Számítógép-Biztonság Design Dokumentum és Tesztelési Terv

1. Bevezető

A Számítógép-biztonság tárgy keretében egy alkalmazást kell készítenünk, aminek készítése során egy biztonságkritikus szoftver teljes tervezési, fejlesztési és tesztelési folyamatát megismerjük. Ebben a dokumentumban az alkalmazás tervezését mutatjuk be.

Az alkalmazás egy online áruház, amiben a CAFF fájlformátumú animált képeket lehet nézegetni. A tervezés során megvizsgáljuk az alkalmazástól elvárt funkcionális és biztonsági követelményeket, megállapítjuk a rendszer elemeit és a köztük zajló adatfolyamokat, vizsgáljuk a lehetséges támadó modelleket és ezek alapján meghatározzuk a szükséges biztonsági funkciókat. Ezek után elkészítjük a rendszer architektúrájának modelljét.

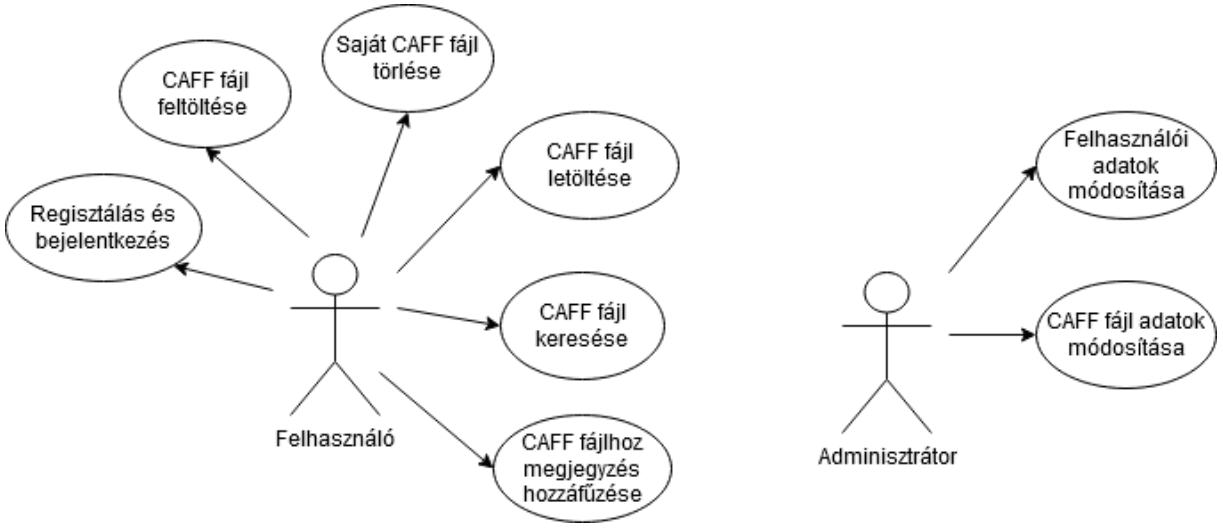
2. Követelmények meghatározása

2.1 Funkcionális követelmények

A tárgy oldalon lévő követelmények alapján az 1. ábrán látható felhasználási szcenáriókat vettük fel.

A felhasználóknak biztosítani, kell hogy regisztráció, majd bejelentkezés után képesek legyenek CAFF fájlokat feltölteni, keresni a kollektív CAFF fájl gyűjteményben, megjegyzéseket fűzni a CAFF fájlokhoz és letölteni azokat. Illetve a saját CAFF fájljaikat tudják törölni.

Az adminisztrátorok, tudják módosítani a felhasználókhoz kapcsolódó adatokat és az összes CAFF fájlhoz tartozó adatokat, illetve törölhetik is bármelyik CAFF fájlt.

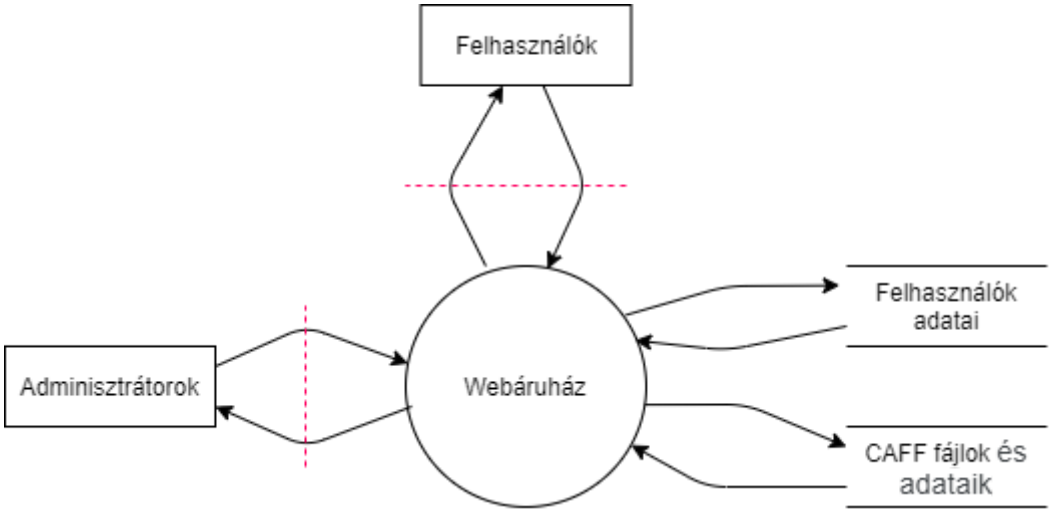


1. ábra: Felhasználói szcenáriók

Az alkalmazásnak egy webszervernek kell lennie, és a teljesítmény érdekében a CAFF fájl feldolgozást C++ nyelven kell megvalósítani.

2.2 Biztonsági követelmények és célok

A funkcionális követelmények segítségével meghatározhatjuk az elkészítendő rendszert nagy vonalakban, illetve annak a környezetét. Két fajta felhasználói szerepkört különböztethetünk meg, az átlag felhasználókat és az adminisztrátor felhasználókat. A felhasználóknak a rendszerrel való interakciója bizalmi kérdéseket vet fel, mivel a tőlük érkező kéréseket nem tudjuk kontrollálni. A 2. ábrán ez piros szaggatott vonallal látható. Továbbá, a rendszernek el kell tárolni a felhasználók adatait, a feltöltött CAFF fájlokat és az adataikat, illetve a felhasználók és a CAFF fájlok kapcsolatát (például tudni kell, hogy ki töltötte fel az adott fájlt).



2. ábra: A rendszer és a környezete

A biztonsági követelmények és célok meghatározásához használjuk az alábbi hat kategóriát: CIA és AAA.

Bizalmasság (confidentiality, read access)

A felhasználók személyes adatait védeni kell a külső entitásoktól. Csak ők maguk és az adminisztrátorok férhetnek hozzá.

A felhasználók és az adminisztrátorok egyaránt láthatják a többi felhasználó által feltöltött CAFF fájlokat és fájlokhoz fűzött megjegyzéseket.

cél: személyes adatokat titkosítva tárolása, adathozzáférést kontrollálása

Integritás (integrity, modification)

A felhasználók csak a saját személyes adataikat, a saját maguk által feltöltött CAFF fájlokat illetve a saját kommentjeiket módosíthatják és törölhetik.

Az adminisztrátorok módosíthatják és törölhetik minden felhasználó személyes adatait, kommentjeit, illetve a feltöltött CAFF fájlokat.

cél: hozzáférést kontrollálása

Elérhetőség (availability, access to system)

A felhasználóknak mindig rendelkezésre kell, hogy álljon a rendszer. Nincsenek időkorlátozások.

cél: állandó elérhetőség biztosítása

Autentikáció (authentication, who is the entity?)

A felhasználók be tudnak regisztrálni az oldalra.

A felhasználók csak bejelentkezés után tudják használni a rendszert.

cél: felhasználó kezelés biztosítása

Autorizáció (authorization, can the entity do that?)

A felhasználók csak vásárlás után tölthetik le a másik felhasználók által hirdetett fájlokat. (Ez egy valós webshop esetén lenne elvárás, a házi feladat során a vásárlás funkciót nem implementáljuk.)

cél: hozzáférés korlátozása, biztonságos fizetési mechanizmus

Auditálás (auditing, proof of performed activities)

A felhasználók és az adminisztrátorok tevékenységét naplózni kell.

