**Projektmunka II.**

GKNB\_INTM005

**Adorján András Bálint – G6I6ZQ**

**Jánoki Lilla – G4O424**

**Nátz Kornél – JOO2S6**

**Pajor Gergő – BXSGQE**

**Konzulens: Kálóczi Imre**

Győr, 2022

**Tartalom**

[1. Projektterv 1](#_Toc120030812)

[1.1. A probléma és a projektcél rövid felvázolása 1](#_Toc120030813)

[1.2. A program tervezett funkcionalitásai 1](#_Toc120030814)

[1.3. Szoftveres felépítés 1](#_Toc120030815)

[1.4. A projekt során használni kívánt eszközök 2](#_Toc120030816)

[1.5. Mérföldkövek 2](#_Toc120030817)

[1.6. Tevékenységek tervezett kiosztása 3](#_Toc120030818)

[1.7. Célok 3](#_Toc120030819)

[1.8. Közös szótár (Glossary) 4](#_Toc120030820)

[2. A kiválasztott probléma 8](#_Toc120030821)

[2.1. Szöveges bemutatás 8](#_Toc120030823)

[2.2. A probléma folyamatábrája 9](#_Toc120030824)

[3. Üzleti igény specifikáció 11](#_Toc120030825)

[3.1. Célok 11](#_Toc120030827)

[3.2. Stakeholderek 11](#_Toc120030828)

[3.3. Rendszer tervezett működése 12](#_Toc120030829)

[4. Követelményspecifikáció 14](#_Toc120030830)

[4.1. Funkcionális követelmények 14](#_Toc120030832)

[4.1.1. Exif adatok kezelése 14](#_Toc120030833)

[4.1.2. Exif adatok szerkesztése 14](#_Toc120030834)

[4.1.3. Képek/Exif adatok feltöltése/letöltése 15](#_Toc120030835)

[4.2. Nem funkcionális követelmények 15](#_Toc120030841)

[4.2.1. Termék követelmények: 15](#_Toc120030843)

[4.2.2. Szervezeti követelmények: 15](#_Toc120030844)

[4.2.3. Külső követelmények 16](#_Toc120030845)

[4.3. Használhatóság 16](#_Toc120030846)

[4.4. Megbízhatóság 16](#_Toc120030847)

[4.5. Rendelkezésre állás 17](#_Toc120030848)

[4.6. Teljesítmény 17](#_Toc120030849)

[4.7. Támogatottság 17](#_Toc120030850)

[5. Technikai specifikáció 18](#_Toc120030851)

[5.1. Architektúra terv 18](#_Toc120030853)

[5.2. Platform tulajdonságok és követelmények 18](#_Toc120030854)

[5.3. Fejlesztői eszközök 18](#_Toc120030855)

[6. Adatbázis terv 19](#_Toc120030857)

[6.1. Szöveges leírás 19](#_Toc120030858)

[6.2. Táblák 19](#_Toc120030859)

[6.3. Adatmodell 21](#_Toc120030860)

[6.4. Egyed-kapcsolat diagram 22](#_Toc120030861)

[6.5. Kulcsok és megszorítások 22](#_Toc120030862)

[6.5.1. Elsődleges kulcsok 22](#_Toc120030870)

[6.5.2. Másodlagos kulcsok 22](#_Toc120030871)

[6.5.3. Megszorítások 22](#_Toc120030872)

[6.6. Tárolt eljárások 23](#_Toc120030873)

[6.7. Továbbfejlesztési ötletek 24](#_Toc120030874)

# Projektterv

## A probléma és a projektcél rövid felvázolása

Csapatunk célja a fotográfusok munkájának segítése a fényképek kezelésének megkönnyítésével. Csapatunk fotográfus tagja és más szakmabeliek is tapasztaltak több különböző, ehhez a témához kapcsolódó problémát. Ezek közé tartozik, hogy a fényképezőgépek nem rögzítenek megfelelő mennyiségű exif adatot; manuális objektívekből egyáltalán nem tudnak adatokat kinyerni; bizonyos információk (pl. szerzői jog és megjegyzés) jóval korlátozottabb formában rögzíthetőek a fényképezőgép segítségével, mint amit a fájlok megengednének; illetve az is, hogy különböző programok más-más exif adattípusokkal kompatibilisek és a kompatibilis adatokat is különböző prioritással kezelik. Ezen problémákat szeretnénk feldolgozni és megoldási javaslatot tenni rájuk egy általunk készített program formájában.

## A program tervezett funkcionalitásai

A programnak képesnek kell lennie az exif adatok kezelésére, tárolására és szerkesztésére. A magasszintű kompatibilitás miatt több gyártó hardveres és szoftveres termékeit és az ezek által használt exif adattípusokat is ismernie kell. A program funkcionalitásai között lesz az exif adat szerkesztés, illetve az exif adatok szerinti szűrés és rendezés.

## Szoftveres felépítés

Megoldásunk egy webes felhasználói kliensből és egy távoli elérésű adatbázis szerverből fog állni. A tervezés során fontos szempont volt a biztonság, ezért megfontoltuk a háromrétegű architektúra kialakítását is, de végül elégségesnek találtuk a Google Firebase titkosítását, így a kétrétegű architektúra mellett döntöttük.

## A projekt során használni kívánt eszközök

Mivel webes alkalmazást fejlesztünk, így a fejlesztés fő eszközei a HTML, CSS és JavaScript nyelvek lesznek. Azon feladatok elvégzésére, amelyre ezek nem alkalmasak vagy célszerűek, C#-ot fogunk használni, de szükség esetén más nyelveket is bevonunk.

Adatbázisnak a korábban is említett Google Firebase rendszert fogjuk használni.

Elsődleges fejlesztési környezetünk a Visual Studio Code lesz.

A projektkezelés támogatására a Trello és a GitHub platformokat fogjuk segítségül hívni.

A chat alapú kommunikációra elsődlegesen az erre a célra létrehozott Messenger csoportban, a megbeszélések pedig Discord-on fognak történi. Amennyiben szükséges személyes találkozókat is szervezünk és más kommunikációs platformokat (pl. Google Meet, Zoom) is bevonunk.

## Mérföldkövek

1. Projektterv dokumentáció.
2. GUI-k tervezése, megvalósítása.
3. GUI-k funkcionálissá tétele.
4. Backend megvalósítása.
5. Rétegek összekapcsolása.
6. Tesztelés, felmerülő hibák kijavítása.

Első feladatunk a projekttervezet részletes, a tantárgyi követelményeknek megfelelő dokumentációja lesz. Ez elengedhetetlen a tantárgy teljesítéséhez, de a projekt végig vezetése során is jó kiindulási alapot fog adni a fejlesztő csapatnak. Ez természetesen nem fogja kitenni a projekt teljes dokumentációját. Azt a projekt során minden hallgató bővíteni fogja és végleges verziója csak a projekt lezárásával fog elkészülni. Második lépés, a felhasználói felületek grafikus megtervezése, majd ezek megvalósítása lesz. Ezután a felhasználói felületek, azaz a frontend oldal funkcionálissá tétele következik. Természetesen a külön funkciók és programrészletek tesztelése is elengedhetetlen, erre is sort fogunk keríteni a fejlesztés során. A negyedik lépés a backend megvalósítása, az adatbázis létrehozása és testre szabása. Amikor ezek mind megvannak, akkor létre lehet hozni a már működő egységek közti kommunikációs interfészeket az együttműködés érdekében. A projektet a program teljeskörű tesztelésével és az esetlegesen felmerülő hibák kijavításával fogjuk zárni.

