

# Špecifikácia softvéru – tvorba internetových aplikácií

*Ivan Havran*

## Základný popis internetovej aplikácie

### O akú aplikáciu sa jedná

AircraftManager je webová aplikácia zameraná na manažovanie lietadiel, letov a letísk. Umožňuje plánovať lety medzi letiskami, údržbu jednotlivých lietadiel či správu samotného letiska. Aplikácia obsahuje priateľivé užívateľské prostredie na interakciu s jednotlivými prvkami systému.

Aplikácia umožní naplánovať let alebo údržbu na daný deň v kalendári. Užívatelia interagujú s aplikáciou z pozície rôznych rolí. Podľa role konkrétneho užívateľa sa určujú práva.

### Kto je cieľová skupina používateľov

Aplikácia cieľi na celý personál zabezpečujúci prepravu lietadlom. Pri vhodne zvolenej štruktúre je možné celý systém plánovania letov a údržby riadiť v jednej aplikácii. Cieľom je poskytovať vyvážený systém – orientovať sa na celkovú aplikáciu a nie na konkrétnu časť (napr. len plánovanie údržby a podobne).

## Používateľské požiadavky

### Popis jednotlivých používateľských rolí

- Admin – správca celého systému
- AirportManager – správca konkrétneho letiska
- AircraftController – správca všetkých lietadiel
- AircraftMaintainer – správca údržby konkrétneho lietadla
- Mechanic – člen tímu mechanikov lietadla
- Pilot – pilot lietadla
- Steward – člen posádky lietadla
- CrewAdmin – správca posádok lietadiel
- Planner – plánovač letov

### Popis prípadov použitia pre jednotlivé používateľské role

Ako **Admin** chcem byť schopný vykonávať všetky operácie, aby som mohol mať nad aplikáciou potrebnú kontrolu.

Ako **Admin** chcem vedieť ľubovoľne manipulovať s rolami ostatných užívateľov, aby som vedel zabezpečiť bezpečné rozdelenie práv.

Ako **AirportManager** chcem byť schopný manažovať všetky zdroje letiska, aby som nad ním mal kontrolu.

Ako **AirportManager** chcem vidieť všetky lety týkajúce sa môjho letiska.

Ako **AirportManager** chcem umožniť konkrétnemu **Planner**-ovi plánovať lety na moje letisko.

Ako **AircraftController** chcem vedieť spravovať všetky lietadlá a ich typy.

Ako **AircraftMaintainer** chcem vedieť plánovať a prezerať naplánovanú údržbu pre stroje ktoré spravujem.

Ako **Steward** chcem vedieť moje naplánované lety.

Ako **Pilot** chcem vidieť moje naplánované lety.

Ako **Mechanic** chcem vidieť program naplánovanej údržby pre moje lietadlá.

Ako **CrewAdmin** chcem manažovať zloženie letových posádok.

Ako **Planner** chcem vedieť plánovať lety pre konkrétne letiská.

Ako **Planner** chcem vidieť všetky naplánované lety, aby som vedel prispôsobiť nový let.

Ako **Planner** chcem vedieť plánovať lety pre konkrétne letiská.

Ako **Planner** chcem vidieť všetkých členov posádok (**Pilot** a **Steward**) aby som mohol naplánovať let.

## Dátový model

### Popis jednotlivých tabuliek a ich polí

Okrem parametrov definovaných v nasledujúcich tabuľkách má každý objekt ešte tri parametre navyše:

- `_id` = interné **id** objektu. Je unikátne v rámci jednej kolekcie. V niektorých prípadoch bude použité ako primárny kľúč. Je to 12-byte BSON typ hexadecimálneho stringu.
- `deleted` = bool hodnota, ktorá signalizuje, či je daný objekt zmazaný. Slúži na bezpečné mazanie dokumentov (soft vs. hard delete)
- `__v`: tzv. **version key**. Objekt má aj svoju internú verziu, ktorá sa zvyšuje pri zmene tohto dokumentu. Slúži na konzistenciu objektov pri asynchrónnych volaniach.

