**Aeroport**

**Descrierea proiectului** : Se va realiza o aplicație web/desktop care să gestioneze activitatea unui aeroport. Aceasta va monitoriza zborurile viitoare, cu ce avion, piloți și însoțitori de zbor se vor desfășura. Aplicația va permite oricărui utilizator să fie la curent cu zborurile ce urmează a fi efectuate, întârzieri estimate sau evetuale zboruri anulate și respectivele reprogramări. Dacă aceștia vor să ia un anumit avion pentru a ajunge la destinația dorită, vor putea achiziționa un bilet în limita locurilor disponibile. De asemenea, după efectuarea zborului, vor putea oferi un feedback companiei cu avionul cărora au călătorit pentru a-i ajuta pe aceștia să îmbunătățească condițiile de zbor. Vor exista și utilizatori cu rol de administrator care vor asigura buna funcționare a aplicației. Ei sunt cei care vor introduce sau elimina un anumit zbor, avion, pilot, asistent de zbor, sau vor actualiza informațiile asociate acestora.

**Tabelele alese împreună cu acțiunile întreprinse de utilizatori în cadrul aplicației**:

**flights** – tabelă ce stochează informații despre zborurile viitoare; administratorii interacționează cu această tabelă, introducând în aplicație un nou zbor, eliminând unul efectuat sau actualizând o anumită informație, urmând ca apoi să anunțe public modificarea făcută; utilizatorii obișnuiți acceseaza aplicația prin logare și se pun la curent, pe baza acestei tabele, cu zborurile viitoare împreună cu detaliile asociate lor

* ID – identificatorul unic al zborului; reprezintă cheia primară a tabelei
* departure\_date – data exactă (zi/lună/an oră minut) la care avionul va decola
* duration – durata totală a zborului în minute
* delay – numărul de minute cu care se estimează o întârziere până la destinația finală
* distance – distanța parcursă în kilometri
* stopovers – numărul de opriri pe care le face avionul în alte aeropoarte până la aeroportul ce constituie destinația finală
* airport\_code – identificatorul unic al aeroportului destinație finală; reprezintă o cheie străină ce modelează legătură one-to-many dintre tabela flights și tabela airports
* airplane\_ID – identificatorul unic al avionului cu care se va desfășura zborul; cheie străină ce stă la baza legăturii one-to-many dintre tabela flights și tabela airplanes

**airplanes** – tabelă ce stochează informații despre avioanele cu care se vor desfașura zborurile; este menținută de către administratori și este folosită pentru a îmbogăți interfața cu utilizatorul cu diverse informații despre avioanele antrenate în zbor

* ID – identificatorul unic al avionului
* first\_class\_seats – numărul de locuri la clasa I
* second\_class\_seats – numărul de locuri la clasa a II-a
* company\_ID – identificatorul companiei căreia îi aparține avionul
* available
  + 1 – avionul este în decursul unui zbor sau este în curs de reparații
  + 0 – avionul este disponibil și i se poate atribui un zbor viitor atâta timp cât nu se suprapune cu un alt zbor viitor al aceluiași avion

**companies** – tabelă ce conține informații despre companiile ale căror avioane figurează în aeroport; utilizatorul va putea furniza companiei, pe baza acestor date, un feedback în funcție de cât de mulțumit a fost de condițiile de zbor

* ID – identificatorul unic al companiei
* name, adress, city, phone\_number, email – datele de identificare ale companiei

**airports** – tabelă ce conține toate aeropoartele în care se poate ajunge din aeroportul curent; utilizatorul va putea cere aplicației să-i furnizeze toate zborurile către destinația dorită, iar aceasta va apela la serverul de baze de date care va interoga baza de date corespunzător

* code – identificatorul unic al aeroportului
* name, adress, city, phone\_number, email – datele de identificare ale aeroportului

**users** – tabelă a cărei bună funcționare depinde de administrator; datele introduse de utilizatorii care doresc să-și creeze cont sunt preluate de aplicație care le trimite la serverul de baze de date, care apoi le va insera în baza de date; piloții și asistenții de zbor sunt introduși de către administratori pe baza certificatelor care atestă pregătirea lor

