



**SAPIENTIA**  
**ERDÉLYI MAGYAR**  
**TUDOMÁNYEGYETEM**  
**Marosvásárhelyi Kar**



**FireForce**

Android alkalmazás

**Fejlesztők:**

Balázs Bence-Adorján  
Marosi Norbert  
Péter Ákos

**Vezető tanár:**

Ferencz Katalin  
Szántó Zoltán

# Tartalomjegyzék

<b>BEVEZETŐ.....</b>	<b>4</b>
<b>A PROJEKT CÉLJA .....</b>	<b>4</b>
<i>Gyors Reagálás .....</i>	<i>4</i>
<i>Közösségi Biztonság.....</i>	<i>4</i>
<i>Önkéntesség Értéke .....</i>	<i>4</i>
<b>PROJEKT ÉRTÉKEI ÉS FŐBB TÉNYEZŐI.....</b>	<b>4</b>
<i>Gyors Reagálás .....</i>	<i>4</i>
<i>Közösségi Szerepvállalás.....</i>	<i>4</i>
<i>Technológia és Innováció.....</i>	<i>4</i>
<b>KÖVETELMÉNY SPECIFIKÁCIÓ .....</b>	<b>5</b>
FELHASZNÁLÓ KÖVETELMÉNYEK. USE-CASE DIAGRAM. ....	5
<i>1.ábra: Use-Case diagram .....</i>	<i>5</i>
ALKALMAZÁS HASZNÁLATA (APP USAGE).....	6
<i>Regisztráció és Bejelentkezés.....</i>	<i>6</i>
<i>Egyesülethez Csatlakozás és Létrehozás .....</i>	<i>6</i>
<i>Tűzesetek Kezelése.....</i>	<i>6</i>
<i>Kommunikáció és Reagálás .....</i>	<i>6</i>
<i>Tűzeset Lezárása .....</i>	<i>6</i>
FELHASZNÁLÓI ALKALMAZÁS RENDSZERKÖVETELMÉNYEK .....	7
FUNKCIONÁLIS RENDSZERKÖVETELMÉNYEK .....	7
NEM FUNKCIONÁLIS RENDSZERKÖVETELMÉNYEK.....	7
<b>TERVEZÉS .....</b>	<b>8</b>
ARCHITEKTÚRA.....	8
<i>2.ábra: Architektúra diagram.....</i>	<i>8</i>
MODULOK LEÍRÁSA .....	9
<b>FELHASZNÁLÓI FELÜLET.....</b>	<b>10</b>
<i>3.ábra: Bejelentkezés felület.....</i>	<i>10</i>
<i>4.ábra: Regisztrációs felület .....</i>	<i>11</i>
<i>5.ábra: Térkép felület.....</i>	<i>12</i>
<i>6.ábra: Menü felület.....</i>	<i>13</i>
<i>7.ábra: Egyesülethez csatlakozás felület .....</i>	<i>13</i>
<i>8.ábra: Egyesület létrehozás felület.....</i>	<i>14</i>
<i>9.ábra: Beállítások felület.....</i>	<i>15</i>
<b>PROJEKTMENEDZSMENT .....</b>	<b>16</b>
<i>10.ábra: GitHub Commit felület .....</i>	<i>16</i>
<i>11.ábra: Projekt felépítés .....</i>	<i>17</i>
FELADATKEZELÉS - KANBAN TÁBLA .....	18
<i>12.ábra: Kanban tábla .....</i>	<i>18</i>
<b>ALKALMAZÁS TESZTELÉSE .....</b>	<b>19</b>
<i>13.ábra: Tesztelési folyamat.....</i>	<i>19</i>
<b>ALKALMAZÁS MŰKÖDÉSE .....</b>	<b>20</b>
FRONT END .....	20
<i>14.ábra: Figma WireFrame terv .....</i>	<i>21</i>

BACKEND .....	22
15.ábra: Backend kódrészlet.....	23
ADATBÁZIS .....	24
16.ábra: Firebase adatbázis regisztrációs tábla.....	24
ADATBÁZIS FÜGGVÉNYEK .....	25
17.ábra: Adatbázis függvénykódok .....	25
<b>TOVÁBBI FEJLESZTÉSI LEHETŐSÉGEK.....</b>	<b>26</b>
1. Riasztások és értesítések fejlesztése: .....	26
2. GPS és térképintegráció fejlesztése.....	26
3. Felhasználói szerepkörök bővítése .....	26
4. Jelentések és analitika .....	26
5. Offline mód támogatása.....	26
6. Felhasználói visszajelzések gyűjtése és integrálása .....	26
7. Bővített biztonsági funkciók.....	26
8. Videó- és hangrögzítés.....	26
9. Önkéntesek közötti kommunikáció.....	26
HIBÁK JAVÍTÁSA .....	27
BÓNUSZ LEHETŐSÉG .....	27
<b>BIBLIOGRÁFIA .....</b>	<b>27</b>

# Bevezető

Az alkalmazás célja, hogy összekapcsolja az önkéntes tűzoltókat és a tűzoltási helyzetekben érintett felek közötti kapcsolatot, erősítve ezzel a közösségi biztonságot és az összefogást a tűzvédelmi ügyekben.

## A Projekt Célja

A FireForce mobilalkalmazás azonnali értesítéseket és információkat nyújt az önkéntes tűzoltóknak a tüzesetekről, lehetővé téve számukra a gyors reagálást és beavatkozást. Az alkalmazás célja, hogy:

**Gyors Reagálás:** Az önkéntes tűzoltókat azonnal tájékoztatja a tűzveszélyes helyzetekről, így lehetővé téve számukra a gyors reagálást és segítségnyújtást.

**Közösségi Biztonság:** A FireForce elősegíti a közösségi biztonságot azáltal, hogy összeköti az embereket és lehetővé teszi számukra a hatékonyabb tűzvédelmi intézkedéseket.

**Önkéntesség Értéke:** Erősíti az önkéntes tűzoltók fontosságát a helyi közösségekben, kiemelve az ő áldozatkész munkájukat és elkötelezettségüket a tűzvédelemért.

## Projekt Értékei és Főbb Tényezői

### A FireForce mobilalkalmazás hozzáadott értékei:

**Gyors Reagálás:** Az alkalmazás valós időben tájékoztatja az önkéntes tűzoltókat a tüzesetekről, lehetővé téve a gyors reagálást és a tűzoltási munkálatok hatékonyabb koordinációját.

**Közösségi Szerepvállalás:** A FireForce elősegíti a közösségi szerepvállalást a tűzvédelmi intézkedésekben és a tüzesetek elkerülésében.

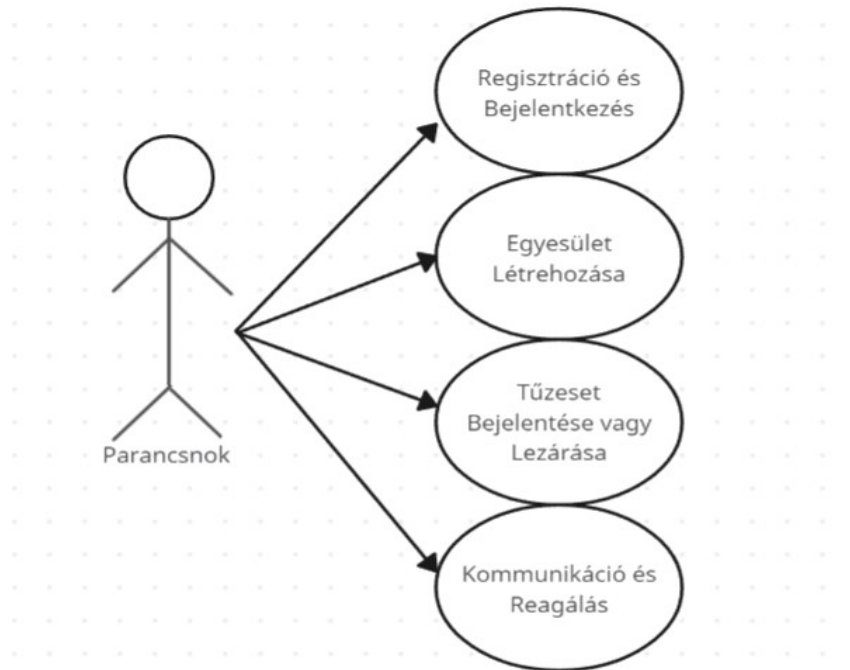
**Technológia és Innováció:** A modern technológia integrálása a tűzvédelmi folyamatokba, ami hatékonyabb és pontosabb értesítést tesz lehetővé a veszélyes helyzetekről.