## Tevékenységek tervezett kiosztása

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Adorján András Bálint | Jánoki Lilla | Nátz Kornél | Pajor Gergő |
| Dokumentáció | X | X | X | X |
| Megbeszélések koordinálása | X |  |  |  |
| Kapcsolattartás | X | X |  |  |
| Folyamatábrák készítése | X | X |  | X |
| GUI-k készítése | X |  | X | X |
| Adatbáziskezelés |  |  | X | X |
| Programozás | X | X | X | X |

A dokumentáció a közös munka és a tantárgy értékelése szempontjából is rendkívül fontos része projektünknek, így mindegyik csapattag foglalkozni fog vele. Ugyanígy kitüntetett szerepet fog kapni a programozás.

A projekt során a csapattagok kommunikálni fognak egymással a korábban említett csatornákon, viszont az egyéb kommunikációval kapcsolatos feladatokat alapvetően a projektvezető fogja ellátni. Ide tartozhat például a megbeszélések koordinálása vagy kapcsolattartás a konzulenssel és esetleges harmadik felekkel.

A folyamatábrák és grafikai elemek elkészítését vegyesen osztottuk fel a csoporttagok között, az adatbázisok készítésére, kezelésére pedig az ebben tapasztaltabb csapattagokat jelöltük ki.

## Célok

Célunk, hogy elkészüljön egy olyan webalapú szoftver, ami nagyban hozzá tud járulni a fotográfusok munkájához a fényképek kezelésének megkönnyítésével.

Reméljük, hogy a projekt elkészítése hozzá fog járulni szakmai fejlődésünkhöz és a konzulensünk is elégedett lesz munkánkkal.

## Közös szótár (Glossary)

**EXIF adat**

Az Exif olyan információk gyűjtőneve, amit jellemzően a digitális fényképezőgépek használnak és az általuk készített képfájlban tárolnak. Célja a fénykép módosítása nélkül olyan információk elhelyezése a képfájlban digitális formátumban, amik a kép készítésének körülményeit írják le.

**Kamera váz**

A fényképező gép teste az objektív és az egyéb kiegészítők nélkül.

**Objektív**

A fotográfiában az objektív a tárgyról valódi képet rajzoló optikai rendszer, mely egy vagy több optikai lencséből, illetve egyes objektívek esetében optikai lencsékből és tükrökből épül fel. Az emberi szemhez hasonlóan működik, a látószögébe eső fényt összegyűjti, majd az optikai tengelyére merőleges érzékeny anyagra (fotófilm, érzékelő lapka) vetíti.

**Érzékeny anyag**

A fényképek rögzítését lehetővé tevő felület. Régebbi fényképezőkben fényérzékeny film vagy lemez, digitális eszközökben egy elektronikus szenzor.

**Gyújtótávolság**

Az objektív frontlencséjének és az érzékeny anyagnak mm-ben mért távolsága. Ha ez a távolság valamilyen mechanikai módon változtatható, akkor zoom-objektívről (zoom-os objektívről), ha nem, akkor fix (gyújtótávolságú) objektívről beszélünk.

**Frontlencse**

Az objektív első, külső optikai lencséje.

**Vintage objektív**

Az objektívek egy speciális fajtája. Fő ismérvük, hogy semmilyen elektromos, illetve elektromosság által vezérelt mechanikus rendszert nem tartalmaznak. Emiatt használatuk, a fókuszálás és a rekesz állítás is teljes mértékben manuális. Általában nagy fényerővel és fix gyújtótávolsággal rendelkeznek, de léteznek zoom-os példányok is.

**Fókuszálás / fókusz állítás**

Ennek segítségével tudjuk beállítani, hogy az elkészült kép mely részei legyenek élesek, hangsúlyosak. Történhet kézzel és szoftveres automatizmusok segítségével is (pl. arcfelismerés).

**Rekeszelés / rekesz állítás**

Az objektívekben a lencserendszer fősíkjának közelébe fényrekeszt (vagy más néven blendét) építenek be. A rekesz segítségével lehet szabályozni a lencsén áthaladó és a gép belsejébe jutó fény mennyiségét, ezzel befolyásolva a kép elkészítéséhez szükséges expozíciós idő hosszát. A legtöbb objektívben ún. íriszrekesz található, amely 5-20 félkör alakú vékony fém- vagy műanyaglemezből (lamellából) áll. Ezen lamellák által kialakított lyuk tágítása, szűkítése a rekeszelés. Minél tágabb a rekesz, annál több fény tud keresztül jutni rajta, minél szűkebb, annál kevesebb.

**Objektív rekeszértéke**

Minél tágabb az objektív belsejében lévő fizikai rekesz, annál kisebb a rekeszértéke, fordítottan pedig minél szűkebb a fizikai rekesz, annál nagyobb a rekeszérték. Az objektívek rekeszértéke lehet fix és változó.

**Objektív fényereje**

Az objektív fényerejének a lamellák által kialakítható legtágabb rekeszértéket nevezik. Fényerős objektívnek az F/2.8 vagy annál kisebb rekeszértékkel rendelkező objektíveket szokták hívni.

**Expozíció**

A fénykép elkészülésének pillanata.

**Stakeholder**

Az üzleti igényspecifikáció a résztvevője.

**Nyers fénykép**

A fénykép módosítás mentes, feldolgozatlan verziója.

**Vaku**

A vaku vagy villanófény egy fényképezésnél használt mesterséges fényforrás, ami az exponálás ideje alatt rövid időre erős fénnyel világítja meg a fényképezni kívánt tárgyat. Használatával jobban megvilágított kép, esetleg különleges képhatások érhetők el.

**Elsődleges kulcs**

Egy relációs adatbázis használatakor minden táblában meg kell jelölni, hogy melyik mező, vagy melyik mezők együttesen az elsődleges kulcsok. Az elsődleges kulcs minden rekordban egyedi.

**Idegen (másodlagos) kulcs**

Az idegen kulcs olyan oszlop, amelyet a másik táblázatban elsődleges kulcsnak neveznek, azaz a táblázat elsődleges kulcsa idegen kulcsnak nevezhető egy másik táblázatban. Lehet, hogy az idegen kulcs duplikált & NULL értékeket tartalmaz, ha a NULL értékeket elfogadják.

**Adatmodell**

Az adatmodell a logikai adatbázis szerkezeti leírását foglalja magába, nevezetesen azonos jellemzőjű információk logikai modellezését a rajtuk végezhető logikai műveletek meghatározását tartalmazza. Az adatmodell tehát mindig szerkezeti és műveleti részből tevődik össze. Az adatmodell feladata, hogy a világban található dolgokról, individuumokról számítógéppel könnyen feldolgozható formálisan leírható adatok tárolásához megfelelő szerkezetet, keretet adjon, illetve ezek lekérdezhetőségét, visszakeresését is biztosítani tudja zárt matematikai alakban.