* ID – identificatorul unic al persoanei
* first\_name, last\_name, date\_of\_birth, adress, city, phone\_number, email – datele personale ale persoanei
* type:
  + user – pentru utilizatori normali care și-au creat cont și interacționează cu aplicația; în momentul în care un utilizator își cumpără bilet, acesta devine și pasager
  + pilot – pilot de avion
  + FA – însoțitor de zbor
  + admin – administrator al bazei de date (operator responsabil cu întreținerea aplicației)
* password – parola contului persoanei
* loged:
  + 0 – utilizatorul nu este logat
  + 1 – utilizatorul este logat

**flight\_attendants** – tabelă ce conține însoțitorii de zbor

* **rank** – atribut ce poate fi:
  + 1 – însoțitor experimentat responsabil cu clasa I
  + 2 - însoțitor experimentat responsabil cu clasa II
  + 3 – asistent de însoțitor de zbor

**pilots** – tabelă ce conține piloții aeroportului

* rank – atribut ce poate fi:
  + 1 – pilot principal, comandant
  + 2 – pilot secundar

**passengers** – tabelă ce contine pasagerii diferitelor zboruri; se află în relație directă cu tabela tickets întrucât o persoană poate deveni pasager abia după achiziționarea unui bilet

* rank – atribut ce poate fi:
  + 1 – pasager ce are rezervat un loc la clasa I
  + 2 – pasager ce are rezervat un loc la clasa a II-a

**tickets** – tabelă ce ține o evidență a biletelor cumpărate de către clienții care interacționează cu aplicația și devin pasageri

* code – numărul unic corespunzător biletului
* price – prețul biletului
* class – bilet de clasa I sau de clasa a II-a
* passenger\_ID – cheie străină ce identifică id-ul persoanei deținătoare a biletului
* flight\_ID – cheie străină ce referențiază detaliile asociate zborului (de exemplu, care este avionul de plecare, aeroportul de plecare și aeropotul în care clientul dorește să ajungă)

**flights\_cache** – un zbor eliminat sau amânat este stocat temporar în această tabelă, acțiune cu care sunt responsabili administratorii; un zbor care a avut loc cu succesc va fi șters din tabela flights și introdus automat în această tabelă, iar în alte condiții (eșec sau amânare), administratorul va fi cel care va introduce zborul în tabelă

* flight\_ID, departure\_date, duration, delay, distance, stopovers, airport\_code, airplane\_ID – detaliile asociate zborului care a fost eliminat sau amânat
* reason – pentru un zbor eliminat are valoarea „success” daca zborul a fost efectuat cu succes sau „failure” dacă, de exemplu, avionul a fost nevoit să aterizeze urgent în urma unor defecțiuni; pentru un zbor amânat motivele pot fi diverse
* rescheduled : data la care un zbor amânat a fost reprogramat; are valoarea null pentru un zbor eliminat

**feedback** – în această tabelă sunt stocate părerile utilizatorilor care și-au alocat timp pentru a aprecia calitatea condițiilor în care au călătorit; interfața aplicației va facilita adăgarea comentariului

* user\_ID – identificatorul userului căreia îi aparține feedback-ul
* company\_ID – identificatorul companiei cu avionul căreia persoana a călătorit
* message – textul ce alcătuiește feedback-ul

**connection\_flights\_pilots, connection\_flights\_attendants, connection\_flights\_passengers** – tabele ce modelează legăturile many-to-many dintre tabelele implicate; administratorii sunt cei care adaugă sau modifică un pilot sau FA de la un anumit zbor, iar pasagerii sunt adaugați automat în momentul achiziționării biletului

**stopovers** – în momentul adăugării unui anumit zbor de către administratori, aceștia sunt responsabili cu adăugarea, dacă există, a informațiilor asociate escalelor în aeropoarte intermediare

* stop\_number – a câta oprire este
* flight\_ID – cheie străină folosită pentru identificarea informațiilor corespunzătoare zborului în cadrul căruia s-a făcut oprirea
* airport\_code – cheie străină ce identifică aeroportul în care a avut loc escala
* time – durata petrecută în aeroportul intermediar
* departure\_date – data exactă (zi/lună/an oră minut) la care se părăsește aeroportul intermediar

**Acțiunile întreprinse de utilizatorii obișnuiți:**

* creare cont
* logare în sistem
* filtrare zboruri după anumite criterii
* selectare zbor
  + vizionare detalii asociate zborului (dată, date despre avion și compania care îl deține, date despre escale)
  + achiziționare bilet
* selectare companie
  + postare feedback
* delogare

