**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](#_Toc116929020)

[1.1. A probléma és a projektcél rövid felvázolása 1](#_Toc116929021)

[1.2. A program tervezett funkcionalitásai 1](#_Toc116929022)

[1.3. Szoftveres felépítés 1](#_Toc116929023)

[1.4. A projekt során használni kívánt eszközök 2](#_Toc116929024)

[1.5. Mérföldkövek 2](#_Toc116929025)

[1.6. Tevékenységek tervezett kiosztása 3](#_Toc116929026)

[1.7. Célok 3](#_Toc116929027)

[1.8. Közös szótár (Glossary) 4](#_Toc116929028)

[2. A kiválasztott probléma 7](#_Toc116929029)

[2.1. Szöveges bemutatás 7](#_Toc116929031)

[2.2. A probléma folyamatábrája 8](#_Toc116929032)

[3. Üzleti igény specifikáció 9](#_Toc116929033)

[3.1. Célok 9](#_Toc116929035)

[3.2. Stakeholderek 9](#_Toc116929036)

[3.3. Rendszer tervezett működése folyamatábra 9](#_Toc116929037)

[4. Követelményspecifikáció 10](#_Toc116929038)

[4.1. Funkcionális követelmények 10](#_Toc116929040)

[4.2. Nem funkcionális követelmények 10](#_Toc116929046)

[4.2.1. Termék követelmények: 10](#_Toc116929048)

[4.2.2. Szervezeti követelmények: 10](#_Toc116929049)

[4.2.3. Külső követelmények: 10](#_Toc116929050)

[4.3. Használhatóság 10](#_Toc116929051)

[4.4. Megbízhatóság 10](#_Toc116929052)

[4.5. Rendelkezésre állás 10](#_Toc116929053)

[4.6. Teljesítmény 10](#_Toc116929054)

[4.7. Támogatottság 10](#_Toc116929055)

[5. Technikai specifikáció 11](#_Toc116929056)

[5.1. Architektúra terv 11](#_Toc116929058)

[5.2. Platform tulajdonságok és követelmények 11](#_Toc116929059)

[5.3. Fejlesztői eszközök 11](#_Toc116929060)

[6. Gazdasági számítások ? 12](#_Toc116929061)

[7. Adatbázis terv 13](#_Toc116929063)

[7.1. Szöveges leírás 13](#_Toc116929065)

[7.2. Táblák 13](#_Toc116929066)

[7.3. Adatmodell 15](#_Toc116929067)

[7.4. Egyed-kapcsolat diagram 15](#_Toc116929068)

[7.5. Kulcsok és megszorítások 15](#_Toc116929069)

[7.5.1. Elsődleges kulcsok 15](#_Toc116929078)

[7.5.2. Másodlagos kulcsok 15](#_Toc116929079)

[7.5.3. Megszorítások 15](#_Toc116929080)

[7.6. Tárolt eljárások 15](#_Toc116929081)

[7.7. Továbbfejlesztési ötletek 15](#_Toc116929082)

[8. GUI terv 16](#_Toc116929083)

# 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)

**Stakeholder**

Az üzleti igényspecifikáció a résztvevője.

**Use-Case diagram**

Magyarul használati eset diagramként is ismert, a felhasználó és a rendszer közti lehetséges interakciók grafikus ábrázolására használt módszer. A Use-Case diagram különböző használati eseteket és különböző típusú felhasználókat mutat be, gyakran más típusú diagramok is kísérik. Az egyes eseteket körök vagy ellipszisek jelölik. Az aktort (azaz a felhasználót) pálcikafigurákként ábrázolja.

**Aktor**

Use-Case diagramban egy szerepkört, a rendszer egy felhasználóját reprezentálja.

**SWOT analízis**

A SWOT analízis (vagy más néven mátrix) egy stratégiai tervezési és stratégiai irányítási elemzéshez használt technika, mely segít egy személynek vagy szervezetnek abban, hogy azonosítani tudják a versenytásakkal szembeni vagy a projekttervezéssel kapcsolatos erősségeket, gyengeségeket, lehetőségeket vagy veszélyeket.

**TCO**

A TCO (Total Cost of Ownership) egy pénzügyi becslés, melynek célja, hogy segítse a vásárlók és a tulajdonosok számára meghatározni egy termék vagy szolgáltatás közvetlen és közvetett költségeit. Ez egy vezetői számviteli eszköz, mely a teljes költségszámításhoz vagy akár az ökológiai, gazdasági számításokhoz is alkalmazható, beleértve a társadalmi költségeket is.

**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.

**GUI**

A grafikus felhasználói felület vagy grafikus felhasználói interfész (angolul graphical user interface, röviden GUI) a számítástechnikában olyan, a számítógép és ember közti kapcsolatot megvalósító elemek összessége, melyek a monitor képernyőjén szöveges és grafikai elemek együtteseként jelennek meg. A grafikus felhasználói felületeken alapvető szerepe van a mutatóeszközök, például az egér használatának, amelyekkel a grafikus felület elemei intuitív módon, a fizikai világ egyfajta modelljeként kezelhetők.

**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.

# A kiválasztott probléma



## Szöveges bemutatás

A projektünk a fényképészeket/fotográfusokat célozza meg. A fényképek elkészültekor, egészen pontosan az exponáló gomb megnyomásakor, legyen az telefonnal vagy fényképezőgéppel készítve ún. EXIF metaadatok 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 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

# Üzleti igény specifikáció



## Célok

## Stakeholderek

**Belső érdekeltek**

**Külső érdekeltek**

## Rendszer tervezett működése folyamatábra

# Követelményspecifikáció



## Funkcionális követelmények



## Nem funkcionális követelmények



### Termék követelmények:

### Szervezeti követelmények:

### Külső követelmények:

## Használhatóság

## Megbízhatóság

## Rendelkezésre állás

## Teljesítmény

## Támogatottság

# Technikai specifikáció



## Architektúra terv

## Platform tulajdonságok és követelmények

## Fejlesztői eszközök

# Gazdasági számítások ?



# 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

## 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

## Tárolt eljárások

## Továbbfejlesztési ötletek

# GUI terv