherheitsanforderungen 7](#_Toc209602390)

[Entwicklungs- und Projektmanagement-Anforderungen 7](#_Toc209602391)

[Rahmenbedingungen 8](#_Toc209602392)

[Technische Anforderungen 9](#_Toc209602393)

[Server-Hardware 9](#_Toc209602394)

[Client-Hardware 9](#_Toc209602395)

[Software 9](#_Toc209602396)

[Sicherheitsanforderungen 9](#_Toc209602397)

[Funktionsbeschreibung der Anwendung 10](#_Toc209602398)

[Anmeldung & Authentifizierung 10](#_Toc209602399)

[Start-/Landeliste (Flugerfassung) 10](#_Toc209602400)

[Pilotenseite 11](#_Toc209602401)

[Fliegen (Zählerstand) 11](#_Toc209602402)

[Admin – Flugzeugverwaltung 12](#_Toc209602403)

[Admin – Benutzerverwaltung 12](#_Toc209602404)

[Admin – Kostenstellen 13](#_Toc209602405)

[Admin – Pilotflüge 13](#_Toc209602406)

[Reservierung 14](#_Toc209602407)

[Berichte & Exporte 14](#_Toc209602408)

[Use Cases 15](#_Toc209602409)

[UC1: Pilot reserviert Flugzeug 15](#_Toc209602410)

[UC2: Pilot startet Flug (Zählerstand) 15](#_Toc209602411)

[UC3: Admin verwaltet Benutzer 16](#_Toc209602412)

[UC4: Wart pflegt Wartung/ARC am Flugzeug 16](#_Toc209602413)

[UC5: Abrechnung exportiert Monatsbericht 16](#_Toc209602414)

[UC6: Anmeldung & RBAC 16](#_Toc209602415)

[UC7: Tower-Mitarbeiter protokolliert Flüge in der Start-/Landeliste 16](#_Toc209602416)

# Einleitung

Das Projekt „Digitales Flugplatzmanagement“ soll die Arbeit am Flugplatz einfacher und moderner machen. Ziel ist es, Verwaltung und Abrechnung digital abzuwickeln. Dafür gibt es mehrere Bereiche, die alle miteinander verbunden sind. Die Benutzer werden zentral verwaltet, mit unterschiedlichen Rollen und Rechten.

Geplant ist eine moderne Webanwendung, die wichtige Aufgaben wie Start- und Landelisten, Pilotenverwaltung, Flugzeugreservierungen und Rechnungsstellung in einem System zusammenfasst. So können alle relevanten Daten schnell und sicher verarbeitet werden.

Das neue System ersetzt die alte, veraltete Software und bringt eine stabile, erweiterbare und einfach zu bedienende Lösung, die sowohl am Computer als auch am Handy genutzt werden kann. Dadurch werden die Abläufe am Flugplatz deutlich verbessert.

# Allgemeines

Dieses Pflichtenheft beschreibt, wie das System „Digitales Flugplatzmanagement“ im Rahmen einer Diplomarbeit entwickelt und umgesetzt wird. Während der Programmierung wird es schrittweise getestet und am Ende direkt im echten Betrieb eingesetzt. Ziel ist es, eine zuverlässige, sichere und leicht zu bedienende Webanwendung zu erstellen, auf die auch per Handy zugegriffen werden kann.

## Teams und Schnittstellen

|  |  |  |
| --- | --- | --- |
| **Rolle(n)** | **Name** | **E-Mail** |
| Projektleiter | Julian Halbmayr | julian.halbmayr@htlwy.at |
| Projektleiter  (Stellvertreter) | Gabriel Deiac | gabriel.deiac@htwly.at |
| Hauptbetreuer | Prof. Ing. Mag. Andreas Schatz | sz@htlwy.at |
| Stellvertretender Betreuer | Prof. Alexander Greil, BEd, MSc | ga@htlwy.at |
| Auftraggeber | DI Dr. Michael Maurer | michael.maurer@gaminside.com |

## Ziele

### Zentrale Daten- und Benutzerverwaltung

* Einrichtung einer gemeinsamen Datenbasis für alle Teilbereiche des Systems (Start-/Landelisten, Pilotenverwaltung, Reservierung, Abrechnung).
* Implementierung eines zentralen Usermanagements mit rollenbasierten Berechtigungen auf Grundlage der Pilotenverwaltung.

