



EÖTVÖS LORÁND TUDOMÁNYEGYETEM  
INFORMATIKAI KAR  
PROGRAMOZÁSELMÉLET ÉS SZOFTVERTECHNOLÓGIAI  
TANSZÉK

**Bejelentett illegális szemétlerakatok  
térképes vizualizációja**

*Témavezető:*  
Cserép Máté András  
egyetemi tanársegéd

*Szerző:*  
Kiszely Gergő János  
programtervező informatikus BSc

*Budapest, 2024*

# EÖTVÖS LORÁND TUDOMÁNYEGYETEM

INFORMATIKAI KAR

## SZAKDOLGOZAT TÉMABEJELENTŐ

### Hallgató adatai:

**Név:** Kiszely Gergő János  
**Neptun kód:** D4CX8O

### Képzési adatak:

**Szak:** programtervező informatikus, alapképzés (BA/BSc/BProf)  
**Tagozat :** Nappali  
Belső témavezetővel rendelkezem

### Témavezető neve:

Cserép Máté  
munkahelyének neve, tanszéke: ELTE IK, Programozáselmélet és Szoftvertechnológia Tanszék  
munkahelyének címe: 1117, Budapest, Pázmány Péter sétány 1/C.  
beosztás és iskolai végzettsége: egyetemi tanársegéd, MSc programtervező informatikus

### A szakdolgozat címe:

Bejelentett illegális szemétlerakatok térképes vizualizációja

### A szakdolgozat témája:

(A témavezetővel konzultálva adja meg 1/2 - 1 oldal terjedelemben szakdolgozat témájának leírását )

A dolgozat célja a TrashOut alkalmazásból nyert hulladékpontok térképes vizualizációja a Duna-vízgyűjtő területén, ahol az adott pontokra kattintva megjelennek annak részletei. Ezeket a pontokat bonyos tulajdonságok alapján (pl. hulladéktípus, ország, stb.), illetve bejelentéstük- vagy frissítésük ideje alapján szűrni. Emellett van lehetőség ezen adatok letöltésére különféle formátumokban (pl. .csv, .xlsx, stb.). A beérkezett adatokat regisztrált szerkesztők, azoknak fiókjait pedig adminisztrátorok egy külön felületen kezelhetik. A program egy adatbázisra és annak webszolgáltatására (későbbiekben back-end), illetve egy felhasználói felületre lesz bontható, melyben az utóbbi egy webes felületet jelent.

A back-end C# programnyelven, a .NET keretrendszer eszközeinek segítségével fog készülni. Ennek az adatbázis része az előbb említett keretrendszer ún. Entity Framework nevű objektum-relációs leképző entitás-keretrendszer, a webszolgáltatás része, amely az adatbázis adatait továbbítja a felhasználói felületnek az ASP.NET szerveroldali-webalkalmazás keretrendszer felhasználásával lesz implementálva.

Budapest, 2023. 11. 29.

# Tartalomjegyzék

<b>1. Bevezetés</b>	<b>3</b>
<b>2. Felhasználói dokumentáció</b>	<b>4</b>
2.1. Rendszerkövetelmények . . . . .	4
2.1.1. Futtatás . . . . .	4
2.1.2. Használat . . . . .	5
2.2. Általános felhasználói tájékoztató . . . . .	5
2.3. Funkciók vendégként . . . . .	6
2.3.1. Felület áttekintése . . . . .	6
2.3.2. Térkép (kezdőoldal) . . . . .	7
2.3.3. Pont részletezése . . . . .	9
2.3.4. Listanézet . . . . .	10
2.3.5. Színséma választása . . . . .	10
2.3.6. Regisztráció . . . . .	11
2.3.7. Bejelentkezés . . . . .	14
2.4. Funkciók felhasználóként . . . . .	15
2.4.1. Pont bejelentése . . . . .	15
2.4.2. Pont módosítása . . . . .	17
2.4.3. Saját profil megtekintése . . . . .	17
2.4.4. Saját profil szerkesztése . . . . .	18
2.5. Funkciók moderátorként . . . . .	19
2.5.1. Pont törlése . . . . .	20
2.6. Funkciók adminisztrátorként . . . . .	20
2.6.1. Felület áttekintése . . . . .	21
2.6.2. Felhasználói fiókok listázása . . . . .	21
2.6.3. Felhasználói fiókok módosítása . . . . .	21
2.6.4. Felhasználói fiókok törlése . . . . .	22

<b>3. Fejlesztői dokumentáció</b>	<b>23</b>
3.1. Megoldási terv . . . . .	23
3.1.1. Követelményelemzés . . . . .	23
3.1.2. Rendszer-architektúra . . . . .	26
3.2. Rétegek . . . . .	27
3.2.1. Perzisztencia-réteg . . . . .	27
3.2.2. DTO-réteg . . . . .	31
3.2.3. Vezérlő-réteg . . . . .	32
3.2.4. Nézet-réteg . . . . .	34
3.2.5. Kliensoldal . . . . .	35
3.3. Megvalósítás . . . . .	36
3.3.1. Projektek szervezése . . . . .	36
3.4. Tesztelés . . . . .	38
3.4.1. Vezérlő-réteg automatikus tesztesetei . . . . .	38
3.4.2. Nézet-réteg manuális tesztelése . . . . .	38
3.5. Fejlesztési lehetőségek . . . . .	38
<b>4. Összegzés</b>	<b>39</b>
<b>Köszönetnyilvánítás</b>	<b>40</b>
<b>A. Szimulációs eredmények</b>	<b>41</b>
<b>Irodalomjegyzék</b>	<b>43</b>
<b>Ábrajegyzék</b>	<b>43</b>
<b>Táblázatjegyzék</b>	<b>44</b>
<b>Algoritmusjegyzék</b>	<b>45</b>
<b>Forráskódjegyzék</b>	<b>46</b>

# 1. fejezet

## Bevezetés

Az illegális szemétlerakatok jelensége modern világunkban egy súlyos probléma, amely nem csak környezetünket károsítja, hanem egészségünkre és közösségeink jólétére is kihathat. Ez a jelenség fokozottan megfigyelhető Magyarországon is, főként folyóink és tavaink, illetve településink határaiban. Ahogy világunk fejlődik és változik, úgy egyre több, ember által okozott szennyezéssel kell szembenéznünk azáltal, hogy megőrizzük környezetünk tisztaságát és egészségét. Habár ehhez az első lépés az emberek figyelmének felhívása, viszont a második mindenkorábban a már természetbe jutott szemét mihamarabbi eltávolítása és nem utolsó sorban a környezet rehabilitálása.

Ezen dolgozatnak a célja is hasonló, amely az előbb említett illegális szemétlerakatok adatbázisba gyűjtése a lakosság segítségével, illetve annak térképes vizualizációja, mely segít azoknak a civil önkénteseknek, akik a környezetvédelem érdekében alkalmilag, vagy akár rendszeresen takarítják és szállítják el ezeket. Ehhez társul egy minimalista, közösségi médiához hasonló felület is, ahol minden regisztrált felhasználó a saját profilján gyűjtheti a már általa bejelentett pontokat. A célt és az implementációt a *Tiszta Tisza Térkép* ihlette, ahol a *TrashOut* nevű alkalmazásból nyert végpontokat tüntetik fel hasonlóan a weboldalukon, mely a weboldal tulajdonosai által évente megrendezett *PET Kupa* című hulladékgyűjtő verseny alkalmával használják azt lehetséges szemétlelőhelyek fellelésére. Míg az előbb említett oldalon csak a pontok megtekintésére van lehetőség, az előbb említett alkalmazásból kinyert adatokkal, addig ebben a programban lehetőség van jóváhagyásos regisztrálás után azok bejelentésére, illetve azok adatainak frissítésére/javítására is.

## 2. fejezet

# Felhasználói dokumentáció

### 2.1. Rendszerkövetelmények

#### 2.1.1. Futtatás

Magának webalkalmazás szerveroldalának a futtatásához a ASP.NET Core Runtime 8.0-ra van szükség, amelyhez szükséges az alábbi operációs rendszerek valamelyike:

- Windows 10, 11 vagy Server 2012+
- 64-bites Ubuntu 20.04+ vagy Debian 11+
- macOS 10.15+ (csak külső, hosztolt szerver esetén támogatott)

A memóriahasználat a felhasználótól nagyban függ, így 0,5 GB-tól akár 4 GB RAM-ra is szüksége lehet a szervernek a zökkenőmentes futáshoz. A merevlemezen történő tárhelyhasználat nagyban függ az éppen bejelentett pontok méretétől és legfőképp az azokhoz csatolt fényképektől. Magához a futtatáshoz elegendő 0,5 GB tárhely is (az ehhez szükséges szoftvereket nem bele értve), viszont az alkalmazás használtat közben ez nagy mértékben fog növekedni az előbb említett okok miatt, így érdemes akár saját, nagy kapacitású merevlemezre telepíteni a szerveroldalt. Az szerver által használt SQL Server adatbázishoz (későbbiekben MSSQL-adatbázis) a Microsoft SQL Server szoftver valamelyik, de lehetőség szerint a legfrissebb, 2019-es verziójára van szükség, melyet a fentebb említett operációs rendszerek bármelyike képes futtatni. Az első üzembe helyezést követően nincsen szükség további rendszergazdai feladatokra.

### 2.1.2. Használat

A program felhasználói felülete egy weboldal. Ennek meglátogatásához elsősorban szükség van egy olyan operációs rendszer telepítésére, majd futtatására, melyre elérhető bármilyen friss, Chromium-alapú webböngésző. Egy ilyen böngésző a Google Chrome, amellyel ajánlott a weboldal látogatása. Hardver- és szoftver követelményei az alábbiak:

- Operációs rendszerek valamelyike:
  - Windows 10, 11 vagy Server 2012+
  - 64-bites Ubuntu 20.04+ vagy Debian 11+
  - macOS 10.15+
  - Android 8.0+ vagy Chrome OS
- Minimális hardver követelmények:
  - Pentium 4+ (vagy más, SSE3-at támogató) processzor
  - 2 (ajánlottan 4) GB RAM

Felhasználóként első üzembe helyezésre nincsen szükség, elég ha a felhasználó meglátogatja az oldalt. Ha még nem regisztrált felhasználói fiókot, akkor bizonyos funkciók eléréséhez regisztrálnia kell.

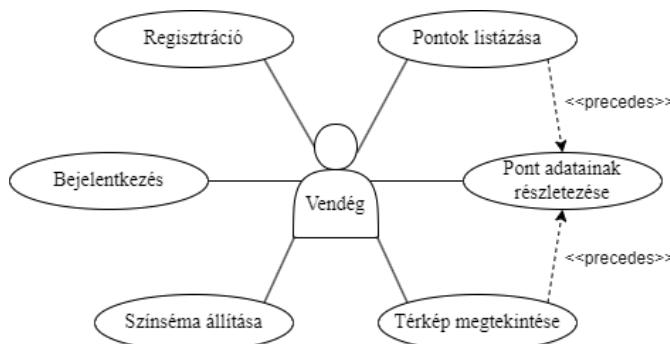
## 2.2. Általános felhasználói tájékoztató

Az weboldal használat közben négy felhasználói típusról beszélhetünk: vendég, regisztrált (és bejelentkezett) felhasználó, moderátor és adminisztrátor. Így vendégeknek számít aki még nem regisztrált, vagy ha igen, akkor nem jelentkezett be fiókjával, moderátorak aki moderátori szintű fiókkal és adminisztrátorak, aki adminisztrátori szintű fiókkal azonosította magát a bejelentkező felületen.

