**FireForce**  
Android alkalmazás

**Fejlesztők: Vezető tanár:**

Balázs Bence-Adorján Ferencz Katalin

Marosi Norbert

Péter Ákos

* **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 tűzesetekről, 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 tűzesetekről, lehetővé téve a gyors reagálást és a tűzoltási munkálatok hatékonyabb koordinációját.
* **Közösségi Szerepvállalás:** A FireForce elősegíti a közösségi szerepvállalást a tűzvédelmi intézkedésekben és a tűzesetek 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 kövtkezők:

* parancsnok(az egyesület létrehozója)
* önkéntes(aki csatlakozott az egyesülethez)

**2DB ÁBRA IDE**

* **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 nem autentikált felhasználók, csak a térképet tudják megnézni, nem képesek egyesületet létrehozni, vagy csatlakozni egysülethez.

* **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 értelem szerűen 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 Fire Force 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,
* Google Maps-hoz teljes hozzaférés,
* 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 átgondoltabb 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 egyszerűségre 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 ehez kapcsolódik a **Felhasználó**, a **Térkép** és az **Adatbázis** is.

Az architektúránk a következő képpen valósult meg:



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 tűzesetet 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 hozzaadható 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 (Folytatás)**