### Sicherer Web-App Zugriff

* Bereitstellung einer sicheren webbasierten Zugriffsmöglichkeit für berechtigte Nutzer, um auch von extern (z. B. Admin, Pilot) auf die Web-App zuzugreifen.
* Umsetzung eines umfassenden Sicherheitskonzepts mit sicherer Authentifizierung, verschlüsselter Datenübertragung und rollenbasiertem Zugriffsschutz.

### Bereitstellung von Systemdiensten

* Entwicklung einer Web-Applikation zur Verwaltung von Start- und Landelisten, Piloten, Reservierungen und Abrechnungen.
* Aufbau einer zentralen Datenbank für Flugzeuge, Piloten, Reservierungen usw.
* Bereitstellung von Exportfunktionen (PDF, CSV) für Berichte und Abrechnungen.

### Skalierbarkeit und Zuverlässigkeit

* Sicherstellung, dass das System jederzeit erweiterbar ist (z. B. zusätzliche Flugzeuge, Vereine oder neue Funktionen).
* Gewährleistung hoher Verfügbarkeit durch stabile Server- und Datenbankarchitektur.
* Durchführung regelmäßiger Datensicherungen mit der Möglichkeit zur vollständigen Wiederherstellung im Fehlerfall.

### Benutzer- und Datenmanagement

* Effiziente Verwaltung der Benutzerrollen (Pilot, Schüler, Wart, Admin, Abrechnungsverantwortlicher) mit klaren Berechtigungen.
* Einheitliche und konsistente Speicherung aller Flug-, Piloten- und Flugzeugdaten.

### Test- und Produktivsystem

* Testbetrieb des Systems in einer Entwicklungsumgebung.
* Bereitstellung im Produktivbetrieb auf dem Server des Auftraggebers.

### Systemsicherheit

* Umsetzung eines mehrstufigen Sicherheitskonzepts mit Authentifizierung, Autorisierung und verschlüsselter Kommunikation.
* Schutz vor unbefugtem Zugriff auf sensible Daten (personenbezogene Daten).

## Funktionale Anforderungen

### Anforderung 1: Zugriff auf Systemdienste

* Web-Applikation: Muss einen sicheren und stabilen Zugriff für berechtigte Nutzer ermöglichen – sowohl lokal am Flugplatz als auch extern über das Internet.
* Zentrale Datenbank (MariaDB): Muss alle relevanten Informationen zu Benutzer, Flugzeugen, Reservierungen und Flügen strukturiert speichern und verwalten.
* Export- und Berichtsfunktion: Muss PDF- und CSV-Exporte für Rechnungsstellung und Berichte bereitstellen.
* Benutzer- und Rollenverwaltung: Muss rollenbasierte Berechtigungen (z. B. Pilot, Schüler, Wart, Admin, Abrechnungsverantwortlicher) unterstützen und deren Rechte systemweit anwenden.

### Anforderung 2: Systemverfügbarkeit und Infrastruktur

* Zugriffssicherheit: Zugriff auf Web- und App-Funktionen muss verschlüsselt (HTTPS) und durch sichere Authentifizierung abgesichert werden.
* Zentrale Verwaltung: Alle Daten, Konfigurationen und Benutzerrechte müssen zentral in der Web-Applikation gepflegt werden können.
* Synchronisation in Echtzeit: Änderungen in Web-App müssen unmittelbar und konsistent in allen verbundenen Systemen verfügbar sein.
* Datensicherung und Wiederherstellung: Regelmäßige Backups müssen automatisch erstellt und sicher gespeichert werden. Es muss ein Konzept zur Wiederherstellung bei Datenverlust existieren.