## 2.3. Funkciók vendégként

Egy vendég felhasználó számára hat funkció érhető el a weboldalon:

- regisztrálás új-, illetve bejelentkezés már meglévő fiókba
- a térkép megtekintése a bejelentett pontokkal (ez a főoldal)
- egy-egy pont adatainak részletesebb megtekintése
- az összes pont listázása
- a weboldal színsémájának ("sötét"- vagy "világos" mód) tetszőleges beállítása

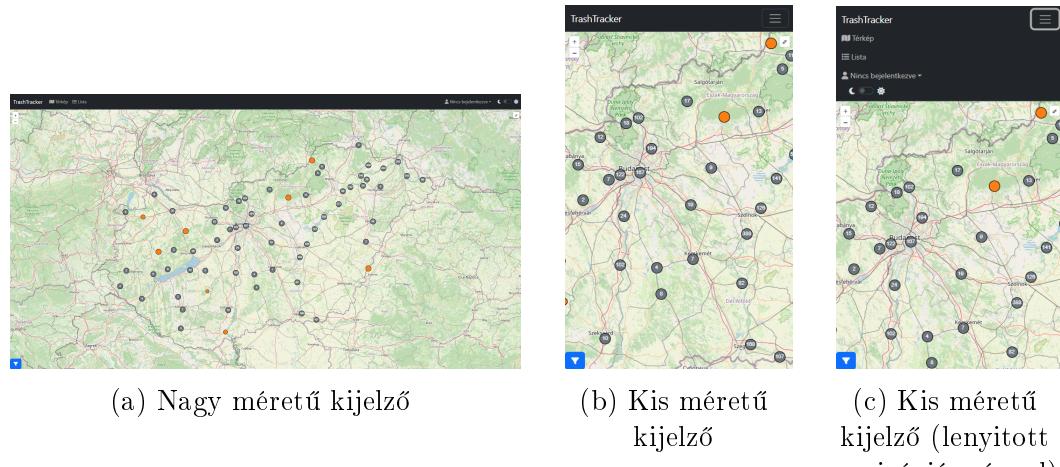


2.1. ábra. Felhasználói esetek vendégként

### 2.3.1. Felület áttekintése

A weboldal látogatása során több felület is megjelenik, viszont ezek általában leírhatóak. minden lap tetején egy navigációs sáv van, melynek segítségével több funkciót is gyorsan elérhetünk. A sáv bal oldalán a weboldal neve, illetve a "Térkép" felirat egyaránt a térképre, azaz a főoldalra irányít, a mellette szereplő "Lista" pedig a pontok listájához. A másik oldalon (jobbról balra) egy fizikai kapcsolót imitáló gomb, melynek segítségével válthatunk az előbb említett színsémák között, illetve amellett egy "Nincs bejelentkezve" feliratú listát lenyitva tudunk a regisztráció, illetve a bejelentkezés közül választani.

Ha a felhasználó nem egy széles képernyőn (pl. asztali számítógép vagy fektetett táblagép) látogatja meg az oldalt, akkor a sáv bár hasonlóan ott van, viszont a könnyebb használat érdekében a menüpontok egy, a jobb oldalon található  $\equiv$  ikonra kattintva lehet ezeket elérni.



2.2. ábra. A felület kinézete a kezdőoldalon különböző méretű kijelzők esetén

Ezeket a képet célszerűbb lenne nagyobb méretben beszúrni. A b) és c) még lehet együtt, de az a) lehetne külön.

### 2.3.2. Térkép (kezdőoldal)

A navigációs sávon a weboldal nevére, illetve a "Térkép" feliratra kattintva azonnal a bejelentett pontok térképére navigálhatunk. Ennek kezelése a már jól ismert térképes alkalmazásokéhoz hasonló. A bal felső sarokban található +, illetve – ikonra kattintva nagyítani, illetve kicsinyíteni lehet, emellett egérrel a görög görgetésével és érintőképernyőn két ujj összehúzásával/széttolásával is lehetőség van. A jobb felső sarokban a térképet teljes képernyős módba is lehet helyezni, melynek ismételt megnyomásával, vagy billentyűzeten az Esc gombbal ki lehet lépni. A bal alsó sarokban található a szűrés gomb, melynek segítségével különböző, a felhasználók által jelentett tulajdonságok szerint van lehetőség.

#### Jelmagyarázat

A térképen a pontok különböző mérettel és színnel jelennek meg, melyeknek céljuk a fontosabb információk lehetőség szerinti kiemelése.

A szemét mennyisége szerint a pont mérete lehet

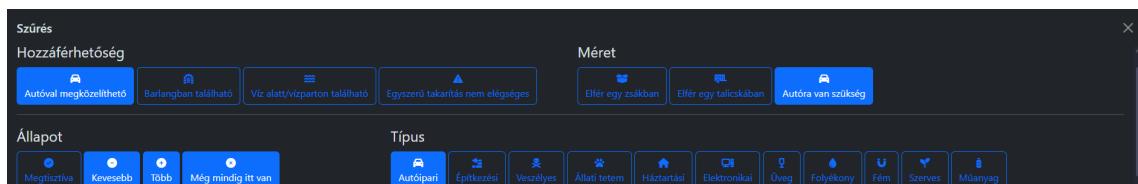
- kicsi (8 pixel), azaz "elfér egy zsákban",
- közepes (10 pixel), azaz "elfér egy talicskában" és
- nagy (12 pixel), azaz "autóra van szükség".

A szemét első bejelentés óta levő állapota szerint a pont színe lehet

- zöld, azaz "megtisztítva",
- sárga, azaz "kevesebb",
- narancs, azaz "még mindig itt van" és
- piros, azaz "több".

## Szűrés

A fentebb említett szűrés gombra kattintva a képernyő alján beúszik egy görgethető menü, amely a pontok tulajdonságai szerinti szűrést teszi lehetővé. A szemét mennyisége és állapota mellett annak típusa és hozzáférhetősége szerint is lehet válogatni. Ezeket a beúszó menün található, tulajdonságok szerint csoportosított "kapcsoló" gombokkal lehet elérni. Ha egy ilyen gomb kék színű, akkor igen, ha átlátszó, akkor pedig nem szeretnénk a térképen látni. Alapértelmezetten az összes ilyen gomb "igen" állapotban van. A gombok között szűrés esetén a logikai vagy művelete értelmezett, tehát ha bejelölésre kerül egy tulajdonság, viszont egy másik nem, akkor attól függetlenül a bejelöletlen attribútum még előfordulhat a megjelenített pontok között, feltéve, hogy legalább egy bejelölt jellemzővel rendelkezik. Ezt a menüt a jobb felső sarokban található gombra nyomva lehet bezárni, a szűrőket megtartva, egy másik oldalra navigálásig vagy az oldal újratöltéséig.



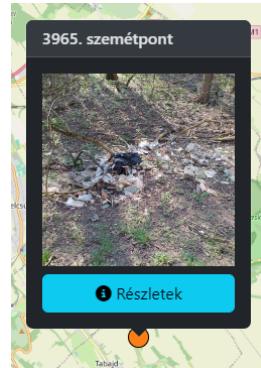
2.3. ábra. A kezdőoldal alján beúszó szűrési menü

## Pontok letöltése

A térképen szereplő bejelentett pontok letöltésére is van lehetőség .csv formátumban. Ehhez a térkép jobb alsó sarkában található ikonnal ellátott gombra kell kattintani, mellyel automatikusan elindul a fájl letöltése. Ez a visszűvel elválasztott formátum könnyen importálható táblázatkezelő szoftverekbe, illetve egyszerűen felhasználható külön, a fájl feldolgozására implementált programokban is.

## Betekintés és részletek

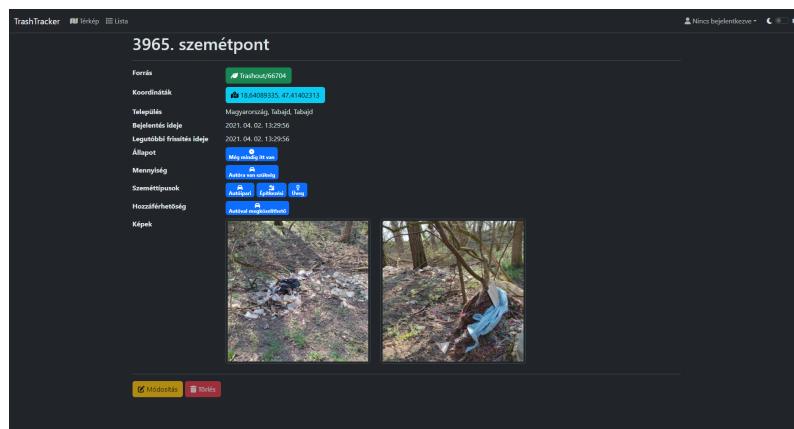
Egy pontra való kattintáskor afelett megjelenik egy modális "buborék", amely minimális információt és egy képet tartalmaz a hozzá tartozó szemétről. Emellett a kis ablak alján található egy "Részletek" gomb, melyre kattintva elnavigálhatunk az azt részletező adatlapra.



2.4. ábra. Példa egy felbukkanó buborékra egy ponton

### 2.3.3. Pont részletezése

Ezt az oldalt az előző bekezdésben említett "Részletek" gombra kattintva, vagy a következő bekezdésben tárgyalt **i** ikonra kattintva lehet elérni. Itt az egyszerű tulajdonságok részletezésén túl megtalálható a pont pontos helyzete, illetve annak forrása (felhasználó esetén bejelentője). Előbbire kattintva a térképen levő helyre, utóbbira pedig annak forrására (TrashOut esetén az ottani adatlapjára, felhasználó esetén annak a profiljára) lehet navigálni.



2.5. ábra. Példa egy szemétpontot részletező adatlapra

### 2.3.4. Listanézet

A pontokat nem csak térképen, hanem listaként is megtekinthetjük. Ennek célja, hogy a bejelentés/frissítés dátuma szerint rendezve könnyedén fel lehessen fedezni a legrelevánsabb szemétlerakatokat. A lista minden sora egy-egy pontot, annak oszlopai pedig annak a fejlécben szereplő tulajdonságát tartalmazzák. Egy adott rekord részletezésére is lehetőség van, a jobb oldalon található **i** ikonra kattintva. Az oldal tetején lehetőség van alapvető szűrésekre is, opcionálisan belefoghatjuk a listába a már megtisztított pontokat, kereshetünk a bejelentő felhasználónévben, illetve az ahhoz tartozó megjegyzéseiben, illetve állíthatjuk az egy oldalon szereplő találatok maximális számát. Ha több találat van, mint ami az oldalon elfér, akkor a lista téjén, illetve alján található nyilakkal és számokkal könnyedén navigálhatunk azok között.

2.6. ábra. Szemétpontok listanézetének felülete

### 2.3.5. Színséma választása

Minden oldal navigációs sávjának jobb szélén található egy kapcsoló mellyel könnyedén állítható az adott oldal színsémája, melynek bal- (🌙) állása sötét, illetve jobb oldali (☀️) állása világos módot jelent. Alapértelmezetten ez az előbbi állapotban van az energiatakarékosabb működés és a szemet jobban kímélő megjelenés miatt. Ez a beállítás szabadon állítható, mely megmarad minden aloldalon a böngészés idejére.