# Követelmény specifikáció

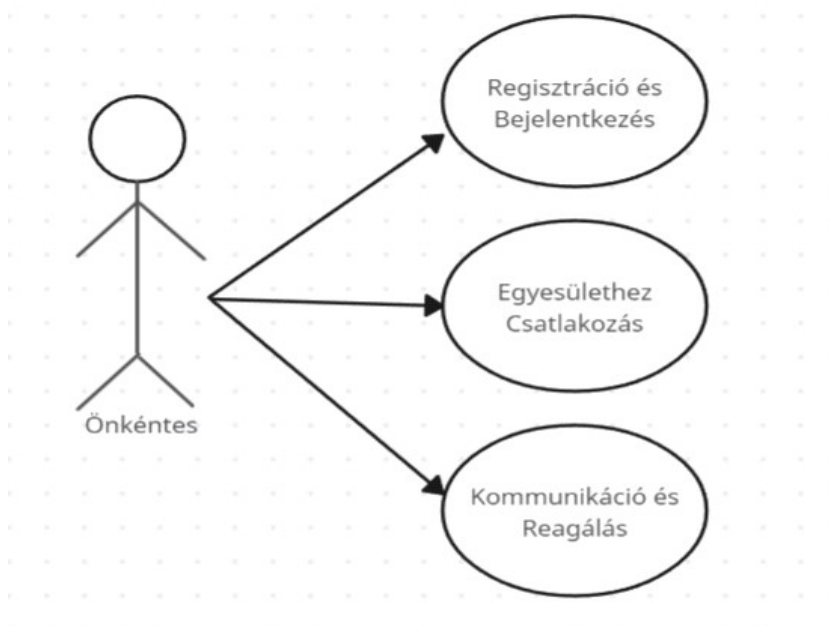
## Felhasználó követelmények. Use-Case diagram.

Az alábbi ábra a használati eset diagramot tartalmazza, amely a felhasználó és a rendszer közötti lehetséges interakciókat mutatja be. A diagram a mi esetünkben kitér a két különböző felhasználókra, ami ebben az a következők:

**parancsnok** (az egyesület létrehozója)



**önkéntes** (aki csatlakozott az egyesülethez)



1.ábra: Use-Case diagram

## **Alkalmazás Használata (App Usage)**

Az alkalmazás használata magában foglalja a felhasználók interakcióját az alkalmazással, valamint az alkalmazás által nyújtott funkciók és szolgáltatások használatát. A FireForce alkalmazás célja a felhasználók gyors és hatékony beavatkozásának lehetővé tétele tűzveszélyes helyzetekben. Az alkalmazás használatakor a felhasználók a következő lépéseket tehetik meg:

### **Regisztráció és Bejelentkezés**

Az alkalmazás lehetővé teszi a felhasználók számára, hogy regisztráljanak vagy bejelentkezzenek. Ez a funkció biztosítja a felhasználók számára az alkalmazásba való belépést és személyes fiókjaik kezelését. Viszont a be nem jelentkezett felhasználók, csak a térképet tudják megnézni, nem képesek egyesületet létrehozni, vagy csatlakozni egyesülethez.

### **Egyesülethez Csatlakozás és Létrehozás**

A regisztrált felhasználók csatlakozhatnak meglévő egyesületekhez, vagy létrehozhatnak saját egyesületeket. Az egyesületek létrehozója lesz az egyesület parancsnoka, és egyedi kódot kap, amellyel más felhasználók csatlakozhatnak az egyesülethez. Egy meglévő egyesülethez bárki csatlakozhat és bármennyien. Ha valaki már egy másik egyesület tagja vagy parancsnoka értelemszerűen nem csatlakozhat még egy egyesülethez.

### **Tűzesetek Kezelése**

Amikor egy tűzesetről értesülnek, az egyesület parancsnoka bejelöli az eseményt az alkalmazásba integrált térképen. Az alkalmazás értesítést küld az összes egyesületi tagnak az aktuális eseményről.

### **Kommunikáció és Reagálás**

Az értesítés után az egyesületi tagok válaszolhatnak az eseményre. Meghatározhatják, hogy részt vesznek-e a tűzesetnél vagy sem. Ez lehetővé teszi a hatékony önkéntesek szervezését és irányítását.

### **Tűzeset Lezárása**

Amikor a tűzeset sikeresen megoldódik, az egyesület parancsnoka lezárja az esetet az alkalmazásban, jelezve, hogy a tűz eloltásra került.

## Felhasználói alkalmazás Rendszerkövetelmények

A FireForce alkalmazás használatához a felhasználónak elsősorban rendelkeznie kell egy okostelefonnal.

Mobil eszközön ajánlott: Android 7.0-nál újabb verzió a Google Maps miatt.

## Funkcionális rendszerkövetelmények

A regisztrált fiókokat két részre oszthatjuk, aki saját egyesületet hozott létre (**parancsnok**) és aki csatlakozott egy már meglevő egyesülethez (**tűzoltó**). A saját felületén mindkét felhasználó képes:

- Regisztráció.
- Bejelentkezés.
- Navigáció az oldalak között.
- Kijelentkezés.
- Egyedi profilkép állítás.
- Egyedi riasztási hang állítás.

Az **Parancsnok** felhasználó képes:

- Tűzesetet bejelenteni.
- Tűzesetet lezárni.

A **Manager** felhasználó képes:

- Reagálni a tűzesetre, hogy kivonul vagy sem.

## Nem funkcionális rendszerkövetelmények

- Felhasználóbarát kezelhetőség,
- Rugalmas, reszponzív felhasználói felület,
- Teljes hozzáférés a Google Maps-hoz,
- Gyorsaság (hatékony algoritmusok és kiváló architektúra),
- Megbízhatóság.

# Tervezés

## Architektúra

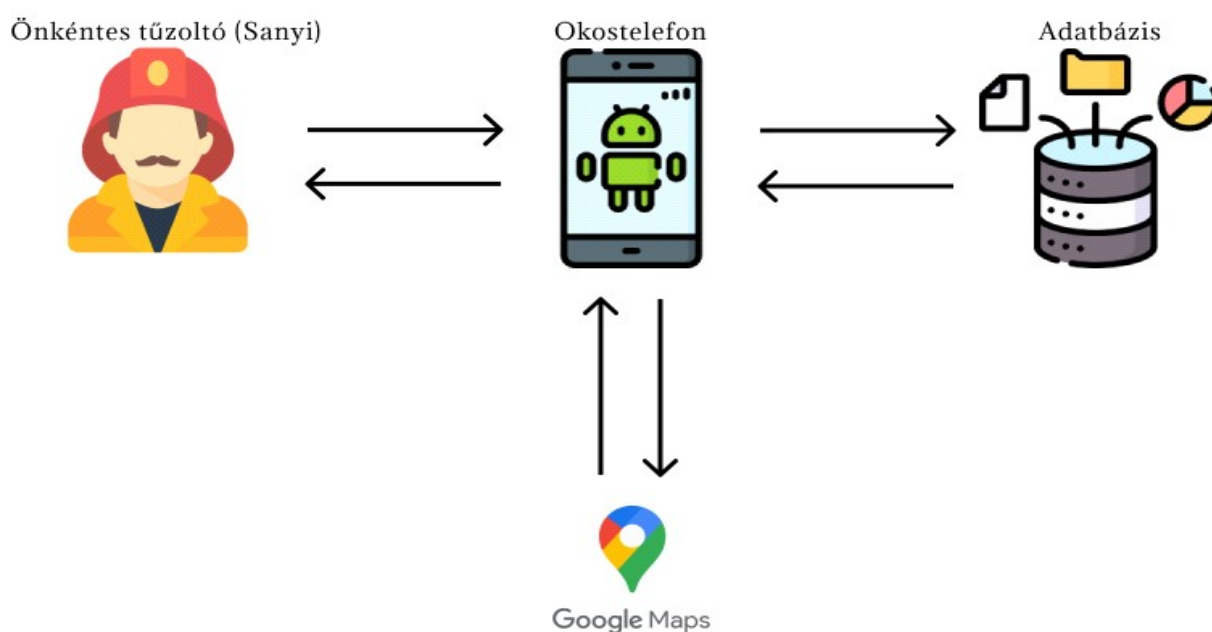
Az elsődleges szempont, amikor szoftverünk architektúráját tervezzük, az a gondos megfontoltság. A tervezés során figyelembe vettük az architektúra letisztultságát és egyszerűségét, ez által könnyen megérthetővé és karbantarthatóvá válik.