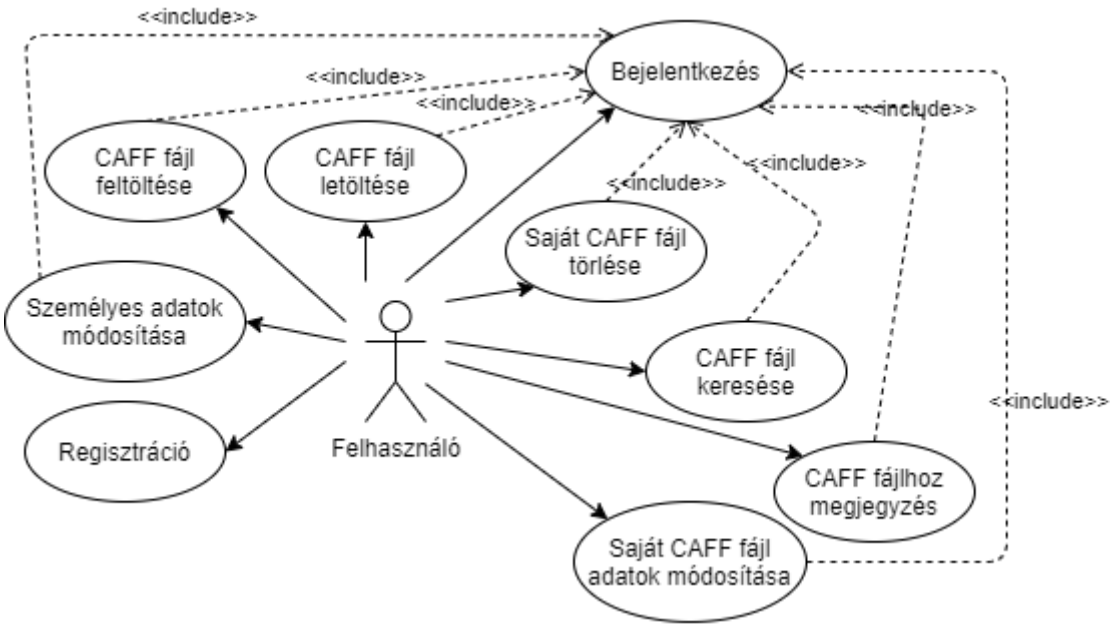
cél: naplózás megvalósítása

2.3 Threat Assessment

A Threat Assessment során először meghatározzuk a rendszer assetjeit, majd megvizsgáljuk, hogy milyen veszélyektől kell megóvnunk ezeket az asseteket. A rendszer assetjeinek a meghatározásához a rendszer szereplőinek a use case-it vesszük sorra. A veszélyek felderítése során a STRIDE keretrendszert követjük.

2.3.1 Assetek megállapítása

Az assetek megállapítását a felhasználók use case-inek vizsgálatával kezdjük. A felhasználóknak hét darab use case-ük van: regisztráció, bejelentkezés, személyes adatok módosítása, CAFF fájl feltöltése, saját CAFF fájl törlése, CAFF fájl letöltése, CAFF fájl keresése és CAFF fájlhoz megjegyzés fűzése. A CAFF fájlokkal való műveletek végzéséhez, illetve az adatainak a módosításához a felhasználónak be kell jelentkeznie az alkalmazásba, tehát ezek extra megkötések a use casek között. A felhasználóhoz köthető use case-k az 3. ábrán láthatóak.



3. ábra: A felhasználóhoz köthető use case-k

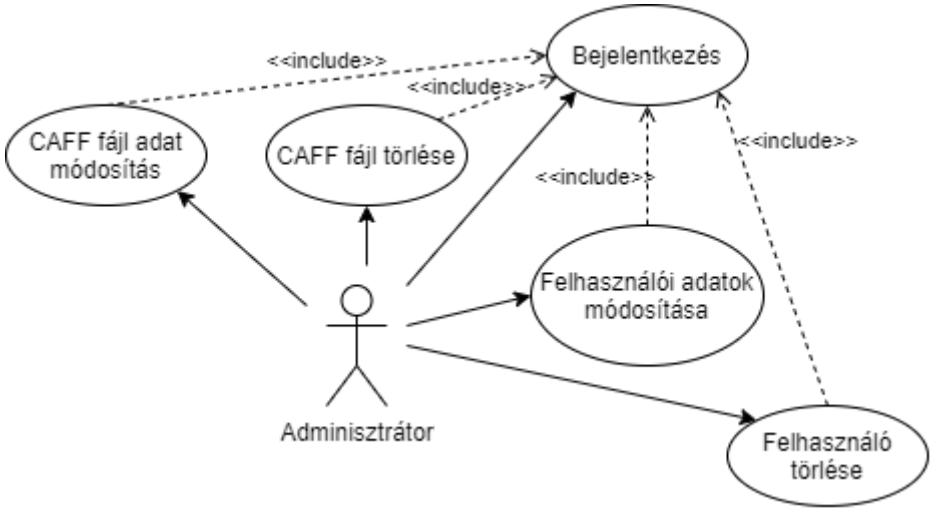
A use case-k alapján szükségünk lesz a következő fizikai assetekre: szerverekre, amiken a webalkalmazás futhat, illetve hálózati eszközökre a távoli elérést biztosítandó. A felhasználó egy emberi asset lesz.

A use casek megvalósításának vizsgálatával határozhatjuk meg, hogy milyen logikai assetekre lesz szükségünk. Az alkalmazás egy online webáruház, ahova a felhasználóknak először regisztrálniuk, majd később, hogy hozzáférjenek a webáruház szolgáltatásaihoz be kell jelentkezniük. Ezek miatt szükség lesz egy adatbázisra, ami a regisztrált felhasználók adatait tárolja, egy adatkezelőre, ami a felhasználói adatbázist kezeli és egy authentikációs komponensre, ami a felhasználók bejelentkeztetését végzi.

A felhasználónak lehetősége van CAFF fájlokat feltölteni a webáruházba. Ebből kifolyólag szükségünk lesz egy adatbázisra, ami a CAFF fájlokat tárolja, és egy komponensre, ami a CAFF fájlokkal kapcsolatos műveleteket végzi. A CAFF fájlok adatait és magukat a CAFF fájlokat külön tároljuk. A CAFF fájlok feldolgozását a CAFF fájl feldolgozó komponens végzi, ami a CAFF fájl kezelő komponenssel kommunikál. A CAFF fájl kezelő komponens kezeli a felhasználó további CAFF fájllal kapcsolatos use caseit, a saját CAFF fájljainak törlését, a letöltést, a keresést és a megjegyzés hozzáfűzését. A CAFF fájl kezelő komponens kommunikál a felhasználói adatkezelővel, hogy a CAFF fájl feltöltéskor hozzáférjen a feltöltő adataihoz.

A személyes adatok módosítása use case miatt szükség van egy hozzáférést szabályzó komponensre, mivel a felhasználó csak a saját adatait módosíthatja. A naplózást is használjuk mint biztonsági funkció, ezért a hozzáférést szabályzó komponens felhasználja a rendszeridőt is.

Következő lépésként hozzávesszük a második szereplőnk, az adminisztrátorhoz tartozó use case-eket, ami a 4. ábrán látható.

