

Event Finder - Dokumentáció

Esemény kereső alkalmazás webes környezetben



Szoftverarchitektúrák házi feladat

2020. ősz

Konzulens:

Tóth Tibor

Készítették:

Gyarmati Bálint - B9KF99

Juhász Péter - EZVYSK

Követelményspecifikáció	3
Feladatkiírás:	3
A fejlesztői csapat:	3
Részletes feladatléírás:	3
Technikai paraméterek:	4
Szótár:	4
Use-case diagram	5
Architektúra és megvalósítás	6
Osztálydiagram	7
User osztály:	7
Esemény osztály:	7
Message osztály:	7
ER diagram	8
Perzisztencia - Adatbázis	9
Üzleti logika - Szerver	9
Megjelenítés - Kliens	10
Működési logika	11
A regisztráció	11
A bejelentkezés	13
Az események megjelenítése	14
Az események kedvencekhez adása	15
Az események eltávolítása a kedvencekből	16
A profile szerkesztése	17
Új esemény létrehozása	18
A kijelentkezés	19
A grafikus felhasználói felület (GUI)	20
A bejelentkezés:	20
Regisztráció	21
A főoldal	23
Profil beállítások	24
Reszponzivitás	25
Az alkalmazás indítása:	26
A szerver (backend) indítása:	26
A kliens (frontend) indítása:	26
Továbbfejlesztési lehetőségek	27

Követelményspecifikáció

Feladatkiírás:

A hallgatók feladata egy társaság kereső alkalmazás tervezése. A felhasználóknak lehetőségük van bejelölni érdeklődési köreiket és nyelvtudásukat.

Az azonos témák iránt érdeklődők egy közös platformon üzenve tudják megbeszélni a programokat. A felhasználók követni tudnak eseményeket, amelyek érdeklik őket.

A fejlesztői csapat:

A csapat tagjai:

csapattag neve	Neptun-kód	E-mail cím
Juhász Péter	EZVYSK	juhasz.peter1337@gmail.com
Gyarmati Bálint	B9KF99	xygyurma@gmail.com

A csapatban dedikált szerepek kiosztását a csapat kis mérete miatt nem tartottuk fontosnak.

Részletes feladatleírás:

A projekt során célunk egy olyan alkalmazás készítése, amely alkalmas a felhasználók által megadott érdeklődési kör és nyelvtudás alapján események keresésére. Az eseményeken belül lehetősége van a felhasználóknak egy közös chat-ben részt venni. Amennyiben érdeklődik egy esemény iránt, azt lementheti, így megjelenik a saját eseménylistájában.

A felhasználónak regisztrációkor meg kell adnia a személyes adatait, érdeklődési köreit, és a beszélt nyelveit - ezek a későbbiekben a profil oldalán szerkeszthetők -, annak érdekében, hogy a számára releváns események jelenjenek meg a kezdőoldalon.

A felhasználók az adott esemény részletes leírásánál megtalálják a komment szekciót, ahol bárki kérdéseket tehet fel a másoknak, vagy szimplán csak a felhasználók kommunikálhatnak egymás között és megoszthatják egymással gondolataikat.

Azokat az eseményeket, amelyeket egy felhasználó korábban megjelölte, mint számára érdekes esemény, a későbbiekben megjelölheti, mint számára irreleváns esemény és eltűnik az eseménylistájából.

A felhasználókat hozhathatnak létre további eseményeket is, amelyek szintén megjelennek az eseménylistában.

Technikai paraméterek:

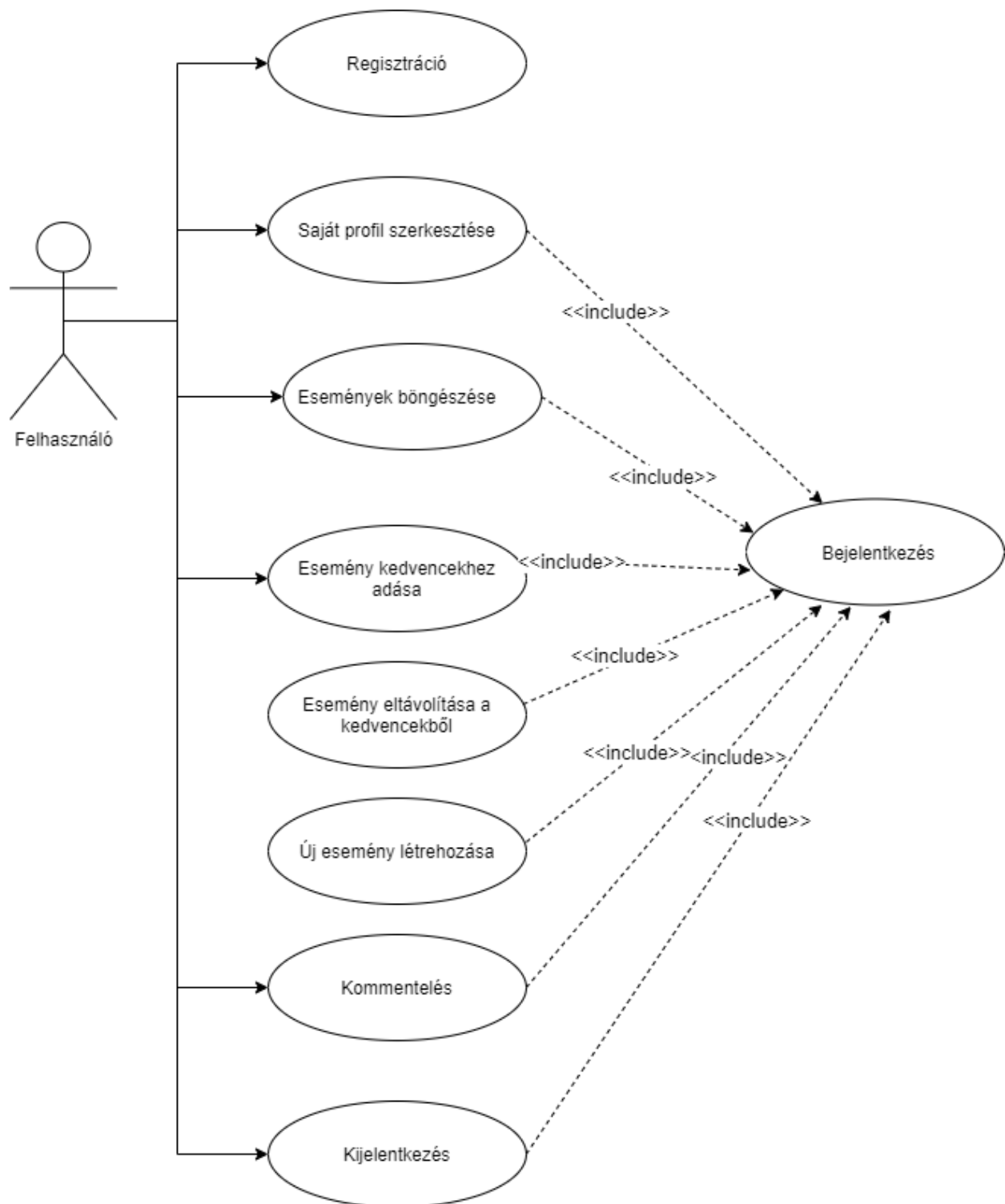
Az projekt egy kliens oldali és egy szerver oldali programból fog állni, a kliens, JavaScript alapú program, lesz amely a webböngészőből érhető el. A szerver .NET alapú, az adatok adatbázisban lesznek eltárolva. A kétirányú kommunikációt websocket technológiával oldjuk meg.

Szótár:

Chat/Kommentszekció: Egy olyan lehetőség, ahol az emberek egymással tudnak kommunikálni és különböző megjegyzéseket tudnak hozzáfűzni az adott eseményekhez, esetleg kérdéseket tehetnek fel azok számára, akiket szintén érdekel az adott esemény.

Esemény: Az esemény a kezdőidőponttól a végidőpontig tart és egy megnevezésből, érdeklődési körből/körökből és egy részletes leírásból épül fel. Az esemény tartalmazza a fent említett commentszekciót és az esemény iránt érdeklődők listáját.

Use-case diagram



Architektúra és megvalósítás

Az architektúra alapvetően három fő elemre oszlik, szerverre, kliensre és egy adatbázisra. A projekt egy szerverből és véges sok kliensből épül fel. Az alkalmazáshoz tartozik egy adatbázis is.