Minél egyszerűbb és megfontoltabb az architektúra, annál könnyebben lehet a szoftvert bővíteni és új funkciókat implementálni. Az egyszerűség és a logikus felépítés segít abban, hogy könnyen eligazodjunk a kódban, megértsük annak működését, és hatékonyan végezhessünk fejlesztéseket. Az egyszerűség nemcsak a jelenlegi feladatok elvégzését teszi könnyebbé, hanem hosszú távon is fenntarthatóbbá teszi a szoftvert.

A jól tervezett architektúra nem csak a fejlesztők munkáját könnyíti meg, de a karbantartás során is előnyöket kínál. Az egységes, átlátható struktúra lehetővé teszi, hogy gyorsan azonosítsuk és javítsuk a hibákat, valamint könnyen alkalmazzuk a frissítéseket és változtatásokat. Ez az előre gondolkodott és egyszerűsége törekvő architektúra alapot képezhet egy hatékony, könnyen kezelhető és jól skálázható szoftverfejlesztéshez.

Ebben az esetben a projektet leegyszerűsítettük 1 részre, ahol az okostelefon van a középpontban és ehhez kapcsolódik a **Felhasználó**, a **Térkép** és az **Adatbázis** is. Az architektúránk a következő módon valósult meg:

## FireForce Architektúra



2.ábra: Architektúra diagram



## Modulok leírása

A fentebb tárgyaltak alapján a szoftverünket 4 fő alkotóelemre lehet osztani:

- **Felhasználó** (*User*)
- **Okos telefon** (*Smart Phone*)
- **Térkép** (*Google Maps*)
- **Adatbázis** (*Database*)

A **felhasználó** a szoftverünk végső felhasználója, aki interakcióba tud lépni a rendszerünkkel. A mi esetünkben 2 féle felhasználó típust tudunk megkülönböztetni:

- **Parancsnok:** Ez a felhasználó létrehozott már egy egyesületet, tud tüzesetet bejelenteni vagy lezárni, a csapatában levő önkétesekkel kapcsolatba tud lépni.
- **Önkéntes Tűzoltó:** Ez a felhasználó, aki megkapja a Parancsnok értesítését és el tudja fogadni vagy visszautasítani a megbízást.

Az **okos telefon** feladata a UI-megjelenítése és a többi komponens összekötése, mivel mindhárom elem elengedhetetlen a program megfelelő működéséhez, így fontos egy Android 7.0 vagy egy frissebb Androidos verzió.

A **térkép** az a jól ismert Google Maps implementálását jelenti a saját programunkba. Megfelelő választás volt számunkra, mivel egy ingyenesen hozzáadható API-ról beszélünk, az árához képest könnyen használható és megbízható.

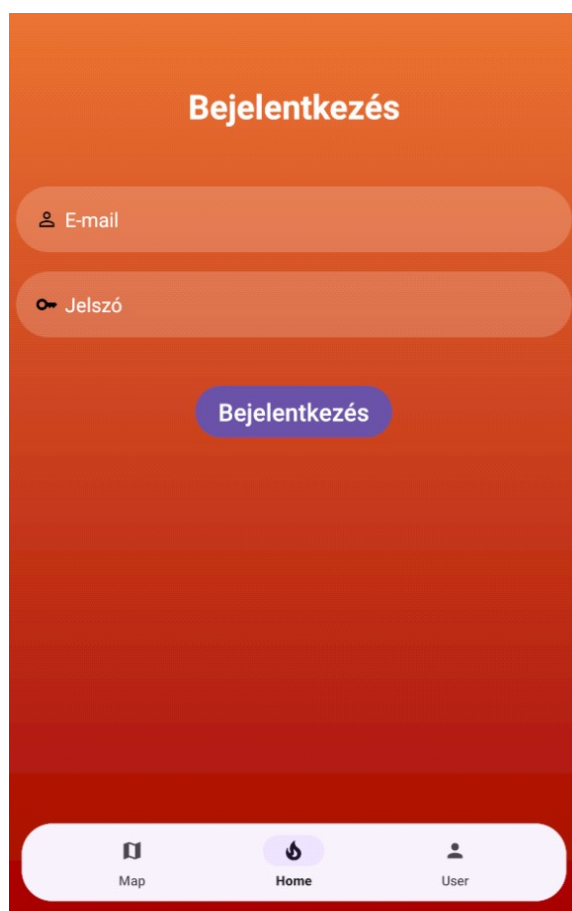
Az **adatbázis** esetünkben a *FireBase*, ami egy NoSQL adatbázis. A mi esetünkben az egyszerű használata miatt volt ez a megfelelő választás, sok tanító videó van és könnyen átlátható. Próbálkoztunk a MongoDB-vel is, de kevesebb sikerrel, így a FireBase maradt a mi megfelelő és végleges választásunk.

# Felhasználói felület

Az "UI" rövidítés a "User Interface" (felhasználói felület) kifejezésre utal. Ez a szoftverek, alkalmazások, weboldalak vagy más interaktív rendszerek azon része, amelyekkel közvetlenül kapcsolatba lépnek és kommunikálnak a felhasználók. Az UI tervezési elemeiből áll, amelyek lehetővé teszik az adatok megjelenítését és a felhasználók interakcióját a rendszerrel.

A mi projektünk esetén egy UI-ról van szó, mivel mindenki ugyanazt lássa, egy apró eltérés van csak a parancsnok és az önkéntesek között.

Az app megnyitásakor a következő oldal fogad, ami a bejelentkezés menü. Itt bármikor belehet jelentkezni egy már meglévő fiókkal.



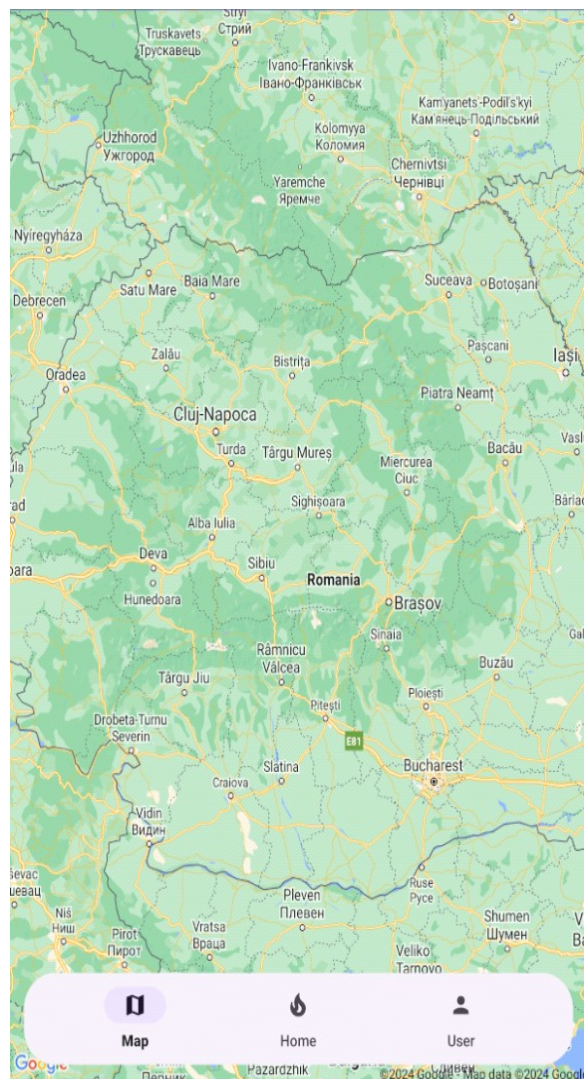
3.ábra: Bejelentkezés felület

A bejelentkezéstől jobbra található meg a regisztráció, ahol bárki szabadon létrehozhat egy fiókot.