**Egyed-kapcsolat diagram**

Az egyed-kapcsolati modell (vagy ER-modell) egy adott tudásterületen belüli, egymással összefüggő dolgokat ír le. Egy alapvető ER-modell egyed-típusokból áll (amelyek osztályozzák az érdekes dolgokat), és meghatározza az egyedek (ezen egyed-típusok példányai) között létező kapcsolatokat. A szoftverfejlesztésben az ER-modellt általában azért alkotják meg, hogy ábrázolja azokat a dolgokat, amelyekre az üzleti folyamatok végrehajtásához egy vállalkozásnak emlékeznie kell. Következésképpen az ER-modell egy absztrakt adatmodellé válik, amely egy olyan adat- vagy információs struktúrát határoz meg, amely egy adatbázisban, jellemzően egy relációs adatbázisban implementálható.

**Tárolt eljárás**

A tárolt eljárás a strukturált lekérdezési nyelv (SQL) utasítások gyűjteménye a hozzá rendelt névvel. Ezeket a tárolt eljárásokat egy relációs adatbázis-kezelő rendszerben (RDBMS) tárolják. Tehát lehetséges az eljárás többszörös meghívása, amely csökkenti a végrehajtási időt. Az adatok eljárásokkal módosíthatók, és a fő előnye az, hogy nem kapcsolódnak egy adott alkalmazáshoz.

# A kiválasztott probléma



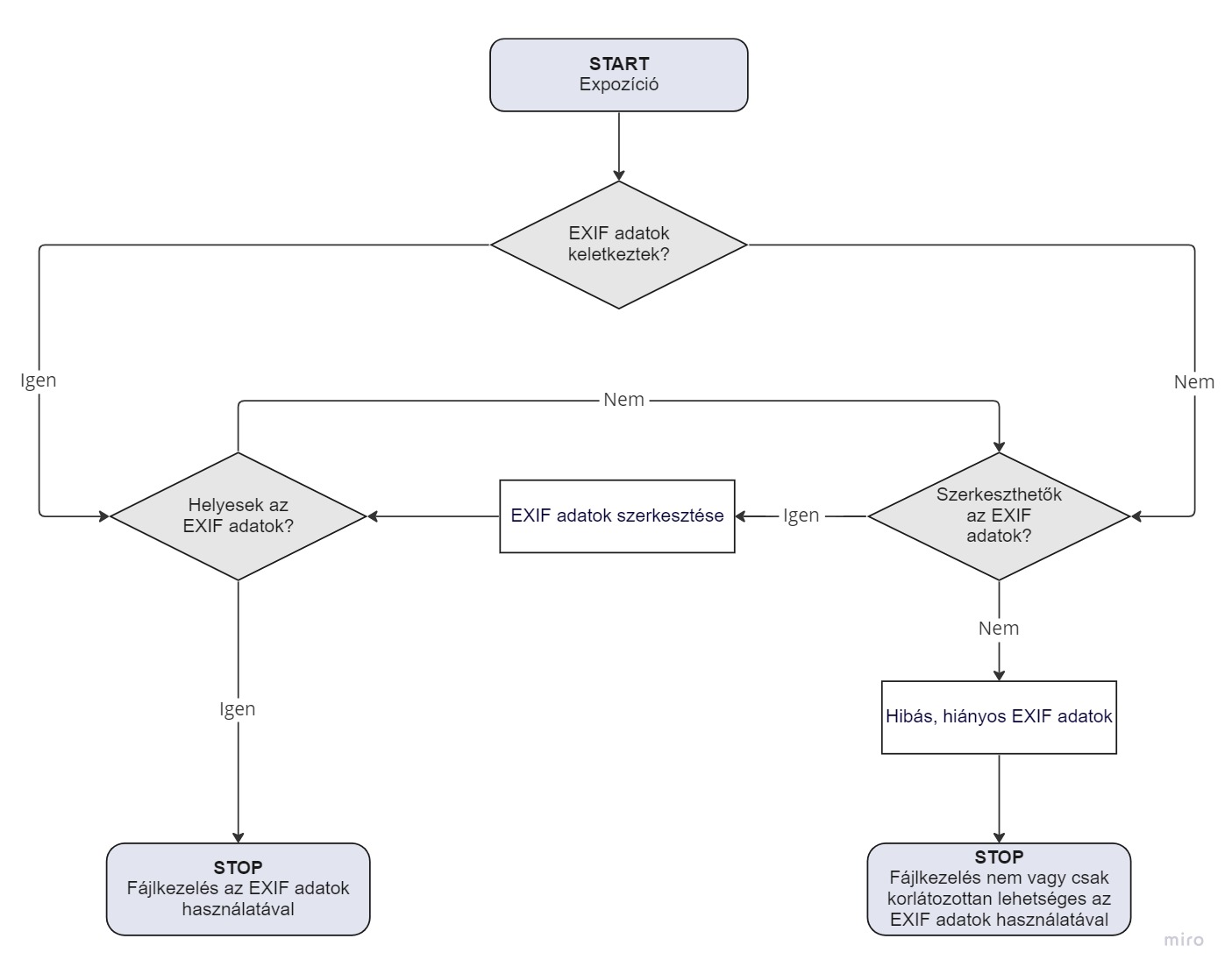
## A probléma szöveges bemutatás

A projektünk a fényképészeket/fotográfusokat célozza meg. A fényképek elkészülésekor, egészen pontosan az exponáló gomb megnyomásakor, legyen az telefonnal vagy fényképezőgéppel készítve EXIF adatok készülnek. Ezek az EXIF adatok különböző paramétereket tartalmaznak az elkészült fényképről, legyenek azok az objektívhez, fényképezőgép vázhoz, vagy vakuhoz kapcsolódó adatok. Így képszerkesztéskor könnyen kiolvashatjuk a kép EXIF adatai közül pl., hogy hány mm-es gyújtótávolságú objektívvel készült el a kép.

Az EXIF adatok, illetve azok felhasználása azonban ennyiben ki is merül a legtöbb esetben. Jó lenne egy olyan alkalmazás/applikáció, amely azt célozza meg, hogy elkészült képeinket egy böngészhető adatbázisba foglalja és azok között az EXIF adatok szerint különböző műveleteket tudjunk végezni. Ilyen műveletek lennének az egyes adatok szerinti szűrés, sorba rendezés, megjelenítés, mind-mind a kereshetőségre funkcionálva.

További probléma, hogy bizonyos fényképezőgép vázak és objektívek (pl. régi vintage objektívek) bajonett átalakító esetén nem rögzítenek egy képhez minden fontos EXIF adatot. Ezt kiküszöbölő megoldás, ha az alkalmazásban lehetőség lenne a már meglévő EXIF adatokat szerkeszteni, a hiányzók esetében pedig új adatokat hozzáadni, esetlegesen törölni.

## A probléma folyamatábrája



**1. ábra - A probléma folyamatábrája**

Az 1. ábrán az expozíció (START) és a fájlkezelés (STOP) közti lehetséges kimentelek láthatók. A kiindulópont minden esetben az expozíció, hiszen az EXIF adatok kezeléséhez először létre kell jönnie egy fényképnek, melyen a későbbiekben dolgozhatunk. A folyamat célja az, hogy a fájlkezelés EXIF adatok használatával történhessen, ám ez a jelen rendszerben nem minden esetben lehetséges, esetleg nem a legjobb minőségben történik.