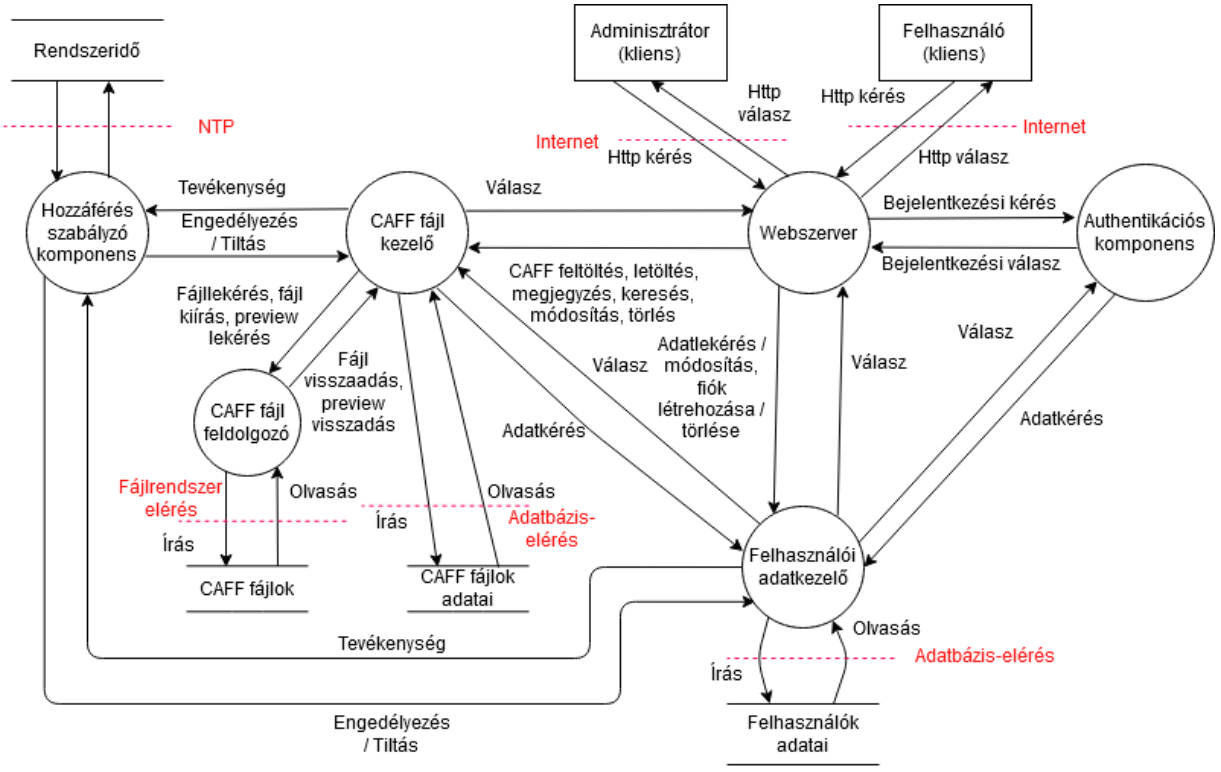


4. ábra: Az adminisztrátorhoz köthető use case-ek

Az adminisztrátor öt darab use case-zel rendelkezik: bejelentkezés, CAFF fájl módosítása, CAFF fájl törlése, felhasználó adatok módosítása és felhasználó törlése. Az adminisztrátornak először be kell jelentkeznie, hogy a többi tevékenységéhez hozzáférhessen.

Az adminisztrátor egy további emberi asset a rendszerbe. Új interakciók kerülnek be a rendszerbe, a CAFF fájl módosítása/törlése illetve a felhasználó törlése miatt. A CAFF fájlok módosítása illetve törlése, egy adminisztrátori jog, ezért a hozzáférést szabályzó komponenstől engedélyt kell kérnie a CAFF fájlhoz kezelő komponensnek ezekhez a műveletekhez.

Mivel nincs több use case, amit még nem elemeztünk, így elkészült a végleges adatfolyam diagram, ami az 5. ábrán látható.



5. ábra: Rendszer adatfolyama az adminisztrátori interakciók után

2.3.2 Támadó modell

A támadó modell kidolgozásához figyelembe kell venni az egyes assetek lehetséges gyengeségeit. A veszélyforrások rendszerezéséhez a STRIDE keretrendszert használtuk, ezzel az egyes veszélyforrás kategóriák könnyen öszerendelhetők az adatfolyam diagram egyes elemeivel. A támadási szcenáriókat (abuse casek) az alábbiakban fejtjük ki:

Megszemélyesítés

- Felhasználó törölni akar más által feltöltött CAFF fájlokat.
- Felhasználó más felhasználó személyes adataihoz próbál hozzáférni.
- Felhasználó hozzá akar férni a CAFF fájlokhoz (letöltés vásárlás nélkül).

Hamisítás

- Felhasználó módosítani akarja a CAFF fájl tulajdonságait (például: ár).
- Felhasználó módosítani akarja a CAFF fájl tartalmát / más fájl-lal helyettesíti.
- Felhasználó módosítani akarja a rendszerhez való jogosultságát.
- Felhasználó módosítani akarja a vásárlási folyamat konfigurációját.

Tevékenységek letagadása

- A felhasználó másik felhasználó account-ját használja a rendszerhez való hozzáféréshez.

- A felhasználó azt állítja, hogy nem kapta meg vásárlás után a CAFF fájlt.
- A felhasználó módosítja a logot.

Információ szivárgás

- A rendszer biztonságával kapcsolatos forráskód részletek közzététele (biztonsági rés).
- Az rendszer működésével kapcsolatos forráskód részletek közzététele (biztonsági rés).
- A felhasználói adatok közzététele.
- A CAFF fájlok közzététele.
- SQL kódok futtatása az adatbázison (SQL injection).
- Adatbázis engedélyekhez való hozzáférés és ennek kihasználása.
- Adatok olvasása a hálózaton.

Szolgáltatás-megtagadás

- Az oldal tartalmának illetéktelen módosítása.
- Denial-of-Service támadások.
- Distributed Denial-of-Service támadások.

Jogosultsági szint emelése

- Az authorizáció megkerülésével a felhasználó más jogosultságot szerez.
- A felhasználó olyan inputokat küld, amit a rendszer nem kezel helyesen.

2.4 Szükséges biztonsági funkcionalitások

A biztonsági követelményeknek való megfeleléshez több biztonsági funkcionalitást kell megterveznünk. A szükséges biztonsági követelményeket a biztonsági követelmények és a támadási szcenáriók alapján alakítottuk ki.

A közösségi fájlmegosztó rendszer használatához szükséges autentikációt megvalósítani. Mivel a tervezett rendszer egy webes rendszer, amit böngészőn keresztül fognak használni a felhasználók, érdemes volt jelszó alapú autentikációt használni, ezért ezt választottuk.

Szerepkörök alapján megkülönböztetünk adminokat és felhasználókat. Fontos, hogy egyes tevékenységek csak bizonyos szerepkörhöz tartozó felhasználók számára legyenek elérhetőek, ezért a tevékenységek megkezdése előtt szükséges, hogy ellenőrizzük azt, hogy az adott felhasználó jogosult-e a tevékenység elvégzésére. Ehhez szerep alapú autorizációs mechanizmust kell implementálnunk.

Fontos, hogy az egyes elvégzett tevékenységekről készüljön napló, hogy visszakereshetőek legyen az egyes műveletek, illetve, azért hogy tudjuk, hogy ki hajtotta végre azt.

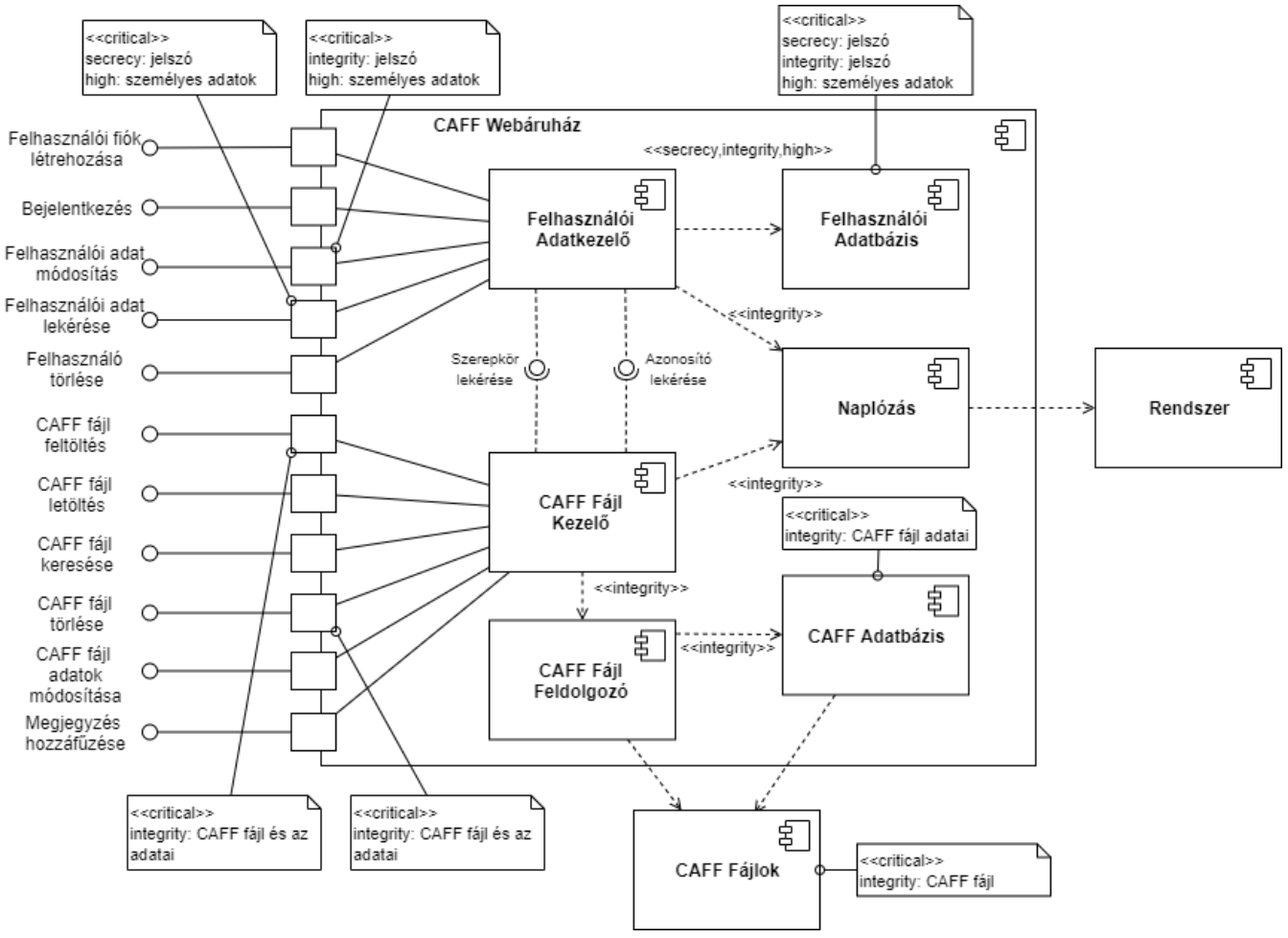
A személyes adatokat és az autentikációhoz szükséges jelszót védenünk kell szivárgás ellen, illetve az illetéktelen hozzáféréstől. A személyes adatokat ennek

érdekében szükséges titkosítani a tárolás és az átvitel során, a jelszavakat pedig szükséges biztonság módon (például: hashelés) tárolni.