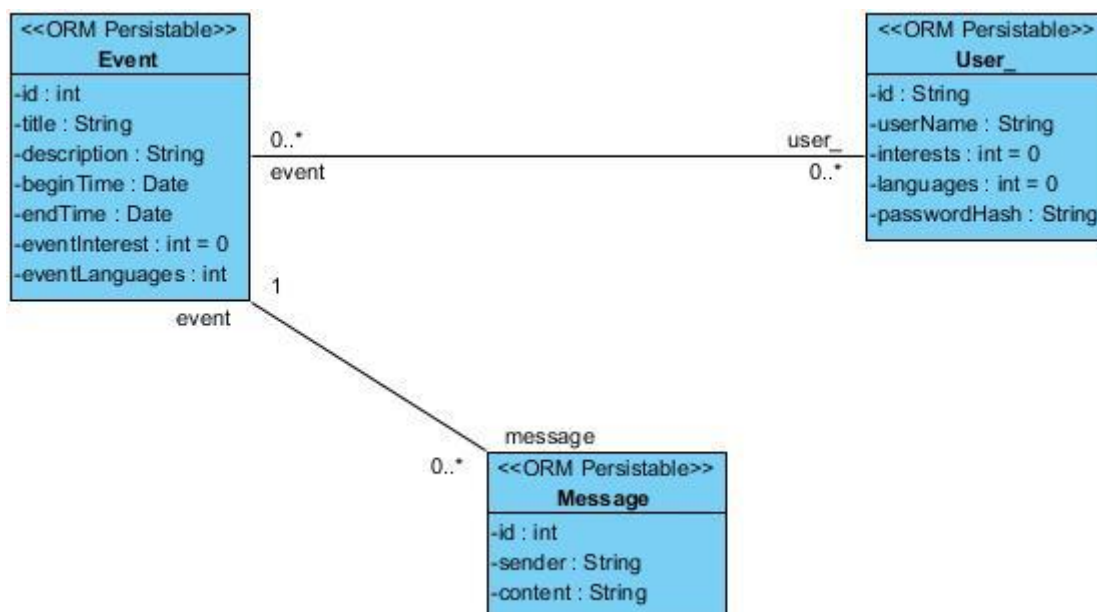
Szerver: A szerver szabályozza az alkalmazás működését. A szerver dönti el, hogy sikeres-e egy adott regisztráció, beléphet-e az adott felhasználó, illetve, hogy mely események jelenjenek meg a kliens oldalon a megadott érkező kör és a nyelvtudás alapján.

Adatbázis: Egy SQL adatbázisban vannak eltárolva, hogy milyen események szerepelnek az alkalmazásban, illetve, hogy milyen regisztrált felhasználók léteznek és melyik eseményhez a felhasználók milyen kommentet írtak.

Kliens: Amíg szerverből csak egyetlen egy szerepel az alkalmazásban, addig a kliensek száma nem korlátozott. Nagyon sok kliens tud regisztrálni az alkalmazásban, amelyeket a szerver szolgál ki. Elsősorban a kliensek eseményeket böngészhetnek és megjelölhetik azokat az eseményeket, amelyek nagyon tetszenek nekik és ezek az események bekerülnek ilyenkor egy kedvencek listába. Ezen felül az adott eseményekhez lehet kommentelni és így a kliensek tudnak egymással kommunikálni. A kliensek létrehozhatnak továbbá új eseményeket is az alkalmazásban.

Ezeket későbbi fejezetekben részletesebben is bemutatjuk.

Osztálydiagram



User osztály:

A User osztály a Microsoft APS.NET Core Identity osztály leszármazottja, onnan öröklí meg az id, userName, passwordHash attribútumokat és tartalmaz még egy érdeklődési, illetve egy nyelvtudás listát is.

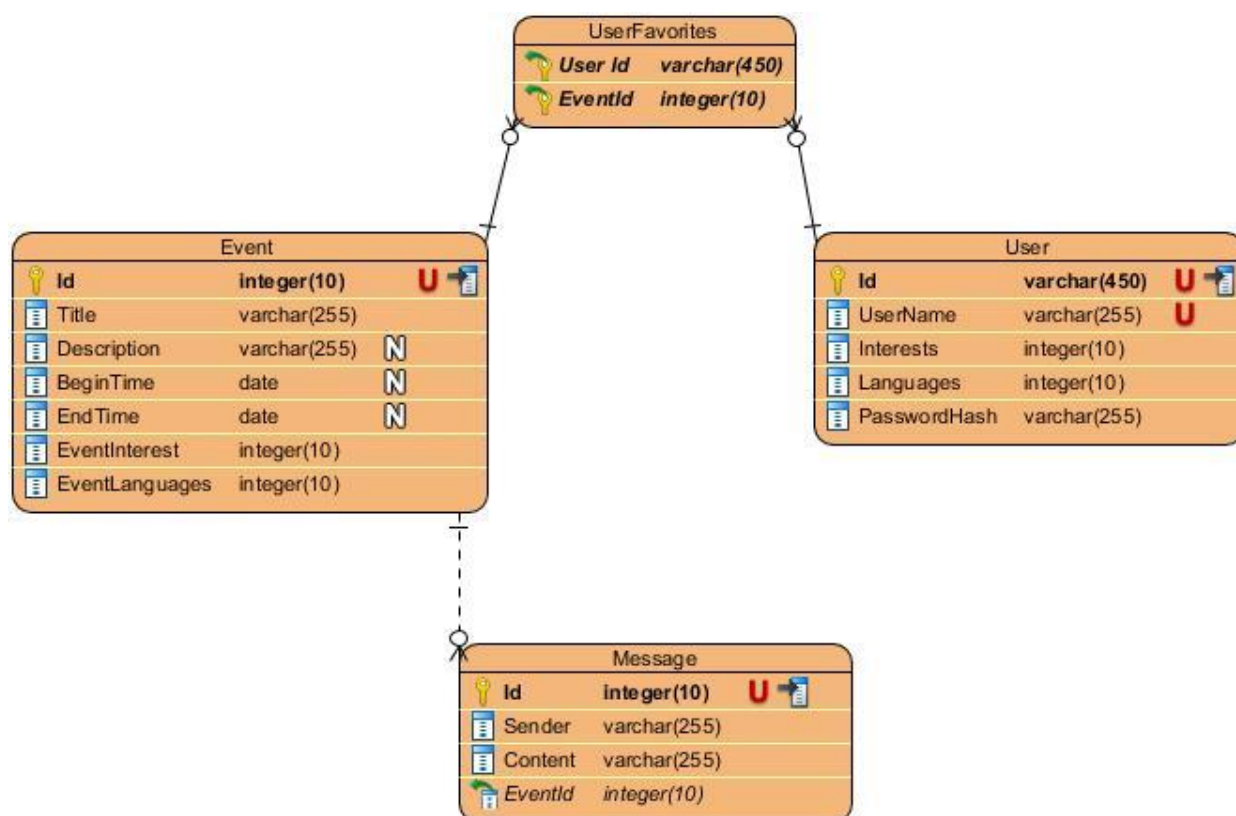
Esemény osztály:

Egy adott esemény egy azonosítóból, egy címből, egy leírásból, egy kezdeti- és egy végidőpontból, illetve érdeklődési körből és nyelvből épül fel.

Message osztály:

Ez az osztály felelős a kommentek létrehozásáért. A komment egy azonosítóból, egy küldőből, aki a felhasználó, illetve az adott tartalomról épül fel.

ER diagram



A fent látható ER-diagram az osztályok közötti viszonyt hivatott reprezentálni. Egy eseményhez több üzenet (komment) is tartozhat, de egy üzenet csak egy eseményhez tartozik (több-egy kapcsolat). Egy eseményt több felhasználó is kedvencekhez tehet, illetve egy felhasználó több eseményt is tehet a kedvencek listájába (ezt a kapcsolatot a UserFavorites valósítja meg).

Perzisztencia - Adatbázis

Az adatbázist az MSSQL adatbázis kezelő rendszerben tároljuk, de magát a kezelését az Entity Framework Core keretrendszerén belül az adat migrálási eszközök segítségével oldjuk meg, így abban a fejezetben fejtjük ki jobban.

Üzleti logika - Szerver

A szerver egy ASP.Net Core alapú alkalmazás, amelyet C# nyelven írtunk meg. A szerver az adatbázissal Entity Framework segítségével kommunikál, melynek a code first megvalósítását alkalmazzuk (azaz a kódban létrehozuk először az entitásokat, majd ebből generáljuk le az adatbázis sémát). Ez a megoldás nem csak az adatbázis létrehozása előtt használható, azaz amennyiben a kódban változtatunk az entitásokon, annak hatására az adatbázisban is megtörténnek a változtatások, amennyiben engedélyezzük a frissült információk migrálását.

