

UNIVERSITATEA „SAPIENTIA” DIN CLUJ-NAPOCA
FACULTATEA DE ȘTIINȚE TEHNICE ȘI UMANISTE,
TÎRGU-MUREȘ
SPECIALIZAREA CALCULATOARE

Sprijinirea administrării online prin
procesarea imaginilor
PROIECT DE DIPLOMĂ

Coordonator științific:





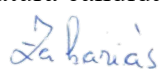
Prof.dr.habil Szilágyi László

drd.Györfi Ágnes

Absolvent:

Zakariás Zsolt

2023

UNIVERSITATEA „SAPIENTIA” din CLUJ-NAPOCA Facultatea de Științe Tehnice și Umaniste din Târgu Mureș Specializarea: Calculatoare		Viza facultății: 
LUCRARE DE DIPLOMĂ		
Coordonator științific: prof. dr. habil Szilágyi László	Candidat: Zakariás Zsolt Anul absolvirii: 2023	
a) Tema lucrării de licență: SPRIJINIREA ADMINISTRĂRII ONLINE PRIN PROCESAREA IMAGINILOR		
b) Problemele principale tratate: <ul style="list-style-type: none">- Studiu bibliografic privind procesarea imaginilor- Realizarea unei aplicații pentru extragerea textelor din imagini- Compararea metodelor folosite		
c) Desene obligatorii: <ul style="list-style-type: none">- Schema bloc al aplicației- Diagrame UML privind software-ul realizat.		
d) Softuri obligatorii: <ul style="list-style-type: none">-Aplicație pentru gestionarea sancțiunilor		
e) Bibliografia recomandată: <ul style="list-style-type: none">- Sonka M, Hlavac V, Boyle R: Image Processing, Analysis and Machine Vision. 4th ed, 2015- https://github.com/tesseract-ocr		
f) Termene obligatorii de consultații: săptămânal		
g) Locul și durata practicii: Universitatea „Sapientia” din Cluj-Napoca, Facultatea de Științe Tehnice și Umaniste din Târgu Mureș Primit tema la data de: 31.03.2022 Termen de predare: 27.06.2023		
Semnătura Director Departament 	Semnătura coordonatorului 	
Semnătura responsabilului programului de studiu 	Semnătura candidatului 	

Declarație

Subsemnata/ul Zakariás Zsolt absolvent(ă) al/a specializării Calculatoare, promoția 2023 cunoscând prevederile Legii Educației Naționale 1/2011 și a Codului de etică și deontologie profesională a Universității Sapientia cu privire la furt intelectual declar pe propria răspundere că prezenta lucrare de licență/proiect de diplomă/disertație se bazează pe activitatea personală, cercetarea/proiectarea este efectuată de mine, informațiile și datele preluate din literatura de specialitate sunt citate în mod corespunzător.

Localitatea, Târgu Mureș

Data: 21.06.2023

Absolvent

Semnătura

Zakariás

Sprijinirea administrării online prin procesarea imaginilor

Extras

Datorită progreselor tehnologice, smartphone-urile și internetul au devenit indispensabile în viața noastră, fiind mereu la îndemână, iar în zilele noastre sunt un fel de asistenți personali, care ne fac viața mai ușoară. Cealaltă mare parte a vieții noastre este încă dominată de gestionarea documentelor personale pe hârtie. Există o barieră perceptibilă între cei doi poli, dintre care unul este caracterizat de mișcare și rapiditate, iar celălalt de lentoare și consum de timp.

Scopul acestei lucrări este de a dezvolta un sistem eficient și rapid pentru a sparge această barieră, oferind un nou mod de gestionarea a sancțiunilor. Metodele tradiționale nu pot ține pasul cu lumea noastră rapidă și pot cauza probleme, cum ar fi procesul greoi de gestionare a amenzilor și a termenilor de plată. Ne-am propus să dezvoltăm un sistem care să răspundă cerințelor tehnologice actuale. În dezvoltarea sistemului, am luat în considerare ambele părți, atât pe cea care emite amenda, cât și pe cea care o primește, și am căutat să realizăm o implementare care să fie eficientă din punct de vedere al timpului și rapidă pentru ambele părți. Utilizarea unei aplicații android pentru a emite sancțiunea s-a dovedit a fi cea mai potrivită, deoarece telefonul este mereu la îndemână și poate oferi utilizatorului acces instantaneu la baza de date. Cea mai evidentă modalitate de implementare a sistemului pentru șoferi a fost utilizarea unei interfețe bazate pe web, deoarece aceasta le-ar oferi o interfață relativ mare pentru a vizualiza și plăti mai ușor amenzi.

Cuvinte cheie: gestionarea sancțiunilor, aplicație android, interfață web.

**SAPIENTIA ERDÉLYI MAGYAR
TUDOMÁNYEGYETEM
MAROSVÁSÁRHELYI KAR
SZÁMÍTÁSTECHNIKA SZAK**

**Az online ügyintézés támogatása
képfeldolgozással
DIPLOMADOLGOZAT**

Témavezető:

Dr. habil Szilágyi László, egyetemi tanár
drd. Győrfi Ágnes

Végzős hallgató:

Zakariás Zsolt

Kivonat

A technológiai fejlődésnek köszönhetően az okostelefonok és az internet elengedhetetlenek lettek az életünkben, hiszen mindig kéznél vannak, és mára már egyfajta személyi asszisztens szerepet betöltve járulnak hozzá az életünk megkönnyítéséhez. Az életünk másik nagy részét még mindig személyes ügyintéзések és papíralapú dokumentumok határozzák meg. Jól érezhető egyfajta gát a két pólus közt melyből az egyikre a folyamatos mozgás, gyorsaság míg a másikra a lassúság, időigényesség jellemző.