Feladatunkban jelenleg nem szükséges megvalósítani a fizetéssel járó folyamatokat, így gyakorlatilag egy közösségi fájlmegosztó rendszerről van szó. Amennyiben ez valóban egy webshop lenne, szükséges lenne valamilyen formában biztosítani azt, hogy a böngészőben megjelenített CAFF fájlok ne legyen elérhetőek olyan felhasználók számára, akik még nem vásárolták meg azt. Ennek érdekében szükséges a fájlok védelme illetve titkosítása a tárolás és az átvitel során. Ezen kívül érdemes a fájlnak csak egy részét kirenderelni a felületre, valamint az eredeti minőségnél alacsonyabb minőségben megjeleníteni a felületen (vízjel, alacsonyabb felbontás).

3. Architektúra tervek

3.1 Struktúra



6. ábra: A rendszer komponens diagrammja

A rendszer komponens diagramja a 6. ábrán látható. A felhasználók felé összesen tizenegy darab interfésze van a rendszernek:

- Felhasználói fiók létrehozása: A felhasználók saját felhasználóval regisztrálhatnak a többfelhasználós rendszerbe.
- Bejelentkezés: A felhasználók ezen az interfészen keresztül tudnak bejelentkezni a rendszerbe
- Felhasználói adat módosítása: A felhasználó módosíthatja a saját adatait vagy az admin módosíthatja más felhasználók adatait ezen az interfészen keresztül.
- Felhasználói adat lekérése: A felhasználó lekérheti a saját adatait.
- Felhasználó törlése: Az admin törölheti a felhasználót ezen az interfészen keresztül.
- CAFF fájl feltöltése: A felhasználók ezen keresztül tölthetik fel a saját CAFF fájljaikat
- CAFF fájl letöltése: A felhasználók ezen keresztül tölthetnek le CAFF fájlokat
- CAFF fájl keresése: Ezen az interfész teszi lehetővé hogy a felhasználók keressenek a CAFF fájlok közt.
- CAFF fájl törlése: A felhasználók törölhetik a saját CAFF fájljaikat vagy az admin törölheti akármelyik CAFF fájlt.
- CAFF fájl adatok módosítása: CAFF fájlhoz tartozó metaadatokat módosíthatja az admin ezen az interfészen keresztül illetve a felhasználó a saját fájljainak adatait képes módosítani.
- Megjegyzések hozzáfűzése: Ezen keresztül tud a felhasználó CAFF fájlokhoz megjegyzéseket fűzni.

A felhasználók adatait a Felhasználói adatbázisban tároljuk. Az adatok kezelését a Felhasználói Adatkezelő végzi. Ezen kívül még CAFF fájlokat kezelünk ezt a CAFF Fájl kezelő végzi. A CAFF fájlok adatai (név,elérési út, stb.) egy CAFF adatbázisban lesznek tárolva. Magukat a CAFF fájlokat a fájl rendszerbe lementve tároljuk. A kimentett CAFF fájlok feldolgozását egy CAFF fájl feldolgozó komponens fogja végezni. Minden folyamat ami a CAFF fájlokat vagy a felhasználókat érinti Naplózásra kerül.

3.2 Viselkedés

A felhasználó regisztrációkor mutatott viselkedést az ábra mutatja be. Amikor beérkezik a regisztrációs kérés, a felhasználói adatkezelő ellenőrzi, hogy a kapott adatokkal létezik-e már felhasználó. Ha létezik, a kérést elutasítja. Ha nem létezik létrehozza az új felhasználót. Az auditálási követelmények szerint minden tevékenységet naplózni kell, ezt a követelményt UML megjegyzésként vettük fel a diagramra az olvashatóság kedvéért.

A rendszer viselkedését bejelentkezéskor 8. ábra mutatja. A kérés beérkezésekor a felhasználói adatkezelő ellenőrzi a kapott felhasználó_név és jelszó párost. Ha nem

egyezik a várt adatokkal, a kérést elutasítja. Ha megegyezik, akkor visszaadja a felhasználó_név alapján a felhasználó adatait.

A 9. ábrán látható a rendszer viselkedése mikor a felhasználó a saját személyes adatait módosítja. A kérés beérkezését követően a felhasználói adatkezelő lekéri a kezdeményező felhasználói_azonosító paraméterét, amivel a megfelelő felhasználó adatait módosítja a beérkező adat értékére.

CAFF fájl feltöltése esetén a rendszer viselkedése a 10. ábrán látható. Ekkor a CAFF Fájl Kezelő lekérdezi a kezdeményező szerepkörét a Felhasználói Adatkezelőtől. Ha a szerepkör Felhasználó, akkor a CAFF fájlt elmenti a szerver oldalon.

A rendszer viselkedése CAFF fájl letöltésekor a 11. ábrán látható. Ekkor a CAFF Fájl Kezelő lekérdezi a kezdeményező szerepkörét a Felhasználói Adatkezelőtől. Ha a szerepkör Felhasználó, akkor visszatér a kért CAFF fájllal.

CAFF fájl keresésekor a 12. ábra mutatja be a rendszer viselkedését. A CAFF Fájl Kezelő lekérdezi a kezdeményező szerepkörét a Felhasználói Adatkezelőtől. Ha a szerepkör Felhasználó, akkor visszatér a keresésnek eleget tevő CAFF fájl listával.

A 13. ábrán az látszódik, hogyan viselkedik a rendszer a felhasználó által CAFF fájlhoz fűzött megjegyzéskor. A CAFF Fájl Kezelő lekérdezi a kezdeményező szerepkörét a Felhasználói Adatkezelőtől. Ha a szerepkör Felhasználó, akkor a megjegyzést létrehozza a kérésnek megfelelő CAFF fájlhoz.