A felhasználók hitelesítésére az ASP.Net Core Identity könyvtárát használjuk fel, így ez a környezet felel, a jelszavak titkosításáért, a felhasználók regisztrálásáért és beléptetéséért.

A felhasználókhöz belépéskor egy JWT token kapnak vissza, amely segítségével hitelesítik magukat az egyes lekérdezések előtt (a token minden kérés mellett elküldik a szervernek). Ennek hiányában, a szerver nem szolgálja ki a kérést.

A szerver és a kliensek között valós idejű kapcsolat van, amelyet az ASP.Net Core SignalR websocket alapú kommunikációjának köszönhet. Ennek segítségével, miután felépült a kliens és a szerver között a kapcsolat, bármelyik fél üzenhet a másiknak, amelyre akár az üzenet feldolgozása következtében azonnal válaszolhat. A SignalR rendszernek köszönhetően a szervernek lehetősége van csak egy bizonyos személy számára küldeni, vagy akár egy csoport, esetleg mindenki számára elküldeni a választ.

Ez hasznos irányított üzenetek esetén, azaz chat jellegű rendszerek megvalósításakor, amelyhez hasonló az események alatti komment szekció működése.

A szerver a következő kéréseket szolgálja ki:

- Felhasználó regisztráció
- Bejelentkezés
- Kijelentkezés
- Saját adataink lekérése
- Saját adataink módosítása
- Események listázása
- Esemény kedvencekbe mentése
- Esemény eltávolítása kedvencekből
- Üzenet küldése egy eseményhez

A kérésekre egy hívás vagy egy sikeres/sikertelen üzenettel válaszol, vagy a lekérdezett adatokkal.

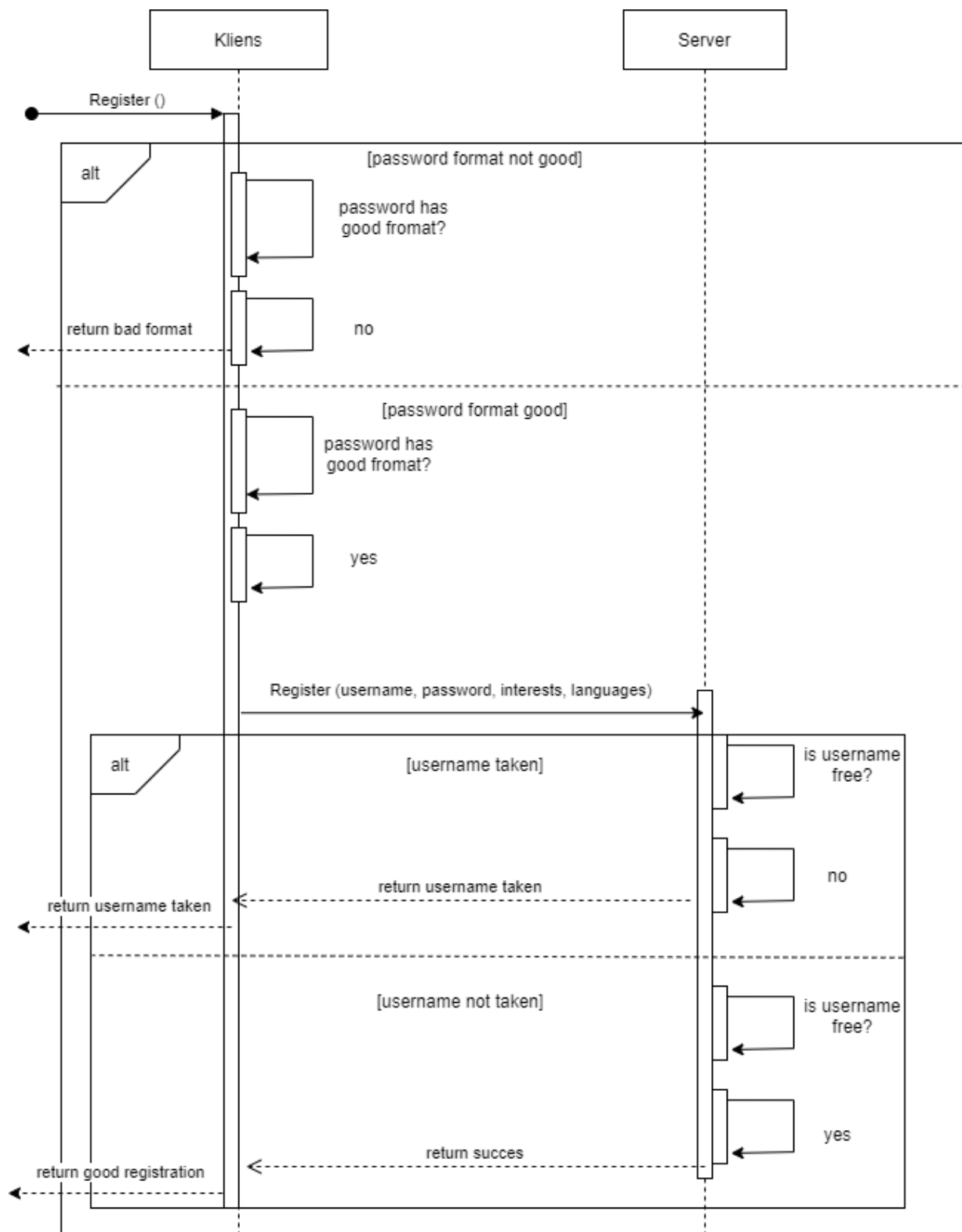
Megjelenítés - Kliens

Kliens oldalon a dinamikus megjelenítésre Angular JavaScriptet használunk. A feladat szempontjából nagyon kényelmes az Angular, hiszen különböző ciklusokat, illetve feltételeket lehet definiálni benne, hogy mi jelenjen meg az oldalon. Az események megjelenítéséhez a Bootstrap „card” osztályát használtuk. A „card” osztály pontosan annyiszor hívódik meg ahány eseményt kap a kliens a szervertől. A kommentszekcióban lévő kommentek megjelenítése is egy Angular for ciklussal történik.

A kliens generálja az előző fejeztben említett kéréseket, illetve kliens oldalon történik minden egyes szerver által küldött válasz feldolgozása és a megjelenítés.

Működési logika

A regisztráció



A regisztráció a fent látható szekvencia diagram alapján zajlik. Először a felhasználónak meg kell adnia egy felhasználónevet, egy jelszót, az érkező körét és a nyelvtudását.

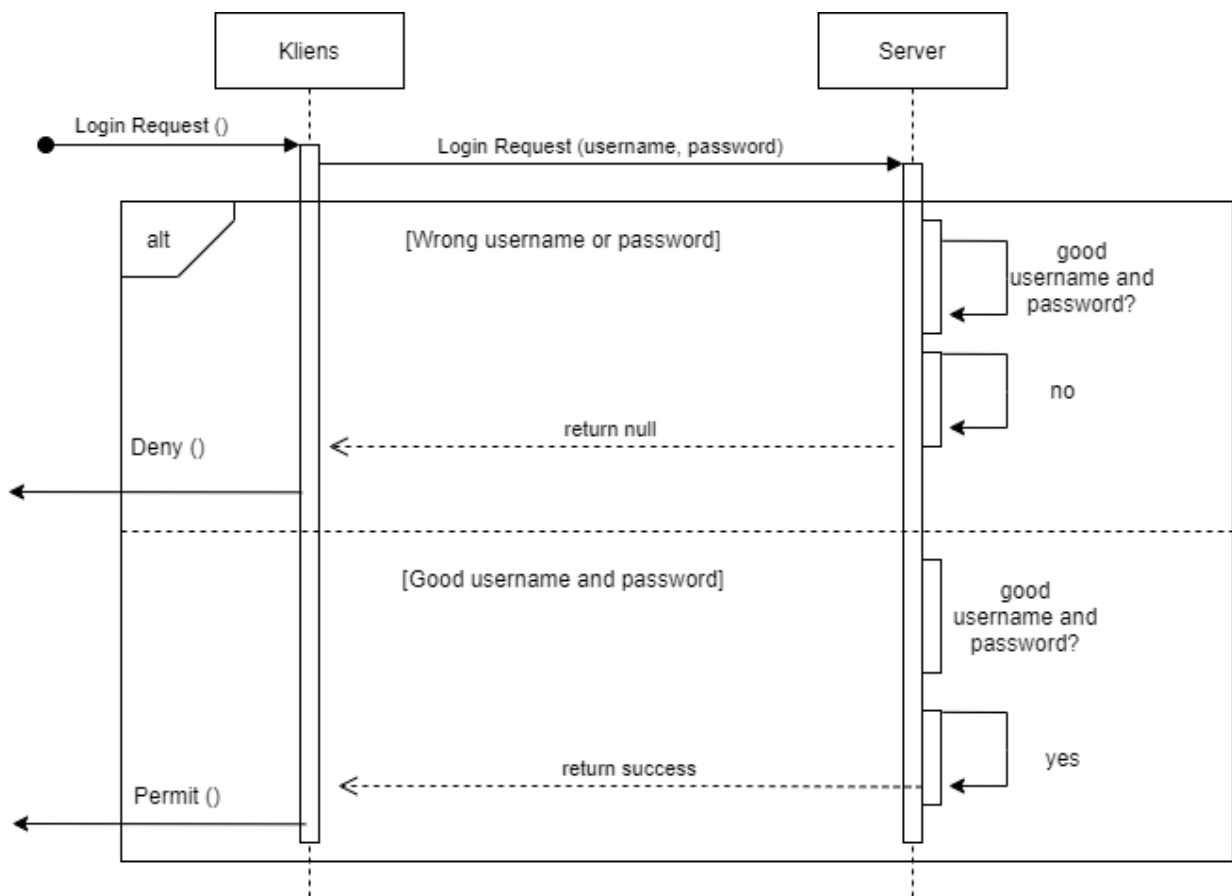
Kliens oldalon megtörténik ezen értékek kiolvasása, majd kétféle különböző jelszóvizsgálat történik. Az első jelszóvizsgálat során összevetjük, hogy a megadott jelszó megüti-e a követelményeket, vagyis legalább 8 karakter hosszú, tartalmaz-e kisbetűt, nagybetűt és számot. A felhasználónak kétszer kell megadni a jelszót egymás után az űrlapon, így ezzel is biztosítva, hogy a felhasználó nem üti el a jelszavát és a regisztráció után képes lesz belépni az alkalmazásba. A második jelszóvizsgálat során leellenőrizzük, hogy a megegyezik-e a két felhasználó által egymás után megadott jelszó. Abban az esetben, ha mindkét jelszófeltételen átmegy a felhasználó a regisztráció során, csak akkor küldjük el a megfelelő paramétereket a szerver számára.