Szemétpontok				
Forrás	Koordinátek	Bejelentés ideje	Címletek	Megjegyzés
Trashout/62460	18.855676 47.424458	2024. 05. 10. 17:40:10	        	Bejelentő: Polgári Termesztettség, Erd  
Trashout/62470	20.132461 47.26568	2024. 05. 01. 9:09:05	        	
Trashout/62469	20.131913 47.265372	2024. 05. 01. 9:07:51	        	
Trashout/62468	20.131669 47.265288	2024. 05. 01. 9:07:05	        	

2.7. ábra. Szemétpontok listanézetének felülete világos módban

### 2.3.6. Regisztráció

További funkciók használatáért (pontok bejelentése és frissítése) a felhasználónak egy felhasználói fiókra van szüksége, mely utóbbija ha nincs, akkor regisztrálnia kell hozzá. Ezt a navigációs sáv jobb oldalán található "Nincs bejelentkezve" feliratra kattintva lenyíló menüben szereplő "Regisztráció" menüpontra nyomva érheti el. Ha az előbbi helyett egy ikonnal társulva egy felhasználó neve szerepel, akkor az már be van jelentkezve, ezzel kapcsolatosan nincsen teendője.

A regisztrációs felületen egy űrlap található, ahol meg kell adni a kívánt felhasználónevet, e-mail címet (elérhetőséggént), jelszavat, illetve opcionálisan egy profilképet. Az adatok kitöltése után az űrlap alján található "Regisztráció" gombra kattintva lehet beadni az adatokat, melyek ha megfelelnek a követelményeknek, akkor a felhasználó a főoldalon találja magát, az új fiókjával bejelentkezve.

The registration form consists of four input fields: 'Felhasználónév' (Username), 'E-mail cím' (Email address), 'Jelszó' (Password), and 'Jelszó megerősítése' (Confirm password). Below these is a file upload section for a profile picture ('Profilkép') with the message 'Fájl kiválasztása' (File selection) and 'Nincs fájl kiválasztva' (No file selected). At the bottom are two buttons: a grey 'Bejelentkezés' (Log in) button and a blue 'Regisztráció' (Registration) button.

2.8. ábra. A regisztrációs felület űrlapja

## Hibaüzenetek

A regisztrációkor megadott adatok több követelménynek is meg kell hogy feleljenek, melyek "megsértésekor" hibaüzenetek jelenhetnek meg az űrlap kitöltése közben vagy után. Az első ilyen követelmény, hogy minden felhasználónévnek egyedinek kell lennie. Ezt az űrlap helyes kitöltése és elküldése után egy "**Ez a felhasználónév már foglalt!**" felirat jelzi.

A szövegben  
ez és a hason-  
lók ne legye-  
nek pirosak,  
inkább pl. dőlt-  
tel szedd, ha ki  
akarod emelni.

The registration form includes an error message at the top: '• Ez a felhasználónév már foglalt!' (This username is already taken!). The rest of the form is identical to Figure 2.8, with input fields for 'Felhasználónév' (username 'felhasznalonev'), 'E-mail cím' (email 'email@example.com'), 'Jelszó' (password), 'Jelszó megerősítése' (confirm password), and a file upload section for a profile picture.

2.9. ábra. Már foglalt felhasználónév esetén megjelenő hibaüzenet

Ha a megadott e-mail cím nem érvényes (azaz nem felel meg annak formai követelményeihez, pl. legyen benne "@" és egy ".") , azt az űrlap kitöltése közben jelzi egy "Érvénytelen e-mail cím!" feliratú üzenet.

The screenshot shows a registration form titled "Regisztráció". The fields are as follows:

- Felhasználónév: felhasznalonev
- E-mail cím: ervenytelenEmail  
Érvénytelen e-mail cím!
- Jelszó: (redacted)
- Jelszó megerősítése: (redacted)
- Profilkép: Fájl kiválasztása | Nincs fájl kiválasztva
- Bejelentkezés (button)
- Regisztráció (button)

2.10. ábra. Érvénytelen e-mail cím esetén megjelenő hibaüzenet

A legtöbb követelmény a jelszavakat érinti, itt több hibaüzenettel is találkozhat a felhasználó. Biztonsági okokból, egy jelszó 6-32 db karakterből kell álljon, melynek tartalmaznia kell legalább egy-egy kis- és nagybetűt, illetve számjegyet is. Ha ennek mégsem felelne meg a megadott jelszó, akkor azt az e-mail címhez hasonlóan egy "A jelszó nem lehet rövidebb 6 és hosszabb 32 karakternél!" vagy "Jelszónak tartalmaznia kell legalább egy kis- és nagybetűt illetve számjegyet!" üzenet jelenik meg, annak függvényében, hogy melyik pontján nem megy át az ellenőrzésen (tehát ha túl rövid a jelszó és pl. nem tartalmaz nagybetűt, akkor csak az előbbi jelenik meg, az utóbbi csak ha a hossz javítása után sem felel meg). Emellett a jelszó megerősítésére is szükség van, azaz meg kell adni a beírt jelszót egy második alkalommal is. Ha ezek nem egyeznének meg, azt is egy hibaüzenet jelzi a felhasználó számára.

The figure consists of three side-by-side screenshots of a registration form. Each screenshot shows a dark-themed interface with input fields for 'Felhasználónév / felhasználónev' (Username / user name), 'E-mail cím / email@example.com' (Email address), 'Jelszó / \*\*\*\*\*' (Password), and 'Jelszó megerősítése / \*\*\*\*\*' (Confirm password). Below these are dropdown menus for 'Profilfény / Fájl kiválasztása' (Profile picture / File selection) and 'Felhasználónév / felhasználónev' (Username / user name). At the bottom are two buttons: 'Bejelentkezés' (Login) and 'Regisztráció' (Registration).

- (a) Nem megfelelő hosszú jelszó hibaüzenete
- (b) Nem megfelelő karaktereket tartalmazó jelszó hibaüzenete
- (c) Nem azonos jelszavak hibaüzenete

2.11. ábra. Érvénytelen jelszó esetén megjelenő hibaüzenetek

### 2.3.7. Bejelentkezés

Ha a felhasználónak már van saját fiókja, akkor ismételt regisztráció helyett beléphet abba, az előző bekezdésben tárgyaltak szerint a "Bejelentkezés" gombra kattintva. Az űrlap hasonló, viszont itt elég csak a felhasználónév (vagy e-mail cím tetszőlegesen), illetve a hozzá tartozó jelszó egyszeri megadására. Ha az adatok egyeznek a regisztrációkor megadottakkal, akkor az űrlap alján található "Bejelentkezés" gombra kattintva a felhasználó a főoldalra lesz átirányítva, a felhasználói fiókjába bejelentkezve. Ellenben, ha nem jár sikerrel, akkor üzenetek tájékoztatják a klienst erről. Ha többszöri próbálkozásra sem jár sikерrel, akkor lehetősége van bejelentkezés helyett újabb felhasználó regisztrálására, vagy egy adminisztrátort megkérve bejelentkezési adatainak megváltoztatására is.

The figure shows a dark-themed login form titled 'Bejelentkezés'. It contains two input fields: 'Felhasználónév vagy E-mail cím' (Username or email address) and 'Jelszó' (Password). Below the fields are two buttons: 'Regisztráció' (Registration) with a person icon and 'Bejelentkezés' (Login) with a right-pointing arrow icon.

2.12. ábra. A bejelentkezéses felület űrlapja

## Hibaüzenetek

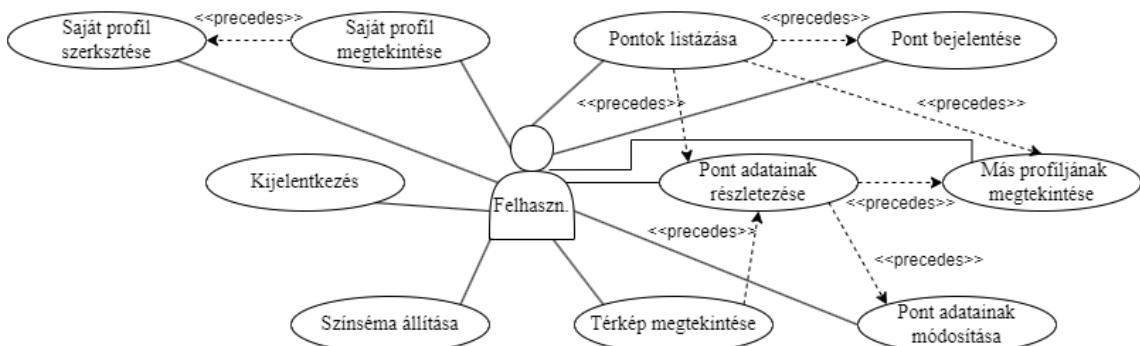
Sikertelen bejelentkezés esetén, azaz nem megfelelő felhasználónév (vagy e-mail cím) és jelszó páros megadásakor biztonsági okok miatt csak egy, a regisztráláskor látottaknál általánosabb "Sikertelen bejelentkezés!" felirat szerepel.

The screenshot shows a dark-themed login form. At the top, a red success message reads "Sikertelen bejelentkezés!" (Successful login). Below it is a text input field labeled "Felhasználónév vagy E-mail cím" (Username or email address) containing "felhasznalo". Underneath is a password input field labeled "Jelszó". At the bottom left is a grey button labeled "Regisztráció" (Registration) with a user icon. On the right is a blue button labeled "Bejelentkezés" (Login) with a right-pointing arrow icon.

2.13. ábra. A bejelentkezéses felület űrlapján szereplő hibaüzenet

## 2.4. Funkciók felhasználóként

Sikeres bejelentkezést (vagy regisztrációt) követően a felhasználó vendég helyett bejelentkezett felhasználónak (vagy fióktípusának megfelelőnek) számít. Ezzel további funkciók válnak elérhetővé, mint például a szemétpontok bejelentése és frissítése, illetve a saját felhasználói fiók profiljának szerkesztése.



2.14. ábra. Felhasználói esetek (bejelentkezett) felhasználóként

### 2.4.1. Pont bejelentése

Új pont bejelentésére a listanézet tetején található **+** ikonra kattintva van lehetőség, mely elnavigál annak az űrlapon meg kell adni a szemét pontos helyzetét koordinátként, amely megadásának segítségül szolgál a szöveges mezőjük melletti "Jelenlegi hely meghatározása", mely megnyomásával a böngésző meghatározza a bejelentőnek a lehető leg pontosabb koordinátáit. A méret jellemzése mellett ajánlott a szemét típusát és hozzáférhetőségét is megadni, melynek kezelése hasonló a térképen való szűrés gombjaihoz. Az utolsó, többsoros szöveges mezőben megjegyzést lehet írni, mely segíti a listanézetben való keresését a pontnak. Az űrlap

alján a "Vissza" gombbal lehet az előbb meglátogatott oldalra navigálni, illetve a "Bejelentés" gombra kattintva is ide lesz átirányítva a felhasználó

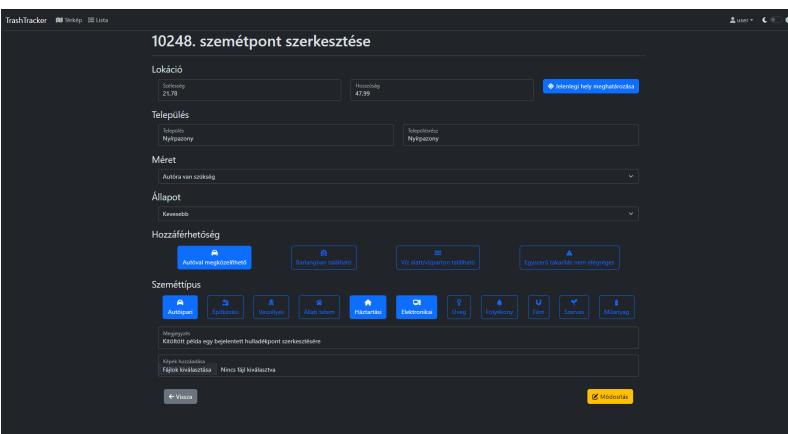
2.15. ábra. Szemétpont bejelentésének űrlapja