A dolgozat célja egy hatékony és gyors rendszer kidolgozása amely révén ezt a gátat feloldjuk, egy újfajta lehetőséget szolgáltatva a büntetések ügyintézésére. A hagyományos módszerek nem képesek lépést tartani a felgyorsult világunkkal, és problémákat okozhatnak, mint például a bírságok nehézkes rendszerezése és az ebből fakadó kifizetésnek az elmúlasztása. A mai technológiai elvárásoknak megfelelő rendszer kidolgozását tűztük ki célul. A rendszer kifejlesztése során figyelembe vettük mindkét félt, aki a büntetést kiosztja illetve aki ezt megkapja, és igyekeztünk egy olyan megvalósításra amely mindkét szerepkör számára időtakarékos és gyors. A büntetés kiszotásához egy androidos applikáció használata tűnt a legmegfelelőbbnek, mivel a telefonunk állandóan kéznél van illetve interneten keresztül azonnali hozzáférést is tud biztosítani a felhasználónak az adatbázishoz. A járművezetőknek szánt rendszer kivitelezéséhez a legkézenfekvőbbnek egy webes felület megvalósítása tűnt mivel ezáltal egy viszonylag nagyobb felületen tudunk jobb rálátást nyújtani számukra a bírságok könnyebb átlátásához illetve kifizetéséhez.

Kulcsszavak: büntetések ügyintézés, androidos applikáció, webes felület

Abstract

Thanks to technological advances, smartphones and the internet have become crucial in our lives, because it is always at our fingertips and also acts as a personal assistant to make our lives easier. The other major part of our lives is still dominated by personal dealings and paper-based documents. There is a perceptible barrier between the two poles, one of which is characterized by constant movement and speed, the other by slowness and time-consumption.

The aim of this thesis is to develop an efficient system to break down this barrier, providing a new way of processing driving penalties. Traditional methods cannot keep up with our fast-paced world and can cause problems such as the difficulty of organizing fines and the resulting delay in payment. We set out to develop a system that meets today's technological requirements. In developing the system, we took into account both parties, the one who hands out the fine and the one who receives it, and we aimed for an implementation that is time-efficient for both roles. The use of an android application to deliver the punishment seemed to be the most appropriate, as our phone is always at hand which grants internet access to the user. The most obvious way to implement the system for drivers was to use a web-based interface, as this would provide them a relatively large interface for a better overview and easier payment of fines.

Keywords: traffic penalty management, android app, web interface

Tartalomjegyzék

1. Bevezető

2. Célkitűzések

3. Szoftver követelmények

3.1. Felhasználói követelmények

3.2. Rendszer követelmények

3.2.1. Funkcionális követelmények

3.2.2. Rendszer követelmények

4. Rendszer részletes leírása

4.1. Adatbázis felépítés

4.2. Optikai karakterfelismerés

4.3. Képfeldolgozó egység, Tesseract OCR felhasználás

4.4. Rendszer szekvencia diagramjai

4.5. A képfeldolgozó egység felépítése

4.5.1. Képfeldolgozás, szöveg kinyerése

4.5.2. Szöveg feldolgozása

4.5.2.1. Személyi igazolvány

4.5.2.2. Vezetői engedély

4.5.3. Kép- illetve szövegfeldolgozás tesztelése

4.5.3.1. 1.módszer - Grayscale, élesítő szűrő illetve Otsu küszöbérték használat

4.5.3.2. 2.módszer - Adaptiv Gauss Thresh Mean C

4.5.3.3. 3.módszer - Adaptiv Thresh Mean C

4.5.3.4. 4.módszer - Simple threshold

4.5.3.5. 5.módszer - Simple Grayscale Image Usage

5. Felhasznált technológiák

5.1. PyCharm

5.2. Android Studio

5.3. Firebase

5.4. HTML

5.5. PHP

5.6. Postman

5.7. GitHub

6. A rendszer által kínált további lehetőségek

6.1. Webes felület

6.2. Android applikáció

7. Fejlesztetőségi lehetőségek

7.1. Webes felület

7.2. Android Applikáció

1. Bevezető

A gyorsan fejlődő technológiának köszönhetően, ami behálózta életünk minden területét, a mai ember már nem tudja elképzelni az életét okostelefon illetve internet hozzáférés nélkül. Napjainkban a digitális eszközök nélkülözhetetlenné váltak, mivel mindig a zsebünkben vannak, azokkal fizetünk, keresgélünk a weboldalakon és nem utolsósorban különböző applikációkat használunk. Ezért fontosnak tartom, hogy minden téren a világunk digitalizációját támogassuk, de sajnos ez legtöbb esetben nem könnyű.

Az ember életét még mindig behálózzák a személyes ügyintézés és olykor az ezzel járó papír alapú dokumentumok. Az állami intézmények digitalizációja messze nem a legkorszerűbb, de bőven akad lehetőség ezeknek a fejlesztésére. Ennek alátámasztására legjobb példa az online fizetőkapuk, amelyek időt tudnak nekünk spórolni de még itt is megjelennek a formanyomtatványok.