The image shows a mobile application registration screen titled "Regisztráció". It features three input fields: "E-mail", "Jelszó" (Password), and "Jelszó újra" (Repeat Password). Below these fields are three checkboxes for "Elfogadom az engedélynyilatkozatot", "Általános szerződési feltételek", and "Adatvédelemi nyilatkozat". A blue "Regisztráció" button is positioned below the checkboxes. At the bottom, there is a navigation bar with three icons: "Map", "Home", and "User", with the "User" icon being highlighted.

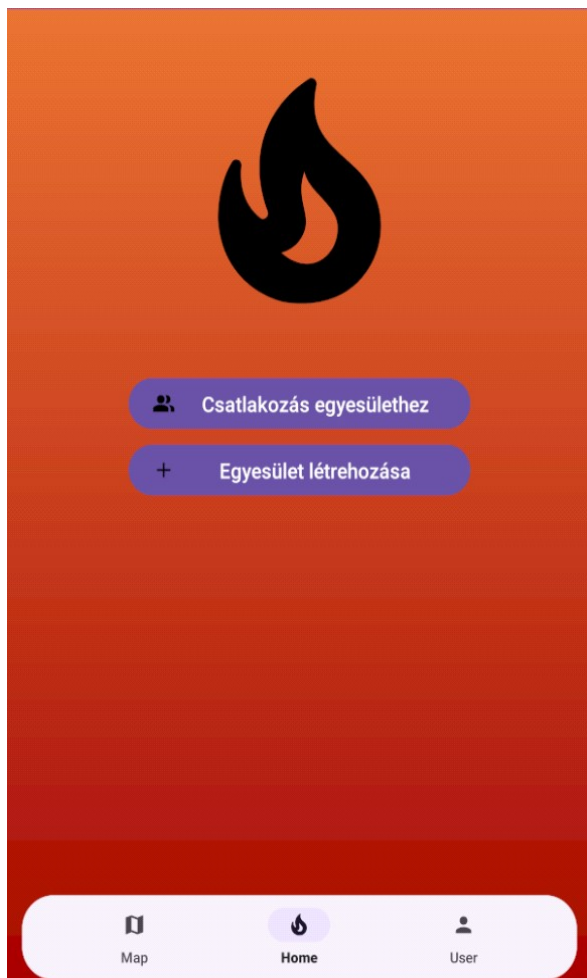
**4.ábra: Regisztrációs felület**

A harmadik oldal, amit egy felhasználó láthat még bejelentkezés nélkül az a Térkép, ami a Google Maps implementálása. Szabadon mozgatható, de még a tűzcsapok sajnos nincsenek bejelölve.



**5.ábra: Térkép felület**

Egyesülethez csatlakozni, vagy egyesületet létrehozni csak a már Bejelentkezett, illetve Regisztrált felhasználók tudnak.



**6.ábra: Menü felület**

Ha az „Csatlakozás egyesülethez” gombra kattintunk, ott egy az alábbihoz hasonló mezőt dob ki, ahol egy egyesület kódját lehet beírni.

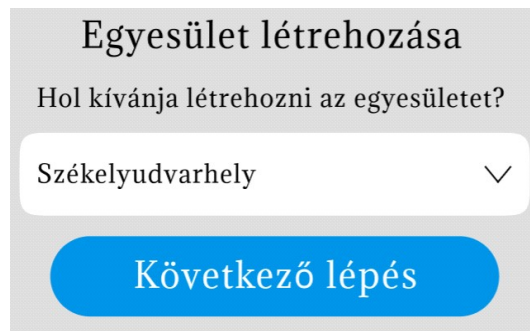
### Egyesülethez csatlakozás

Egyesület csatlakozási kódja

Csatlakozás

**7.ábra: Egyesülethez csatlakozás felület**

Ha viszont saját egyesületet szeretnél létrehozni, akkor a következőképpen valósíthatod meg.



**Egyesület létrehozása**

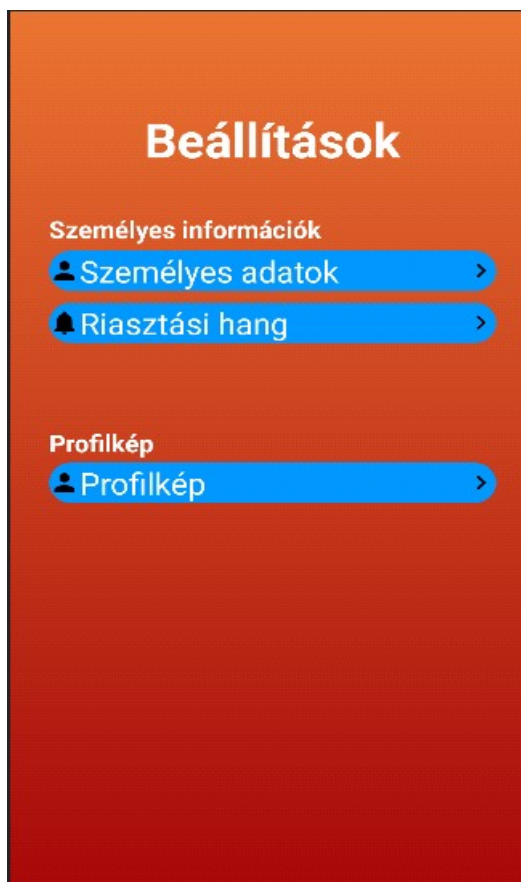
Hol kívánja létrehozni az egyesületet?

Székelyudvarhely ▾

**Következő lépés**

**8.ábra: Egyesület létrehozás felület**

Bejelentkezés után elérhető lesz egy Beállítások menüpont is, ahol kedv szerint lehet állítani a következő három dolgot:



**Személyes adatok**

Neved

Családnév Keresztnév

Telefonszámod

+407XXXXXXXXX

Csatlakozás

**Profilkép választás**

Grid of 6 firefighter profile icons:

- Icon 1: Firefighter with yellow helmet and black uniform.
- Icon 2: Firefighter with red helmet and yellow uniform.
- Icon 3: Firefighter with yellow helmet and red uniform.
- Icon 4: Firefighter with orange helmet and black uniform.
- Icon 5: Firefighter with red helmet and red uniform.
- Icon 6: Firefighter with yellow helmet and red uniform.