## Hibaüzenetek

Egy pont bejelentésekor egyedül a koordináták megadása kötelező, de erősen ajánlott további adatok leírása, illetve fénykép csatolása is. Ha a felhasználó nem ad meg egy kötelező adatot vagy beküldés előtt törli azt, akkor az űrlap egy "**<adat> megadása kötelező!**" hibaüzenettel jelzi azt az adott mező alatt. A koordinátákkal kapcsolatosan vannak triviális követelmények is, mely szerint nem lehet kevesebb, mint -180 és több, mint 180, mely megsértésekor egy "**A koordináta nem lehet kevesebb, mint -180° és több, mint 180°.**". Emellett az elfogadott formátum a tizedesvessző (tehát nem a -pont) használata. A megjegyzésbe bármit lehet írni, viszont biztonsági okokból az nem lehet több, mint 2000 karakter, mely túllépése esetén egy "**2000 karakternél nem lehet hosszabb a megjegyzés!**". A képekkel kapcsolatosan van a legtöbb technikai követelmény, melyek kizárolag fájlok, azon belül is kép, ahol csak ".jpg" (vagy ".jpeg"), illetve ".png" formátumúak lehetnek, amelyek nem léphetik túl az 1 MB méretet. Ezen pontok megsértésekor rendre "**Az objektum nem fájl!**", "**A fájl nem egy kép!**", "**Ez a kép formátuma nem támogatott! Támogatott formátumok: ".jpg", ".jpeg", ".png".**", illetve "**Ez a kép túl nagy! Maximális méret: 1024KB.**" hibaüzenetekkel találkozhat a felhasználó. **A bejelentés csak akkor végelesedik, ha hiba nélkül sikerül megadni az adatokat!**

### 2.4.2. Pont módosítása

Egy már meglévő pont adatainak módosítására az azt részletező adatlap alján található "Módosítás", vagy a listanézet egyes soraiban található  ikonra kattintva érhető el. Az ezen az oldalon található űrlap hasonló, a pont létrehozáskor használtéhoz, a szemét állapotának lenyíló menüpöt leszámítva, melynek megadása létrehozáskor még értelmetlen lenne. Emellett a meglevő képek felülírása helyett az itt megadtak (a hozzátartozó mező leírásához hűen) a ponthoz hozzáadásra kerülnek, a régieket megtartva.



The screenshot displays the '10248. szemétpont szerkesztése' (Edit point 10248) page. At the top, there are sections for 'Lokáció' (Location) and 'Település' (Municipality). Below these are dropdown menus for 'Méret' (Size), 'Állapot' (Status), and 'Hozzáférhetőség' (Accessibility). A prominent blue button labeled 'Ausztria megtörzseni' (Delete Austria) is visible. The 'Szemétpont típus' (Point type) section contains a large grid of icons representing different point types, with several being highlighted in blue. At the bottom, there are sections for 'Magyarország' (Hungary) and 'Közvetlen belépés a profilba' (Direct access to the profile), along with a 'Mentés' (Save) button.

2.16. ábra. Szemétpont módosításának űrlapja

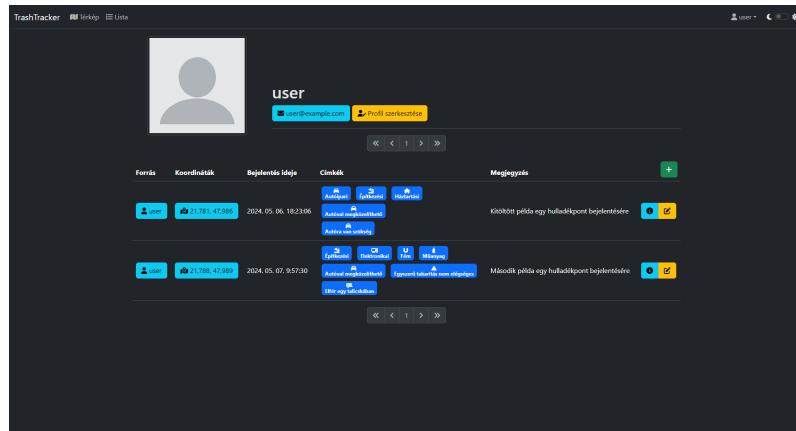
### Hibaüzenetek

Az itt megjelenő hibaüzenetek azonosak a 2.4.1 bekezdésben tárgyaltakkal.

### 2.4.3. Saját profil megtekintése

A felhasználó regisztrálásával mindenki egy saját profil oldalra is szert tesz. A saját megtekintéséhez a navigációs sáv jobb oldalán, a felhasználónévre kattintva lenyíló menüben a "Saját profil" menüpontra nyomva van lehetőség.

Minden profiloldal fejlécében a bal oldalon a profilkép (ha van, ellenben egy "Nincs profilkép" felirat), mellette pedig a felhasználó neve, illetve elérhetősége (e-mail címe) található. Ha a saját proflunk böngésszük, akkor az elérhetőségünk mellett találjuk a "Profil szerkesztése" gombot, mely értelemszerűen annak a szerkesztését segítő űrlapra navigál. Ez alatt az adott felhasználó által bejelentett pontok listanézete található, amely használata azonos a 2.3.4 bekezdésben tárgyaltakkal, a szűrési lehetőségeket leszámítva.



2.17. ábra. Példa egy felhasználói profilra

### Más felhasználó profiljának megtekintése

Egyszerű felhasználói fiókkal nincs lehetőség további felhasználók "felderítésére", csak az általuk bejelentett pontokon levő hivatkozásokkal. Mind a pontok listanézetében (2.3.4 bekezdés), illetve azok adatlapján (2.3.3 bekezdés) találhatóak ezek, ahol az adott fejezetekben tárgyaltak szerinti ikonnal ellátott gombokra kattintva van lehetőség.

#### 2.4.4. Saját profil szerkesztése

A saját profil megtekintésekor (2.4.3 bekezdés) a "Profil szerkesztése" gombra kattintva van lehetőség a felhasználónév, e-mail cím és profilkép módosítására. Ehhez is egy űrlap áll a rendelkezésre, mely a gombra kattintáskor jelenik meg, az előbb említett adatok mezőivel. Az alján található "Módosítás"-ra nyomva azok a mezők, melyek tartalmaznak adatot, és ha tartalmaznak akkor újat, módosulni fognak, feltéve ha hiba nélkül megy végbe az.

(a) Profil szerkesztésének űrlapja
(b) Szerkesztés utáni profiloldal

2.18. ábra. Profil szerkesztése és annak eredménye

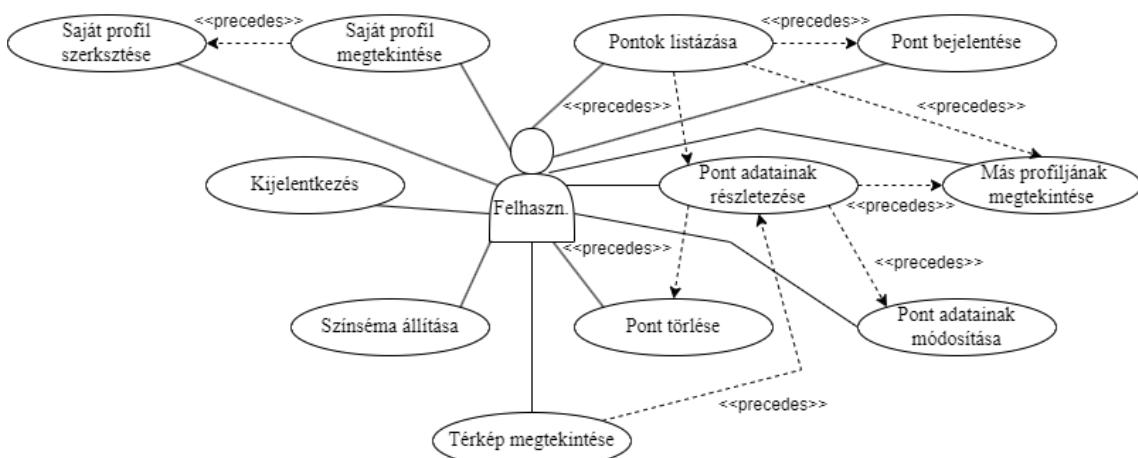
## Hibaüzenetek

Az űrlap elküldésekor van lehetőség, hogy sikeres módosítás helyett hibaüzenetet kapunk (ilyenkor az adatok nem módosulnak még). Mivel minden felhasználónévenek és e-mail címnek egyedinek kell lennie, így ha azok mégsem lennének a módosítás után, akkor rendre "A felhasználónév már foglalt!", illetve "Az e-mail cím már foglalt!" üzenetekkel találkozhat a felhasználó. Emellett ez e-mail címnek érvényes formátumúnak kell lennie, és a profilképnek is meg kell felelnie a regisztráláskor definiáltakkal, mely a 2.3.6 bekezdésben van tárgyalva.

2.19. ábra. Profil szerkesztésének űrlapja

## 2.5. Funkciók moderátorként

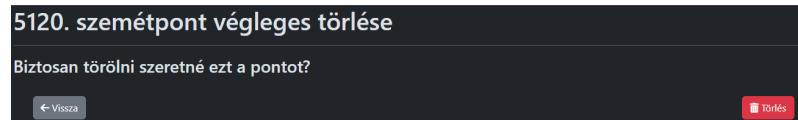
A moderátor szintű felhasználói fiókok mindenre jogosultak, mint az egyszerű felhasználók, viszont képesek bejelentett pontok törlésére, ezzel az esetleges "spam" moderálására.



2.20. ábra. Felhasználói esetek moderátorként

### 2.5.1. Pont törlése

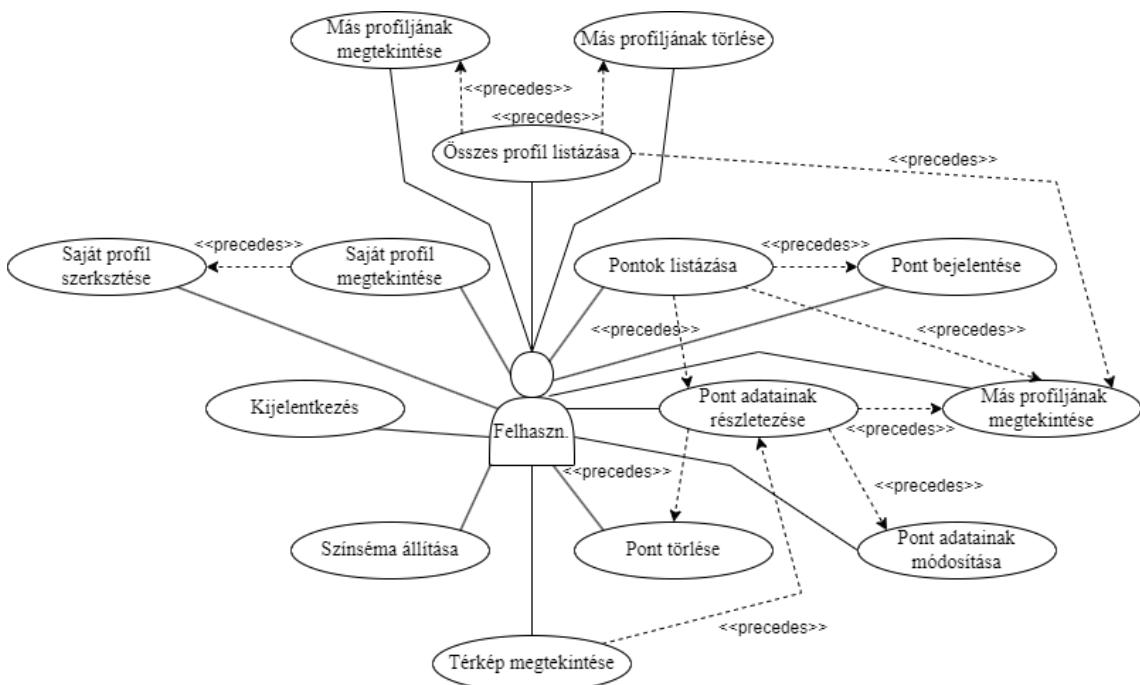
Egy pont törlésére azok listanézetéből a ikonra vagy annak adatlapján található "Törlés" gombra kattintva van lehetőség. A "balesetek" elkerülése érdekében először egy megerősítő oldalra navigál, ahol "Vissza" és "Törlés" opciók közül lehet választani, melyek rendre az előző oldalra navigálnak (törölés nélkül), illetve véglegesen törlik az adott pontot.