A szerver kiolvassa a kapott értékeket és összeveti a kapott felhasználónevet a rendszerben lévőekkel és, ha már valaki korábban választotta azt a felhasználónevet, akkor visszaszól, hogy a megadott felhasználónév már foglalt és válasszon másikat az regisztráló.

Abban az esetben, ha a felhasználó olyan felhasználónevet adott meg, amely még nem foglalt, akkor a szerver elmenti a megadott adatokat az adatbázisba és visszaszól a kliensnek, hogy rendben, sikeres volt a regisztráció.

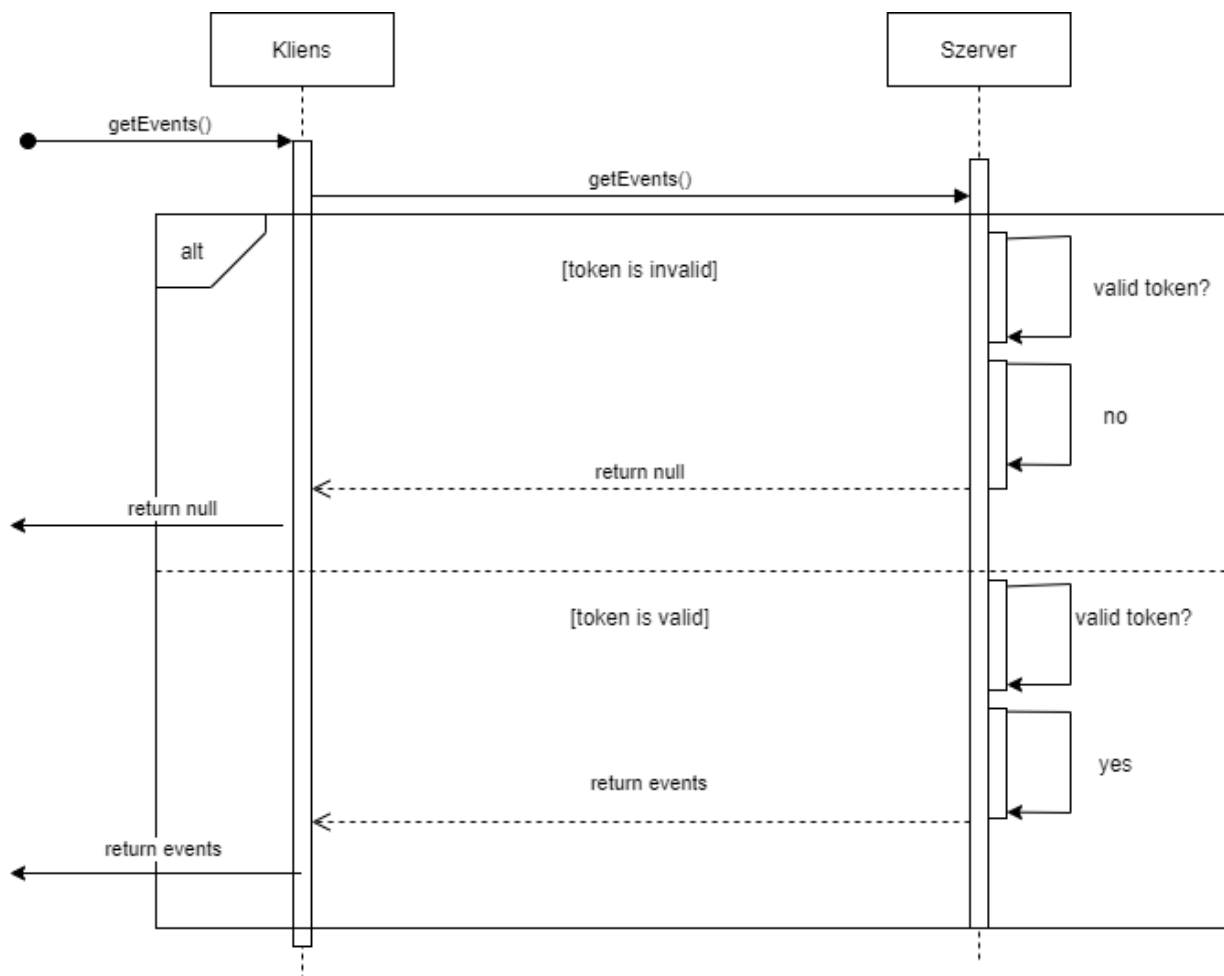
Ekkor a kliens visszairányítja a felhasználót a bejelentkező oldal és megkezdődhet a profilba bejelentkezés.

A bejelentkezés



A bejelentkezés a fent látható szekvenciadiagram alapján zajlik. Először a kliens beüt egy felhasználónevet és egy jelszót a két input mezőbe és rányom a Login gombra. Ezzel jelzi a kliens számára, hogy valaki be szeretne jelentkezni az alkalmazásba és megtörténik a kliens oldalon a megadott értékek kiolvasása. Amint ez megtörtént ekkor jelez a kliens a szervernek és a kliens elküldi a szervernek a felhasználónév és jelszó párost. A szerver kiolvassa a kapott értéket és megnézi, hogy egyáltalán létezik-e ilyen felhasználónév és, ha igen, akkor a megadott jelszó megegyezik-e azzal, amit ő tárol a felhasználóról. Abban az esetben, ha a felhasználónév vagy a jelszó nem jó, akkor egy null üzenetet küld vissza, ezzel jelezve a kliensnek, hogy nem megfelelőek a megadott paraméterek. Abban az esetben viszont, ha megfelelőek az értékek, akkor belép a felhasználó a fiókjába.