#### User

- `username`: str
- `password`: str
- `fullName`: str
- `email`: str
- `roles`: Role[]

#### Role

- `name`: str
- `concerningId`: str (id/code ktoreho sa tyka dana rola – moze byt aj prazdny string (Admin, Pilot))

#### Aircraft

- `code`: str – slúži aj ako primary key

- aircraftTypeCode: str
- homeAirportCode: str

### **Airport**

- name: str
- code: str – slúži aj ako primary key
- address: str
- managerId: str

### **AircraftType**

- name: str
- code: str – slúži aj ako primary key
- weight: float
- height: float
- width: float
- numberOfPlaces: int

### **AircraftCrew**

- name: str
- mainPilotId: str
- secondPilotId: str
- memberIds: str[]

### **MechanicCrew**

- name: str
- memberIds: str[]
- aircraftTypeCodes: str[]
- homeAirportCode: str – domovské letisko tejto skupiny

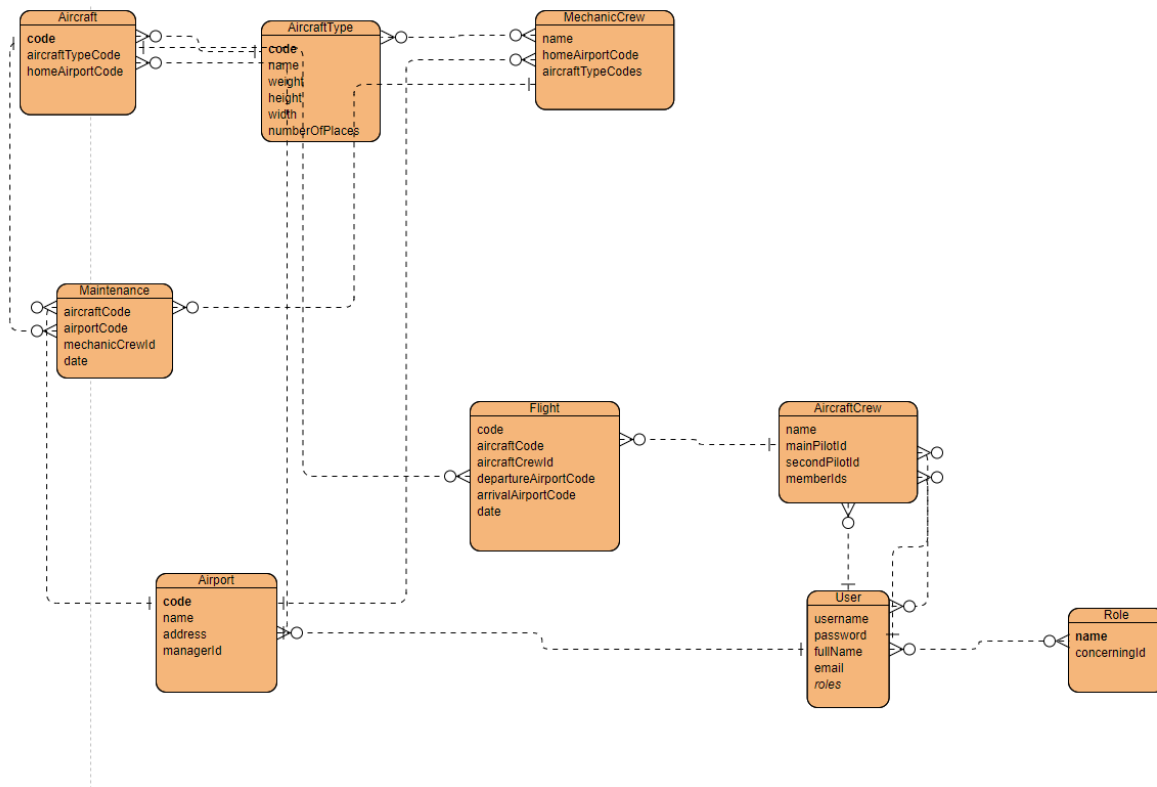
### **Flight**

- code: str
- aircraftCode: str
- aircraftCrewId: str
- departureAirportCode: str
- arrivalAirportCode: str
- date: Date

### **Maintanance**

- aircraftCode: str
- airportCode: str
- mechanicCrewId: str
- date: Date

## Entitno-relačný diagram popisujúci vzťahy tabuliek v databáze



## Technologické požiadavky

Aplikácia je vybudovaná na báze MERN stacku (MongoDB, Express, React, Node.js) – preto sa od nich odvíjajú aj ostatné technológie. Pri tvorbe tech-stacku som sa inšpiroval týmto videom - <https://www.youtube.com/watch?v=CvCiNeLnZ00>.

S väčšinou technológií som sa stretol v mojej bývalej práci, kde som na tomto stacku vyvíjal jeden rok. Niektoré z technológií boli obalené custom knižnicami, ale práca s nimi nebola výrazne rozdielna (Redux forms, Mongoose). Pri autentikácii alebo hostingu som ale volil na základe videa (keďže obe tieto knižnice som používal minimálne – boli v kompetencii iných tímov a boli z väčšiny vytvorené na mieru)

### Client-side

Na client-side využijem React. FE časť aplikácie je oddelená v samostatnom repozitári. Redux použijem na tvorbu forms, ktorými sa budú obsluhovať niektoré operácie.