Be kell látnunk, hogy a papíralapúság sok időt emészt fel az ember értékes idejéből, még akkor is mikor nem mi kell kitöltsük, mint egy közúti büntetés esetében. Ez több szempontból is rossz módja az adatok rögzítésére. A már hangsúlyozott idővesztésen kívül fennáll annak a veszélye, hogy ezek egy példányban kerülnek hozzánk és megsemmisülés vagy elvesztés esetén rengeteg utánajárást igényel egy esetleges hiteles másolat beszerzése ezzel lassítva az ügyintézés. Nem utolsósorban ezeknek a rendszerezése némi erőfeszítést igényel.

Annak ellenére, hogy a legtöbb dokumentum papír formájában készül el végső soron egy digitalizációs folyamaton esnek át, adatbázisba kerülnek a rajta feltüntetett személyes adatok. Ezek a módszerek nem csak egyfajta biztonsági hálót nyújtanak de megkönnyítik a keresést valamint bárholn is elérhetőek lehetnek.

A dolgozat keretein belül az esetleges büntetés és annak a hatékony illetve gyors ügyintézésére kerestünk megoldást.

A hagyományos módszerek időigénye, analóg megvalósítása nem egyeztethető össze felgyorsult világunkkal ami eredményezhet elfelejtett kifizetendő bírságot ami késedelmi díj befizetést is von maga után valamint a nehezen megvalósítható rendszerezést amely szükséges az átláthatósághoz.

Egy ilyen rendszer megvalósításával szeretnénk hozzájárulni az egyre gyorsabban digitalizálódó világunk fejlődéséhez.

2. Célkitűzések

A dolgozat célja egy olyan korszerű rendszer kidolgozása amely eleget tesz a technológiai elvárásoknak napjainkban. Elsősorban maga a közúti büntetési folyamat hatékonyságának növelésére törekszünk a digitalizáció révén. A hatékonyság alatt az időtakarékoságot és az ezzel járó gyorsaságot értjük.

Fontos, hogy a terv kivitelezése során olyan erőforrásokat vegyünk igénybe amelyek nem terhelik az adott személyt anyagilag, mindenki számára elérhető valamint mindig kéznél van. Ezeket a szempontokat figyelembe véve tettük le a voksunkat az okostelefon mellett, mivel napjainkban többnyire minden személy rendelkezik ilyen eszközzel.

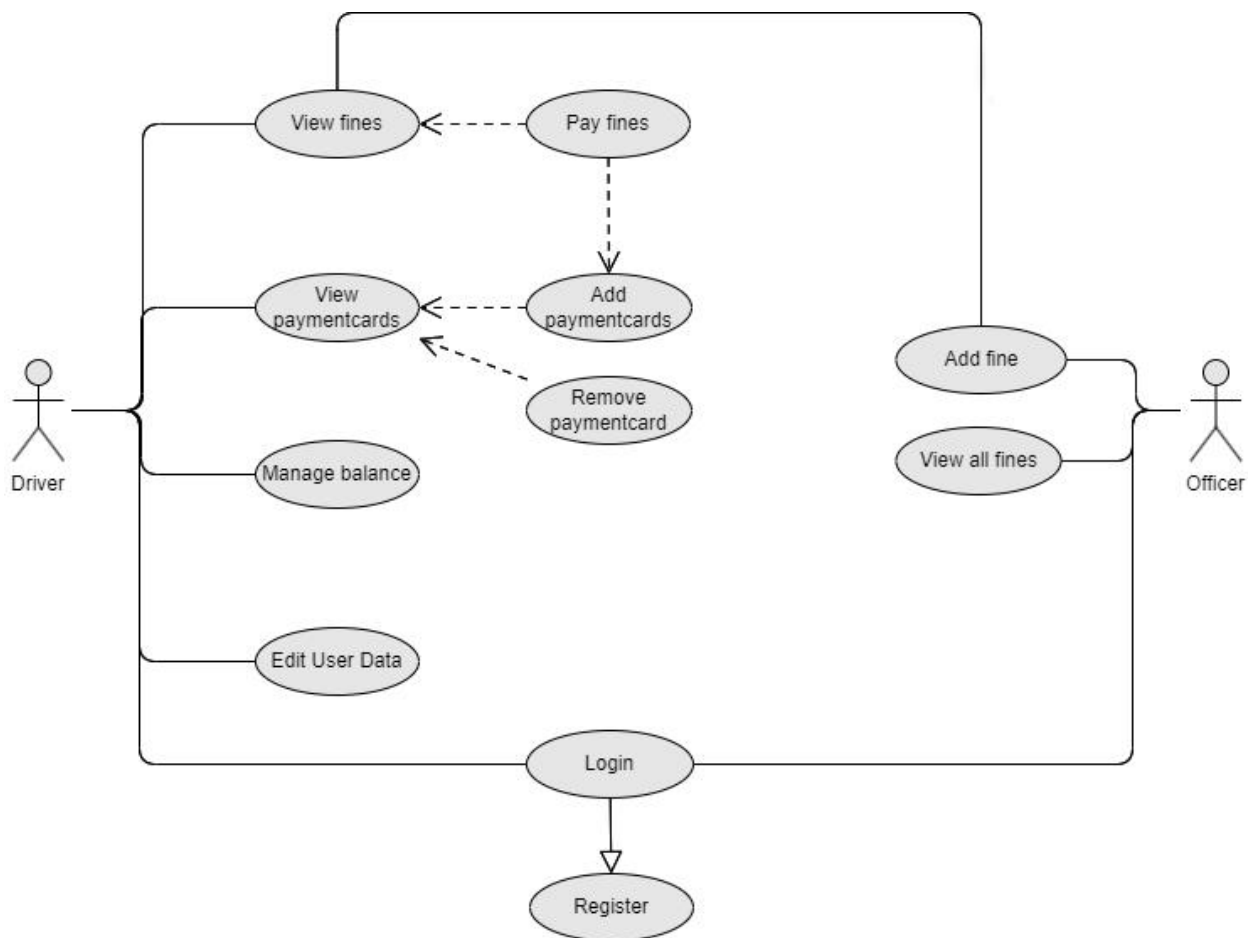
Ezen kívül célunk, hogy ne csupán a büntetést kiosztó személy munkáját könnyítsük hanem a másik fél helyzetét is. Ennek megvalósítására egy webes felület által nyújtott előnyök kiaknázása mellett döntöttünk. Ennek a felületnek a használatával könnyen kifizethetjük valamint számon tarthatjuk a büntetéseket.