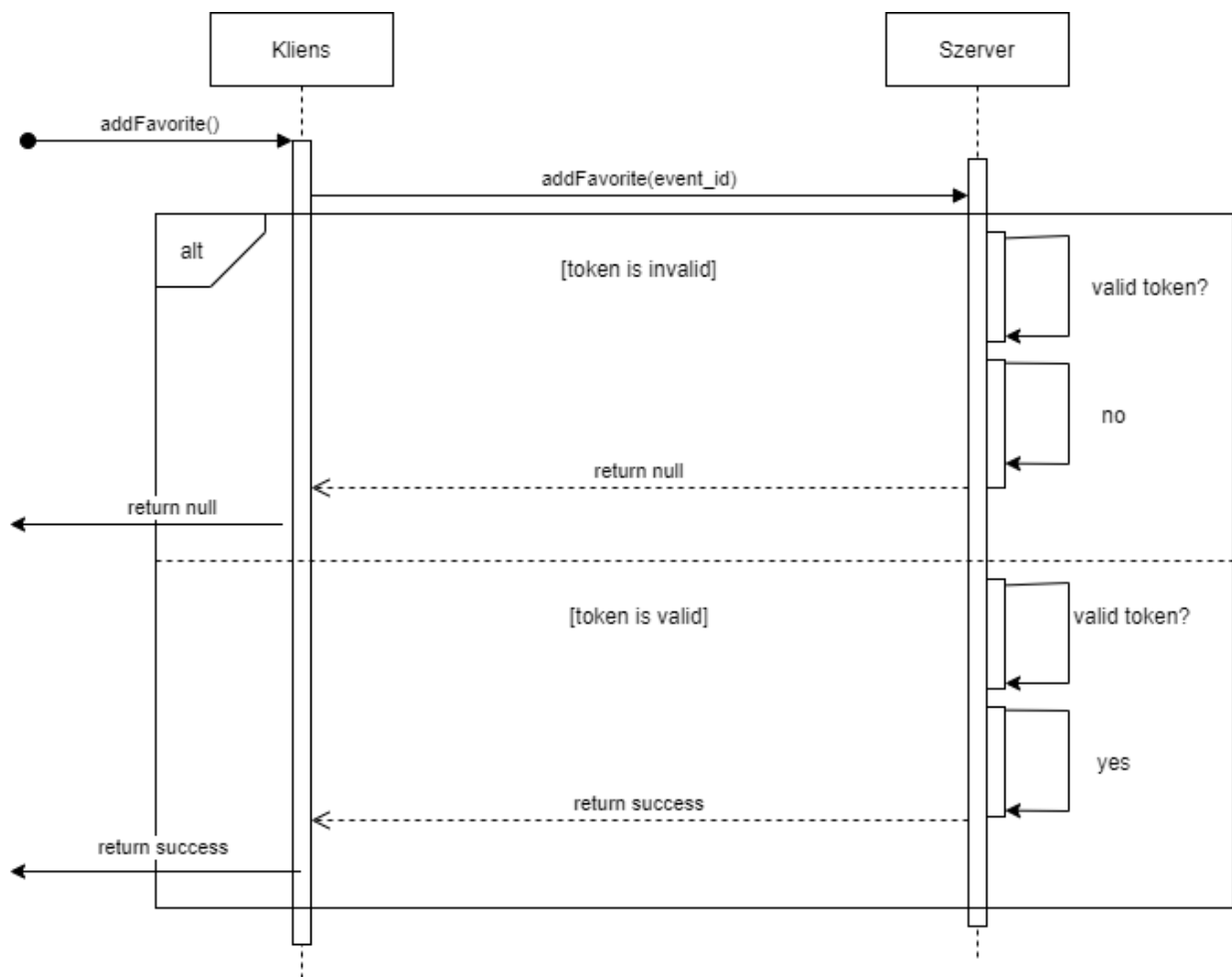
Az események megjelenítése



Az események megjelenítése a következőképpen zajlik. Bejelentkezés után a kezdőlap fogadja a felhasználókat, ahol megjelennek azok az események, amelyek megfelelnek a regisztráció során megadott érdeklődési körnek és nyelvtudásnak. Természetesen, ha valaki nem adott meg semmilyen érdeklődési kört vagy nyelvtudást, akkor nem fog ilyenkor megjelenni semmi. Ezt bármikor javíthatja a profil beállításoknál, azonban ezt ezen dokumentáció később taglalja. A szerver a klienstől kapott token által tudja, hogy melyik felhasználónak kell átadni az eseményeket.

Az események megjelenítése bejelentkezéshez kötött, így ha valaki nincs bejelentkezve, akkor nem fog semmi megjelenni, hiszen nincs megadva valid felhasználó.

Az események kedvencekhez adása



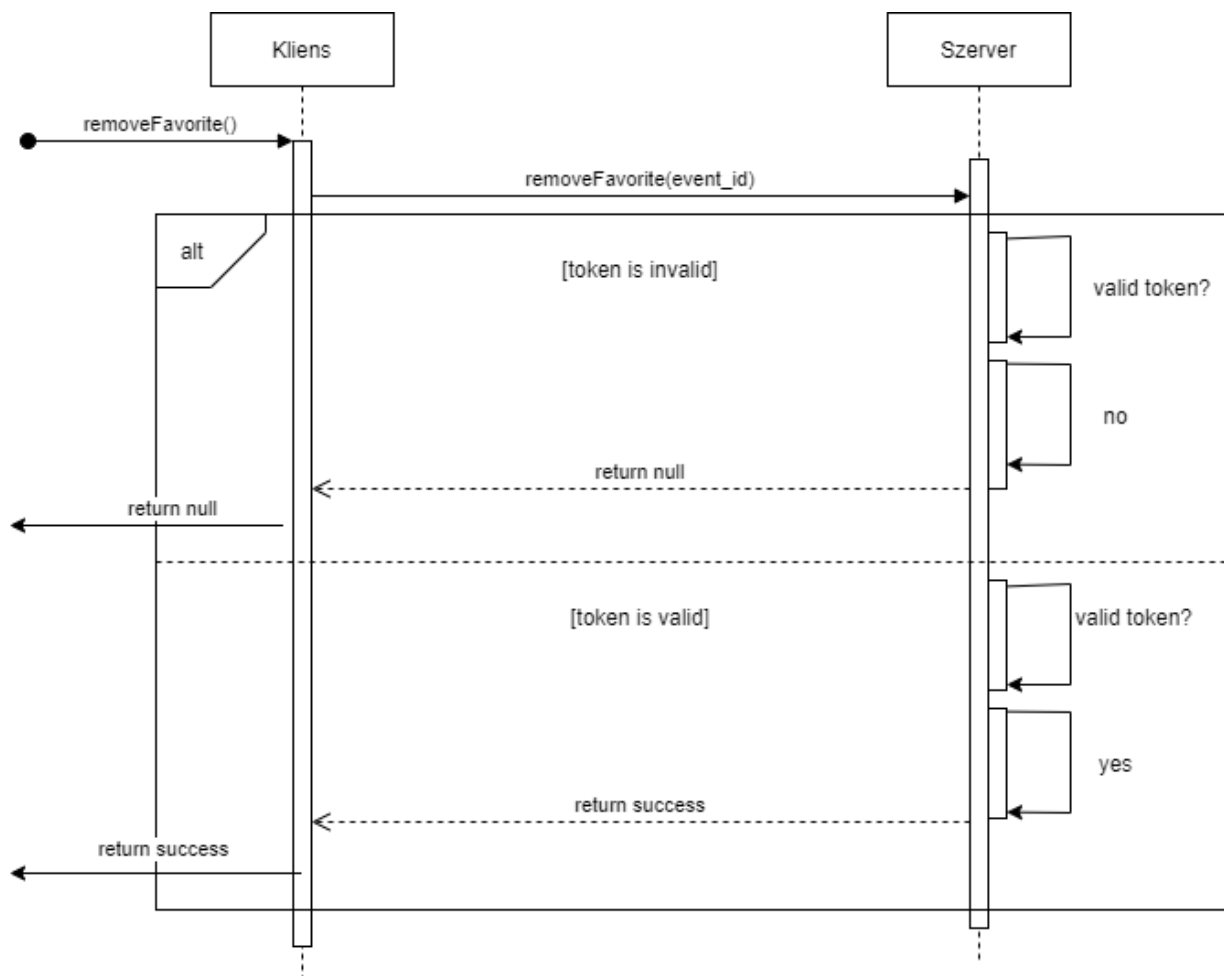
Abban az esetben, ha egy felhasználónak megtetszik egy esemény, akkor hozzáadhatja a kedvencekhez, így bármikor később meg tudja nézni az adott eseményt és nem kell újra keresgélni az eseménylistában.

A kedvencekhez adás is token alapján történik, így nem történhet meg, hogy más felhasználóhoz kerül az esemény.

A felhasználó az adott eseményen lévő kedvencek gombra kattintva tudja hozzáadni a listához az adott eseményt. A kedvencekhez adás után a felhasználó a Favorites gombra kattintva megtekintheti a kedvenceit.

A felhasználók el is tudják távolítani az eseményeket a kedvencekből, de ezt a következő ábrán mutatjuk be.