**Acțiunile întreprinse de administratorii bazei de date (operatori aeroport):**

* logare în sistem
* adăugare companie/avion/zbor/aeroport
* ștergere companie/avion/aeroport
* update informații companie/avion/zbor/aeroport
* ștergere zbor deja efectuat
* declarare zbor amânat
  + introducere motiv amânare
* asociere între zbor și piloți
* asociere între zbor și însoțitori
* delogare

**Constrângerile din/între tabele:**

**flights**

* ID – constrângere de tip cheie primară
* airport\_code – constrângere de tip cheie străină ce referențiază atributul code din tabela airports
* airplane\_ID – cheie străină ce referențiază atributul ID din tabela airplanes
* departure\_date – constrângere de tip not null
* duration – constrângere de tip not null
* distance – constrângere de tip not null

**airplanes**

* ID – cheie primară
* company\_ID – cheie străină ce referențiaza atributul ID din tabela companies
* available – contrângere de tip check, atribut limitat la două valori, 0 și 1
* first\_class\_seats – not null
* second\_class\_seats – not null

**airports**

* code – cheie primară
* adress - constrângere de tip unique
* phone\_number - constrângere de tip unique
* email – constrângere de tip unique și not null
* name – not null
* city – not null

**companies**

* ID – cheie primară
* name – not null
* city – not null
* email – not null și unique

**users**

* ID – cheie primară
* type – check, limitat la „pilot”, „FA”, „user”, „admin”
* first\_name – not null
* last\_name – not null
* date\_of\_birth – not null
* city – not null
* email – not null și unique
* phone\_number - unique
* password – not null

**pilots, flight\_attendants, passengers**

* pilot\_ID, FA\_ID, passenger\_ID – chei străine referențiind atributul ID din tabela persoană
* rank – check limitat la valorile {1,2}/{1,2,3}/{1,2} pentru piloți, însoțitori de zbor, și respectiv pasageri

**connection\_flights\_pilots, connection\_flights\_attendants, connection\_flights\_passengers**

* flight\_ID, pilot\_ID, FA\_ID, passenger\_ID – chei străine referenșiind atributele ID din tabelele corespunzătoare

**stopovers**

* flight\_ID – cheie străină ce referențiază atributul ID din tabela flights
* airport\_code – cheie străină ce referențiază atributul code din tabela airports
* stop\_number – not null
* time – not null
* departure\_date – not null

**feedback**

* person\_ID, company\_ID – chei străine referențiind atributele ID din tabelele corespunzătoare
* message – not null

**tickets**

* code – cheie primară
* passenger\_ID - cheie străină referențiind atributul ID din tabela passengers
* flight\_ID – cheie străină referențiind atributul ID din tabela flights
* price - not null
* class – not null

**flights\_cache**

* flight\_ID – cheie primară

**Tipuri de legături:**

* între aiplanes și companies – many-to-one (o companie are mai multe avioane; un avion aparține unei singure companii)
* între aiplanes și flights – one-to-many (un zbor se realizează cu un singur avion; un avion poate fi antrenat în mai multe zboruri viitoare, pe date diferite)
* între flights și pilots – many-to-many (un zbor poate avea unul sau doi piloți; un pilot poate fi înregistrat în mai multe zboruri)
* între flights și flight\_attendants – many-to-many (un zbor poate avea minim un însoțitor de zbor și maxim cinci; un FA poate fi înregistrat în mai multe zboruri)
* între flight și passengers – many-to-many (un zbor are loc cu mai mulți pasageri; un pasager e posibil să aibă bilete pentru zboruri distincte)
* între airports și flights – many-to-many (un zbor are o singură destinație finală, dar poate avea și destinații intermediare; zboruri diferite pot avea aceleași destinații
* între passengers și tickets – one-to-many (biletul aparține unui singur pasager; un pasager poate achiziționa mai multe bilete)
* între flights și tickets – many-to-one (un zbor poate avea maxim atâtea bilete achiziționate câte locuri disponibile are; un tichet are un singur identificator de zbor trecut pe el)
* între feedback și persons – one-to-many (un feedback aparține unui singur utilizator; o persoană poate oferi oricâte feedback-uri dorește)