### Anforderung 3: Sicherheitsmaßnahmen

* Authentifizierung und Autorisierung: Zugriff auf Funktionen und Daten muss abhängig von der Benutzerrolle gesteuert werden (Role-Based Access Control).
* Verschlüsselte Datenübertragung: Alle Daten zwischen Client (Web/App) und Server müssen per HTTPS verschlüsselt übertragen werden.
* Anwendungssicherheit: Schutz vor typischen Angriffen wie SQL-Injection.
* Zugriffsschutz: Unbefugter Zugriff auf sensible Daten muss verhindert werden.

## Nichtfunktionale Anforderungen

### Allgemeine Anforderungen

* Verfügbarkeit: Das System (Web-Applikation, Datenbank) muss für berechtigte Nutzer rund um die Uhr verfügbar sein.
* Benutzerfreundlichkeit (UI/UX): Die Oberfläche muss klar strukturiert und optisch ansprechend sein. Das Layout soll einheitlich und leicht bedienbar sein, optimiert für PC, Laptop und mobile Endgeräte.
* Performance: Alle Anfragen und Datenabrufe sollen möglichst verzögerungsfrei verarbeitet werden.
* Plattformunabhängigkeit: Die WebApp muss auf verschiedenen Endgeräten (Laptop/PC, Tablet, Smartphone) funktionsfähig sein.

### Qualitäts- und Sicherheitsanforderungen

* Datensicherheit: Kommunikation zwischen Client (Web/App) und Server muss über HTTPS erfolgen.
* Datenschutz: Sämtliche Datenerfassungen, -speicherungen und -übertragungen müssen den geltenden Datenschutzgesetzen entsprechen.

### Entwicklungs- und Projektmanagement-Anforderungen

* Agiles Vorgehen: Entwicklung nach agilen Methoden mit regelmäßigen Abstimmungen, Feedbackrunden und flexibler Anpassung der Arbeitsschritte.
* Projektorganisation: Nutzung von Projektmanagement-Tools zur Aufgabenverteilung, Statusverfolgung und Dokumentation des Fortschritts.
* Saubere Code-Architektur: Trennung von Datenzugriff, Logik und Darstellung zur besseren Wartbarkeit („Clean Code“-Prinzipien).
* Backup-Konzept: Regelmäßige Datensicherungen und ein funktionierendes Wiederherstellungssystem müssen implementiert werden.

## Rahmenbedingungen

<Gantt-Diagramm.xlsx>

## Technische Anforderungen

### Server-Hardware

* Webserver zur Bereitstellung der Anwendung und Verarbeitung der Benutzeranfragen.
* Datenbankserver für die Speicherung aller Flugdaten, Benutzerinformationen und Flugzeugen (MariaDB).
* Backup-Speicher für regelmäßige Datensicherungen.

### Client-Hardware

* Zugriff auf das System per Desktop-PC, Laptop oder Smartphone über einen aktuellen Webbrowser (z. B. Chrome, Firefox, Edge, Safari).
* Keine spezielle App erforderlich, mobile Nutzung erfolgt über eine optimierte Weboberfläche.

### Software

* Betriebssystem (Server): Windows 11
* Webserver: Apache.
* Programmiersprache: C#
* Datenbank: MariaDB
* Backup-Software: Automatisierte Sicherungslösung mit zeitgesteuerten Backups.
* Sicherheitssoftware: SSL-Zertifikat (HTTPS), Firewall-Konfiguration am Server.

### Sicherheitsanforderungen

* Zugriffsschutz: Rollenbasierte Berechtigungen, sichere Benutzeranmeldung.
* Verschlüsselung: HTTPS-Verschlüsselung für alle Datenübertragungen.
* Sicherheitskonzept: Schutz vor SQL-Injection.
* Backup- und Wiederherstellungskonzept: Regelmäßige Datensicherung und Möglichkeit zur vollständigen Wiederherstellung im Fehlerfall.