Ezen célkitűzések megvalósítása esetén egy hatékony rendszert tudunk biztosítani a felhasználók számára, mindezek mellett számos továbbfejleszthetőségi lehetőséget rejt magában a megoldásunk.

3.Szoftver követelmények

3.1. Felhasználói követelmények

A rendszert használók között két felhasználó típust különíthetünk el, büntetést kiosztó, rendőrségi alkalmazott, valamint a büntetést kapó, autóvezető. Ezekről a felhasználóktól elvárt követelményeknek a könnyebb átláthatósága érdekében létrehoztunk egy használati eset (use case) diagramot, lásd 3.1 ábra. Ezt követően részletezzük a követelményeket szerepkörönként.



3.1 ábra. A rendszer használati diagramja.

Járművezető/hétköznapi felhasználó

- Regisztráció révén a felhasználó a rendszer része lesz így majd be tud bármikor jelentkezni.
- Listázhatja a meglévő büntetéseket, valamint kifizetheti őket.
- Kezelheti a bankkártyáit valamint az egyenlegét. Az egyenleg segítségével tudja kifizetni a büntetéseket.
- Módosíthatja a megadott személyes adatait valamint jelszót.

Rendőrségi egység tagja/hivatali felhasználó

- A regisztrációt ebben az esetben maga a hivatal végzi el, a felhasználó a belépési adatokat megkapva tud a fiókjába bejelentkezni.
- Kioszthat új, illetve megtekinthet meglévő bírságokat.

3.2. Rendszer követelmények

3.2.1. Funkcionális követelmények

Járművezető/hétköznapi felhasználó

- Regisztráció: teljes név, személyi szám (CNP - cod numeric personal), megye kiválasztása egy legördülő listából, a személyi igazolványon szereplő lakcím, érvényes e-mail cím, jelszó és annak megismétlése révén.
- Bejelentkezés: az e-mail cím és jelszó megadásával a felhasználó tudjon belépni a profiljára.
- Amennyiben a felhasználó szeretne jelszót váltani, ezt megteheti a profil beállításoknál.
- Megtekinteni a felhasználó büntetéseit táblázatba valamint rákattintva egyre bővebb információk megtekintése.
- A profilhoz tartozó egyenleg kezelése, pénzegység feltöltése valamint kiutalása egy megadott bankkártyára.
- Bankkártya hozzárendelése a felhasználó profiljához vagy adott esetben annak eltávolítása tőle.
- A profil hitelesítése céljából a személyi igazolványnak fényképes formátumú feltöltése.

Rendőrségi egység tagja/hivatali felhasználó

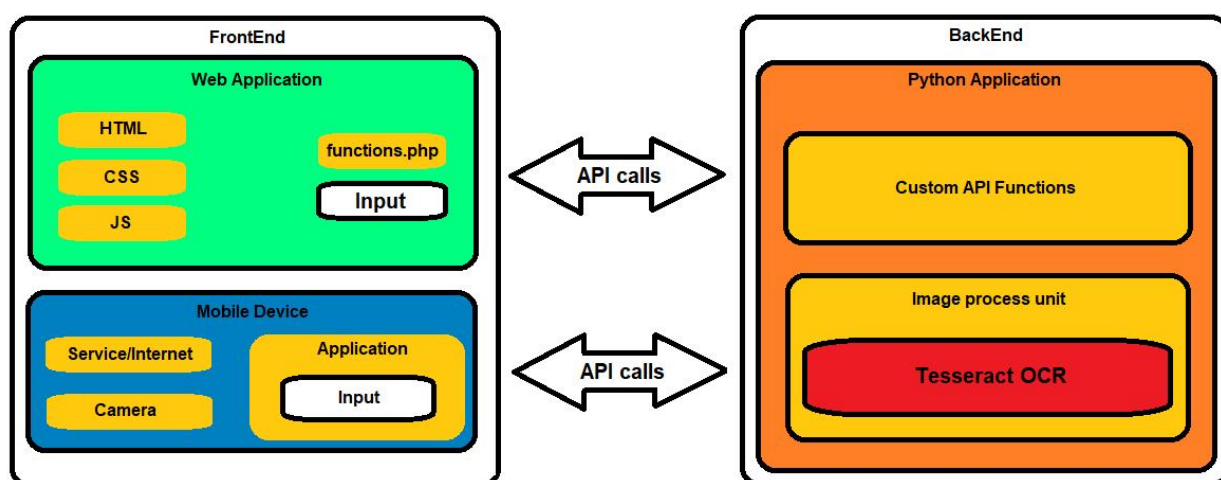
- Bejelentkezés: a jelvéyszám megadásával való bejelentkezés.
- Megtekinteni az összes kiosztott büntetést, ezeket egy görgethető listába szervezve.
- Fénykép feltöltése a büntetés kiosztásához tartozó formanyomtatványhoz szükséges adatok kiszűrése érdekében.
- Formanyomtatvány kiállítása illetve mentése az adatbázisba.

