

Fejlesztési feladat

Tartalom

1. Szcenárió ismertetése.....	1
2. Probléma ismertetése.....	1
2.1. Kapcsolattartó entitás.....	2
2.2. Listázás felhasználói eset.....	2
2.3. Részletezés felhasználói eset.....	2
2.4. Létrehozás felhasználói eset.....	3
2.5. Módosítás felhasználói eset.....	3
2.6. Törlés felhasználói eset.....	3
3. Feladat.....	4

1. Szcenárió ismertetése

Adott egy Single-Page Application (pl. Angular) és egy hozzá tartozó, Spring Boot-ban készült backend service, ami REST API-t szolgáltat a frontend alkalmazás számára.

Az alkalmazás egy login képernyőből és néhány, bejelentkezést követően elérhető háttérfelületből áll. A bejelentkezés és a hitelesítés JWT használatával történik (a feltételezett megvalósítás lényegtelen).

2. Probléma ismertetése

Az ügyfél igény az, hogy az alkalmazást kiegészítsük új felületekkel, ahol kapcsolattartók kezelésére van lehetőség. Egy új menüpont alól elérhető a listázó felület, ahonnan további műveleteket tud kezdeményezni a felhasználó.

Szükséges műveletek listája:

- Kapcsolattartók listázása
- Kapcsolattartó részletezése
- Kapcsolattartó létrehozása
- Kapcsolattartó módosítása
- Kapcsolattartó törlése

A kapcsolattartókat a rendszer a hozzá tartozó adatbázisban tárolja el.

2.1. Kapcsolattartó entitás

Az ügyfél által kezelni kívánt kapcsolattartói adatokat az alábbi táblázat tartalmazza.

Adat neve	Típus	Leírás
Vezetéknév	szöveg	
Keresztnév	szöveg	
E-mail cím	szöveg	Csak e-mail cím formátum lehet
Telefonszám	szöveg	Opcionális; Csak E-164 formátum lehet
Cég	hozzárendelés	Kiválasztható egy darab cég az adatbázisból
Megjegyzés	szöveg	Opcionális; Tetszőleges megjegyzést tartalmazhat, plaintext vagy HTML formátumban
Státusz	enumeráció	Választható: aktív, törölt

Fejlesztői oldalról egészítsük ki a fenti adatokat további adatokkal:

- egyedi, generált azonosító
- létrehozás ideje
- utolsó módosítás ideje

2.2. Listázás felhasználói eset

A felhasználó egy listát szeretne látni a rendszerben lévő aktív kapcsolattartókról. Ennek folyamata:

1. A felhasználó kiválasztja a listázó menüpontot
2. Az alkalmazás lekérdezi az első oldalhoz tartozó kapcsolattartók listáját. Az oldalon megjelenik egy listázó táblázat, az alábbi oszlopokkal:
 - a. Teljes név
 - b. Cég neve (hozzárendelt céghez tartozó név)
 - c. E-mail cím
 - d. Telefonszám
3. A táblázat csak az aktív státuszú kapcsolattartókat jeleníti meg, név szerint ABC sorrendben és egyszerre mindig tízet.
4. A felhasználónak lehetősége van tízesével előre és hátra lapozni a találatok között, illetve oldalszám alapján ugrálni. Ilyenkor az alkalmazás újabb lekérdezéseket hajt végre, ahol a kérés paraméterként tartalmazza a megjeleníteni kívánt oldalszámot.
5. A táblázatban a teljes név hivatkozásként is funkcionál, ami átvezet a kapcsolattartó adatlapjára (részletezés felhasználói eset).

2.3. Részletezés felhasználói eset

A felhasználó a kapcsolattartó összes adatát látni szeretné egy adatlapon. Ennek folyamata:

1. A felhasználó megnyitja a listázó oldalt
2. Lapozással megkeresi a megtekinteni kívánt kapcsolattartót, majd rákattint a nevére
3. Az alkalmazás lekérdezi a kiválasztott kapcsolattartóhoz tartozó részletek és megjeleníti egy tetszőleges elrendezésben. A megjelenített adatok:
 - a. Vezetéknév
 - b. Keresztnév
 - c. Cég neve

- d. E-mail cím
 - e. Telefonszám
 - f. Megjegyzés
 - g. Létrehozás ideje
 - h. Utolsó módosítás ideje
4. Az aloldalon szerepel egy törlés gomb, aminek a megnyomására – egy megerősítő kérdés jóváhagyását követően – az alkalmazás törlési kérést intéz és a rendszer végrehajtja a törlési műveletet.