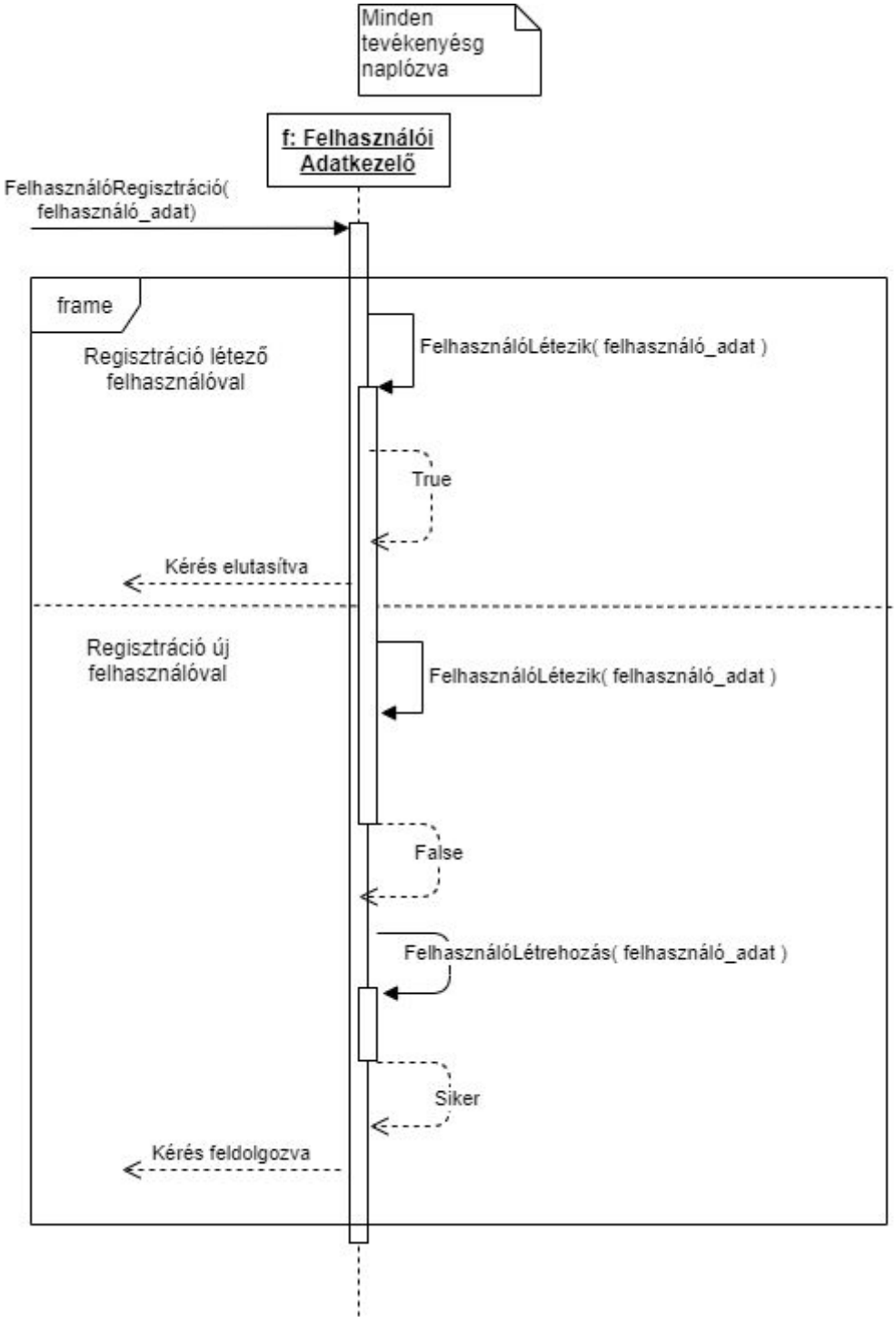
A rendszer viselkedése adminisztrátor általi CAFF fájl módosításakor a 14. ábrán látható. A kérés fogadása után, a CAFF Fájl Kezelő lekérdezi a felhasználó szerepkörét a Felhasználói Adatkezelőtől. Ha a szerepköre Adminisztrátor, akkor a kiválasztott CAFF fájl adatait a kérésnek megfelelően módosítja.

Adminisztrátor általi CAFF fájl törléskor a rendszer viselkedése a 15. ábrán látható. A kérés fogadása után, a CAFF Fájl Kezelő lekérdezi a felhasználó szerepkörét a Felhasználói Adatkezelőtől. Ha a szerepköre Adminisztrátor, akkor a kiválasztott CAFF fájlt törli a rendszerből.

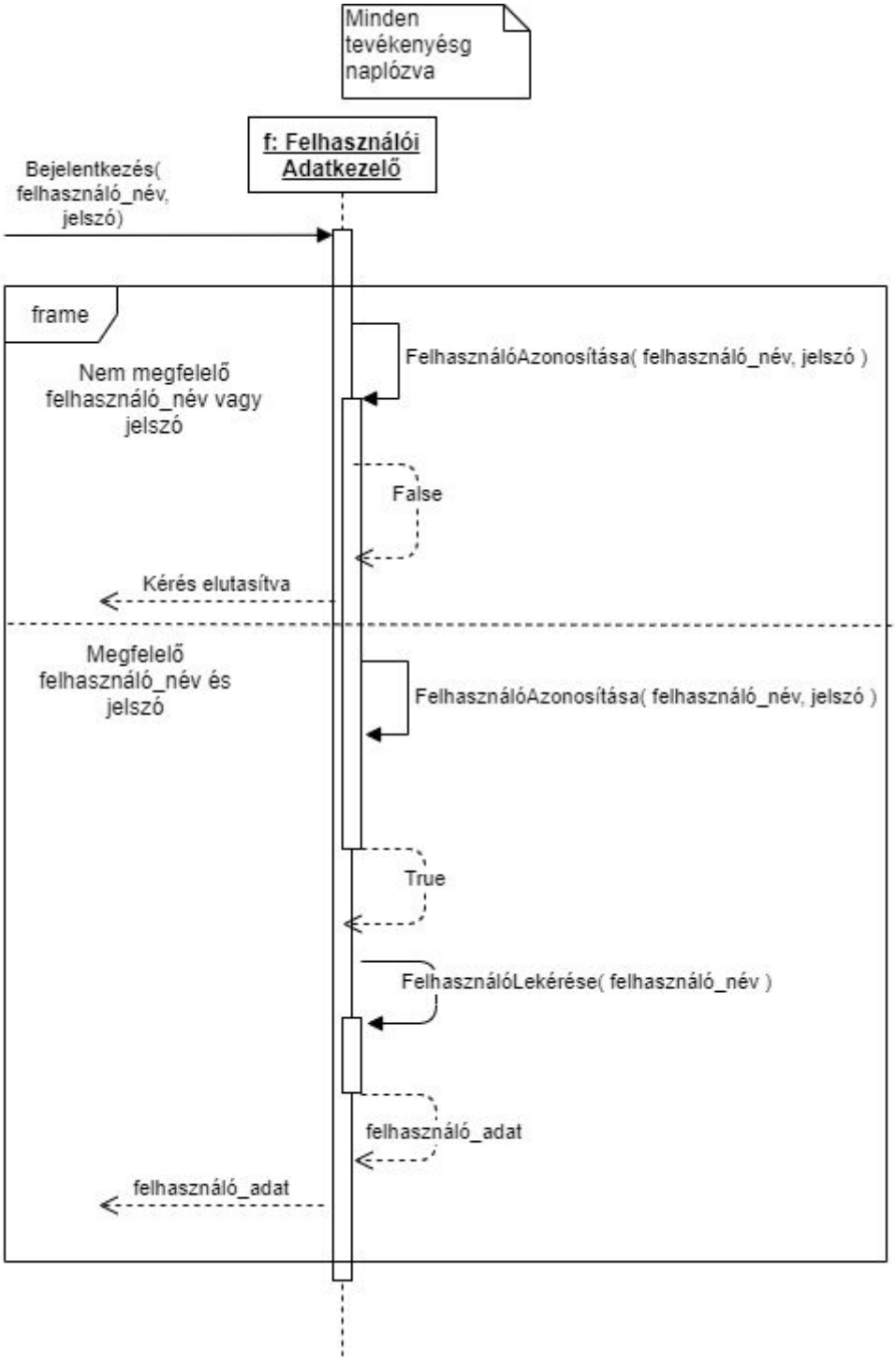
A 16. ábrán látható a rendszer viselkedése adminisztrátor által felhasználó adatainak módosításakor. A kérés fogadása után, a Felhasználói Adatkezelőt ellenőrzi a felhasználó szerepkörét. Ha a szerepköre Adminisztrátor, akkor a paraméterként kapott felhasználó_azonosító által meghatározott felhasználó adatai beállítódnak a szintén paraméterként kapott adat értékének megfelelően.

Adminisztrátor általi felhasználó törlésekor a renszer viselkedése a 17. ábrán látható. Ekkor a Felhasználói Adatkezelő a felhasználó szerepkörét ellenőrzi. Ha a

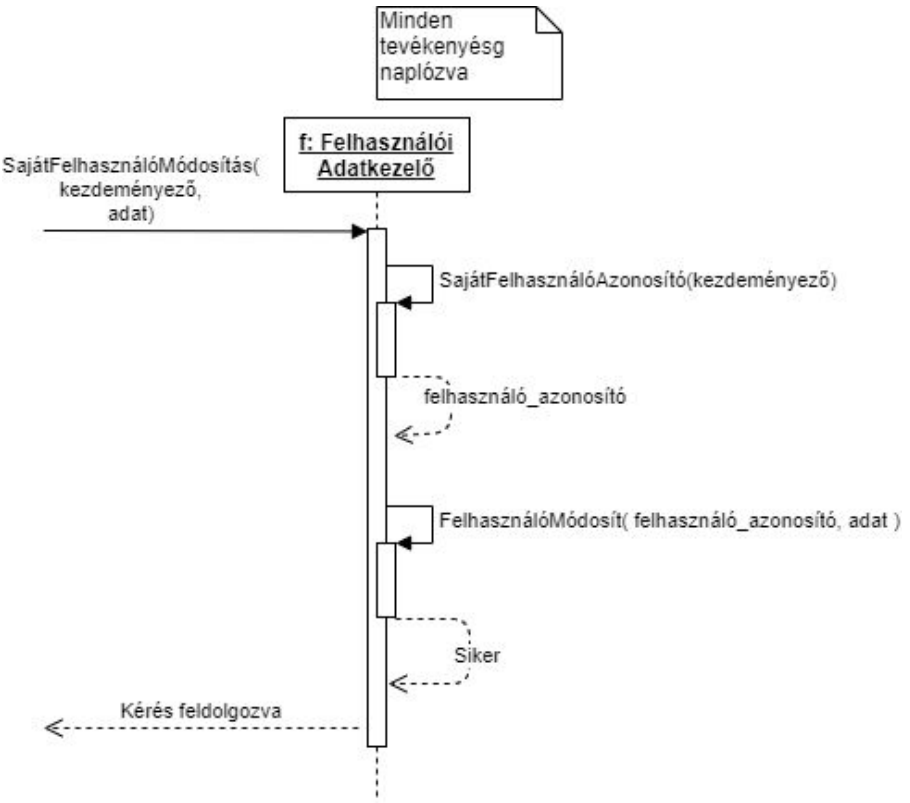
kezdeményező felhasználó Adminisztrátor szerepkörrel rendelkezik, akkor a felhasználó_azonosító által meghatározott felhasználó törlődik a rendszerből.



