**InfAir**

Készítette: Báló Krisztina, Rest Balázs

**Bevezetés**

A mai világban az utazás elég nagy szerepet tölt be az emberek életében. Ma már a világ minden pontja elérhető a repülés segítségével.

A repülőterek zsúfoltsága miatt kényelmetlenné vált a repülő jegyek intézése a repülőtereken. Nagy könnyebbséget jelent a felhasználóknak, hogy az interneten keresztül tudják intézni a repüléssel kapcsolatos ügyeiket.

A weboldalunk célja egy olyan felület biztosítása, ahol a felhasználó repülőjegyeket tud foglalni, és ezzel megkönnyíti ,illetve meggyorsítja a repüléssel kapcsolatos fáradalmakat.

**Felhasználói dokumentáció**

Előszőr is be kell regisztrálnunk, illetve be kell jelentkeznünk, hogy használhassuk a weboldat.

A bejelentkezés után weboldal fő oldalával találkozunk, ami az aktuálisan akciós repülőjáratokat láthatjuk. Innen válogathatunk, hátha megtetszik nekünk egy utazási lehetőség, és olcsóbban elutazhatunk valamerre a világban. Itt helyben nem foglalhatunk még magunknak repülőjegyet.

A menüsorban láthatunk pár gombot, melyekkel a weboldal egyéb részeihez navigálhatunk. A jegy foglaláshoz menjünk a Book felületre.

A Book felületén kereshetünk szűrhetünk az elérhető repülőjegyek között. Válasszuk ki a SearchBy listában milyen paraméterre szeretnénk keresni:

* Startpoint Melyik városból indul a repülő.
* Endpoint Mi a járat cél városa.
* Launch Mikor indul a repülő.
* Arrival Mikor van a tervezett érkezése a járatnak.
* Cost Mennyibe kerül a repülő járat.
* Sale Mennyi akció van a járaton.

Ezután Írjuk be a kereset értéket a Search For-ba. Például: Kiválasztjuk a Startpoint opciót és megadjuk a London várost. A weboldal kilistázza nekünk az összes járatot, ami Londonból indul.

A járatok alján található Book gomb megnyomásával foglalhatunk jegyet a járatra. A gomb megnyomására egy ablak urgik fel nekünk, ahol további extra szolgáltatásokat adhatunk a repülőjegyünkhöz:

* Specail seat Speciálisan kényelmes ülés.
* Next to window Ablak mellé kérjük a helyünket.
* Bigger foot space Nagyon teret kérünk a lábunknak.
* Extra suitcase Extra bőröndöt is vihessünk magunkkal.
* Bagcar Nagy bőröndökhöz kocsit kérünk a könnyebb szálitásért.

Az ok gomb lenyomásával el is mentettünk a foglalásunkat.

A foglalásainkat a My Book felületen nézhetjük át.

A weboldal kilistázza az összes lefoglalt jegyünket. Itt lemondhatjuk, törölhetjük a foglalásunkat a Delete gomb segítségével, vagy módosíthatjuk, a korábban kiválasztott extra szolgáltatásokat.

**Fejlesztői dokumentáció**

**Program áttekintése:**

A program 2 részből áll:

* Egy java alapú rest applikáció a backend.
* Egy Angulár alapú webes UI felület.

A program futtatásához mennie kell, hogy kiszolgálhassa a felhasználói felületet, erre a SpringBoot-ot használjuk.

A mappa szerkezet fő részből áll az src és a client mappákból. Az src tartalmazz a backendet és a client a GUI-t. További mappák és egyéb fájlok a verziókezeléshez használt git-ből, illetve a project manage-elérséhez használt maven-ből származnak.

**Backend:**

Az src mappában található. Azon belül 2 további mappa a main és a test. A java konvenciós szabályainak megfelelő mappa struktúrát használunk. A main-en belül található a java, illetve a resources mappa.

A resources mappában található az adatbázis feltöltésért felelős sql parancsok.

A java/hu/elte/InfAir mappában található a backend forrás fájljai.

Az InfAirApplication.java A program belépési pontja a main. Elindítja a Spring applikációt és vele a servert is.

A controller mappában találhatóak a Controller osztályok melyek foglalják az API hívásokat.

* BookController.java
* FlightController.java
* ServiceController.java
* UserController.java

A BookController fogadja a foglalásokkal kapcsolatos hívásokat. A /books-szal érhető el.

Lehetőségei Get esetén:

* /{id}
  + Vissza adja az id-hoz tartozó Book adatait.
* /Userid/{userid}
  + Vissza adja az userid-hoz tartozó User id-jét.
* /Flightid/{flightid}
  + Vissza adja az flightid-hoz tartozó Flights id-jét.
* /Serviceid/{serviceid}
  + Vissza adja az serviceid-hoz tartozó Service id-jét.

Lehetőségei Post esetén:

* Létrehoz egy foglalást

Lehetőségei Delete esetén:

* /{id}
  + Kitörli az adott id-hez tartozó foglalást

A FlightController fogadja a járatkal kapcsolatos hívásokat. A /flights-szal érhető el.

