

Backend-Business-Logik – vollständiger Entwurf (v1)

Inhalt

Backend-Business-Logik – vollständiger Entwurf (v1)	1
1. Klassen	1
2. Zustände & Übergänge (State Machines).....	5
2.1 Flugzeug.....	5
2.2 Reservierung.....	5
2.3 Pilotenflug.....	5
2.4 Start-/Landeliste (Movement)	6
3. Validierungen.....	6
4. Berechnungen	6
4.1 Blockminuten.....	6
4.2 Flugkosten.....	6
4.3 Landegebühr.....	6
4.4 Steuern & Beträge	6
5. Rollen.....	7
6. Fehlercodes (Beispiele)	7
7. Konfigurierbarkeit.....	7
8. Offene Punkte (optional, falls später gebraucht).....	7

1. Klassen

t_kostenstelle

- Id
- Name
- Strasse
- PLZ
- Ort
- Aktiv

t_benutzer

- Id
- Vorname
- Nachname
- Email
- Passwort
- KostenstelleId → *t_kostenstelle.Id*

t_rolle

- Id
- Rollentyp

t_benutzerrolle

- BenutzerId → *t_benutzer.Id*
- RolleId → *t_rolle.Id*
- **Primärschlüssel:** (BenutzerId, RolleId)

t_preiskategorie

- Id
- Kategorie
- Preis

t_flugart

- Id
- Flugtyp

t_flugzeugtyp

- Id
- Flugzeugtyp
- PreisId → *t_preiskategorie.Id*

t_flugzeug

- Id
- Kennzeichen
- Zaehlerstand (DECIMAL(6,1))
- Grounded (BOOLEAN)
- FlugzeugtypId → *t_flugzeugtyp.Id*
- KostenstelleId → *t_kostenstelle.Id*
- DatumARC
- DatumFlugzeitwartung
- Wartungsart

t_flugminutenkosten

- Id
- FlugartId → *t_flugart.Id*
- FlugzeugId → *t_flugzeug.Id*
- KostenMinuten
- **Unique:** (FlugartId, FlugzeugId)

Vererbungsstruktur

t_flug (*Basistabelle*)

- Id
- FlugzeugId → *t_flugzeug.Id*
- FlugartId → *t_flugart.Id*
- StartTs
- EndeTs
- Bemerkung
- Status ∈ {Geplant, StartErfasst, EndeErfasst, Storniert}

t_reservierung (*is-a t_flug*)

- Id → *t_flug.Id*
- Pilot (VARCHAR)
- Datum
- Gastname
- Gastkontakt
- Status ∈ {Offen, Erledigt, Storniert}

t_pilotenflug (*is-a t_flug*)

- Id → *t_flug.Id*
- Pilot (VARCHAR)
- StartZaehler (DECIMAL(6,1))
- EndZaehler (DECIMAL(6,1))
- DauerMinuten (berechnet)

t_startlandeliste

- Id
- Einflug (BOOLEAN)
- Ausflug (BOOLEAN)
- Startzeit
- Landezeit

- Bemerkung
- Bezahlte (BOOLEAN)
- Pilot1
- Pilot2
- FlugzeugId → *t_flugzeug.Id*
- FlugartId → *t_flugart.Id*

2. Zustände & Übergänge (State Machines)

2.1 Flugzeug

- **Zustände:** Aktiv, Gegroundet
- **Übergang Gegroundet→Aktiv:** nur Rolle {Wart, Admin} ändern
- **Nebenwirkungen:** Reservieren/Starten blockieren

2.2 Reservierung

- **Zustände:** Offen → {Erledigt | Storniert}
- **Regeln:**
 - Kollisionen verboten
 - Flugzeug muss Aktiv sein zur Anlage
 - Storno: Slot freigeben, referenzierte Flüge trennen

2.3 Pilotenflug

- **Zustände:** StartErfasst → EndeErfasst
- **Start:** Zählerstand == letzter Zählerstand (sonst Fehler; Override nur {Wart, Admin})
- **Ende:** Zählerstand := eingabe

2.4 Start-/Landeliste (Movement)

- **Zustände pro Zeile:** Offen → Beahlt
- **Regeln:**
 - Startzeit \geq Landezeit (falls Start vorhanden)
 - EF/AF konsistent (EF=Einflug? AF=Abflug?)
 - Checkliste erfüllt
 - Landegebühr berechnen, Aktionen: Neu/Kopieren/Schlepp/Touch&Go, Rechnungen erzeugen

3. Validierungen

- **Zeiten:** endzeit > startzeit; Reservierung innerhalb Tagesgrenzen

4. Berechnungen

4.1 Blockminuten

blockmin = Startzähler – Endzähler

4.2 Flugkosten

preis_minute = in Flugzeug oder in der Preiskategorie

flugkosten = blockmin * preis_minute

4.3 Landegebühr

landung_basis = flugzeug preis

4.4 Steuern & Beträge

netto = sum(positionen)

steuer = netto * steuersatz

brutto = netto + steuer

5. Rollen

Aktion	Normal/Schüler	Wart	Abrechnung	Admin
Login, eigene Reservierungen/Flüge ansehen	✓	✓	✓	✓
Reservierung anlegen/ändern (aktiv. Flzg)	✓	✓	✓	✓
Zähler-Override bei Startfehler	✗	✓	✗	✓
Flugzeug Grounding ändern	✗	✓	✗	✓
Preismodell/Preis-Override setzen	✗	✓	✓	✓
Rechnungen erzeugen/versenden	✗	✗	✓	✓
CRUD	✗	✗	✗	✓

6. Fehlercodes (Beispiele)

- RESERVATION_CONFLICT – Zeitkollision
- AIRCRAFT_GROUNDED – Aktion auf gegroundetem Flugzeug
- COUNTER_MISMATCH – Startzähler passt nicht
- ROLE_FORBIDDEN – Rolle verletzt

7. Konfigurierbarkeit

- Preislisten versioniert ab Datum
- Pflichtfelder je Klasse

8. Offene Punkte (optional, falls später gebraucht)

- DSGVO: Datenminimierung, Löschkonzept