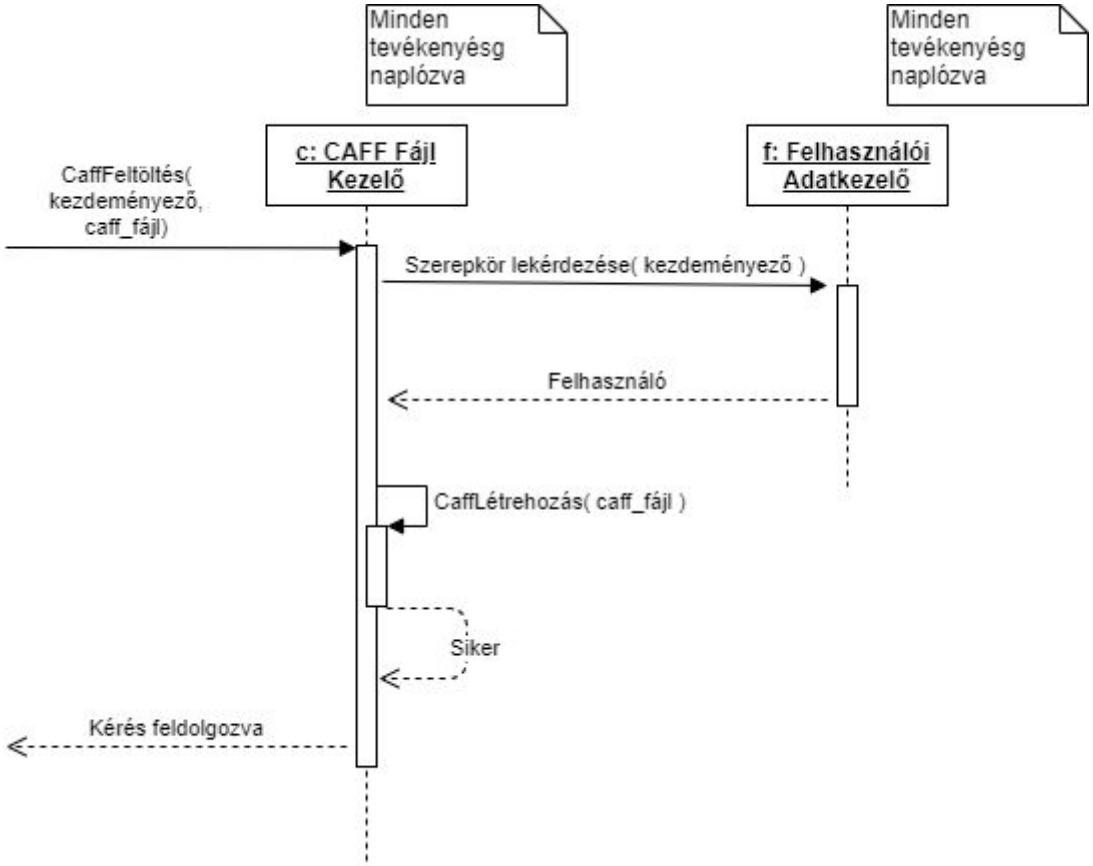
7. ábra: A rendszer viselkedése felhasználó regisztrációkor



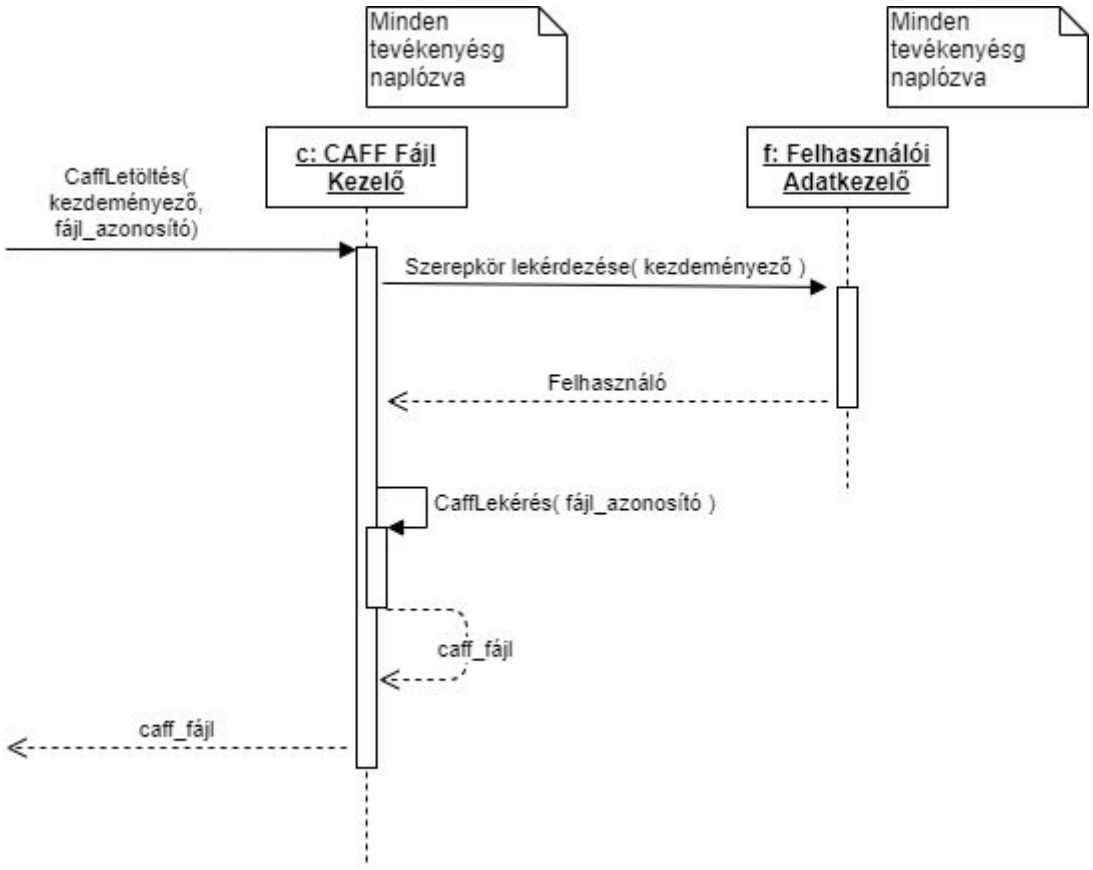
8. ábra: A rendszer viselkedése belépéskor