Kezdetben meg kell vizsgálnunk, hogy egyáltalán keletkeztek-e EXIF adatok az expozíció során.

Amennyiben keletkeztek, meg kell állapítanunk, hogy ezen adatok helyesek-e, amennyiben igen, természetesen a folyamat lezártnak is tekinthető, hiszen képesek vagyunk a fájlkezelésre a megfelelő EXIF adatok használatával. A helyzet azonban a legtöbb esetben nem ilyen egyszerű. Amennyiben ezen adatok ugyan létrejöttek, de nem megfelelők/helyesek, akkor meg kell néznünk, hogy szerkeszthetők-e. Ugyanígy kell tennünk, amennyiben automatikusan nem keletkeztek EXIF adatok. Ha az adatok szerkeszthetők, visszajutunk a „Helyesek az EXIF adatok?” kérdéshez, innen pedig le is zárhatjuk a folyamatot vagy amennyiben még maradt hiba az adatokban, ismételt javítással korrigálhatjuk hibáinkat.

Amennyiben az adatok nem szerkeszthetők, ám nem helyesek vagy hiányosak, akkor sajnos nincs más lehetőségünk, ezen adatokkal kell tovább mennünk a folyamatban és a „Fájlkezelés nem, vagy csak korlátozottan lehetséges EXIF adatok használatával” pontba érkezünk, mely korántsem ideális. Megoldásunkkal ezt az állapotot szeretnénk elkerülhetővé tenni.

# Üzleti igény specifikáció



## Célok

A program elsődleges célja a fotográfusok munkájának segítése a fényképek kezelésének megkönnyítésével. A program képes az exif adatok kezelésére, tárolására és szerkesztésére. A kompatibilitás miatt több gyártó hardveres és szoftveres termékeit és az ezek által használt exif adattípusokat is ismeri. A program funkcionalitásai között megtalálható az exif adat szerkesztés, illetve az exif adatok szerinti szűrés és rendezés.

## Stakeholderek

Projektünk elsődleges belső érdekeltjei az applikáció fejlesztői, akik elkészítik és a továbbiakban is karban tudják majd tartani, fejleszteni az applikációt. Mellettük belső érdekeltek lehetnek még az esetleges befektetők. A külső érdekeltek körébe tartoznak a fényképészek, akik igénybe szeretnék majd venni a szolgáltatást.

**Belső érdekeltek**

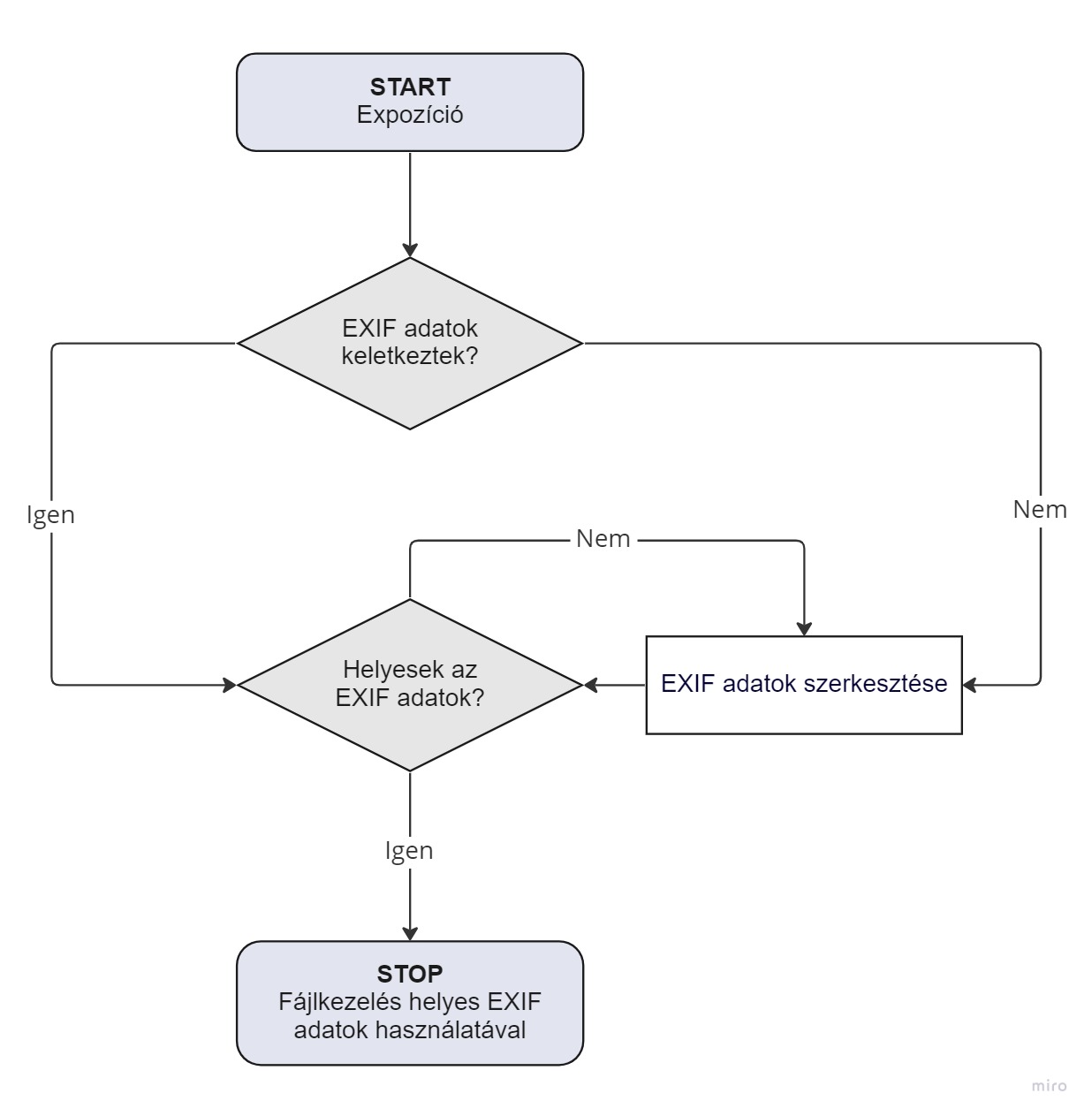
A fejlesztők nélkül nem jöhetne létre és nem is működhetne az applikáció. Feladatuk megtervezni a célnak megfelelően működő alkalmazást. Későbbiekben is szükség lesz a munkájukra a program karbantartása és esetleges fejlesztések kivitelezése kapcsán.

A befektetők kulcsszerepet játszhatnak az applikáció létrehozásának, fenntartásának és továbbfejlesztésének finanszírozásában. Minél népszerűbb lesz az alkalmazás, annál több emberhez juthat el és annál több fotós veheti igénybe a szolgáltatásokat. Később az alkalmazást ki is lehet bővíteni és további szolgáltatásokat biztosítani.