3.2.2. Rendszer követelmények

- Mind a webes felület, mind a mobilos applikáció használata internet hozzáférést igényel.

- Az applikáció használatára szükség van minimum Android 12 Snow Cone (31 API szint) eszközre.
- Az adatok tárolása valós idejű, tábla alapú, relációs adatbázisban történik. A fejlesztés valamint tesztelés keretein belül XAMPP program által lokálisan futtatott SQL alapú adatbázist használtunk.
- A Hivatali felhasználó által használt eszköznek rendelkeznie kell egy hátlapi kamerával a vezetői engedély szkennelése érdekében.

4. Rendszer részletes leírása



4. ábra. A rendszer architektúrája.

Az adatok tárolására illetve szolgáltatására SQL alapú adatbázis áll az alkalmazás illetve a webes felület rendelkezésére. Az adatbázissal való kommunikációt REST API (representational state transfer, application programming interface) hívásokon keresztül valósítottuk meg mindkét felhasználói rendszer esetében. Azért tartottuk fontosnak a REST API kommunikáció megvalósítását mivel így a felhasználói modulok jól elkülönülnek a rendszer többi részétől. Ezáltal lehetőséget biztosít a jövőt tekintve, amennyiben szeretnénk a jelenleg használt technológiákat lecserélni teljesen újra, növelve a rendszerünk időszerűségét. A REST API-t két listába tudjuk sorolni, web felület (lásd 4.1-es táblázat) illetve a mobil alkalmazás (lásd 4.2-es táblázat) által használtak.

API (path)	Ellátott feladat
/register-user	Felhasználó regisztrálása.
/login-user	Felhasználó bejelentkezése.
/get-user-information	Felhasználó adatainak lekérdezése.
/update-user-details	Felhasználó adatainak frissítése.
/update-user-password	Felhasználó jelszavának frissítése.
/add-payment-card	Bankkártya hozzáadása a felhasználó fiókjához.
/delete-payment-card	Bankkártya inaktíválása a felhasználó fiókjától.
/deposit-to-balance	Kredit hozzáadása a felhasználóhoz.
/withdraw-from-balance	Kredit visszautalása egy kiválasztott bankkártyára.
/get-user-payment-card	A felhasználóhoz tartozó bankkártyák lekérése.
/get-user-balance	A felhasználó egyenlegének lekérdezése
/get-state-fines	A felhasználó büntetéseinek lekérdezése.
/get-state-fine	Egy kiválasztott büntetése lekérdezése.
/pay-state-fine	Egy kiválasztott büntetése kifizetése.
/get-transactions	Tranzakciók listázása.
/tesseract-ocr	A képfeldolgozás megvalósítása.

4.1 táblázat. A webes felületen használt REST API-k listája.

API (path)	Ellátott feladat
/app/get-state-fines	Lekérni az összes kiosztott büntetést.
/app/send-photo	Képfeltöltés az adatok szűrésére.
/app/send-form-data	A kiosztandó büntetése érvényesítése, adatbázisba való tárolás.

4.2 táblázat. A mobil alkalmazás által használt REST API-k listája.

A képfeldolgozó egységet a nyílt forráskódú Tesseract OCR (Optical Character Recognition) szövegfelismerő motor valósítja meg. A Tesseract használatához szükséges telepíteni a programot az adott számítógépre, a mi esetünkben Windows 10 operációs rendszert használó számítógépen történt a telepítés, de elérhető macOS illetve Ubuntu operációs rendszert használó számítógépeken is.

4.1. Adatbázis felépítés

Az adatok tárolása a már említett SQL (Structured Query Language) típusú adatbázis révén valósul meg. Legfőbb előnye az adatok strukturált, relációs sémák szerinti tárolása táblákba rendezve. Ezen kívül még számos előnnyel rendelkeznek az SQL adatbázisok, például rugalmasság vagy skálázhatóság, könnyed integráció, adatbiztonság stb.