2.21. ábra. Pont törlésének megerősítő űrlapja

## 2.6. Funkciók adminisztrátorként

Az adminisztrátori szintű fiókok mindenre jogosultak, tehát az előbb említett funkciók mellett képes a felhasználói fiókok listázására, azok módosítására, illetve törlésére is.



2.22. ábra. Felhasználói esetek adminisztrátorként

### 2.6.1. Felület áttekintése

A 2.3.1 bekezdésben tárgyalt navigációs felületen adminisztrátorként egy újabb hivatkozás válik elérhetővé "Felhasználók" névvel. Erre való kattintással elérhető az összes felhasználói fiók listázása.

### 2.6.2. Felhasználói fiókok listázása

Ez a felület a 2.6.1 bekezdésben tárgyalt navigációs sávban található menüponttal érhető el, ha adminisztrátori szintű fiókkal van a felhasználó bejelentkezve. Kezelése hasonló a 2.3.4 bekezdésben tárgyalt szemétpontok listanézetével. A lista tetején ugyanúgy megtalálható a már megismert szűrő beállítások, ahol felhasználó vagy e-mail cím alapján lehet keresni, illetve a találatok lapszámát állítani. Ez alatt található a lista, ahol minden fiók felhasználónévét, e-mail címét és regisztrálási idejét (ha alapértelmezetten generált fiók, akkor az érték "0001. 01. 01. 0:00:00") lehet megtalálni, regisztrálás ideje szerint sorrendben (azaz a legfrissebb fiókok vannak előbb). Emellett minden rekord végén található három gomb, mely rendre az adott fiókhöz tartozó profil megtekintésére, -szerkesztésére és törlésére szolgál.

Felhasználónév	E-mail cím	Regisztrálás ideje
user	user@example.com	2001. 01. 01. 0:00:00
admin	admin@example.com	2001. 01. 01. 0:00:00
moderator	moderator@example.com	2001. 01. 01. 0:00:00

2.23. ábra. Felhasználói fiókok listanézetének felülete

### 2.6.3. Felhasználói fiókok módosítása

A módosítás űrlapja hasonló, a 2.4.4 bekezdésben tárgyaltakkal, viszont itt lehetőség van egy adott felhasználói fiók hozzáférési szintjét állítani egy lenyíló lista segítségével. Biztonsági okok miatt a saját felhasználó jogosultsági köre nem módosítható, az erre való lista nem jelenik meg az űrlapon.

The screenshot shows a user profile editing form titled "user nevű felhasználó szerkesztése". It contains fields for "Felhasználónév" (user) and "E-mail cím" (user@example.com). Below these are dropdown menus for "Felhasználó típusa" (User type) and "Fájl kiválasztása" (File selection), both currently set to "Nincs fájl kiválasztva" (No file selected). At the bottom are "Vissza" (Back) and "Módosítás" (Edit) buttons.

2.24. ábra. Felhasználói fiókok szerkesztési űrlapja adminisztrátorként

#### 2.6.4. Felhasználói fiókok törlése

A törlés folyamata is megegyezik a 2.5.1 bekezdésben tárgyalt pontéval. Biztonsági okok miatt, a saját felhasználót törölni nem lehet, a listában is az erre mutató gombhivatkozás nem kattintható.

The screenshot shows a confirmation page titled "user nevű felhasználó végleges törlése". It asks "Biztosan törölni szeretné ezt a felhasználót?". At the bottom are "Vissza" (Back) and a red "Törölés" (Delete) button.

2.25. ábra. Felhasználói fiókok törlésének megerősítő űrlapja

### **3. fejezet**

## **Fejlesztői dokumentáció**

### **3.1. Megoldási terv**

#### **3.1.1. Követelményelemzés**

A weboldal használatakor négy típusú felhasználó különböztethető meg: vendég, bejelentkezett felhasználó, moderátor és adminisztrátor. Az előző fejezetben hasonlóan, itt is ezek alapján történik meg a funkcionális követelmények definiálása.

##### **Vendég**

- Térkép megtekintése:
  - Alapvető irányítási lehetőségek: mozgatás, forgatás, nagyítás/kicsinyítés
  - Az összes pont helyzetének megfelelő megjelenítése, fontosabb információk jelölésével
  - Pontra kattintáskor sorszám, kép és megjegyzés megtekintése, hivatkozás annak adatlapjára
- Pont részletezése:
  - minden információ megjelenítése
  - közvetlen hivatkozások a feltöltő profiljára, a térképen való megtekintésre és az adatlap megtekintésére
- Pontok listázása:
  - Pontok közötti szűrés, lapméret állítása
  - Fontosabb információk megjelenítése

- Közvetlen hivatkozások a feltöltő profiljára, a térképen való megtekintésre és az adatlap megtekintésére
- Színséma választása:
  - Sötét és világos mód közötti váltás bárhol
- Regisztráció:
  - Felhasználónév megadása
  - E-mail cím megadása
  - Jelszó megadása és megerősítése
  - Profilkép megadása
  - Regisztráció gombra nyomva:
    - \* sikér esetén: automatikus bejelentkeztetés az új fiókba
    - \* sikertelenség esetén (hiányzik egy kötelező adat, vagy felhasználónév/e-mail cím már foglalt): hibaüzenet megjelenítése a sikertelenség okával feltüntetve
- Bejelentkezés:
  - Felhasználónév megadása
  - Jelszó megadása
  - Bejelentkezés gombra nyomva:
    - \* sikér esetén: bejelentkeztetés a fiókba
    - \* sikertelenség esetén (hiányzik egy adat, vagy felhasználó-jelszó páros nem megfelelő): hibaüzenet megjelenítése a sikertelenség okát NEM feltüntetve

### **Bejelentkezett felhasználó**

- minden, amit egy vendég is tud (értelemszerűen a regisztráció és bejelentkezés kivételével)
- Kijelentkezés:
  - Kijelentkezés gombra kattintva kijelentkezteti a felhasználót és elnavigálja a kezdőoldalra
- Új pont bejelentése:
  - Koordináták (kötelező) megadása, saját koordináták automatikus meghatározása
  - Település és -rész megadása
  - Méret megadása

- Hozzáférhetőség megadása
- Szemét típusának megadása
- Opcionális megjegyzés megadása
- Képek hozzáadása
- Bejelentés gombra nyomva:
  - \* sikeres esetén: pont létrehozása és elnavigálás az előzőleg látogatott aloldalra
  - \* sikertelenség esetén (hiányzó koordináták, érvénytelen adatok megadása): hibaüzenet megjelenítése a sikertelenség okáról feltüntetve
- Pont szerkesztése:
  - minden bejelentéskor megadott adat frissítése
  - Új állapot megadása
  - Módosítás gombra nyomva: hasonló funkcionálitás a bejelentésekkel kapcsolatos Bejelentés gomb esetén
- Saját profil megtekintése:
  - Profilkép megjelenítése
  - Felhasználónév megjelenítése
  - E-mail cím megjelenítése
  - Hivatkozás saját profil szerkesztésére
  - Saját bejelentett pontok listázása (hasonló módon az összes pont listázásához)
- Saját profil szerkesztése:
  - minden regisztráláskor megadott adat frissítése
  - Módosítás gombra nyomáskor:
    - \* sikeres esetén: felhasználó adatainak módosítása és elnavigálás az előzőleg látogatott aloldalra
    - \* sikertelenség esetén (foglalt felhasználónév vagy e-mail cím, érvénytelen kép megadása): hibaüzenet megjelenítése a sikertelenség okáról feltüntetve

## **Moderátor**

- minden, amit egy bejelentkezett felhasználó (ezzel együtt vendég) is tud
- Pont törlése
  - Megerősítő üzenet a törlésről, igen/nem válasszal:

- \* igen esetén: pont végleges törlése, ezzel többé nem megjelenítve a térképen, sem a listanézeten
- \* nem esetén: visszanavigálás az előzőleg látogatott aloldalra

#### Adminisztrátor

- minden, amit egy moderátor is tud
- Felhasználói fiókok listázása:
  - Felhasználónév megjelenítése
  - E-mail cím megjelenítése
  - Regisztrálás idejének megjelenítése
  - Hivatkozások az adott fiók profiljára, módosítására és törlésére
- Felhasználói fiókok módosítása:
  - Saját profil szerkesztéskor megadható adatok megadása
  - Új szerepkör megadása
  - Módosítás gombra nyomáskor: a saját profil szerkesztéséhez hasonló viselkedés
- Felhasználói fiókok törlése:
  - Megerősítő üzenet a törlésről, igen/nem válasszal:
    - \* igen esetén: a fiók végleges törlése, ezzel többé elérhetetlené téve a fiókot, ha be van jelentkezve, akkor kijelentkeztetésre kerül azonnal
    - \* nem esetén: visszanavigálás az előzőleg látogatott aloldalra

#### 3.1.2. Rendszer-architektúra

A weboldal elsősorban C# nyelven, az ASP.NET Core keretrendszer használatával ún. MVC (Model-View-Controller, azaz Modell-Nézet-Vezérlő) architektúrában van megvalósítva. Másodsorban JavaScript nyelven is, egyes oldalak kliensoldali dinamikusságáért, túlnyomó részt a térképes megjelenítés esetében. Maguk, a statikus oldalak tartalmai C# segítségével összeállított HTML oldalakból állnak, melynek stílusbeli megjelenéséért a CSS stílusleíró nyelvben írt Bootstrap 5.3 keretrendszer felel.

A felhasználó egy webböngészőben létesít kapcsolatot az MVC architektúra vezérlőrétegével, mely segítségével a model-réteg bizonyos műveleteit meghívva az adatokban állapotváltozást ér el. Ezen változásokat a nézet-réteg közvetlen, vagy közvetve

Kellene pár  
szakirodal-  
mi hivatkozás  
azért ide.

egy nézetmodell (vagy DTO, azaz Data Transfer Object, magyarul adatátviteli objektum) segítségével nézetet (jelen esetben egy weboldalt) szolgáltat vissza.

Egy ilyen vezérlőhöz érkező kéréskor a program példányosítja a megfelelő vezérlőosztályt és az ahhoz szükséges egyéb objektumokat, majd a nézet elkészítése és visszaküldése után ezek megsemmisítésre kerülnek. Emellett előfordul, hogy nem egy nézet lesz a válasz, hanem egy akció eredmény, mely kiegészülhet egy eredménnyel.