## Funktionsbeschreibung der Anwendung

### Anmeldung & Authentifizierung

* Login mit E-Mail/Passwort (verschlüsselt, HTTPS).
* Rollenbasierter Zugriff (Pilot, Wart, Admin).
* Ein Bild, das Text, Screenshot, Wolke, Himmel enthält.

  KI-generierte Inhalte können fehlerhaft sein.Fehlermeldungen bei Fehlversuchen.

### Start-/Landeliste (Flugerfassung)

* Übersicht der Flüge des gewählten Tages inkl. Status (EF/AF), Zeiten, Personen, Flugart, Kostenstelle, Gebühren, bezahlt.
* Aktionen: Neuer Flug, Kopieren, Schleppzug, Start, Touch & Go, Landung, Berichte, Rechnung Landegebühren, Rechnung Benzin, Löschen, Einstellungen (Admin).
* Filter/Suche, Sortierung, Export (CSV/PDF).

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

KI-generierte Inhalte können fehlerhaft sein.

### Pilotenseite

* Auswahl eines Flugzeugs; sofortige Anzeige der Verfügbarkeit (im Kalenderformat).
* Ein Bild, das Text, Screenshot, Himmel, Cartoon enthält.

  KI-generierte Inhalte können fehlerhaft sein.Aktionen: Reservieren, Fliegen (Direktstart).

### Fliegen (Zählerstand)

* Eingabe und Validierung des Zählerstands; Plausibilitätsprüfung gegen letzten Stand und erwartete Werte.
* Start/Ende-Buttons; Fehlermeldung mit Korrekturvorschlag bei Abweichungen.

Ein Bild, das Text, Wolke, Screenshot, Himmel enthält.

KI-generierte Inhalte können fehlerhaft sein.

### Admin – Flugzeugverwaltung

* CRUD für Flugzeuge mit Kennzeichen, Typ, Kostenstelle, Zählerstand, Landekosten, Verfügbarkeit.
* Wartungsverwaltung: ARC-Datum, Flugzeitwartung, Art der Wartung, Freigabestatus; Statusfärbung (grün/gelb/rot).

Bild wird überarbeitet

### Admin – Benutzerverwaltung

* CRUD für Benutzer (NR, Name, E-Mail, Rolle, Kostenstelle).
* Ein Bild, das Text, Software, Computersymbol, Webseite enthält.

  KI-generierte Inhalte können fehlerhaft sein.Filter und Massenaktionen (Hinzufügen/Löschen).

### Admin – Kostenstellen

* Ein Bild, das Text, Software, Webseite, Computersymbol enthält.

  KI-generierte Inhalte können fehlerhaft sein.CRUD mit Stammdaten (Name, Straße, PLZ, Ort).

### Admin – Pilotflüge

* Übersicht/Filter der Flüge je Pilot inkl. Kennzeichen, Zeiten, Flugart, Kostenstelle.
* Filterdialoge (global)
* Ein Bild, das Text, Software, Zahl, Schrift enthält.

  KI-generierte Inhalte können fehlerhaft sein.Einheitliche Filter-Overlays für Benutzer, Flugzeuge, Kostenstellen, Pilotflüge, Reservierungen mit Zurücksetzen/Abbrechen/Anwenden.

### Reservierung

* Erstellen/Bearbeiten/Löschen von Reservierungen: Datum, Start-/Endzeit, Flugart, optional Gastname/-kontakt.
* Kollisionsprüfung mit bestehenden Reservierungen und Wartungsfenstern.
* Ein Bild, das Text, Screenshot, Software, Zahl enthält.

  KI-generierte Inhalte können fehlerhaft sein.Anzeige der Reservierungen in der Admin- und Pilot-Ansicht.

### Berichte & Exporte

* PDF/CSV-Export für Abrechnungen.
* Monatsreport je Kostenstelle/Flugzeug.