A projekt keretein belül olyan személyes adatokat használunk, valamint tárolunk amelyek a GDPR (General Data Protection Regulation) vonatkozásában személyes adatnak minősülnek, például a vezetéknév illetve keresztnév, személyi szám, személyi igazolvány és vezetői engedély száma. Ezeken kívül még szükségesnek tartottuk a bankkártyával kapcsolatos néhány jellemző információ felhasználását annak érdekében hogy a rendszer teljeskörű lefedettséget nyújtson. Az általunk kigondolt rendszer adatbázisának szerkezete a 4.3-as ábrán tekinthető meg.

allamvizsga payment_card id : int(11) fk_personal_code : text card_number : text card_holder : text month_year : text security_code : text active : int(11)	allamvizsga fine_transaction id : int(11) fk_state_fine_id : int(11) fk_balance_id : int(11) amount : int(11) fine_payment_date : timestamp	allamvizsga payment_card_transaction id : int(11) fk_payment_card_id : text transaction_type : text amount : float transaction_date : timestamp
allamvizsga person id : int(11) personal_code : text first_name : text last_name : text county : int(11) address : text email_address : text password : text phone_number : text	allamvizsga state_fine id : int(11) fk_personal_code : text first_name : text last_name : text fine_date : datetime fine_amount : float fine_type : text fine_description : text state_fine_state : text	allamvizsga user_balance id : int(11) fk_person_id : int(11) balance : float

4.3. ábra. Az adatbázis felépítése táblák szerint.

4.2. Optikai karakterfelismerés

A Optikai Karakterfelismerés egy olyan képfeldolgozási módszer amely lehetőséget nyújt a nyomtatott vagy kézzel írt szövegek automatikus felismerésére és digitalizációjára. Az ML (Machine Learning) rendszerek esetében a karakterfelismerés karakterről karakterre történik és azokat kategorizálja a rendszer. Ezzel szemben az OCR (Optical Character Recognition) rendszerek a képen megjelenő szöveget sorról sorra veszi így lehetővé teszi különböző méretű sorok feldolgozását ahogyan ezt az alábbi tanulmány keretein belül is említik [1]. A Tesseract régebbi implementációjai a karakter alapú szövegfeldolgozást alkalmazza viszont az általunk is használt újabb verzió az úgynevezett “End-to-end recognition” módszert alkalmazza. Ez a módszer egész karakterláncokat próbál meg felismerni. Emellett fontos hangsúlyozni, hogy itt nem történik szegmentáció a kép előfeldolgozását illetően.

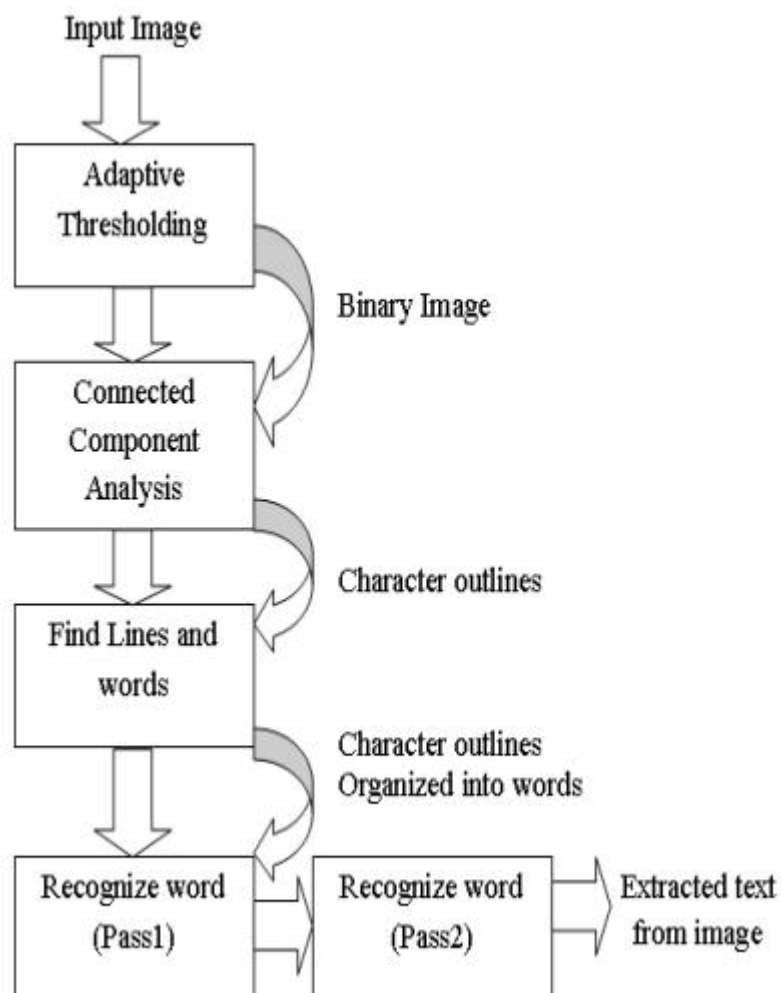
4.3. Képfeldolgozó egység, Tesseract OCR felhasználás

Egy képfeldolgozó egység esetében a legfőbb kritérium, hogy minél több hasznos adatot szolgáltatson a kép minőségétől függően. Mivel a feldolgozó egység eredményessége

nagyban függ az adott kép minőségétől, ahogyan ezt az alábbi tanulmányban is kifejtették [2], feladatunk, hogy előfeldolgozás révén a lehető legjobb minőségű képet szolgáltatassunk a szövegfelismerő motornak.

Maga a Tesseract az egy nyílt forráskódú szövegfelismerő motor melyet kezdetben a Hewlett-Packard Labs fejlesztett ki majd ezt a későbbiekben az open-source közösség átvette és fejlesztette a Google támogatásával.