Csatlakozás

**Riasztási hang**

Riasztási hang

☒ Hang 1

☐ Hang 1

☐ Hang 1

Csatlakozás

9.ábra: Beállítások felület

# Projektmenedzsment

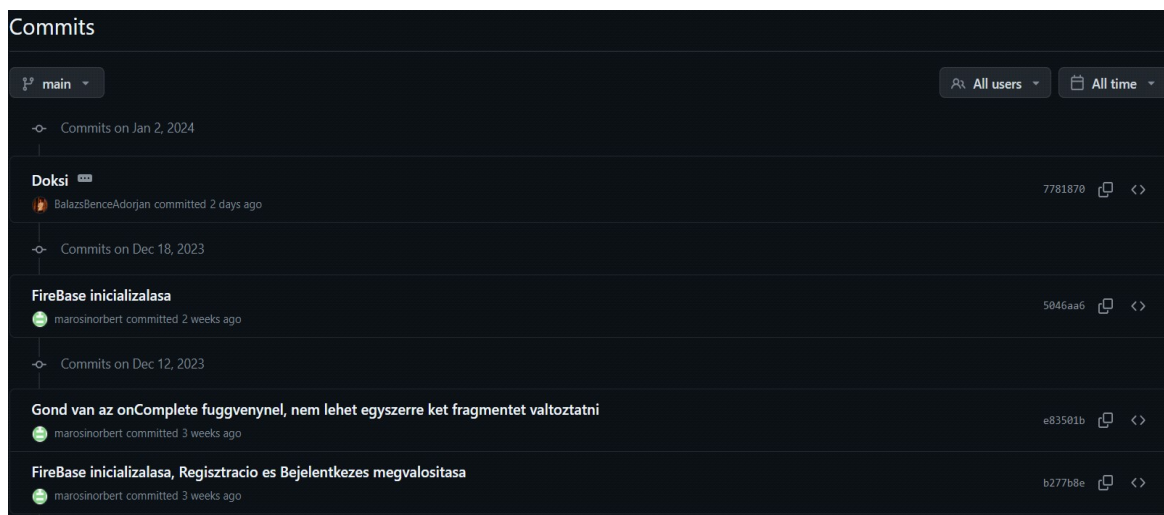
Az alkalmazás fejlesztése hosszú folyamat, ami időnként több hónapot is igénybe vehet. Ebben az időszakban a projekt különböző verziókkal bővíthet, amelyek eltérő ágakon vagy branch-eken fejlődnek tovább. Azért, hogy pontosan nyomon követhessük ezeket a változásokat időrendi sorrendben, elengedhetetlen olyan verziókezelő rendszerek használata, mint például a Git.

A Git egy nyílt forráskódú, elosztott verziókezelő szoftver, amely lehetővé teszi számunkra, hogy átlátható módon kövessük és kezeljük a fejlesztés során bekövetkező módosításokat.

Az egyik legelterjedtebb és ingyenes tárhelyszolgáltatás jelenleg a GitHub. Ez az internetes tárhelyszolgáltatás kifejezetten szoftverfejlesztéshez és verziókezeléshez készült, kihasználva a Git rendszerét. A platform lehetővé teszi a Git elosztott verziókezelését, hibakövetést, szoftverfunkciók kérését, feladatkezelést, folyamatos integrációt és minden projekt számára readme oldalakat biztosít. A jelen projekt is ezt a szolgáltatást használja, és megtalálható a következő adattárakban (repository-ban):

<https://github.com/marosinorbert/FireForce>

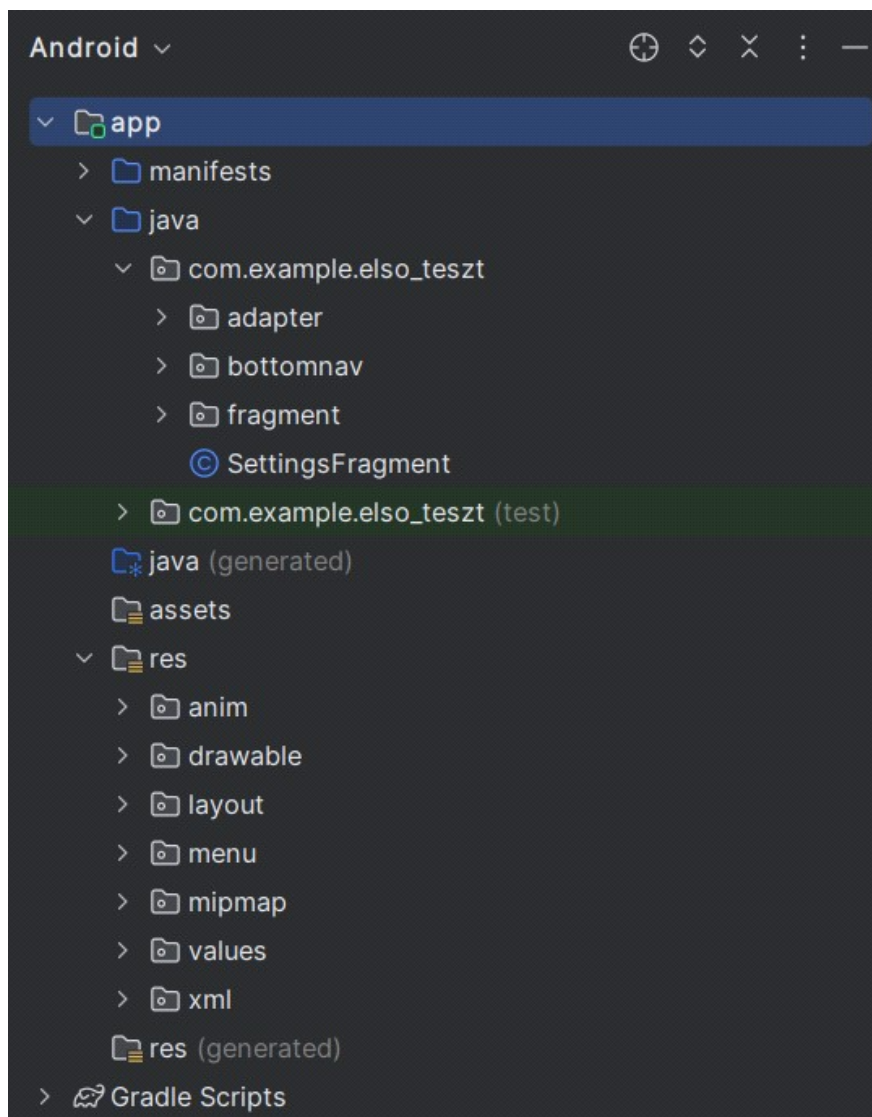
A mi csapatunk leginkább összeül és közösen írta a kódokat, mert nekünk az kézenfekvőbb volt, de a projekt kisebb fejlődései a commit-ok által is megfigyelhető. Itt egy példa a fejlődés követésére:



10.ábra: GitHub Commit felület



Mi a projektünket Android Studioban készítettük, azon belül is AndroidApp-ot hoztunk létre. A projekt felépítése a következő képen figyelhető meg:



**11.ábra: Projekt felépítés**

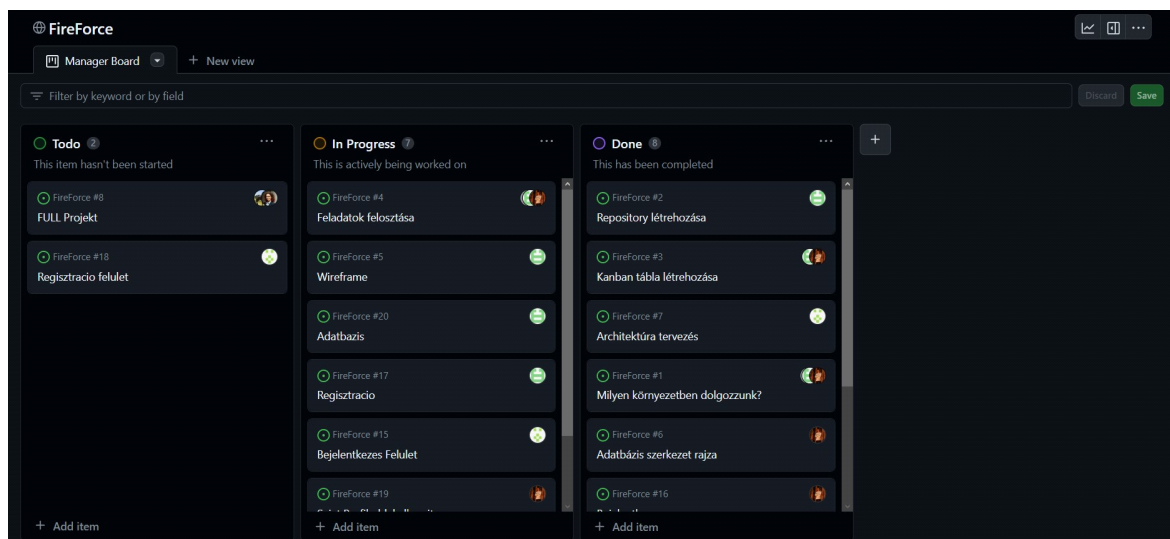
## Feladatkezelés - Kanban tábla

Egy projekt fejlesztése közben rendkívül fontos, hogy a feladatok ki legyenek osztva a csapat tagja között. Egy nagyon effektív eszköz erre, amivel mindezt meg lehet valósítani az a *Kanban* tábla.