**Külső érdekeltek**

A fényképészek érdeke, hogy egy fotózás után a lehető leggyorsabban elérhetővé váljanak az általa készített képek. A program segítségével lényegesen gyorsabban lehet az elkészült képeket kezelni, és az azokhoz tartozó adatokat módosítani.

## A rendszer tervezett működése

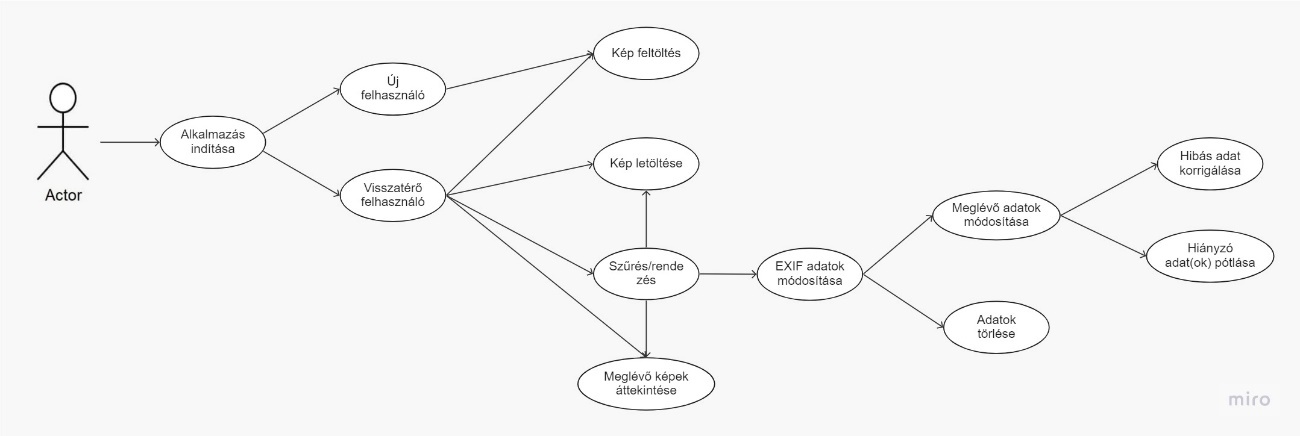


**2. ábra - A rendszer tervezett működésének folyamatábrája**

A 2. ábrán látható a projekt által kínált megoldással való fájlkezelés folyamata. A korábbiakhoz hasonlóan expozícióval (START) indul a folyamat, mely során EXIF adatok keletkezhetnek. Innen kiindulva meg kell vizsgálni, hogy ténylegesen keletkeztek-e ilyen adatok. Amennyiben keletkeztek, fontos szempont, hogy helyesek-e, tehát módosításra szorulnak-e. Amennyiben helyes adatok keletkeztek, nincs is további teendő eljutunk a fájlkezelés helyes EXIF adatok használatával (STOP) mezőhöz és a folyamat lezártnak tekinthető.

A program szempontjából érdekesebb ág, mikor a keletkezett adatok hibásak, hiányosak, esetleg egyáltalán nem is jöttek létre. Ebben az esetben szükségessé válik ezen mezők szerkesztése. A korábbi ábrával ellentétben itt a programnak köszönhetően ezen hiányzó/hibás elemek minden esetben korrigálhatók. A megalkotott rendszernek köszönhetően elkerülhető, egy a probléma folyamatábráján még jelen lévő végkimenet, melyben a fájlkezelés hibás adatokkal kell, hogy történjen, mely egy korántsem ideális eshetőség. Az ábra alapján belátható, hogy az elkészült applikáció nagyban megkönnyíti a fájlkezelést és jóval kényelmesebbé válik a fotósok munkája, minden esetben elérhető a helyes EXIF adatokkal történő fájlkezelés.

Ugyanezen folyamat látható a 3. ábrán Use-case diagram formájában.



**3. ábra - A rendszer tervezett működésénem Use-case diagramja**

# Követelményspecifikáció



## Funkcionális követelmények

## Exif adatok kezelése

**Előfeltétel:**

* Nyers fénykép feltöltése
* Internet elérés.

**Folyamat:**

A folyamat akkor indul el, amikor felhasználó belép a rendszerbe. A kezelő felületen keresztül képes adott exif adatokra keresni, az adatokat szűrni és rendezni.

**Hibák:**

* Nincs internet kapcsolat
* Nem működik az Applikáció által használt szerver/adatbázis

## Exif adatok szerkesztése

**Előfeltétel:**

* Nyers kép feltöltése az adatbázisba
* Internet elérés

**Folyamat:**

A folyamat akkor indul el, amikor a felhasználó a kezelőfelületen keresztül módosítani szeretné a képhez tartozó adatokat. A módosítások végrehajtása utána a változtatások mentésre kerülnek

**Hibák:**

* Az exif adat írásvédett
* Nem működik az Applikáció által használt szerver/adatbázis

## Képek/Exif adatok feltöltése/letöltése

**Előfeltétel:**

* Internet elérés

**Folyamat:**

A folyamat akkor indul el, amikor a felhasználó fájlokat szeretni feltölteni, vagy adott esetben letölteni. A kezelő felületen keresztül lehetőség van egy, vagy akár több kép feltöltésére is. Ugyanakkor a felhasználó is le is tudja tölteni az adatbázisban tárolt képeit

**Hibák:**

* Nincs internet
* Megszakad a kapcsolat
* Nem működik az Applikáció által használt szerver/adatbázis



## Nem funkcionális követelmények



### Termék követelmények:

**Teljesítmény követelmények:**

* Applikációt futtatni képes eszköz (mobiltelefon)
* Megfelelően optimalizált alkalmazás

**Használhatósági követelmények:**

* Internetkapcsolat
* Applikáció előre feltelepítése

### Szervezeti követelmények:

**Implementációs követelmények:**

* Applikáció univerzális használhatósága.
* Jelenlegi operációsrendszerekkel kompatibilis applikáció tervezése.
* Elfogadható válaszidő.

### Külső követelmények

Abc

## Használhatóság

Egy gyakorlott felhasználó könnyedén felfedezheti az alkalmazást, figyelembe véve a hasonló szoftverek felhasználói felületét. Vannak felhasználók, akik a menüket és vannak, akik az ikonokat részesítik előnyben. A két grafikai kialakítást egybeolvasztva lenne megvalósítva az app.

Egy kezdő felhasználó számára az eszközrendszere több helyről elérhető és segítségükre van a súgó rendszer is, ami a megfelelő ikonra (egy kérdőjelre) kattintva megtalálható.

Egy, a felhasználók kérdései alapján felépített online súgó és egy fórum, amely megteremtheti a folyamatos kapcsolatot a készítőkkel, nem csak a használhatóságot, hanem a későbbi továbbfejlesztést is megkönnyítheti.

A felület letisztult, könnyen használható és átlátható legyen. A rendszer legyen látványos és figyelemfelkeltő a kellő minimalizmussal.

Az alkalmazás jól optimalizált, nem foglal a kelleténél több helyet, és nem igényel a kelleténél több nyersanyagforrást (CPU, RAM, GPU).

Felesleges adatokat ne tároljunk.