Annak ellenére, hogy a Tesseract rendszer is végez bizonyos képfeldolgozási lépéseket, lásd 4.4 ábra, az eredményesség érdekében, ezt kiegészítettük a kép szűrővel való előfeldolgozásával.



4.4 ábra. Tesseract OCR architektúrája [1]

A képfeldolgozást nehezíti a személyi igazolvány valamint a vezetői engedély jellegzetes dizájnja. Mindkét okmány esetében sajátos elemeket használtak a kivitelezésben így az eredmények is eltérőek a szövegfelismerést illetően. A vezetői engedély esetében szerencsésebb a helyzet mivel az ott használt dizájn elemek illetve a pasztell színek választása nem tompít a megjelenő szöveg élességén, nem olvad a kettő egybe a rendszer számára. Sajnos ez nem mondható el a személyi igazolványról, ahol maga a dizájn illetve a színválasztás nagyon domináns ez által a Tesseract nehezen tud kiszűrni hasznos adatot erről. Viszont megjelenik a személyin egy rész, ahol tiszta fehér háttéren megjelenítve sűrítettek pár lényeges információt, például személy név, széria illetve széria szám valamint nyomokban a személyi számot (CNP). Ezt a részt az esetek többségében sikerül pontosan beazonosítania a szövegfelismerő motornak.

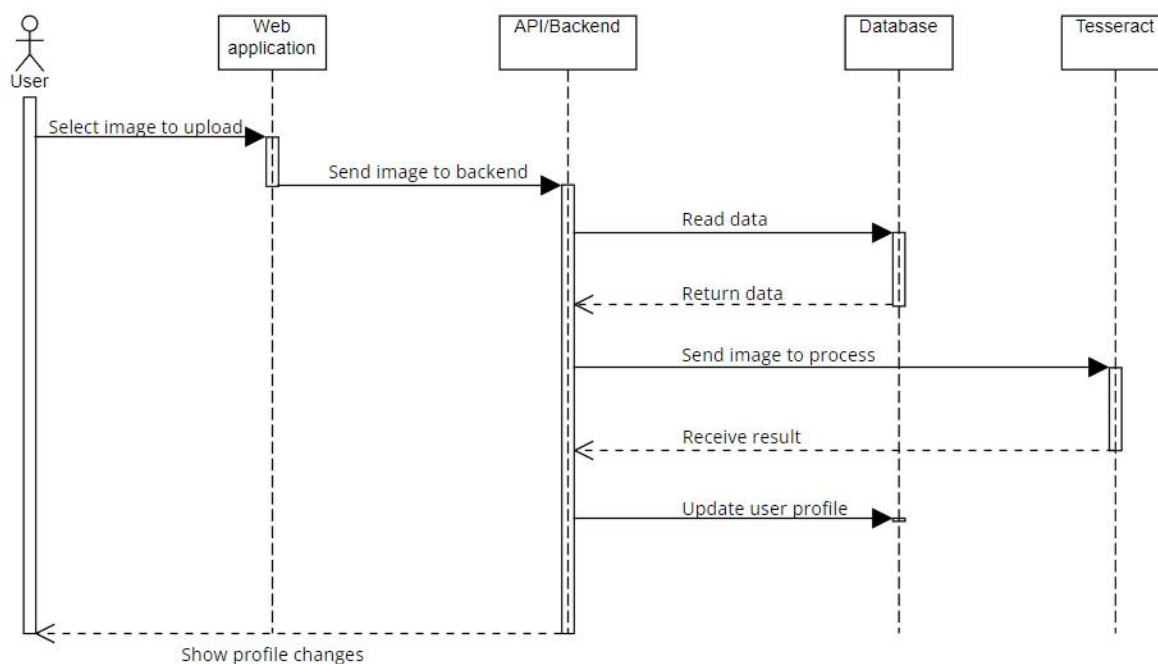
Felmerült ebben az esetben a szegmentálás használata de mégsem alkalmaztunk mivel a tesztek során volt eset amikor sikerült a dizájn háttérrel rendelkező részből is felhasználható adat kinyerésére.

Belátható, hogy a képfeldolgozás eredménye elég bizonytalan tud lenni ezért a Tesseract által adott szövegekből reguláris kifejezések segítségével szűrtük ki a számunkra hasznos információkat. Mivel a fentebb említett adatok amelyek szükségesek egy személy azonosítására meghatározott formával rendelkeznek, lásd a személyi szám jelen esetünkben 13 számjegyből áll, nem jelentett nagy gondot ezeknek a kiszűrése reguláris kifejezések által. Előfordulhat viszont, hogy a rendszer a kép gyengébb minősége miatt a 0 (zéró) számjegy helyett kicsi vagy nagy o betűt ismert fel. Mivel az általunk használt módszer nagyon leszűkíti a keresett adat formáját, az előbb felvázolt esetben sikertelennek minősül az adat kiszűrése a képről ez által kénytelenek vagyunk egy újabb képet készíteni és újrafeldolgozni.