## 3.2. Rétegek

### 3.2.1. Perzisztencia-réteg

Ennek a rétegnak, mint neve is sugallja, az adatok tárolása a célja, mely nem hagyja elveszni azokat a program leállítása esetén. Ehhez egy MSSQL-adatbázis áll a rendelkezésre, melyet az alkalmazás implementációjához használt Entity Framework Core (röviden EF) objektum-relációs leképzőrendszer hoz létre ún. migráció alkalmával, illetve létesít kapcsolatot a program és az adatbázis között. Ez a rendszer a programban definiált osztályokat felelteti meg az adatbázis tábláinak oly módon, hogy egy osztály tulajdonsága lesz a relációs tábla egy-egy oszlopa (vagy más néven attribútuma), illetve az osztály példányai (azaz objektumai) az adatbázis egy-egy sora (vagy más néven rekordja).

Angol kifejezéseket érdemes tipográfiaileg, pl. dőlttel szedve kiemelni.

Az adatbázis felépítése *code-first* megközelítéssel történik, ami azt jelenti, hogy a program C# nyelvű forráskódjából áll elő az adatbázis. A leképzőrendszer nem csak az osztályok tulajdonságait veszi figyelembe, hanem azoknak adattípusait, illetve egyedi megszorításait is megfelelteti MSSQL-oldalon. Emellett felismeri az osztályok tulajdonságainak neve alapján a táblák közötti relációkat is (egy az egyhez, egy a többhöz és több a többhöz kapcsolatokat).

Emellett ilyen kapcsolatok esetén lehetőség van ún. navigációs tulajdonságok deklarálására is, mely az idegen kulcs forráskód oldali használata helyett közvetlen a kapcsolatban álló objektumra mutat (melyek értelemszerűen nem jelennek meg az adatbázisban, ott továbbra is csak az idegen kulcs található meg), ezzel megkönnyítve annak fejlesztését. A legjobb teljesítmény érdekében ezek a navigációs tulajdonságok lista kiértékeléssel vannak ellátva, mely azt jelenti, hogy csak akkor lesz beletöltve valós érték, amikor arra szükség van és/vagy explicit kérjük azokat. Amíg ezek nem kerülnek betöltésre, addig ezek helyén egy helyettes (idegen szóval proxy) tervezési

minta által készített leszármaztatott proxy lesz.

Azt, hogy mely osztályok definiálnak egy-egy táblát, azt egy adatbázis-kontextus osztályban kell felsorolni, melyen keresztül lehetőség lesz az adatbázissal való kommunikációra is. Emellett nem csak a táblák felsorolása, hanem az azokra vonatkozó megszorítások is itt szerepel. A programkódban történő adatmódosítások perzisztálását külön eljárás meghívásával kell elérni, mely tulajdonképpen a követett módsításokat menti az adatbázisba.

Nem csak táblákat, hanem lekérdezéseket is lehet C# kód ról SQL-re fordítani, mivel a programnyelv támogatja az abba beágyazott lekérdezéseket (LINQ-et) is. Ilyen lekérdezések a kontextuson való meghívásával a rendszer automatikusan SQL-re fordítva hajtja végre azokat.

## **Adatbázis**

Az előző 3.2.1 alszakaszban tárgyaltak szerint az adatbázis a `TrashTracker.Data.Models` névterben található `TrashTrackerDbContext` adatbázis-kontextus osztály alapján definiált táblákból áll össze. Az ebben az osztályban `DbSet` generikus tulajdonságokkal felsorolt táblák az alábbiak:

- `Trashes` (típusa `DbSet<Trash>`): a bejelentett pontokat, illetve azok információját tartalmazza
- `TrashImages` (típusa `DbSet<TrashImage>`): a bejelentett pontokhoz tartozó (akár több) képeket tartalmazza, egy `Trashes` rekordhoz több `TrashImages` rekord tartozhat
- `UserImages` (típusa `DbSet<UserImage>`): a felhasználókhöz tartozó profilképet, illetve azokhoz társuló egyéb adatokat tartalmazza

Az adatbázis-kontextus osztály az ASP.NET Core `IdentityDbContext` osztályából származik le, melynek típusparaméterei a `TrashTrackerUser` (ez definiálja a felhasználók attribútumait), illetve a `TrashTrackerIdentityRole` (ez definiálja a szerepkörök adatait) osztályok, így további táblák kerülnek létrehozásra a felhasználó és szerepkörkezelés céljára:

- `AspNetUsers` tábla, mely az alapértelmezett tulajdonságok (pl. felhasználónév, e-mail cím és jelszó) mellett `TrashTrackerUser` osztály által definiált tulajdonságokat egybefoglalva tárolja az összes regisztrált felhasználó adatait. Egy az egyhez kapcsolatban áll a `UserImages` táblával, mivel egy felhasználónak egyszerre csak egy profilképe lehet.

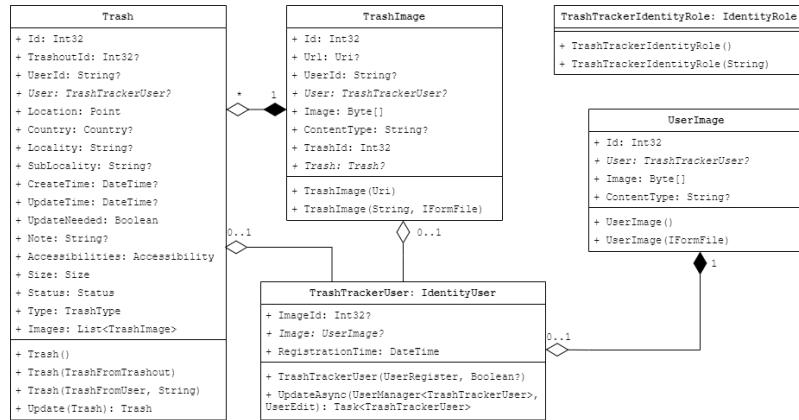
- **AspNetRoles** tábla az összes, jelen esetben felhasználó, moderátor és adminisztrátor szerepköröket tárolja.
- **AspNetUserRoles** tábla több a többhöz kapcsolatot alakít ki a felhasználók és a hozzájuk tartozó szerepkörök között, egy-egy felhasználó (-kulcs) és szerepkör (-kulcs) páros segítségével.
- **AspNetUserTokens** tábla, mely a bejelentkezés-kezelőnek biztosít perziszenciát a felhasználókhöz társított ún. tokenekkel, melyekkel bejelentkezés után a felhasználónak nem kell minden aloldal látogatás során újra és újra bejelentkeznie, hanem ezzel igazolja, hogy valóban ő az, aki bejelentkezett.

*Megjegyzés.* A fentebbi felsorolásban csak a program szempontjából releváns táblák lettek felsorolva. Az ASP.NET Core több táblát is definiál (pl. felhasználókhöz vagy szerepkörökhöz jogokhoz **AspNetUserClaims** és **AspNetRoleClaims** táblákat), viszont ezek nincsenek felhasználva a program működése során.

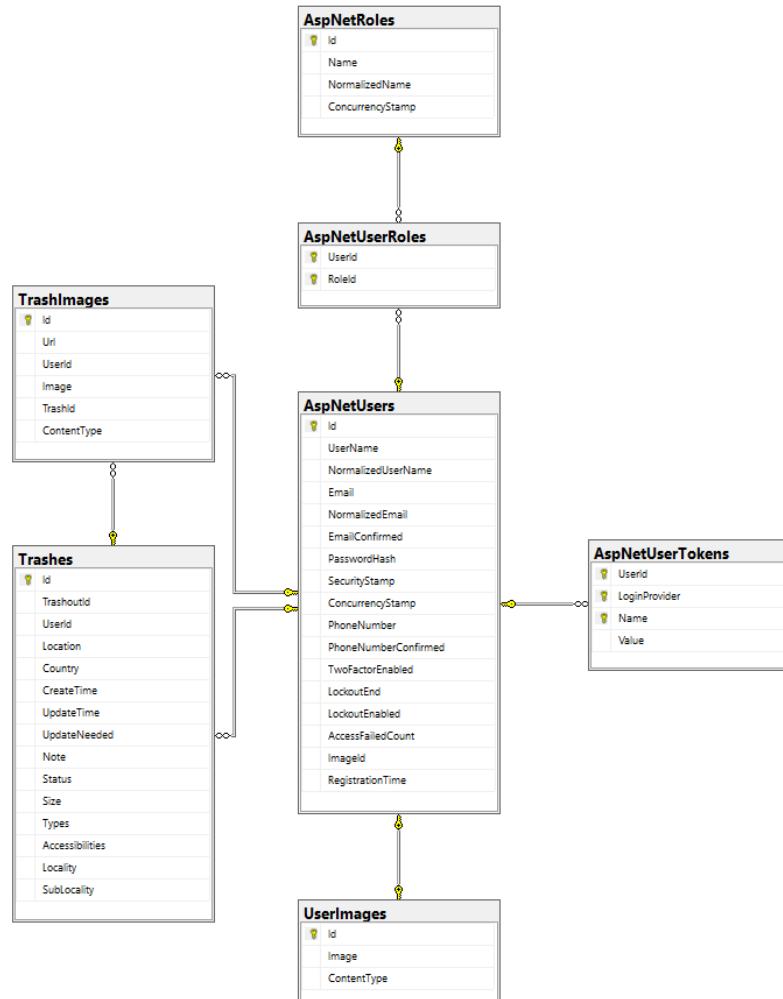
A szemétpontokon található címkeknek (pl. típus, méret) külön felsoroló adattípus került definiálásra, melyek biztosítják, hogy az adott attribútum csak bizonyos értékeket vehessen fel. Ezek között vannak olyan felsoroló típusok, melyek egyszerre több értéket is felvehet, a 2-es szám hatványainak tulajdonságát kihasználva. Ezek a felsorolások a **TrashTracker.Data.Models.Enums** névtérben találhatóak:

- **Accessibility**: a megközelíthetőségeket sorolja fel
- **Size**: a méretet tartalmazza
- **Status**: a mennyiség változását tartalmazza a bejelentés óta
- **TrashType**: a szemét típusait sorolja fel

### 3. Fejlesztői dokumentáció



(a) Az adatbázis osztályai forráskód oldalon



(b) Az adatbázis sémája SQL oldalon

3.1. ábra. Az adatbázis sémája leképzés előtt és után

Olyan méretű-  
ek legyenek a  
képek, hogy ha  
A4-es méretben  
nyomtatásra  
kerülne, olvas-  
ható legyen.