## Megbízhatóság

**Biztonság:** Fontos, hogy minden felhasználó csak ahhoz a részhez férhessen hozzá, amihez jogosultsága van. Hiszen nem engedhetjük meg, hogy egy egyszerű felhasználó mások által feltöltött fájlokat töröljön, vagy egy alkalmazás karbantartó hozzáférjen a fényképészek adataihoz. Ezért kulcskérdés a felhasználók megfelelő szerepkörbe való osztása mely az adminisztrátor feladata. Valamint egy olyan azonosítási procedúrával kell rendelkeznie a rendszernek mely megkerülhetetlen, és egyértelműen azonosítja a felhasználót.

**Karbantarthatóság:** Az esetleges hibák fellépésekor, akár a tesztelés, akár a felhasználói használat során (amennyiben ezt jelzik a fejlesztők felé) a hiba detektálása és javítása azonnal megkezdődik. A hiba súlyosságától függően elképzelhető, hogy a rendszert le kell kapcsolni a hibajavítás idejére. Kisebb hibák esetén elég egy csak a hibás fájlok kicserélésének idejére leállítani az alkalmazást futtató webszervert.

**Hordozhatóság:** A szoftvert (kliens oldali) alkalmazóknak lehetősége van operációs rendszert, vagy platformot váltani, ezt ugyanis a platformfüggetlen fejlesztés lehetővé teszi.

**Hibakezelés:** Az alkalmazás szabályos használata esetén nem fordul elő hibajelenség, nem jelenik meg hibaüzenet, minden szakszerűen működik. Amennyiben a felhasználó hibás adatot ad meg, a program azt megfelelően kezeli, és biztosítja annak javítását.

**Jogszabályok:** Jogszabályok betartása. A felhasználó adatainak biztonságának megőrzése

## Rendelkezésre állás

A rendszer alapvetően a munkában elterjedt idősávokban vannak a legmagasabb kihasználtsági csúcsok, „office hour” vagy akár a 2 műszakos idősávok alatt. Ezen időszakokban lehet a legnagyobb rendszer terhelésre számítani.

A rendszer felügyeletét reggel 8 órától este 10ig biztosítani szükséges, az ezen kívül eső intervallumon csak a következő napra lehet a problémákat egy automatikus hangrögzítő segítségével leadni.

**MTBF (Mean Time Between Failure):** A rendszer kritikus követelménye, hogy az esetleges hibák/leállások között a lehető legtöbb idő teljen el, azaz a lehetőleg minimálisnak kell lennie a felléphető hibák számának, melyek a működést befolyásolják. Statisztikák alapján egy leállás átlag ideje 30 perc. És ilyen mértékű leállás fél évente egyszer történik, tehát az MTBF együtthatónk egy évre számolva 16-ra számítható.

**MTTR (Mean Time To Repair):** Ebben az esetben is azt kell figyelembe venni, hogy a rendszer mennyi ideig van használhatatlan állapotban, de ebben az esetben fontos szempont a hogy a leállások alatt mennyi javítás volt szükség elvégezni annak érdekében, hogy a rendszer újra működőképes legyen. Ebben az esetben is , mint az MTBF esetében a rendszer érdeke, hogy a lehető legalacsonyabb hibával működjön. A rendszer karbantarthatósága egyszerű, mert az applikáción kívül a webszervert kell az adatbázissal karban tartani. A webszerverről a biztonság kedvéért naponta egyszer hajnali 3-kor biztonsági mentés készül, továbbá havonta egyszer újraindítják.

## Teljesítmény

A rendszer biztonságának megőrzése érdekében folyamatosan figyeli önmagát biztonsági problémák szempontjából és állapotjelentést ad, amely megjelenik egy oldalon. Az állapotjelentés figyelmeztet a négy fontos területtel kapcsolatos általános problémákra és javaslatokat ad a kijavításukra. A rendszer különböző módon reagál alacsony és magas terhelésre.

A teljesítmény akkor optimális, amikor a felhasználó zökkenőmentesen és bármiféle lassúlás nélkül tudja használni a rendszert.

Teljesítményvesztés oka lehet hardveres vagy szoftveres. A rendszer folyamatosan számon tartja a háttérben futó feladatok erőforrástól elvett mennyiségét, ha valamelyik túl sokat használ, akkor elmenti a rendszer aktuális állapotát, majd megpróbálja a folyamatot leállítani, ezzel elérve az optimális teljesítményt. Erről a folyamatról a felhasználó értesítést kap.

A teljesítményvizsgálatot fontos elvégezni az adatbázisra és az operációs rendszerre egyaránt. A rendszert hatalmas mennyiségű munka terheli, annyira, hogy a rendszer már nem működik egy adott pillanatban. Ez lehetővé teszi számunkra a rendszer szűk keresztmetszeteinek azonosítását. Az adatbázis-tesztek során a leggyakoribb problémák a következők: A rendszerre gyakorolt nagy nyomás szerepet játszik az adatbázis-tranzakciók állapotának meghatározásában. Ha az adatbázis szoftver hibás elképzeléssel rendelkezik a tranzakciók állapotáról, akkor azonnal következetlenek lesznek. A vizsgálat elvégzése után a régi teszt adatokat meg kell tisztítani, és teret kell adni az új tesztadatokhoz. Az operációs rendszernél hasonlóan kell eljárni.

## Támogatottság

**Felhasználói támogatottság:** Biztosítjuk a feladatok ellátása közben kialakult hardveres és szoftveres hibák kivizsgálását és azonnali elhárítását.

Az információk különböző módon jelennek meg. Súlyos hiba esetén egy felkiáltó jel mellett piros, nagy betűkkel írja ki, ha probléma merült fel a rendszerrel. Kisebb hiba esetén narancssárga felkiáltó jel mellett narancssárga, közepes méretű betűkkel írja ki a hiba okát. További információkat fekete színnel írat ki a képernyőre.

# Technikai specifikáció



## Architektúra terv

**A tervezett rendszer a következő összetevőkből áll:**

* Adatbázis
* Adatbázis szerver
* Tűzfal
* Web szerver
* Web kliens

**Az architektúrában az adatforgalom szempontjából a következőképpen zajlik:**

Az applikáció a web szerverrel kapcsolatba lép és lekéri a szükséges adatokat, innen kétféleképpen alakulhat. Az első eset, hogy a kérés eredményének meghatározására nem szükséges az adatbázis. Ebben az esetben a web szerver azonnal tud válaszolni. A másik eset, ha szükséges az adatbázis, ilyenkor a web szerver továbbítja a kérést az adatbázis szerver felé. Miután megkapta az adatbázisból a szükséges adatot, elküldi a web szervernek, végül pedig a szerver elküldi a választ az web kliensnek.

Gondolkodás után a böngészőt választottuk ki a megfelelő platformoknak. Azért erre esett a választás, mert így kényelmesen, operációs rendszertől függetlenül érhetőek el a szükséges információk, a saját számítógépről.