## Use Cases

### UC1: Pilot reserviert Flugzeug

Akteure: Pilot (Primär), System

Vorbedingungen: Pilot ist angemeldet UND Flugzeugdaten vorhanden

Hauptablauf:

1. Pilot öffnet Pilotenseite und wählt ein Flugzeug.
2. System zeigt Verfügbarkeit des Flugzeugs.
3. Falls das Flugzeug belegt ist, kann der Pilot den Kalender des ausgewählten Flugzeugs öffnen.
   * Der Kalender zeigt bestehende Reservierungen dieser Woche und freie Slots.
4. Pilot wählt Datum, Start- und Endzeit, Flugart.
5. System prüft auf Kollisionen; bei Erfolg speichert es die Reservierung.
6. Bestätigung wird angezeigt.

Alternativen:

* 3a) Pilot möchte ein anderes Flugzeug wählen → Dropdown-Menü für Flugzeugwechsel; nach Auswahl wird der Kalender für das neue Flugzeug angezeigt.
* 4a) Kollision → Hinweis und Auswahl einer anderen Zeit.
* 4b) Wartung → Reservierung blockiert.

### UC2: Pilot startet Flug (Zählerstand)

Akteure: Pilot (Primär), System  
Vorbedingungen: Reservierung vorhanden ODER Flugzeug frei; Pilot angemeldet  
Hauptablauf:

1. Pilot öffnet „Fliegen“.
2. System zeigt letzten Zählerstand.
3. Pilot gibt aktuellen Zählerstand ein und klickt auf Start.
4. System validiert und legt Flugstart an.  
   **Alternativen:**

* 3a) Abweichung → Fehlermeldung mit korrektem Vorschlag
* 3b) keine Berechtigung → Zugriff verweigert

### UC3: Admin verwaltet Benutzer

Akteure: Admin  
Vorbedingungen: Admin angemeldet  
Hauptablauf: Liste öffnen → Filter anwenden → Benutzer hinzufügen/bearbeiten/löschen → Änderungen speichern

### UC4: Wart pflegt Wartung/ARC am Flugzeug

Akteure: Wart, Admin  
Vorbedingungen: Flugzeug existiert  
Hauptablauf: Flugzeug aufrufen → Wartungsdaten bearbeiten → Statusfarben aktualisieren → Freigabestatus setzen

### UC5: Abrechnung exportiert Monatsbericht

Akteure: Abrechnungsverantwortlicher  
Hauptablauf: Start-/Landeliste filtern → Export als CSV/PDF → Datei speichern/weitergeben

### UC6: Anmeldung & RBAC

Akteure: Nutzer, System  
Hauptablauf: Login → Rolle ermitteln → Navigation und Berechtigungen entsprechend freischalten

### UC7: Tower-Mitarbeiter protokolliert Flüge in der Start-/Landeliste

Akteure: Tower-Mitarbeiter (Primär), System

Vorbedingungen: Tower-Mitarbeiter ist angemeldet UND Start-/Landeliste ist geöffnet

Hauptablauf:

1. Flugzeug kommt zur Start- oder Landemeldung.
2. Tower-Mitarbeiter legt in der Start-/Landeliste einen neuen Eintrag an.
3. Eingabe von Piloten, Flugzeug, Flugart, Kostenstelle, Uhrzeit, ggf. Gastdaten.
4. Bei Start wird der Status auf „gestartet“ gesetzt, bei Landung auf „gelandet“.
5. System speichert die Daten und aktualisiert die Liste in Echtzeit.

Alternativen:

* 3a) Touch & Go → zusätzlicher Eintrag oder Markierung im aktuellen Flug.
* 3b) Schleppflug → Eingabe zusätzlicher geschleppter Flugzeuge.
* 4a) Nachträgliche Korrektur → Änderungen sind möglich, werden aber im Änderungsprotokoll dokumentiert.