### 3.2.2. DTO-réteg

A `TrashTracker.Data.DTOs` névterben találhatóak az ún. DTO-k, magyarul adatátviteli osztályok. Ezeknek célja az adatbázisban található összes adat helyett csak az adott kontextusban szükséges adatokra való projektálás (pl. listanézet esetén nincs szükség a képekre, ezzel jelentős adatforgalmat megtakarítva). Emellett az egyes adatoknak itt vannak definiálva a validációs kritériumai és -hibaüzenetei (pl. a koordináták -180 és 180 között kell legyenek), ha azok bemeneti adatokként is szolgálnak, ellentétben nincs rá szükség, mivel feltehetjük, hogy a már adatbázisban szereplő adatok lekérdezéskor megfelelnek a megszorításoknak). Az adatátvitelhez használt objektumok az alábbiak:

`TrashTracker.Data.DTOs.In` (ki- és bemenő adatokhoz, validációkkal)

- `TokenFromGoogle`: a `TrashOut`-hoz való hozzáféréshez tartalmazza a tokent
- `TrashEdit`: egy bejelentett pont szerkeszthető tulajdonságait tartalmazza, melyből frissíthető egy `Trash` objektum
- `TrashFromTrashout`: egy `TrashOut`-ról letöltött pont adatait tartalmazza, melyből előállítható egy `Trash` objektum
- `TrashFromUser`: egy felhasználó által bejelentett pont adatait tartalmazza, melyből előállítható egy `Trash` objektum
- `UserEdit`: egy felhasználói profil szerkeszthető adatait tartalmazza, melyből frissíthető egy `TrashTrackerUser` objektum
- `UserLogin`: egy felhasználó bejelentkezéshez szükséges felhasználónév (vagy e-mail cím) és jelszó párosát tartalmazza
- `UserRegister`: egy felhasználó profil regisztráláshoz szükséges/megadható adatait tartalmazza, melyből előállítható egy `TrashTrackerUser` objektum

`TrashTracker.Data.DTOs.Out` (kizárolag kimenő adatokhoz, validációk nélkül)

- `TrashDetails`: egy bejelentett pont adatlapján szereplő adatokat (azaz a képeket is) tartalmazza, mely egy `Trash` objektumból áll elő
- `TrashIndex`: egy bejelentett pont listanézetén szereplő adatokat tartalmazza, mely egy `Trash` objektumból áll elő
- `TrashMap`: egy bejelentett pont térképen levő koordinátái és szűrési tulajdon-ságait tartalmazza, mely egy `Trash` objektumból áll elő
- `TrashMapDetails`: egy bejelentett pont térképen levő adatlapján szereplő ada-tokat tartalmazza, mely egy `Trash` objektumból áll elő

- **UserDetails**: egy felhasználói profil oldalán szereplő adatokat tartalmazza, mely egy **TrashTrackerUser** objektumból áll elő
- **UserIndex**: egy felhasználói profil listanézetén szereplő adatokat tartalmazza, mely egy **TrashTrackerUser** objektumból áll elő

### 3.2.3. Vezérlő-réteg

Ez a réteg felel minden beérkező kérés kiszolgálásáért, melyre több vezérlőosztály és a bennük található akció metódusok állnak rendelkezésre. Ezek az akciók általában a vezérlő és annak akciójából, illetve (ha van) paramétereiből összeállított "/kontroller/akcio?param1=ertek1&param2=ertek2" formájú útvonalon érhetőek el.

Minden akcióhoz tartozik egy annotáció, mely megmondja mely HTTP-kérésre fussenak le. Ha egy akciót csak bizonyos szerepkörrrel lehet elérni, akkor ahhoz is szerepel egy annotáció, mely megmondja melyik az a legalacsonyabb szerepkör, mellyel hozzáférhető az.

A kérés típusától független minden akció ad vissza valamilyen választ, mely lehet egy nézet, HTTP-állapotkód vagy egy JSON formátumú objektum, mely utóbbi a térképnek szolgáltatott adatokhoz szükséges.

Minden vezérlőt az ASP.NET Core keretrendszer példányosít, mely a HTTP-kérés beérkezésekor történik meg, mely példányon a megfelelő akciót futtatja le. A létrehozáshoz szükséges konstruktor paramétereit (pl. adatbázis-kontextus vagy felhasználó-kezelő) szintén ez a keretrendszer szolgáltatja ún. függőségi befecskedezéssel.

**HomeController**: A főoldal, azaz a térképen található adatok szolgáltatásáért felelős.

- **/OnMap**: az összes pont koordinátáit és szűrési tulajdonságait adja vissza **TrashMap** objektumokban
- **/OnMapDetails**: egy, a térképen való pontra kattintáskor megjelenő ablakban szereplő adatok adja vissza egy **TrashMapDetails** objektumban

**TrashesController**: Az összes ponttal kapcsolatos, nem térképen megjelenő adatok nézeteinek szolgáltatásáért felelős

- **/Trashes?pageNumber=1&pageSize=100&searchString=abc&showCleaned=false**: A pontok lapozható listanézetét adja vissza, melynek valamennyi paramétere

opcionális. A paraméterek rendre az adott oldalszám, az oldal méret (hány pont legyen egy oldalon), a keresési szöveg, illetve a megtisztított pontok listázásának logikai értéke. Mivel a paraméterek opcionálisak, így ezek tetszőlegesen kihagyhatóak.

- **/Trashes/Details/*id*:** Egy ponthoz tartozó adatlapnak a nézetét adja vissza, ahol az *id* egy kötelező paraméter, mely meghatározza hányas sorszámú pontnak az adatlapja kerül visszaadásra.
- **/Trashes/Create:** GET kéréskor egy pont létrehozásához szükséges űrlap nézetét, POST kéréskor, benne az űrlap adataival pedig siker esetén egy átirányítást az előző oldalra, ellenben a GET kéréskor kapott űrlap nézetét a már kitöltött adatokat és hibaüzeneteit adja vissza.
- **/Trashes/Edit/*id*:** GET kéréskor egy pont módosításához szükséges űrlap nézetét, POST kéréskor, benne az űrlap adataival pedig siker esetén egy átirányítást az előző oldalra, ellenben a GET kéréskor kapott űrlap nézetét a már kitöltött adatokat és hibaüzeneteit adja vissza.
- **/Trashes/Delete/*id*:** GET kéréskor egy pont törléséhez tartozó megerősítő kérdés nézetét adja vissza, POST kéréskor pedig siker esetén átirányítást az előző oldalra.

**UserController:** Az összes felhasználói fiókkal kapcsolatos, felhasználó-, bejelentkezés- és szerepkör-kezelésért felelős.

- **/User?pageNumber=1&pageSize=100&searchString=abc:** A felhasználói fiókok lapozható listanézetét adja vissza, melynek valamennyi paramétere opcionális. A paraméterek rendre az adott oldalszám, az oldal méret (hány fiók legyen egy oldalon), illetve a keresési szöveg. Mivel a paraméterek opcionálisak, így ezek tetszőlegesen kihagyhatóak.
- **/User/Details?userName=felhasznalo:** Egy felhasználói fiókhoz tartozó profiloldalnak a nézetét adja vissza, ahol az *userName* egy kötelező paraméter, mely meghatározza kinek a profilja kerül visszaadásra.
- **/User/Register:** GET kéréskor a fiók létrehozásához szükséges regisztrációs űrlap nézetét, POST kéréskor, benne az űrlap adataival pedig siker esetén egy átirányítást az főoldalra (illetve az új fiókba bejelentkezve), ellenben a GET kéréskor kapott űrlap nézetét a már kitöltött adatokat és hibaüzeneteit adja vissza.
- **/User/Login:** GET kéréskor a fiókba bejelentkezéshez szükséges űrlap nézetét,

POST kéréskor, benne az űrlap adataival pedig siker esetén egy átirányítást az főoldalra (illetve a fiókba bejelentkezve), ellenben a GET kéréskor kapott űrlap nézetét a már kitöltött adatokat és hibaüzeneteit adja vissza.

- `/User/Delete/id`: GET kéréskor felhasználói fiók törléséhez tartozó megerősítő kérdés nézetét adja vissza, POST kéréskor pedig siker esetén átirányítást az előző oldalra.

#### 3.2.4. Nézet-réteg

Ez a réteg felel az egyes akciók által visszaadott nézetek definiálásáért. A `TrashTracker.Web.Views` névterben található osztályok nem a C# hagyományos szintaxisával, hanem a Razor-ével íródtak, mely tulajdonképpen egy `.html` formátumú fájl, benne C# kód részletekkel, ebből is adódik ezen fájlok kiterjesztésének a `.cshtml` neve.

controller vagy vezérlő, de ez a "kontroller" inkább ne legyen

Minden nézet kontrollerenként van csoportosítva, és minden nézetet visszaadó akciót pontosan egy `.cshtml` formátumú nézetfájlja van. Ezek mellett a vannak ún. parciális nézetek is, melyek az egyes oldalakon szereplő, több helyen is ismétlődő tartalmat emelik ki külön fájlba, ezzel elkerülve az önmétlést. Ezen parciális nézetek fájljainak nevei az `_` karakterrel kezdődnek, melyeknek többsége a Shared mappában található (pl. navigációs sáv, illetve az abban lévő fiók menüje).

##### Home

- **Index:** A főoldalon található térkép, illetve az ahhoz tartozó szűrési menüt tartalmazó nézet. A térkép viselkedése nem itt, hanem a `.../wwwroot/js/home.js` fájlban van definiálva, mivel a térképes megjelenítés dinamikája nem szerver-, hanem kilensoldalon történik a JavaScript nyelven íródott OpenLayers könyvtárcsomag segítségével.

##### Trashes

- **Index:** A bejelentett pontok listanézetét megjelenítő nézet. Magát a listát az ebből az osztályból meghívott `_IndexList` parciális nézet tartalmazza, ez csupán a szűrési menüt definiálja.
  - `_IndexList:` A bejelentett pontok listanézetének a listáját tartalmazó parciális nézet, melynek tetején és alján található lapszámozás külön, `_Pagination` nevű parciális nézetben kap helyet.
- **Details:** Egy bejelentett pont adatlapját megjelenítő nézet.

- **Create:** Egy pont bejelentésének űrlapját megjelenítő nézet.
- **Edit:** Egy bejelentett pont módosításának űrlapját megjelenítő nézet.
- **Delete:** Egy bejelentett pont törlését megerősítő üzenetét megjelenítő nézet.

#### User

- **Index:** A felhasználói fiókok listanézetét megjelenítő nézet. Magát a listát az ebből az osztályból meghívott `_IndexList` parciális nézet tartalmazza.
  - `_IndexList:` A felhasználói fiókok listanézetének a listáját tartalmazó parciális nézet, melynek tetején és alján található lapszámozás külön, `_Pagination` nevű parciális nézetben kap helyet.
- **Details:** Egy felhasználói fiók profiloldalát megjelenítő nézet, mely tartalmazza a `Trashes/_IndexList` parciális nézetet, mely az adott fiók által bejelentett pontokat listázza.
- **Login:** A bejelentkezési űrlapot megjelenítő nézet.
- **Register:** A regisztrálási űrlapot megjelenítő nézet.
- **Edit:** A profil szerkesztésének űrlapját megjelenítő nézet. Adminisztrátori fiókkal kiegészül az adott felhasználó szerepkörének módosításával.
- **Delete:** Egy felhasználói fiók törlését megerősítő üzenetét megjelenítő nézet.

### 3.2.5. Kliensoldal

Vannak olyan aloldalak (túlnyomó részben a főoldal/térkép), melyhez külön JavaScript nyelven írt kliensoldali programkódra is szükség van. Ezek a forráskódok a `TrashTracker.Web` projekt `wwwroot/js` könyvtárában találhatóak.

A `Home/Index` nézethez készült és hivatkozott `home.js` állomány tartalmazza magának a térkép objektumának a definícióját, illetve a szűrési menükben található gombok megnyomása esetén azt a függvényt, mely gombok állapota szerint változnak a térképen található pontok, melyek megfelelnek a feltételeknek. Emellett a pontra való kattintáskor megjelenő modális buborék is megjelenítésére szolgáló függvény is itt kap helyet.

A `Trashes/Index` nézet tetején található szűrési beállításokhoz készült a `index.js` segédprogram, mely segít az oldalra navigáláskor az adott szűrési feltételeket a hozzá megfelelő mezőbe beilleszteni.

A `Trashes/Create` és `Trashes/Edit` nézethez készült `position.js` fájl tartalmazza a "Jelenlegi hely meghatározása" gombra kattintáskor történő helymeghatározást,

illetve annak megfelelő mezőibe való beillesztést.

Emellett a színséma alkalmazásáért és aloldalakon történő navigálások esetén való perzisztálásáért felel a `site.js` fájl, mely a közös, navigációs sávot is tartalmazó `Shared/_Layout` nézetben van beszúrva, ezzel minden oldalon betöltésre kerül.