A *Kanban* tábla GitHubba van beépítve, így hatékony volt ezzel dolgozzunk, ez egy olyan online eszköz, amelyet a projektkezelésre vagy a feladatok nyomon követésére használnak. Könnyen létrehozható, átlátható és módosítható. Több oszlop is létrehozható, de mi 3 táblával is beértük. A mi tábláink ezeket tartalmazzák:

- **Todo:** Ebben az oszlopban az ötleteinket és az elvégzendő feladatainkat írjuk, amelyeket a későbbiekben valósítunk meg.
- **In Progress:** Itt azok a feladatok vannak, amelyeken jelen pillanatban is dolgozunk vagy nincs teljesen befejezve.
- **Done:** A névből is adódóan ebben az oszlopban a teljesen kész, elvégzett feladatok vannak.

Így néz ki a mi Kanban táblánk a GitHubon jelen pillanatban:

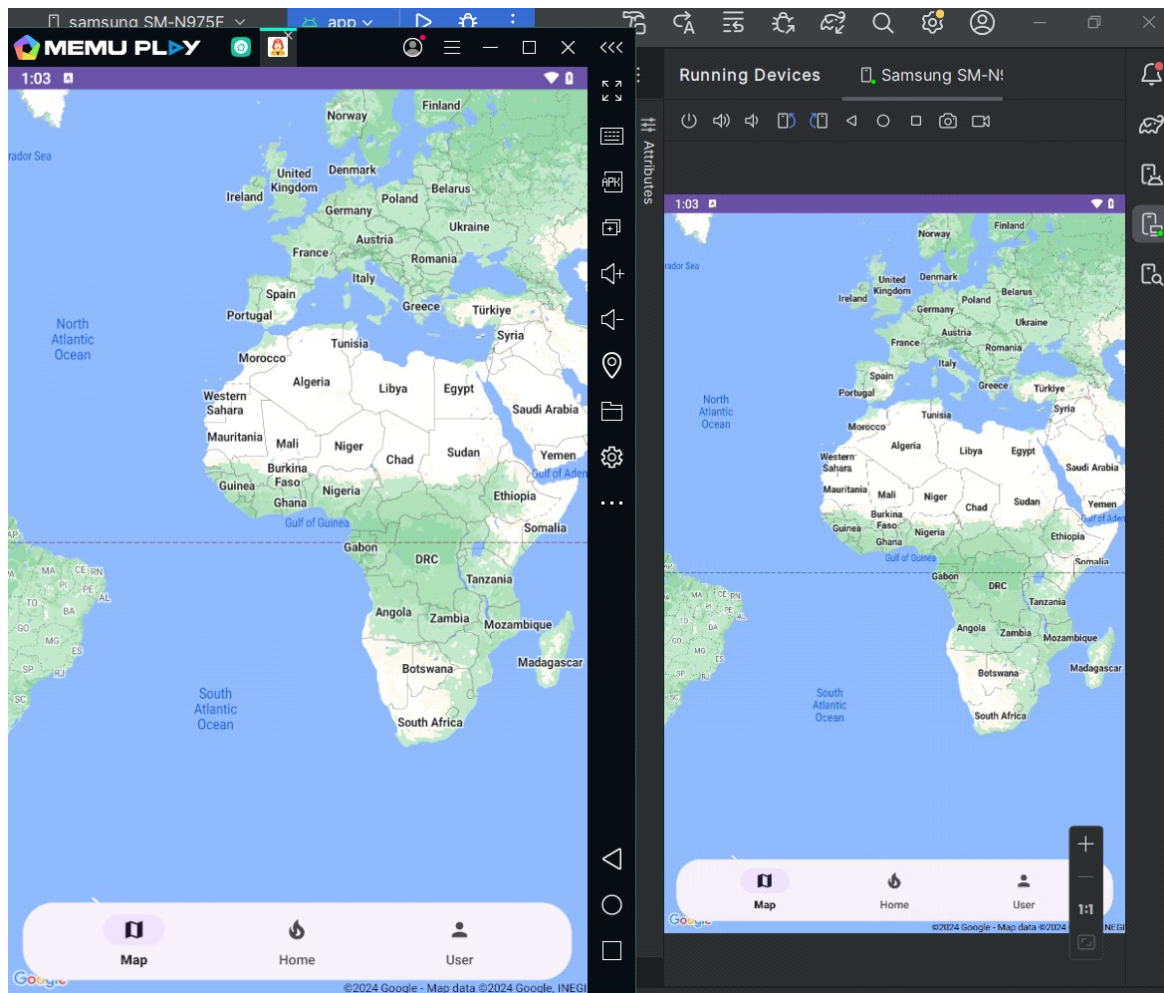


12.ábra: Kanban tábla

Egyik táblából a másikba könnyen át lehet húzni a feladatokat, így amint valamin dolgozunk, vagy egy feladat elkészült máris a hozzá kirendelt táblába kerül. Ki lehet osztani, hogy ki melyik feladaton dolgozzon és ezt bármikor lehet módosítani. Érdekességként a projektvezetőinkre bízunk a projekt elkészítését, de nem végeztek el a feladatukat.

## Alkalmazás tesztelése

A teszteket az AndroidStudios belül végeztük el, sokáig virtuális telefont használva, de az utóbbi időben áttértünk a fizikai eszköz használatára. Volt csapattag, aki nem tudta virtuálisan futtatni az eszközt, így egy külső program segítségével a MEMu-val tudott futtatni az AndroidStudióban egy fizikai-ként érzékelt virtuális eszközt:



13.ábra: Tesztelési folyamat

# Alkalmazás működése

Az alkalmazás három nagy részre osztható:

- front end (UI)
- backend
- adatbázis

## Front End

A FireForce alkalmazás front end része a felhasználók elsődleges interakciós pontja, ahol az önkéntes tűzoltók kapcsolatba lépnek az alkalmazással. Az interfész ezeken a pontokon keresztül kínál lehetőséget az események megtekintésére, az azokra való reagálásra és a tevékenységek rögzítésére.