A teljesítménytesztelés segít a rendszerek megfelelő karbantartásában és a hibák kijavításában mielőtt a problémák elérnék a rendszerfelhasználókat. Segít fenntartani az alkalmazások hatékonyságát, válaszképességét, méretezhetőségét és sebességét az üzleti követelményekhez képest. Ha hatékonyan működik, a teljesítménytesztelésnek meg kell adni a szűk keresztmetszetek megszüntetéséhez szükséges diagnosztikai adatokat, amelyek gyenge teljesítményt eredményeznek. Szűk keresztmetszet akkor fordul elő, ha az adatfolyam megszakad vagy leáll, mert nincs elegendő kapacitás a számítási feladat kezeléséhez. A terheléstesztelés a rendszer teljesítményét méri a számítási feladatok növekedésével. Azonosítja, hogy az alkalmazás hol és mikor omlik össze, így az éles környezetbe való szállítás előtt kijavíthatja a problémát.

## Platform tulajdonságok és követelmények

**A rendszer futtatásához ajánlott hardverkörnyezet:**

Az alkalmazás használatához szükség van egy szabadon választott böngészőre. Ebből adódóan bármely olyan hardver képes futtatni a programot, amely képes futtatni az álltalános felhasználásra kínált böngészőket.

**A rendszer futtatásához szükséges szoftverkörnyezet:**

Egy platform független böngészőből elérhető az applikáció. A hétköznapi használatra kínált böngészők (Firefox, Chrome, Opera, Edge) képesek probléma nélkül kezelni a szoftvert.

## Fejlesztői eszközök

Az alapvető platformunk a chat alapú kommunikációhoz az erre a célra létrehozott **Messenger** csoport. Itt folyik a mindennapos kommunikáció, és itt szoktuk összegyűjteni a kérdéseket és a megoldandó feladatokat. A gyakorlati órák után tartott személyes megbeszéléseken, kívül minden vasárnap este tartunk online megbeszélést, amihez a **Discord** programot használjuk. Általában itt kerülnek elfogadásra az adott heti és kiosztásra a következő heti feladatok, továbbá itt vitatjuk meg a felmerülő kérdéseket is.

Az **MS Word** programban történik a projekt dokumentum vezetése, szerkesztése. A dokumentumok kezelésére, tárolására az erre a célra létrehozott **GitHub** repository-t használjuk. Itt található a projekt dokumentum legfrissebb verziója és a kiegészítő, fejlesztést segítő fájlok is. A bevált gyakorlat szerint a projekttagok elküldik a munkáikat a projektvezetőnek, aki ezeket elfogadja és beilleszti a megfelelő formátumban a közös projekt dokumentumba. A folyamatábrák és use case diagrammok a **Miro** program ingyenes, online verziójával készültek.

Szoftveres háttérnek a **Visual Studio Code**-ot és a HTML, CSS és Java Script programozási nyelveket és környezeteket választottuk.

Adatbázis tervezéséhez **Google Firebase**-t használunk.



# Adatbázis terv

## Szöveges leírás

A féléves feladatunkban egy fényképek kezelésére alkalmas webapplikációt fogunk elkészíteni. Az adatbázisunkban eltároljuk, a fényképezőgép váz, az objektív és az elkészült kép adatait, illetve a felhasználó MAC címét, mint egyedi azonosítót. Az adatok eltárolásába regisztrációnál a felhasználónak bele kell egyeznie.

Alapvetően 4 tábla van: **User**, **Body**, **Lens**, **Picture**. A User tábla tárolni fogja a felhasználó MAC címét, mint egyedi azonosítót és mivel a felhasználó minden interakció végrehajtója, így a többi tábla ID rekordja is a User táblában kerül külön tárolásra majd. A Body táblában a fényképezőgép váz tulajdonságai kerülnek tárolásra a váz ID-ja mellett, mint a fényképezőgép márka, modell típusa, expozíciós idő, expozíciós program, fénymérési mód, szoftver és a vaku módja. A Lens tábla tárolni fogja az objektív ID-ját, a rekeszértékét és a fókuszálási távolságot. A fénykép azonosítóján kívül (PictureID), a Picture tábla tárolni fogja az összes képre vonatkozó tulajdonságot, mint a kép létrehozási dátuma, a tömörítés, a kép szélessége és magassága, fájl mérete, fájl típusa, a fényérzékenység, a színtér és a szerző.

Egyelőre ennyi adatot tárol az adatbázis, viszont az applikáció fejlődése során lehetőség lesz újabb típusú adatokat rögzíteni.

## Táblák

|  |
| --- |
| **Relációs séma** |
| USER (UserID, BodyID, LensID, PictureID, MacAddress) |
| BODY (BodyID, Company, BodyType, ExpoTime, ExpoProgram, MeteringMode, Software, FlashMode) |
| LENS (LensID, Aperture, FocalLength) |
| PICTURE (PictureID, CreateDate, Compression, Width, Height, FileSize, FileType, Iso, ColorSpace, Author) |

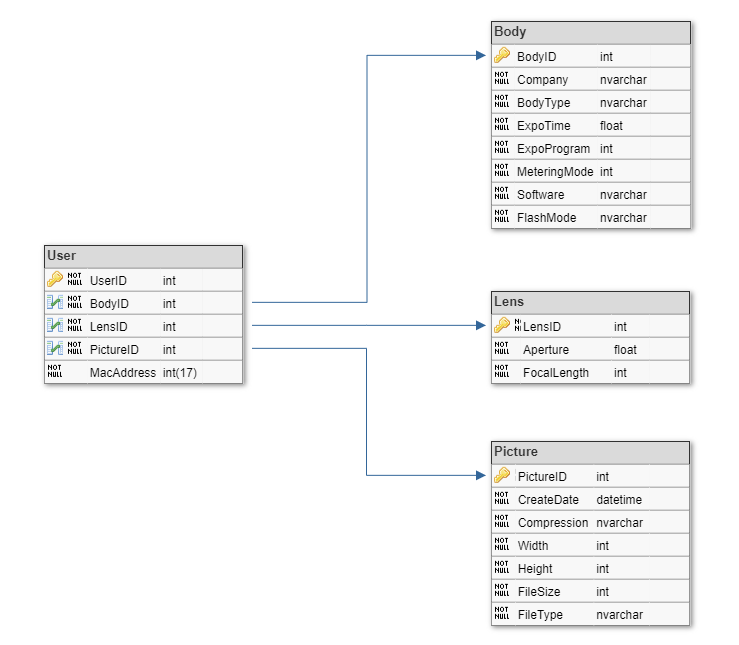
|  |
| --- |
| **USER tábla** |
| UserID [int] NOT NULL |
| BodyID [int] NOT NULL |
| LensID [int] NOT NULL |
| PictureID [int] NOT NULL |
| MacAddress [int](17) NOT NULL |

|  |
| --- |
| **BODY tábla** |
| BodyID [int] NOT NULL |
| Company [nvarchar] NOT NULL |
| BodyType [nvarchar] NOT NULL |
| ExpoTime [double] NOT NULL |
| ExpoProgram [int] NOT NULL |
| MeteringMode [int] NOT NULL |
| Software [nvarchar] NOT NULL |
| FlashMode [nvarchar] NOT NULL |

|  |
| --- |
| **LENS tábla** |
| LensID [int] NOT NULL |
| Aperture [double] NOT NULL |
| FocalLength [int] NOT NULL |