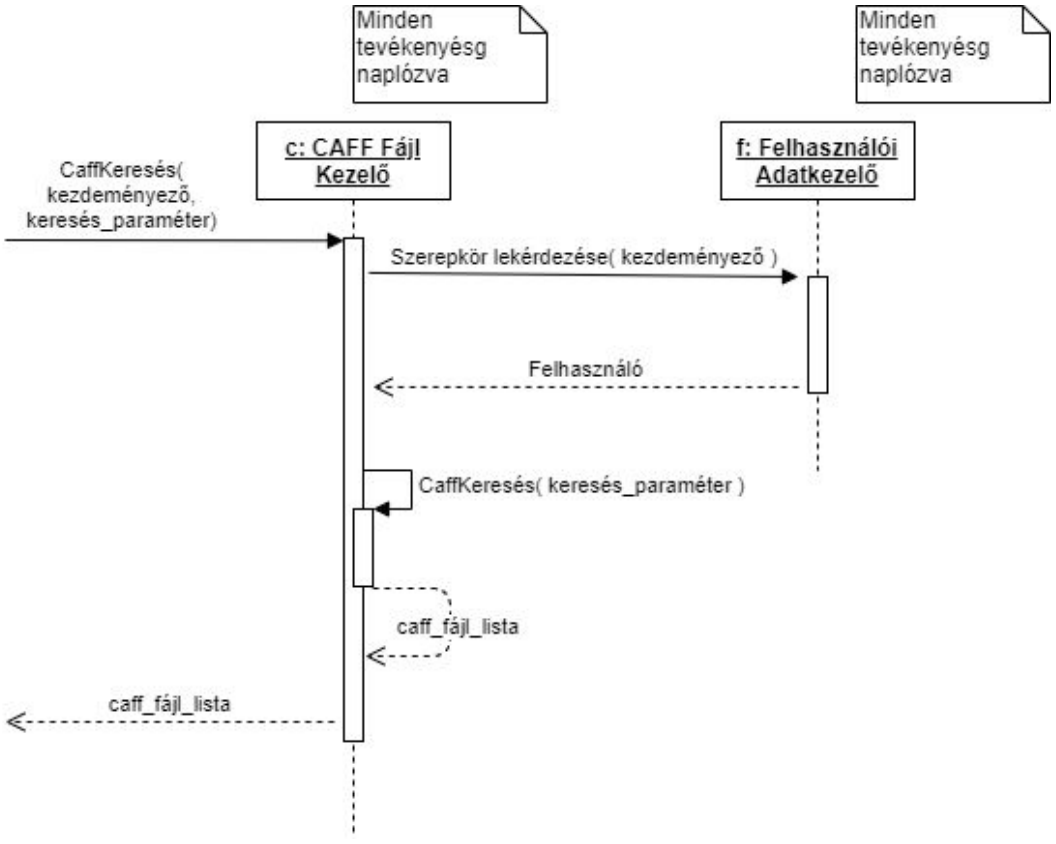
9. ábra: A rendszer viselkedése felhasználó személyes adatainak módosításakor



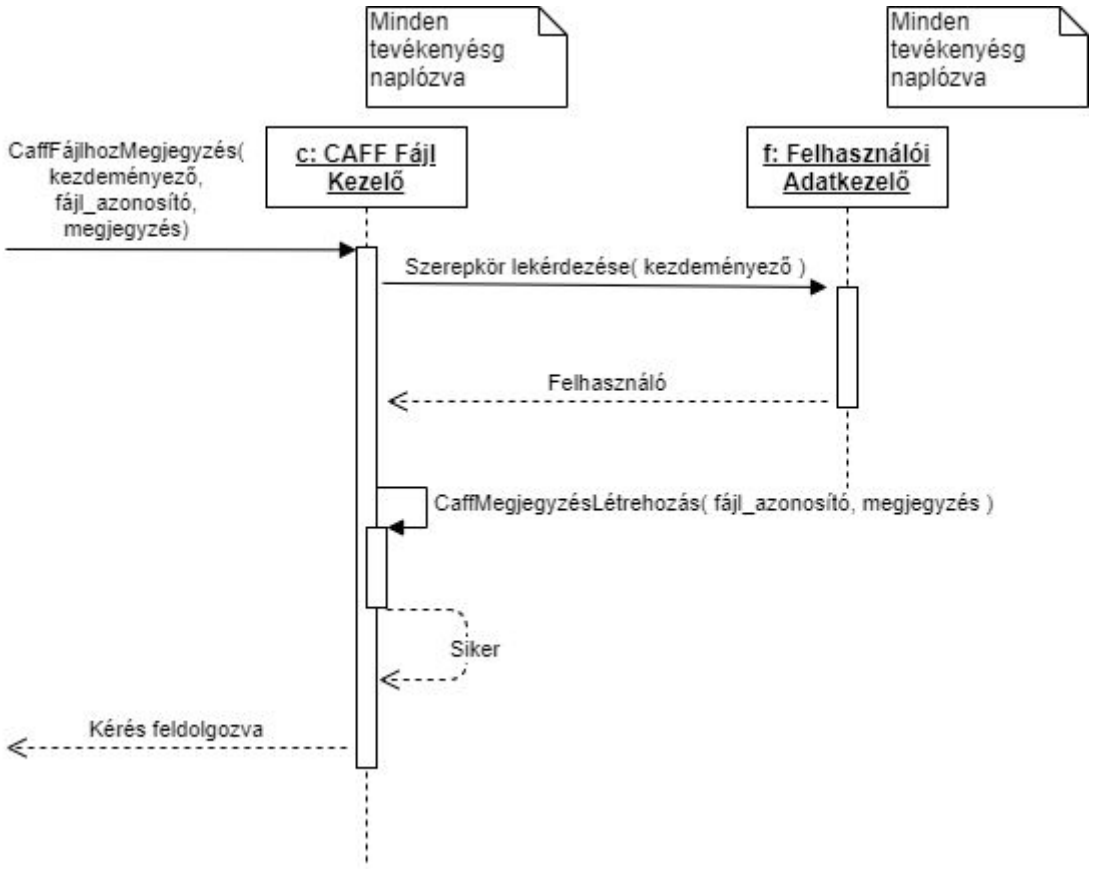
10. ábra: A rendszer viselkedése CAFF fájl feltöltésekor



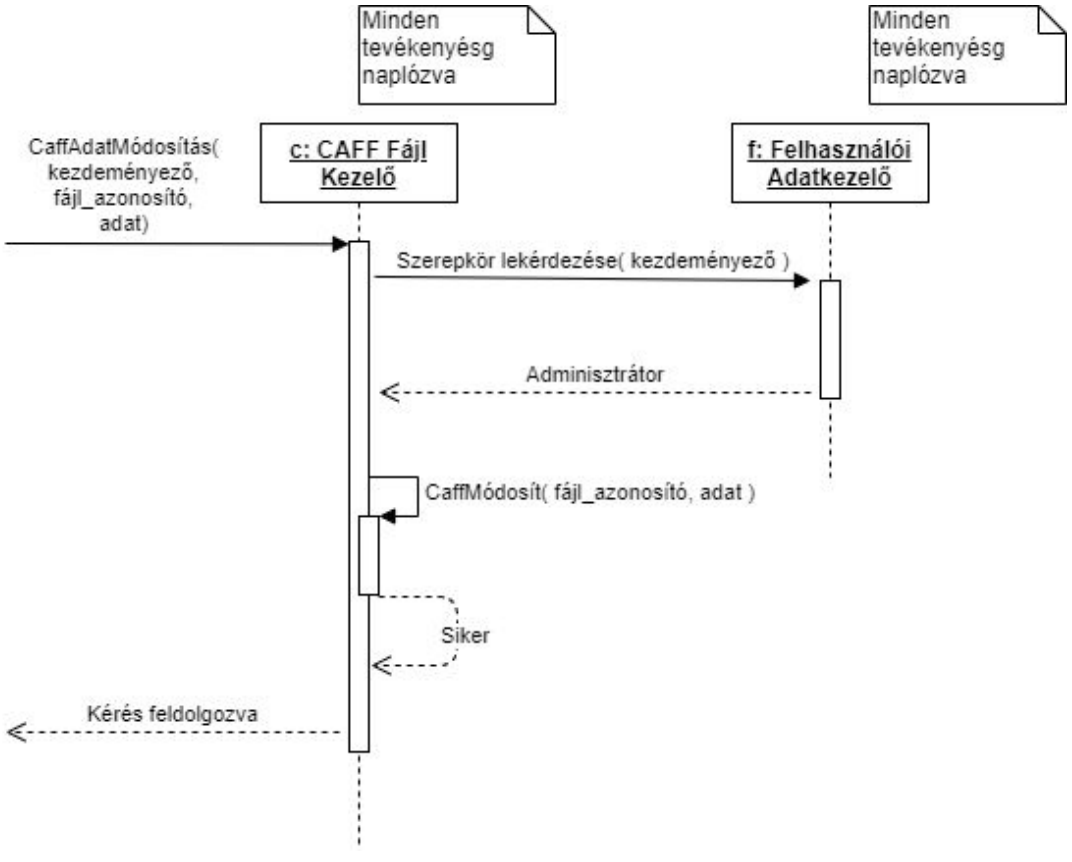
11. ábra: A rendszer viselkedése CAFF fájl letöltésekor