Lehetőségei Get esetén:

* /{id}
  + Vissza adja az id-hoz tartozó Flight adatait.
* /Startpoint/{startpoint}
  + Vissza adja az startpointből induló Flightokat.
* /Endpoint/{endpoint}
  + Vissza adja az endpointba érkező Flightsokat.
* /Launch/{launch}
  + Vissza adja az adott időpontba induló Flightsokat
* /Arrival/{arrival}
  + Vissza adja az adott időpontba érkező Flightsokat
* /Cost/{cost}
  + Vissza adja az adott értékbe kerülő Flightsokat
* /Sale/{sale}
  + Visszaadja az akciós járatokat

Lehetőségei Post esetén:

* Létrehoz egy új járatot

Lehetőségei Patch esetén:

* /{id}
  + Módosítja az adott id-vel rendelkező járat adatait
* /{id}/sale/{sale}
  + Beállít egy új akciós árat

Lehetőségei Delete esetén:

* /{id}
  + Törli az id-hoz tartozó járatot.

A ServiceController fogadja az extra szolgáltatásokkal kapcsolatos hívásokat. A /books-szal érhető el.

Lehetőségei Get esetén:

* /{id}
  + Vissza adja az id-hoz tartozó servicet.
* /””
  + Vissza az összes servicet.
* /specialseat/{specialseat}
  + Vissza adja azokat a service-eket, amikben kérték a specialseat opciót.
* /nexttowindow/{nexttowindow }
  + Vissza adja azokat a service-eket, amikben kérték a nexttowindow opciót.
* /boggetfootspace/{ boggetfootspace }
  + Vissza adja azokat a service-eket, amikben kérték a boggetfootspace opciót.
* /extrasuicase/{ extrasuicase }
  + Vissza adja azokat a service-eket, amikben kérték a extrasuicase opciót.
* /bagcar/{ bagcar }
  + Vissza adja azokat a service-eket, amikben kérték a bagcar opciót.

Lehetőségei Post esetén:

* /””
  + Elmenti a requestbody-ban kapott servicet

Lehetőségei Patch esetén:

* /{id}
  + Módisítja az idhoz tartozó servicet a requestbody-ban kapott service-szel.

Lehetőségei Patch esetén:

* /{id}
  + Tőrli az id-hoz tartozó servicet.

A repository mappában találhatóak az adatbázis repositorik.

A model mappában az adatbázik tábláit representáló java osztályok.

**Frontend:**

Nodejs, angulárral lett fejlesztve.

A domain mappában találhatóak a backenddel történő üzenetváltásnál használt adatszerkezetek reprezentációi.

Az egész felületet componentek segítségével építettük fel.

* Menüsor:
  + A menüsor maga egy komponens amiben gombok találhatóak. A gombok megnyomásával az oldal törzse megváltozik, erre a routing.module-t használjuk.
* HomePage:
  + A menüsoron található InfAir megnyomásával a homepage component töltődik be.
  + A felület célja hogy akciós repülőjáratokat mutasson a felhasználónak amikor belép az oldalra.
  + A hompage további 1 componentet tartalmaz a homepage-list-et. amely további homepage-tile componenteket tartalmaz.
  + A hompage-tile-ok jelenítik meg az egyes lepülőjáratok adatait.
  + Az adatok lekérdezésére a filghtservice-t használjuk, ami az api FilghtControllerével kommunikál.
* Book:
  + A Book felületen lehet keresni, szűrni a járatok közt és foglalni a járatokra jegyet.
  + A kitudja választani a felhasználó a kereset paramétert és megadni egy keresett értéket.
  + A keresés után a serverről lekérdezzük a járatokat és megjelenítjuk a homepage-hez hasonlóan a search-result-list-tel és a search-result-tile-lal.
  + Minden járatnak a felületén (search-result-tile) található egy Book gomb melynek a megnyomásával feldobjuk a felhasználónak a book-dialog.html-ot megjelenítő dialog ablakot.
  + Ebben az ablakban a felhasználó kiválaszthatja milyen extra szolgáltatásokat akar hozzá adni a foglalásához.
  + A dialog OK gombjának a megnyomásával elmentődnek az adatok. Először is az extraServoce.service segítségével elmentjük a serveren a szolgáltatásokat. Majd a visszakapott ID-val létrehozunk egy foglalást és elmentjük a serveren a book.service segítségével.
* MyBook
  + Ezen a felületen láthat a felhasználó egy összegzést a foglalásaival kapcsolatban.
  + A book.service segítségével lekérdezzük a felhasználó foglalásait. A foglalások alapján lekérdezzük a járatokat és a service-eket is.
  + A foglalásokat tudja törölni, amit azonnal a serveren is tőrlünk.
  + Valamint tudja módosítani a service-eket a foglaláshoz. A módosításokat mentjük a serveren is.