### 3.3. Megvalósítás

A webalkalmazás forráskódja túlnyomó részben C# nyelven, az ASP.NET Core keretrendszer segítségével lett implementálva. Ennek oka, hogy a keretrendszer támogatja az MVC architektúrát, amely tökéletesen megfelel a webes környezetnek. Az előbb említett keretrendszer mellett az Entity Framework objektum-relációs leképzőrendszer is nagy segítség, mind a fejlesztést, illetve futtatást is hatékonyabbá téve. Ezen technológiák futtatására csak a .NET Core Runtime-ra van szükség, mely elérhető Windows, Linux és macOS rendszereken egyaránt.

A forráskód rétegekre bontása jelentősen elősegíti a jövőbeli továbbfejleszthetőségét a programnak, pl. egy perzisztencia- és DTO-rétegre épülő webszolgáltatással (Rest API-val) lehetőség van akár egy asztali alkalmazás fejlesztésére is, az előbb említett rétegek minimális módosításával, illetve azok redundanciája nélkül. Emellett a már meglévő funkciók is úgy lettek implementálva, hogy lehetőség szerint azokhoz a legkevesebb refaktorálásra legyen szükség (pl. új attribútumok-, új szerepkörök hozzáadása).

#### 3.3.1. Projektek szervezése

A megvalósított forráskód két projektben kapott helyet.

Az első a `TrashTracker.Data`, mely tartalmazza a perzisztencia- és DTO-rétegeket, ezzel egy adatszolgáltatási projektként leszeparálva. Ennek oka, hogy az előző 3.3 szakaszban említett továbbfejlesztési lehetőség esetén ez a projekt akár módosítás nélkül is képes feladatának az ellátására.

A második a `TrashTracker.Web`, mely tartalmazza magát a webalkalmazást, az előzőleg említett projekten keresztül kinyert adatok kontrollerekkel való továbbítását a nézeteknek. Ennek oka, hogy az előbb említett továbbfejlesztés esetén ezek a kódrészek nem relevánsak, illetve ehhez a projekthez akár fejleszthető saját adatszolgáltatási projekt, mely kompatibilis ezzel az alkalmazással.

## Adatszolgáltatás

### Webalkalmazás

Az alkalmazás belépési pontja a projekt gyökérkönyvtárában található `Program` osztály, mely felkonfigurálja, majd elindítja azt. A különféle szolgáltatásokat egy `WebApplicationBuilder` objektumba gyűjti össze, melybe először az adatbázis kontextusát regisztrálja annak *connection string*-jével. Itt van megadva a földrajzi adatok használata, illetve a lusta betöltés proxyjainak bekapcsolása.

Ezt követően felkonfigurálja a használt felhasználó- és szerepkörtípusokat, mely a rendre a `TrashTrackerUser` és `TrashTrackerIdentityRole` típusok mellett kikötéseket is tartalmaz egy-egy felhasználó adataira (pl. hány és milyen karakterekből álljon egy felhasználónév vagy jelszó).

Ezt követően a `TrashTracker.Web.Utils` névterben egyedileg készített `PolicyBuilder` statikus osztály a webalkalmazást építő objektumba legenerálja a szerepköröket, illetve azok hierarchiáját, mely szerepkörök a `TrashTracker.Data.Models.Defaults` névterben definiált `Roles` felsoroló adattípusban találhatóak, azok sorrendje adja az "alá-/fölérendeltséget". Ezek a szerepkörök szabadon bővíthetők az előbb említett felsoroló módosításával.

Ezután történik a vezérlők és az azokhoz tartozó nézetek regisztrációja, illetve a `TrashOut`-tal kapcsolatot definiáló `TrashoutService` és az azt időszakosan meghívó `TrashoutTimedHostedService` szolgáltatás.

Az "összeszedett" szolgáltatások itt *build*-elésre kerülnek, mely a `WebApplicationBuilder`-ból egy `WebApplication` objektumot hoz létre. Itt történik meg a `TrashOut`-tal való kapcsolat létrehozása, a HTTP kérések HTTPS-re való átirányításának-, a végpontok *routing*-jának-, az autentikációknak és autorizációknak a bekapcsolása, illetve a főoldal útvonalának megadása. A forráskód fájljának végén kerül az adatbázis inicializálásra, mely "éles" környezetben az adatbázis migrációját, teszt környezetben pedig annak alapértelmezett- és minta adatokkal való ellátását biztosítja. Emellett a már előbb említett szerepkörök létrehozásra/frissítésre kerülnek, illetve minden szerepkörhöz (melyhez addig nem volt) egy alapértelmezett fiók is regisztrálásra kerül.

A `Program` osztály lefutását követően a legutolsó sorban található eljárás blokkolja az azt végrehajtó szálat (a leállításig), mellyel ténylegesen elindítja a webalkalmazást.

## **3.4. Tesztelés**

A vezérlő-rétegben található végpontok egy, a `TrashTracker.Web.Test` nevű XUnit projektben található automatikus egységesztekkel lett tesztelve. Ezek az egységesztek a legtöbb végpont bemeneti és kimeneti specifikációja szerint teszteli azok helyes viselkedését, illetve a perzisztenciában bekövetkezett változások helyességét.

A nézet-rétegben található nézetek manuálisan, a 2 fejezetben szereplő felhasználói eseteket figyelembe véve lettek tesztelve, mely folyamata a 3.4.2 alszakaszban van naplózva.

### **3.4.1. Vezérlő-réteg automatikus tesztesetei**

### **3.4.2. Nézet-réteg manuális tesztelése**

## **3.5. Fejlesztési lehetőségek**

## 4. fejezet

### Összegzés

Lorem ipsum dolor sit amet, consectetur adipiscing elit. In eu egestas mauris. Quisque nisl elit, varius in erat eu, dictum commodo lorem. Sed commodo libero et sem laoreet consectetur. Fusce ligula arcu, vestibulum et sodales vel, venenatis at velit. Aliquam erat volutpat. Proin condimentum accumsan velit id hendrerit. Cras egestas arcu quis felis placerat, ut sodales velit malesuada. Maecenas et turpis eu turpis placerat euismod. Maecenas a urna viverra, scelerisque nibh ut, malesuada ex.

Aliquam suscipit dignissim tempor. Praesent tortor libero, feugiat et tellus porttitor, malesuada eleifend felis. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Nullam eleifend imperdiet lorem, sit amet imperdiet metus pellentesque vitae. Donec nec ligula urna. Aliquam bibendum tempor diam, sed lacinia eros dapibus id. Donec sed vehicula turpis. Aliquam hendrerit sed nulla vitae convallis. Etiam libero quam, pharetra ac est nec, sodales placerat augue. Praesent eu consequat purus.

# Köszönetnyilvánítás

Amennyiben a szakdolgozati / diplomamunka projekted pénzügyi támogatást kapott egy projektből vagy az egyetemtől, jellemzően kötelező feltüntetni a dolgozatban is. A dolgozat elkészítéséhez segítséget nyújtó oktatók, hallgatótársak, kollégák felé is nyilvánítható külön köszönet.

## A. függelék

### Szimulációs eredmények

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque facilisis in nibh auctor molestie. Donec porta tortor mauris. Cras in lacus in purus ultricies blandit. Proin dolor erat, pulvinar posuere orci ac, eleifend ultrices libero. Donec elementum et elit a ullamcorper. Nunc tincidunt, lorem et consectetur tincidunt, ante sapien scelerisque neque, eu bibendum felis augue non est. Maecenas nibh arcu, ultrices et libero id, egestas tempus mauris. Etiam iaculis dui nec augue venenatis, fermentum posuere justo congue. Nullam sit amet porttitor sem, at porttitor augue. Proin bibendum justo at ornare efficitur. Donec tempor turpis ligula, vitae viverra felis finibus eu. Curabitur sed libero ac urna condimentum gravida. Donec tincidunt neque sit amet neque luctus auctor vel eget tortor. Integer dignissim, urna ut lobortis volutpat, justo nunc convallis diam, sit amet vulputate erat eros eu velit. Mauris porttitor dictum ante, commodo facilisis ex suscipit sed.

Sed egestas dapibus nisl, vitae fringilla justo. Donec eget condimentum lectus, molestie mattis nunc. Nulla ac faucibus dui. Nullam a congue erat. Ut accumsan sed sapien quis porttitor. Ut pellentesque, est ac posuere pulvinar, tortor mauris fermentum nulla, sit amet fringilla sapien sapien quis velit. Integer accumsan placerat lorem, eu aliquam urna consectetur eget. In ligula orci, dignissim sed consequat ac, porta at metus. Phasellus ipsum tellus, molestie ut lacus tempus, rutrum convallis elit. Suspendisse arcu orci, luctus vitae ultricies quis, bibendum sed elit. Vivamus at sem maximus leo placerat gravida semper vel mi. Etiam hendrerit sed massa ut lacinia. Morbi varius libero odio, sit amet auctor nunc interdum sit amet.

Aenean non mauris accumsan, rutrum nisi non, porttitor enim. Maecenas vel tortor ex. Proin vulputate tellus luctus egestas fermentum. In nec lobortis risus,

sit amet tincidunt purus. Nam id turpis venenatis, vehicula nisl sed, ultricies nibh. Suspendisse in libero nec nisi tempor vestibulum. Integer eu dui congue enim venenatis lobortis. Donec sed elementum nunc. Nulla facilisi. Maecenas cursus id lorem et finibus. Sed fermentum molestie erat, nec tempor lorem facilisis cursus. In vel nulla id orci fringilla facilisis. Cras non bibendum odio, ac vestibulum ex. Donec turpis urna, tincidunt ut mi eu, finibus facilisis lorem. Praesent posuere nisl nec dui accumsan, sed interdum odio malesuada.

# Ábrák jegyzéke

2.1.	Felhasználói esetek vendégként . . . . .	6
2.2.	A felület kinézete a kezdőoldalon különböző méretű kijelzők esetén . . . . .	7
2.3.	A kezdőoldal alján beúszó szűrési menü . . . . .	8
2.4.	Példa egy felbukkanó buborékra egy ponton . . . . .	9
2.5.	Példa egy szemétpontot részletező adatlapra . . . . .	9
2.6.	Szemétpontok listanézetének felülete . . . . .	10
2.7.	Szemétpontok listanézetének felülete világos módban . . . . .	11
2.8.	A regisztrációs felület űrlapja . . . . .	12
2.9.	Már foglalt felhasználónév esetén megjelenő hibaüzenet . . . . .	12
2.10.	Érvénytelen e-mail cím esetén megjelenő hibaüzenet . . . . .	13
2.11.	Érvénytelen jelszó esetén megjelenő hibaüzenetek . . . . .	14
2.12.	A bejelentkezéses felület űrlapja . . . . .	14
2.13.	A bejelentkezéses felület űrlapján szereplő hibaüzenet . . . . .	15
2.14.	Felhasználói esetek (bejelentkezett) felhasználóként . . . . .	15
2.15.	Szemétpont bejelentésének űrlapja . . . . .	16
2.16.	Szemétpont módosításának űrlapja . . . . .	17
2.17.	Példa egy felhasználói profilra . . . . .	18
2.18.	Profil szerkesztése és annak eredménye . . . . .	18
2.19.	Profil szerkesztésének űrlapja . . . . .	19
2.20.	Felhasználói esetek moderátorként . . . . .	19
2.21.	Pont törlésének megerősítő űrlapja . . . . .	20
2.22.	Felhasználói esetek adminisztrátorként . . . . .	20
2.23.	Felhasználói fiókok listanézetének felülete . . . . .	21
2.24.	Felhasználói fiókok szerkesztési űrlapja adminisztrátorként . . . . .	22
2.25.	Felhasználói fiókok törlésének megerősítő űrlapja . . . . .	22
3.1.	Az adatbázis sémája leképzés előtt és után . . . . .	30

# Táblázatok jegyzéke

# Algoritmusjegyzék

# Forráskódjegyzék