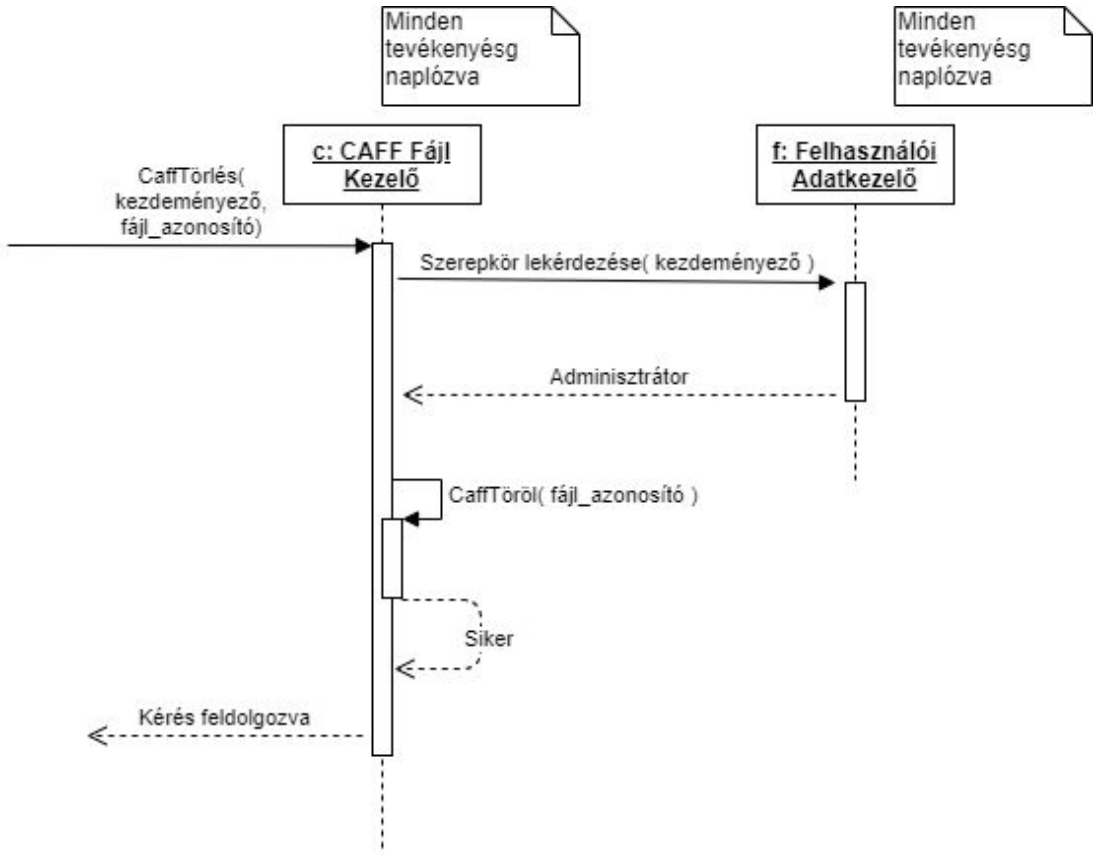
12. ábra: A rendszer viselkedése CAFF fájl keresésekor



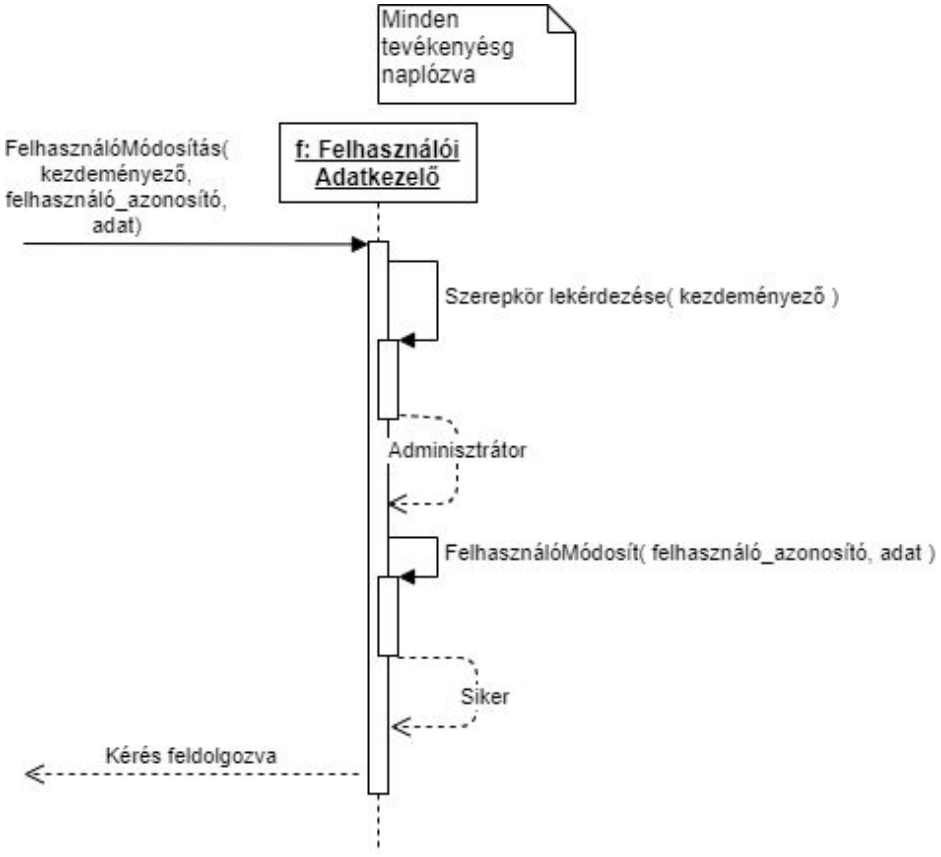
13. ábra: A rendszer viselkedése CAFF fájlhoz fűzött megjegyzéskor



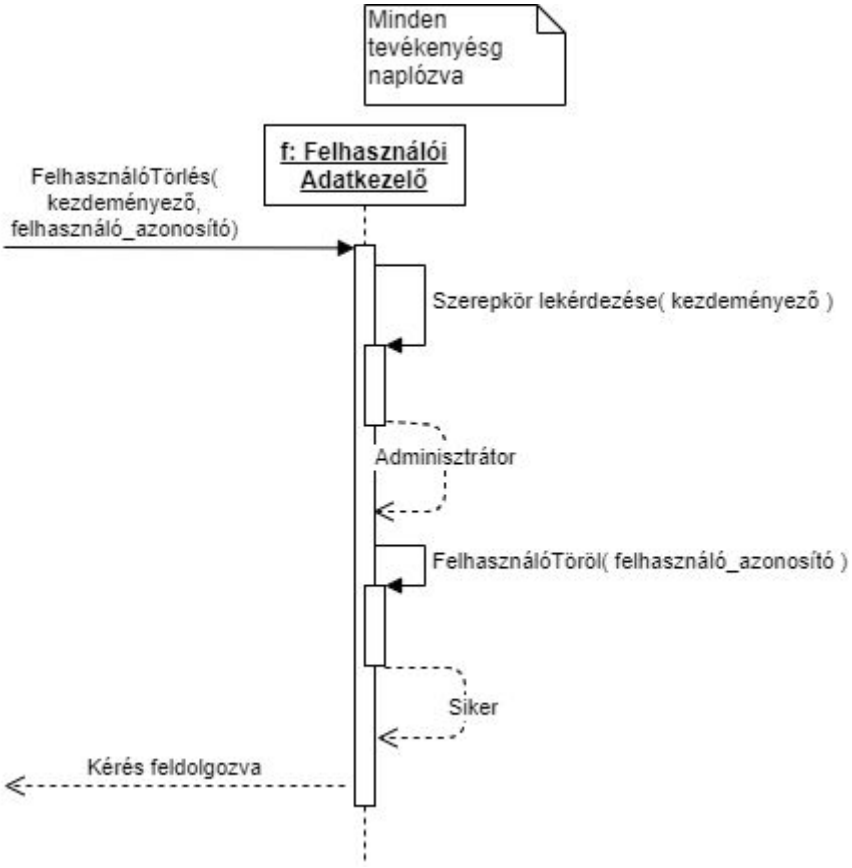
14. ábra: A rendszer viselkedése adminisztrátor által CAFF fájl módosításkor



15. ábra: A rendszer viselkedése adminisztrátor által CAFF fájl törlésekor



16. ábra: A rendszer viselkedése adminisztrátor által felhasználó adatainak módosításakor



17. ábra: A rendszer viselkedése adminisztrátor által felhasználó törlésekor

4. Tesztelési terv

Az alkalmazás megfelelő viselkedését tesztelést segítő eszközökkel és általunk írt unit teszttekkel fogjuk ellenőrizni. A CAFF fájlokat feldolgozó natív C++ komponens tesztelésére az AFL fuzzoló eszközt, és a memória ellenőrző Valgrind eszközt fogjuk használni.

A C# .NET backend és az Angular frontend teszteléséhez unit teszteket fogunk írni, hogy teszteljük a legfontosabb funkciókat.