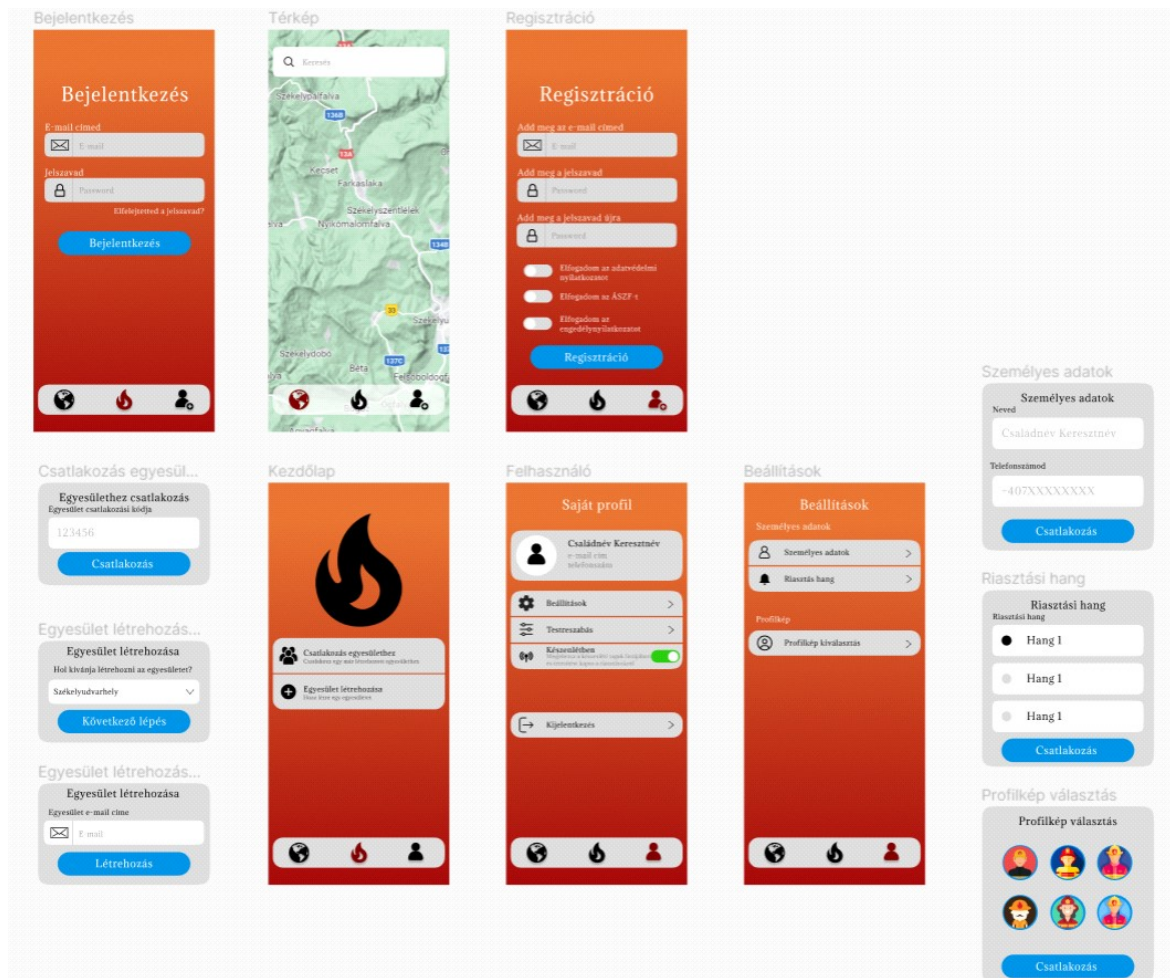
Az alkalmazás front end felülete intuitív navigációt kínál a felhasználóknak, lehetővé téve számukra, hogy könnyen böngésszék az eseményeket, és válaszoljanak azokra. A felhasználóbarát felület megkönnyíti az önkéntesek számára, hogy azonnal értesüljenek a tűzoltási eseményekről, és egyszerűen reagáljanak rájuk.

Az alkalmazás esztétikus dizájnnal és átgondolt elrendezéssel rendelkezik, amely lehetővé teszi a könnyű áttekintést és a gyors cselekvést. A kezelőfelület intuitív elemeket használ, például gombokat, listákat és kereső funkciókat, amelyek lehetővé teszik a felhasználók számára, hogy gyorsan és hatékonyan kezeljék a rendszer funkcióit.

Emellett az alkalmazás front end része a felhasználók számára lehetőséget kínál az események térképen történő megtekintésére és a tűzoltási helyszínek pontos meghatározására. A térképes megjelenítés átlátható és segíti az önkénteseket abban, hogy gyorsan eligazodjanak és hatékonyan koordinálják a segítségnyújtást.

A FireForce front end része tehát összpontosít az egyszerű navigációra, az események gyors megjelenítésére és a felhasználóbarát felület biztosítására, hogy az önkéntesek hatékonyan és gyorsan reagálhassanak a tűzoltási eseményekre.

Az egyik első lépés az UI megtervezése volt Figma-ban, amely a következő képen tekinthető meg:



14.ábra: Figma WireFrame terv

## **Backend**

A FireForce alkalmazás hátsó része, azaz a backend, a rendszer működésének motorja. Ez az alkalmazás rejtett része, amely a felhasználók által nem közvetlenül látható, viszont létfontosságú a teljes funkcionalitásért és az adatkezelésért.

A backend felelős az adatbázis-kezelésért, az adatok tárolásáért és azok hatékony lekérdezéséért. Ez az alkalmazás része kezeli az összes tűzoltási eseményhez kapcsolódó információkat, például az események leírását, helyszínét, státuszát és az önkéntes tűzoltók adatait.

Az alkalmazás backendje biztosítja az események kezelését és rögzítését, valamint az összes interakciót az adatbázissal. Ezen keresztül történik az önkéntesek regisztrációja és az eseményekhez történő csatlakozás, valamint az események státuszának és részleteinek frissítése.

A backend működése biztosítja az alkalmazás stabilitását és hatékonyságát. Ez az alkalmazás része gondoskodik arról, hogy az adatok biztonságosak legyenek, és az összes funkció zavartalanul funkcionáljon a frontenddel történő összeköttetés során.



Összességében a backend felelős az adatok tárolásáért, kezeléséért és biztosítja azok elérhetőségét a frontend számára. Ez a rész teszi lehetővé a FireForce alkalmazás teljes funkcionalitását és hatékony működését.

```
plugins {
    id("com.android.application")
    id("com.google.android.libraries.mapsplatform.secrets-gradle-plugin")
    id("com.google.gms.google-services")
}

android {
    @@ -33,11 +34,23 @@ android {
}

dependencies {
    // Import the Firebase BoM
    implementation(platform("com.google.firebase:firebase-bom:32.7.0"))

    implementation("androidx.appcompat:appcompat:1.6.1")
    implementation("com.google.android.material:material:1.8.0")
    implementation("androidx.constraintlayout:constraintlayout:2.1.4")
    implementation("com.google.android.gms:play-services-maps:18.1.0")

    // TODO: Add the dependencies for Firebase products you want to use
    // When using the BoM, don't specify versions in Firebase dependencies
    implementation("com.google.firebase:firebase-analytics")
    implementation("com.google.firebase:firebase-auth:22.3.0")

    // Add the dependencies for any other desired Firebase products
    // https://firebase.google.com/docs/android/setup#available-libraries

{
    "project_info": {
        "project_number": "1094277280345",
        "project_id": "fireforce-72922",
        "storage_bucket": "fireforce-72922.appspot.com"
    },
    "client": [
        {
            "client_info": {
                "mobilesdk_app_id": "1:1094277280345:android:f985666e115ad267b2b748",
                "android_client_info": {
                    "package_name": "com.example.elso_teszt"
                }
            },
            "oauth_client": [],
            "api_key": [
                {
                    "current_key": "AIzaSyD8sq5teDOPHgaoES771KlIrRwDrNq8h3Q"
                }
            ],
            "services": {
                "appinvite_service": {
                    "other_platform_oauth_client": []
                }
            }
        }
    ]
}
```