2.4. Létrehozás felhasználói eset

A felhasználó létre szeretne hozni egy új kapcsolattartót. Ennek folyamata:

1. A felhasználó megnyitja a listázó oldalt
2. Rákattint a listázó felett elhelyezkedő Létrehozás gombra
3. Az alkalmazás megjelenít egy létrehozás űrlapot az alábbi mezőkkel:
 - a. Vezetéknév, szöveges beviteli mező
 - b. Keresztnév, szöveges beviteli mező
 - c. Cég, kiválasztás legördülő menüvel
 - d. E-mail cím, szöveges beviteli mező
 - e. Telefonszám, szöveges beviteli mező
 - f. Megjegyzés, szöveges beviteli mező, többsoros
4. Az űrlap alatt elhelyezkedő mentés gomb megnyomását követően az alkalmazás kérdésben elküldi az űrlap tartalmát. A rendszer ellenőrzi a felküldött adatokat és csak abban az esetben hozza létre a kapcsolattartót, ha sikeres volt az ellenőrzés. A validációs szabályok a következők:
 - a. Minden nem opcionális mezőhöz kötelező érkeznie kell adatnak
 - b. E-mail cím csak e-mail cím formátumú szöveg lehet
 - c. A telefonszám csak E-164 formátumú szöveg lehet
 - d. Az érkező cég azonosítónak létező céghez kell tartoznia
 - e. A státusz értéke csak a megengedett értékkészlet elemeit veheti fel.
5. Siker esetén egy aktív státuszú kapcsolattartó jön létre.
6. Hibás felküldés esetén, a rendszer sikertelen státuszkóddal válaszoljon a kérésre.

2.5. Módosítás felhasználói eset

A felhasználó módosítani szeretne egy létező kapcsolattartót. Ennek folyamata:

1. A felhasználó megnyitja a listázó oldalt
2. Lapozással megkeresi a megtekinteni kívánt kapcsolattartót, majd rákattint a nevére
3. Rákattint az adatlapon található Szerkesztés gombra.
4. Az alkalmazás lekérdezi a kapcsolattartóhoz tartozó adatokat, majd betölti egy megjelenített módosítás űrlapba. Az űrlap mezői megegyeznek a létrehozási űrlap mezőivel.
5. A mentés hasonlóképp működik, mint a létrehozás esetén.

2.6. Törlés felhasználói eset

A felhasználó módosítani szeretne egy létező kapcsolattartót. Ennek folyamata:

1. A felhasználó megnyitja a listázó oldalt

2. Lapozással megkeresi a megtekinteni kívánt kapcsolattartót, majd rákattint a nevére
3. Rákattint az adatlapon található Törlés gombra.
4. Az alkalmazás törlési kérést generál. A rendszer megkeresi a kiválasztott kapcsolattartót és a státuszát töröltre állítja.

3. Feladat

A feladat, hogy készíts megvalósítást az ismertetett felhasználói esetekhez az alkalmazás backend oldalán. A megvalósításnak REST API szeretnénk, a kommunikáció formátumának JSON-t. Minden felhasználói eset egy végpontnak megfeleltethető.

Hitelesítéssel, jogosultságkezeléssel nem foglalkozunk.

Kérések és megjegyzések a megvalósítással kapcsolatban:

- Szolgáltatunk egy Spring Boot-ban készült alaprendszert, amihez a megoldást hozzá kellene fejleszteni. Az átadott repository tartalmaz egy README.md fájlt telepítési és használati instrukciókkal.
- Az elkészült végpontokhoz szeretnénk dokumentációt kapni.
- Az elkészült kódhoz szeretnénk, ha írnál teszt eseteket.
- Kérjük, hogy telefonszám validációhoz a Libphonenumber (com.googlecode.libphonenumber) programkönyvtárat használd fel.