### Server-side

Štandardný Node.js.

### Interface client-side

RTX Query na optimalizovanú komunikáciu medzi obidvoma stranami klient-server.

## Hosting

Na hosting aplikácie plánujem použiť **Render** (<https://render.com/>). Na ukážku funkčnej aplikácie by mal postačovať a rovnako nepredpokladám problém s integráciou.

## Datábaza

MongoDB vo verzii 5.0.14 – vytvorený cluster vo free verzii na AWS (funkcionalitu poskytuje priamo MongoDB, len je potrebné nastaviť). MongoDB je NoSQL databáza, namiesto tabuliek používa kolekcie, v ktorých sa vyskytujú dokumenty (jednotlivé objekty).

Následné pripojenie a práca s databázou prebieha za pomoci frameworku Mongoose vo verzii 6.9.1. Táto knižnica poskytuje jednoduchšiu prácu (optimalizované volania, validácia, jednoduchšie pripojenie k databáze).

## Zoznam podporovaných prehliadačov

Základnú podporu bude mať aplikácia pre prehliadače Google Chrome a Mozilla Firefox. V prípade dostatku času rozšírim aj podporu pre Safari, ale táto úloha bude optional.

## Časový plán

### 12.3 – 19.3.2023 Week 1

- Backend:
  - Základný setup aplikácie – server-side 2h
  - základné CRUD operácie pre všetky modely 6h
  - otestovanie základných operácií pomocou Postman-a 2h
- Frontend:
  - v tomto cykle nie programovanie, iba návrhy jednotlivých stránok 2h

### 19.3 – 26.3.2023 Week 2

- Backend:
  - Z začať pracovať na komplexnejších use-casoch (vyžadujúce prechádzanie viacerých kolekcii a pod.) 2h
- Frontend
  - Úvodná stránka – v budúcnosti sa bude vyžadovať prihlásenie – usporiadať layout na tejto stránke 3-4 h (predpokladám možno aj vyšší čas kvôli malej skúsenosti s frontendom)
  - Z začať pracovať na stránke po prihlásení – úvodná stránka ktorá bude mať naľavo menu s jednotlivými sekciami (aspoň 1-2 h v závislosti na rýchlosti práce na prvom bode plánu)

### 26.3 – 2.4.2023 Week 3

- Backend
  - dokončenie use-casov, ktoré budú v beta verzii 3h
  - prepojenie client-server side 2h
- Frontend

- Výrazne pokročiť s ostatnými stránkami (jednotlivé sekcie) a začať pracovať na forms pre vytváranie objektov 4h

## **2.4 – 9.4.2023 Week 4**

- Backend
  - Deploynúť aplikáciu na hostingovú službu 2-3h
  - Vyladenie autentikácie, ktorá je rozpracovaná z minulého cyklu 2-3h
- Frontend
  - Dokončiť forms a stránky pre beta verziu 4h

## **11.4.2023 Beta projekt**

## **9.4 – 16.4.2023 Week 5**

- Backend
  - Role-based access control – vytvoriť a prepojiť systém kontroly práv so skutočnými use-cases 5h
  - Integrovať autentikáciu a autorizáciu 3h
- Frontend
  - prepojiť stránku s prihlasovaním na implementovaný backend 3h

## **21.4.2023 Feedback k beta projektom**

## **16.4 – 23.4.2023 Week 6**

- Backend
  - Dokončiť všetky use-cases navrhnuté v špecifikácii 4h
- Frontend
  - prepojiť stránku s prihlasovaním na implementovaný backend 3h

## **23.4 – 30.4.2023 Week 7**

Zpracovanie zozbieranej spätnej väzby. ?h

Záverečný polishing problémových funkcií (ktoré budú vyšpecifikované v reporte z Week 6) ?h

## **2.5.2023 Odovzdanie plnej verzie**

## **Future work**

V budúcnosti by bolo možné systém rozšíriť aj o ďalšie časti – napr. plánovanie dopravnej letovej prevádzky, rozšírenie na viac zamestnancov letiska a podobne).

Ďalším možným rozšírením by bola akási deľba jednej inštancie aplikácie na viaceré – rozdeliť ich napríklad podľa regiónov a po integrácii prepojiť len nutné súčasti systému.