**15.ábra: Backend kódrészlet**

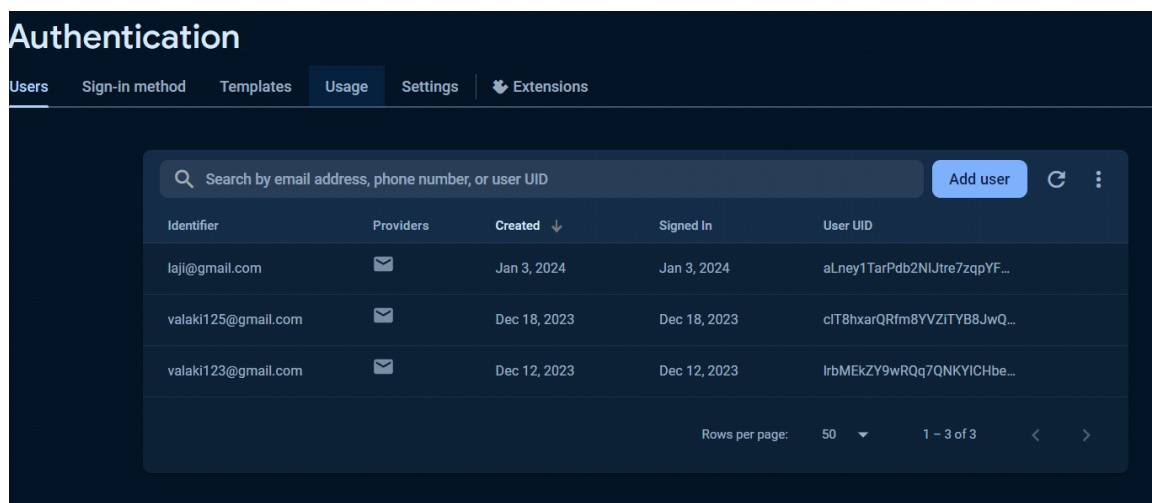
## Adatbázis

Az alkalmazásunk háttértárolása és adatkezelése során a Firebase adatbázist választottuk. A Firebase az egyik vezető valós idejű, NoSQL alapú adatbázis, mely szorosan integrálódik a Google Firebase platformmal. Ennek az adatbázisnak az a fő célja, hogy lehetővé tegye az alkalmazásoknak az adatok valós idejű szinkronizációját és a felhasználók interaktív kapcsolatát az adatokkal.

Ez a strukturált adatgyűjtemény biztosítja az alkalmazásnak a hatékony tárolást és a gyors adatelérést. A Firebase rendszerén belül a NoSQL adatbázis lehetővé teszi az adatok könnyű és rugalmas kezelését, amelyeket különböző entitások közötti kapcsolatok szervezésekor is alkalmazhatunk.

Az alkalmazásban használt Firebase adatbázis lehetőséget biztosít a valós idejű adatszinkronizációra, így az összes felhasználó számára az azonnali frissítéseket és változásokat jelenthetjük be. Ezáltal a felhasználók számára a legaktuálisabb adatokat tudjuk szolgáltatni, mindezt a Firebase dinamikus és skálázható adatkezelési képességeivel.

A következő képen szeretnénk bemutatni a regisztrációnk tesztelését az adatbázisba:



The screenshot shows the 'Authentication' section of the Firebase console, specifically the 'Users' tab. It displays a table of registered users with columns for Identifier, Providers, Created, Signed In, and User UID. There are three users listed, all using email as a provider. The table includes a search bar at the top and pagination controls at the bottom.

Identifier	Providers	Created	Signed In	User UID
laji@gmail.com	📧	Jan 3, 2024	Jan 3, 2024	aLney1TarPdb2NIJtre7zqpYF...
valaki125@gmail.com	📧	Dec 18, 2023	Dec 18, 2023	cIT8hxrQRfm8YVZiTYB8JwQ...
valaki123@gmail.com	📧	Dec 12, 2023	Dec 12, 2023	lrbMEkZY9wRQq7QNKYICHbe...

16.ábra: FireBase adatbázis regisztrációs tábla



## Adatbázis függvények

És itt található meg a regisztrációnak a függvénye.

```
Button regisztracioGomb = view.findViewById(R.id.regisztracio_gomb);

regisztracioGomb.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {

        EditText emailEditText = view.findViewById(R.id.email);
        EditText passwordEditText = view.findViewById(R.id.jelszo);

        String email = emailEditText.getText().toString().trim();
        String password = passwordEditText.getText().toString().trim();

        // Regisztráció
        registerUser(email, password);

    }
});

return view;

Regisztrációs függvény hozzáadása
private void registerUser(String email, String password) {
    mAuth.createUserWithEmailAndPassword(email, password)
        .addOnCompleteListener(requireActivity(), new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                if (task.isSuccessful()) {
                    // Sikeres regisztráció
                    Toast.makeText(requireContext(), "Sikeres regisztráció", Toast.LENGTH_SHORT).show();

                    // Ide adhatsz hozzá kódot, amit a sikeres regisztráció esetén szeretnél végrehajtani
                } else {
                    // Sikertelen regisztráció
                    Toast.makeText(requireContext(), "Sikertelen regisztráció: " + task.getException().getMessage(), Toast.LENGTH_SHORT).show()
                }
            }
        });
}
```

**17.ábra: Adatbázis függvénykódok**

## További fejlesztési lehetőségek

1. **Riasztások és értesítések fejlesztése:** A rendszer további értesítési funkciókkal bővíthető, például testre szabható riasztásokkal, amelyek lehetővé teszik az önkéntesek számára az esetleges vészhelyzetek azonnali értesítését.
2. **GPS és térképintegráció fejlesztése:** Az alkalmazás térképkezelése tovább fejleszthető azáltal, hogy pontosabb helymeghatározást biztosít, és lehetővé teszi az önkéntesek pontosabb helyszínre való navigálását a tűzoltási helyszínre.
3. **Felhasználói szerepkörök bővítése:** Az alkalmazás szerepköröit lehet bővíteni vagy differenciálni azáltal, hogy különböző jogosultságokat és lehetőségeket biztosítunk az egyes felhasználók számára (pl. vezetők, támogatók stb.).
4. **Jelentések és analitika:** Egy részletes jelentési funkció bevezetése lehetővé tehetné a vezetők számára az események, beavatkozások és egyéb tűzoltási tevékenységek elemzését és kiértékelését.
5. **Offline mód támogatása:** Az alkalmazás lehetne felkészítve arra, hogy bizonyos esetekben offline módban is elérhető legyen, különösen olyan területeken, ahol nincs megbízható internetkapcsolat.
6. **Felhasználói visszajelzések gyűjtése és integrálása:** Az alkalmazásba beépített visszajelző rendszer lehetőséget adna az önkénteseknek és felhasználóknak arra, hogy visszajelzéseikkel és javaslataikkal hozzájáruljanak az alkalmazás fejlesztéséhez és javításához.
7. **Bővített biztonsági funkciók:** Az alkalmazás biztonságát tovább lehetne fejleszteni, például az adatvédelmi szabályozások szigorításával és az esetleges biztonsági rések kijavításával.
8. **Videó- és hangrögzítés:** Az alkalmazásban lehetőség lenne videó- és hangrögzítésre, ami segíthet az események dokumentálásában és későbbi elemzésében, illetve további tanulási lehetőségek kialakításában.
9. **Önkéntesek közötti kommunikáció:** Egy beépített kommunikációs rendszer lehetőséget adhat az önkéntesek közötti gyors és hatékony kommunikációra. Ez lehetne chat, vagy akár csoportos hívások funkcionálitása az alkalmazáson belül, amely megkönnyítené az eseményekkel kapcsolatos koordinációt és információcsere lehetőségét.

## Hibák javítása

A szoftverfejlesztés során elengedhetetlen fontosságú a hibamentes működés biztosítása. Bár egyetlen szoftver sem mentes a hibáktól, ez a webalkalmazás sem kivétel. Jelenleg az alábbi ismert hibák (bugok) találhatóak a kód bázisban:

- Van amikor a térkép rosszul jeleníti meg a helyszínt.
- Nem küld értesítést egy másik eszközre
- Még elfordulnak kisebb apró hibái, amik javításra szorulnak.

## Bónusz lehetőség

Ha sikerül egy 100%-ban megbízható, hibáktól mentes és teljesen működő programot létrehozni, akkor szívesen odaadnánk a környékünkön megtalálható önkéntes tűzoltóknak, hogy könnyebbé tegyük vagy segítsük a bátor munkájukat.

## Bibliográfia

A dokumentáció elkészítéséhez felhasznált források:

ChatGPT: <https://openai.com/blog/chatgpt>

FireBase: <https://firebase.google.com>

DrawSql: <https://drawsql.app/teams/fireforce/diagrams/fireforce>

Github: <https://github.com>

Wikipedia: <https://en.wikipedia.org>