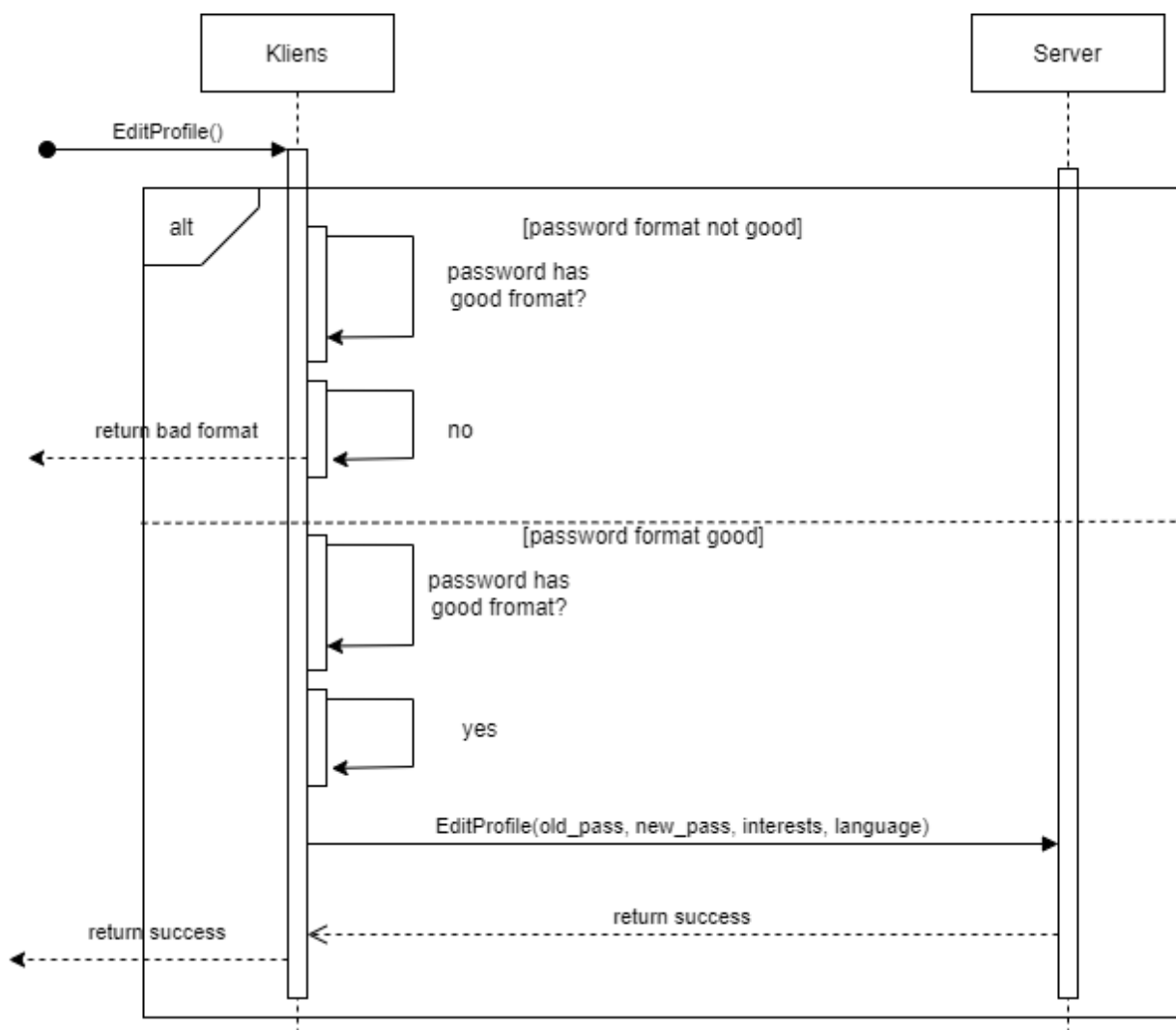
Az események eltávolítása a kedvencekből



Ahogy korábban már említettük a felhasználók el is tudják távolítani a kedvencekből az adott eseményeket. Szintén token-hez kötött az eltávolítás.

Maga az eltávolítás úgy történik, hogy az adott eseményen lévő kedvencekhez adás gombot kell újra megnyomni. Ezt nem feltétlen kell egyből a kedvencekhez adás után, bármikor megteheti a felhasználó később is.

A profile szerkesztése



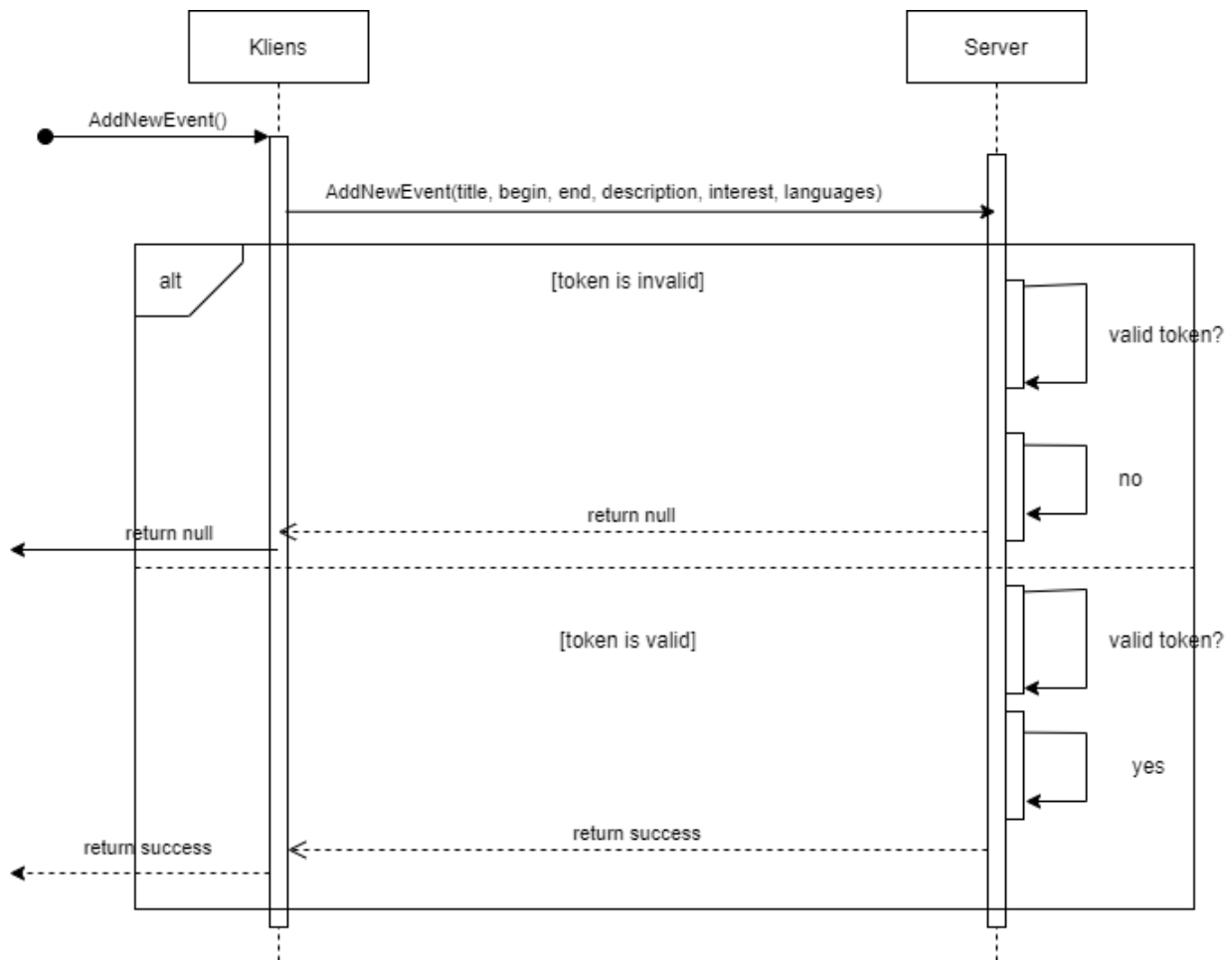
Természetesen utólag lehet szerkeszteni a regisztrációnál megadott paramétereket a felhasználóneven kívül. Jelszó változtatásra, érdeklődési kör, illetve nyelv változtatásra van lehetőség.

Továbbra is figyelni kell, hogy a jelszó megfelelő formátumban érkezzen, vagyis tartalmazzon legalább egy kisbetűt, egy nagybetűt, egy számot és legalább 8 karakter hosszú legyen.

Abban az esetben, ha a jelszó helyes formátumban van megadva, akkor kliens oldalon megtörténik az adatokat kiolvasása és ezeket küldjük el a szervernek.

A szerver, amint megváltoztatta a dolgokat, még visszaszól, hogy sikeres volt-e a tranzakció.