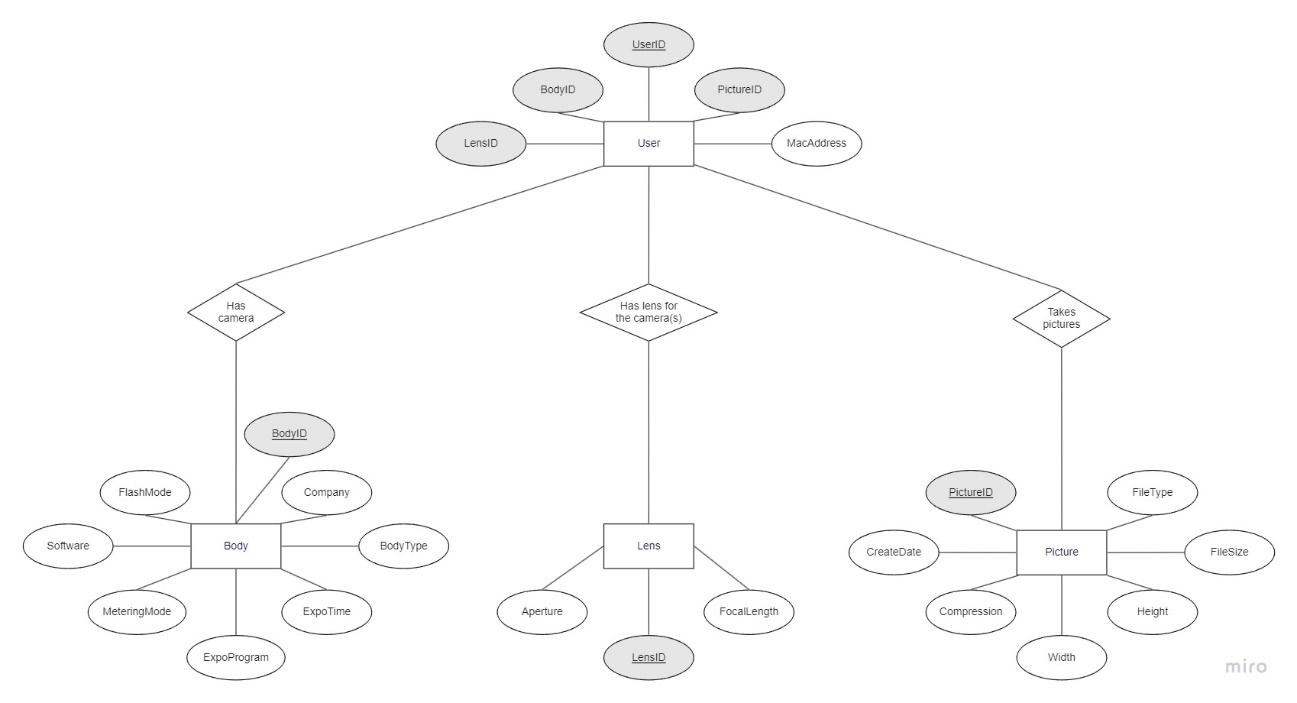
|  |
| --- |
| **PICTURE tábla** |
| PictureID [int] NOT NULL |
| CreateDate [datetime] NOT NULL |
| Compression [nvarchar] NOT NULL |
| Width [int] NOT NULL |
| Height [int] NOT NULL |
| FileSize [int] NOT NULL |
| FileType [nvarchar] NOT NULL |
| Iso [int] NOT NULL |
| ColorSpace [boolean] NOT NULL |
| Author [nvarchar] NOT NULL |

## Adatmodell



4. ábra - Adatmodell

## Egyed-kapcsolat diagram



5. ábra - Egyed-kapcsolat diagram

## Kulcsok és megszorítások



### Elsődleges kulcsok

**USER tábla:** UserID

**BODY tábla:** BodyID

**LENS tábla:** LensID

**PICTURE tábla:** PictureID

### Másodlagos kulcsok

**USER tábla:** BodyID, LensID, PictureID

### Megszorítások

**USER tábla:**

* A MAC cím 17 karakterből kell álljon (00:1B:44:11:3A:B7 formátum)
* A MAC cím minden 3. karaktere „:” karakter kell legyen

**BODY tábla:**

* Az expozíciós idő 1/16 000 másodperctől egészen végtelenig terjedhet, emiatt legalább 0.0000625 (1/16 000) értéknek kell lennie
* Az expozíciós program [0, 8] intervallumba kell essen, mivel EXIF adatként 0, 1, 2, 3, 4, 5, 6, 7, 8 értéket vehet csak fel
* A fénymérési mód [0, 6] intervallumba kell essen, illetve a {255} érték is megengedett (ez jelképezi az egyéb módot)

**LENS tábla:**

* A rekeszérték [0, 32] intervallumba kell essen
* A fókusztávolság minimum 0 értéket kell felvegyen

## Tárolt eljárások

Az adatbázisban tárolt eljárások a program üzemeltetését segítik.

* CreateUser: Az inputként kapott MAC cím alapján automatikusan hoz létre egy új felhasználót. A UserID kiosztása automatikusan történik.
* AddBody: Jelen esetben azt feltételezzük, hogy egy felhasználónak több fényképezőgépe lehet. A kép EXIF adatból kiolvasva automatikusan felveszi a még nem ismert fényképezőgépet és felveszi azt új kameraként az adatbázisba. A BodyID kiosztása automatikusan történik.
* AddLens: Itt szintén feltételezzük, hogy egy fényképezőgéphez nem csak egy objektívet használhat a felhasználó, így ezt a kép EXIF adatokból kiolvasva automatikusan felveszi új objektívként. A LensID kiosztása automatikusan történik.
* AddPicture: Képeket feltöltve a webalkalmazásba automatikusan létrehoz egy új képet az adatbázisban. A PictureID kiosztása automatikusan történik.

## Továbbfejlesztési ötletek

Továbbfejlesztési ötletek is felmerültek, amelyek az alkalmazás használatának könnyítését és fejlesztés szempontjából az átláthatóságot segítik:

* **FLASH tábla az adatbázisba:** Ez lényegében egy funkcióbővítési lehetőség, ugyanis így az alkalmazás az alap EXIF adatokon kívül fizikailag a fényképezőgépre kerülő vakuk adatait is képes lenne kezelni.
* **Felhasználókezelés kibővítése:** Jelen pillanatban a felhasználók egy MAC cím által vannak megkülönböztetve a programban. Ki lehetne bővíteni olyan funkciókkal a felhasználók kezelését, mint felhasználó profil, felhasználónév-jelszó, személyes adatok megadása, mint pl. e-mail cím megadási lehetőség. Ez azért is hasznos, mert így nem lenne MAC címre korlátozva a felhasználói profil. Így pl. másik számítógépről felhasználónév-jelszó párossal bejelentkezve is betöltődne az adott profilhoz tartozó kép galéria és az EXIF adatok, ezzel sokkal nagyobb szabadságot adva a felhasználóknak.
* **EXIF adatok kezelésének kibővítése:** Több száz darab EXIF adatot is tud rögzíteni akár egy fényképezőgép egy kép tulajdonságaihoz. Jelenleg az alkalmazásunk a legfontosabbakat kezeli, de még több idő ráfordításával akár az összes EXIF adatot lelehet így fedni.