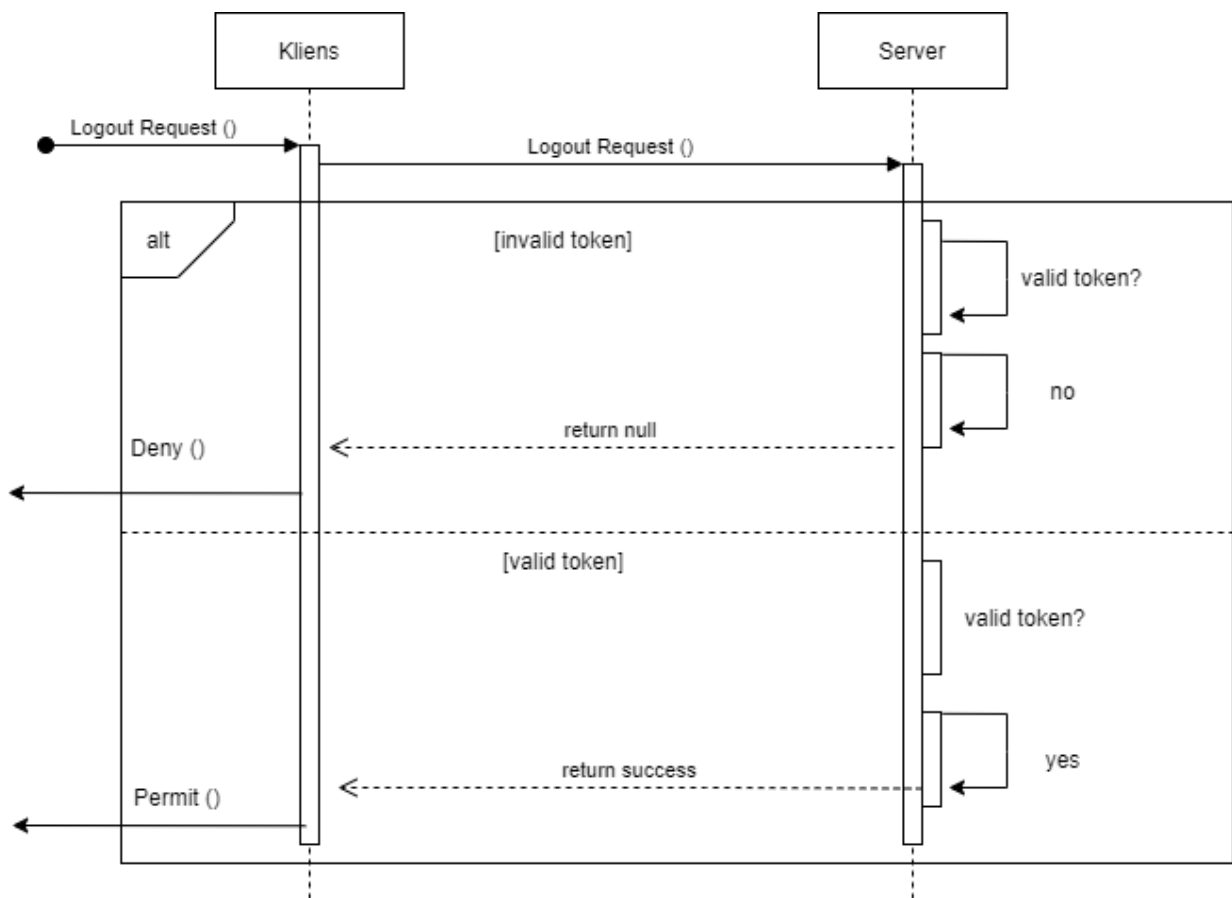
Új esemény létrehozása



Az új esemény létrehozásánál is megtörténik a token ellenőrzése, hogy megfelelő-e az adott token. Amikor egy felhasználó új eseményt akar létrehozni, akkor meg kell adnia, hogy mi legyen a címe, mikor kezdődik az adott esemény, mikor lesz vége az eseménynek, meg kell adni az eseményről egy hosszabb leírást, illetve meg kell határoznia, hogy melyik érkező körbe tartozik, és milyen nyelven vagy nyelveken kerül megrendezésre az esemény.

A szerver miután elmentette az adatbázisába az új eseményt visszaszól a kliensnek, hogy sikeres volt a tranzakció.

A kijelentkezés



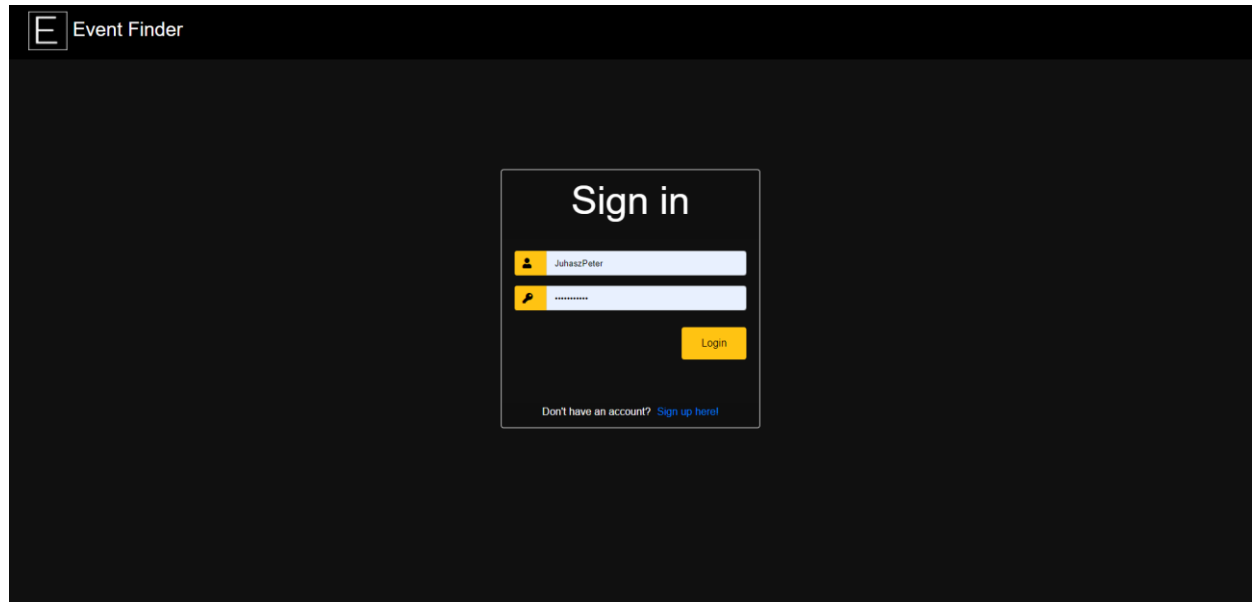
A kijelentkezés a következőképpen zajlik. A kliens tokenje elküldésre kerül a szervernek, innen tudja a szerver, hogy melyik kliens akar kijelentkezni.

Amint ez megtörtént, válaszol a kliensnek, hogy sikeres volt és a kliens visszairányítja a bejelentkező felületre.

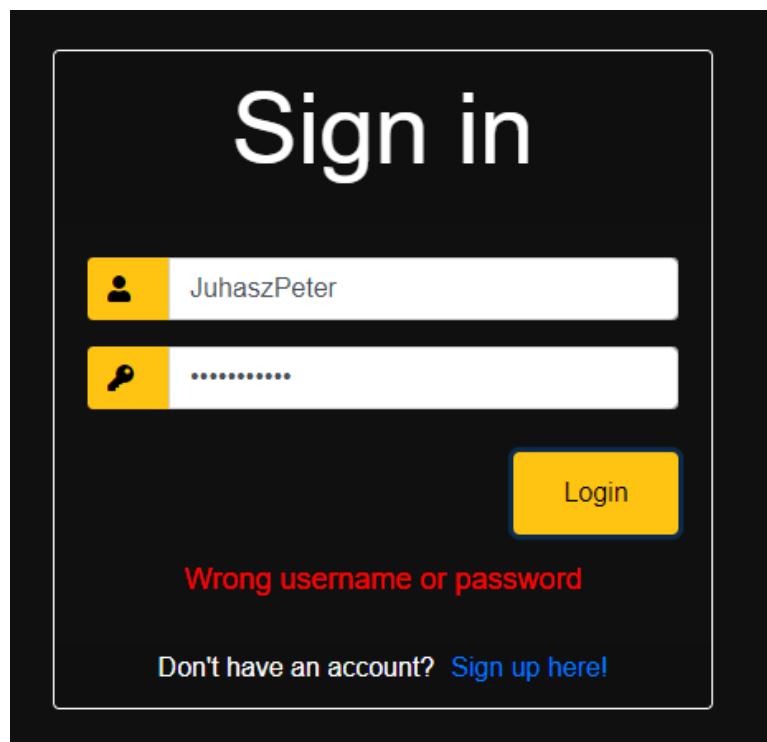
A grafikus felhasználói felület (GUI)

Ebben a fejezetben mutatjuk be, hogy is néz ki az általunk megálmodott és megalkotott alkalmazás.

1. A bejelentkezés:



Abban az esetben, ha nem jó a megadott felhasználónév, vagy jelszó, akkor az a következőképpen néz ki:



2. Regisztráció

E

Event Finder

Registration

username

password

confirm password

What are you interested in?

☐ I like watching sports.

☐ I like playing sports.

☐ I like watching movies in cinema.

☐ I like going to museum.

☐ I like hiking.

☐ I like cooking.

Which language(s) do you speak?

☐ I speak English.

☐ I speak German.

☐ I speak French.

☐ I speak Spanish.

☐ I speak Russian.

☐ I speak Hungarian.

Finish

Abban az esetben, ha nem megfelelő a jelszóformátum az a következőképpen látszódik:

Registration

JuhaszPeter

.....

.....

Password is matching

What are you interested in?

☒ I like watching sports.

☒ I like playing sports.

☐ I like watching movies in cinema.

☐ I like going to museum.

☒ I like hiking.

☒ I like cooking.

Which language(s) do you speak?

☐ I speak English.

☐ I speak German.

☒ I speak French.

☐ I speak Spanish.

☒ I speak Russian.


☐ I speak Hungarian.


Wrong password format


Finish

Abban az esetben, ha az adott felhasználónév foglalt, akkor az pedig így jelenik meg:

Registration

 JuhaszPeter





Password is matching

What are you interested in?

☒ I like watching sports.

☒ I like going to museum.

☒ I like playing sports.

☒ I like hiking.

☒ I like watching movies in cinema.

☒ I like cooking.

Which language(s) do you speak?

☒ I speak English.

☒ I speak Spanish.

☒ I speak German.

☒ I speak Russian.

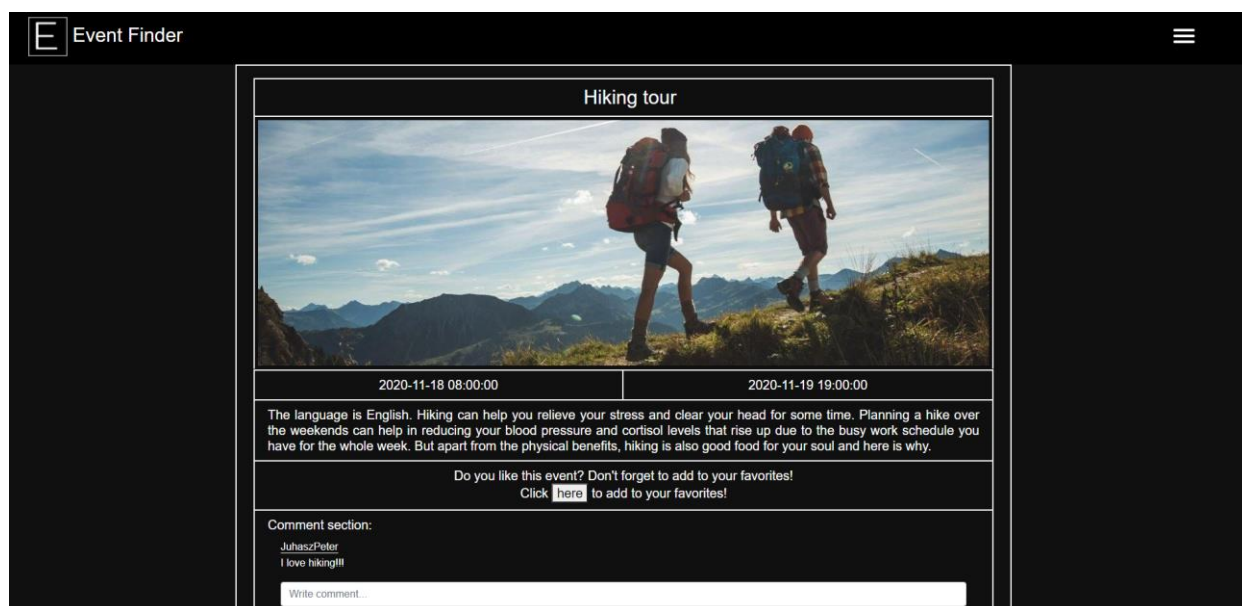
☒ I speak French.

☒ I speak Hungarian.

Username already taken

Finish

3. A főoldal



A felhasználóra jellemző események egy görgethető panelben jelennek meg. A Write comment... mezőben tudnak kommentet írni az adott eseményekhez. A navbar jobb oldalán található hamburger menü segítségével lehet navigálni a főoldal, a kedvencek, az új esemény létrehozása és a profil beállítások között. Ezen felül a kijelentkezés is ott található meg.

4. Profil beállítások

Account settings:

Username:

JuhaszPeter

Old password:

Edit

New password:

Confirm new password:

What are you interested in?

☒ I like watching sports.

☐ I like going to museum.

☒ I like playing sports.

☒ I like hiking.

☒ I like watching movies in cinema.

☐ I like cooking.

Which language(s) do you speak?

☒ I speak English.

☐ I speak Spanish.

☒ I speak German.

☒ I speak Russian.

☒ I speak French.

☒ I speak Hungarian.

Finish

A felhasználók itt tudják megváltoztatni a jelszavukat, az érdeklődési körüket, illetve a nyelvtudásukat. A finish gomb megnyomása után már frissülnek is a profil beállítások.

5. Új esemény létrehozása

Add new event

Title:

Beginning:

yyyy-mm-dd --:--

Ending:

yyyy-mm-dd --:--

Description:

Field of interests

☐ Watching sport.

☐ Museum tour.

☐ Playing sport.

☐ Hiking tour.

☐ Watching movie in cinema.

☐ Cooking lesson.

Which language(s)?

☐ English.

☐ Spanish.

☐ German.

☐ Russian.

☐ French.

☐ Hungarian.

Add

Az új esemény létrehozása a fenti képen látható módon történik és a websocket-nek köszönhetően egyből meg is jelenik oldal frissítés nélkül azok számára, akiknek belepaszsol az érdeklődési körébe.

Reszponzivitás

A grafikus felhasználói felületet megvalósítottuk részponzívan, így az alkalmazás máshogyan néz ki például laptopon, telefonon vagy tabletten megnyitva. Ehhez a Bootstrap media queries használtuk, amely attól függően, hogy mekkora méretben van nyitva az ablak, attól függően

használja a CSS beállításokat. Ügyeltünk arra is, hogy telefonon megnyitva az alkalmazást ne legyenek a szövegek, illetve a gombok kicsik, még álló helyzetben sem.

Az alkalmazás indítása:

1. A szerver (backend) indítása:

A szerver elindításához szükség van az images mappára amelyet a bináris szerver fájlal megegyező mappában kell elhelyezni. Ezt alap esetben publikálásnál, illetve fordításkor is megoldja a Visual Studio. A lefordított fájlt elég elindítani, melyet követően a szerver pár másodpercen belül már elérhető. A jelenlegi beállítások a localhost:5000-es porton indítják el a szervert, de ezt módosíthatjuk a konfigurációban, de ekkor természetesen a kliens konfigurációját is frissíteni kell a megfelelő működés érdekében. Konfigurációs fájlokban tároljuk még az adatbázis elérési helyét, illetve a szerver titkos kulcsát is, amely segítségével a tokeneket hozza létre. Az indításhoz szükség van a Microsoft .Net Core runtime minimum 3.1-es verziójára.

2. A kliens (frontend) indítása:

Az indítás megoldható például a python3 segítségével, amely képes lokális webszervert indítani a default 8000-es porton. Valamilyen konzolos felületen be kell navigálni a frontend mappába és ott kiadni az alábbi parancsot:

```
python3 -m http.server
```

Továbbfejlesztési lehetőségek

- Az események érdeklődési köre is lehetne dinamikus, így lehetőség lenne olyan érdeklődési kör megadására, amely korábban még nem szerepelt.
- Lehetne definiálni különböző role-okat, ez alatt gondolunk User és Admin role-ra. Különböző jogokkal bírnának a felhasználók. Lenne a mezei felhasználó, aki csakis kizárólag böngészni tudja az adott eseményeket, és azokhoz tud kommentelni. Azonban az admin ezzel ellentétben tudna feltölteni új eseményt a honlapra, ezzel együtt tudna törölni is eseményeket, illetve az admin képes lenne adott kommenteket is törölni.
- Esetleg meg lehetne valósítani olyat is, hogy az adott felhasználók tudjanak egymásnak írni privát üzeneteket is, ezzel is növelve a kommunikációs lehetőségek számát.