4.4. Rendszer szekvencia diagramjai

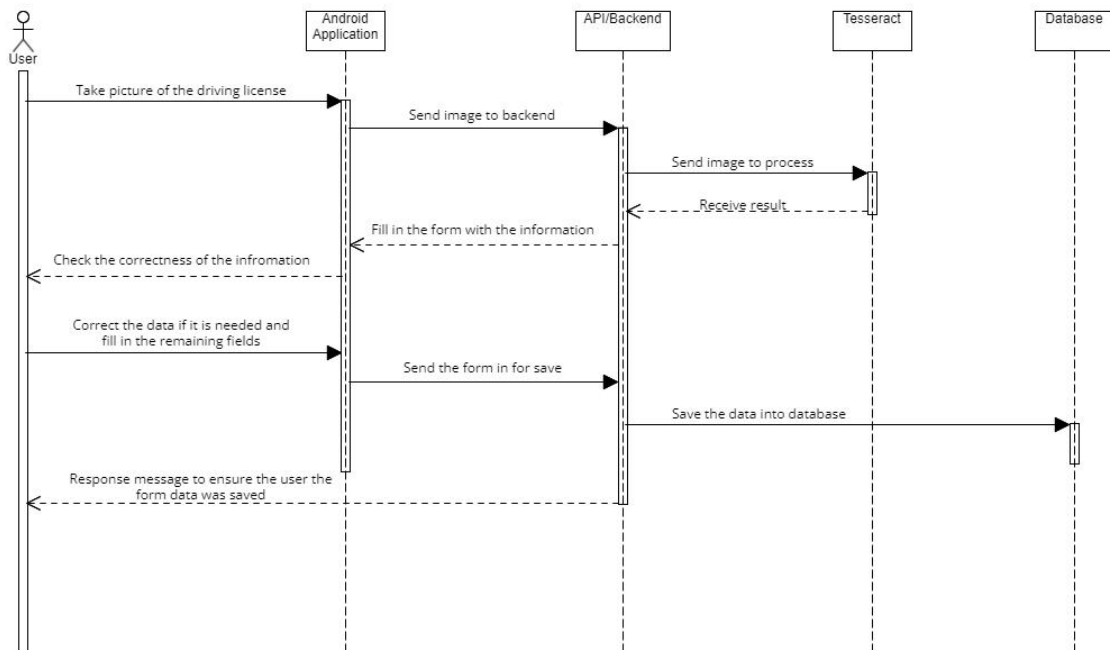
Az alábbi szekvencia diagramok révén mutatjuk be a rendszerünk legfőbb moduljainak működését. A 4.5 ábra mutatja be hogyan tudja a felhasználó a weboldalon regisztrált profilját hitelesíteni; kiválaszt egy képet a személyi igazolványáról vagy vezetői engedélyéről, ezt feltölti egy API hívás segítségével, ezt követően a Backend lekérdezi a felhasználó adatait, a képet átadja a Tesseract szövegfelismerő motorjának ami visszaad

egy szöveggyűjteményt a képen talált szövegekkel. Kiszűri a hasznos adatot és összeveti az adatbázisban tárolt, már megadott adatokkal és ez szerint módosítja a felhasználó profiljának hitelességét.



4.5 ábra. Felhasználó profil hitelesítés

A második diagram, lásd 4.6 ábra, révén szemléltetjük hogyan tud a felhasználó egy büntetést kiosztani. Készít egy képet a büntetendő személy vezetői engedélyéről, ezt feltölti egy API hívás segítségével, ezt a Backend átadja a Tesseract szövegfeldolgozónak. A kiszűrt szövegeket visszaküldi a felhasználónak illetve behelyettesíti az űrlap megadott szövegmezőit. Ezek ellenőrzése valamint a megmaradt szövegmezők kitöltése után a felhasználó elküldi az űrlapot ismét egy API hívás révén. Amint az adatok mentése megvalósult, a felhasználó kap egy visszajelzést mentés sikerességéről.



4.6 ábra. Űrlap kitöltése

4.5. A képfeldolgozó egység felépítése

A rendszer alapjául szolgáló egységet két fő részre tudjuk felosztani, egy amely a képfeldolgozást hajtja végre illetve a másik amely a képfeldolgozás eredményeként kapott szövegen hajt végre adatszűrést. A következőkben ezt a két egységet mutatjuk be illetve, hogy az egység által kinyert hasznos információkat milyen célokra használjuk fel.

4.5.1. Képfeldolgozás, szöveg kinyerése

A fő feldolgozást a Tesseract OCR hajtja végre az általunk felépített rendszerben viszont ez sem végez tökéletes munkát ezért implementáltunk egy kiegészítő eljárást amellyel a kép élességén igyekeztünk javítani. Mielőtt a képet átadnánk a beépített algoritmusnak a képen végrehajtottunk egy threshold-ot. A módszer lényege, hogy a képet bináris fájlként kezelve végrehajtottunk egy vágást egy megadott küszöbérték szerint a képpontban. Ezzel a megközelítéssel a képen élesebbek lettek a fehér illetve fekete képpontok közötti különbségek, jobban elkülönítve a szöveget a háttértől. A legmegfelelőbb küszöbérték meghatározás érdekében több tesztet is végeztünk különböző képeken illetve más-más küszöbértékkel azonban a 127-et használva értük el a legjobb eredményeket.

Ezen kívül azokon a képeken amelyek nem bizonyosultak eredményesnek megpróbáltunk adaptív vágást végezni Adaptiv Thresh Mean illetve Gaussian C eljárást használva. Az Adaptiv Thresh Mean C egy általunk megadott méretű ablakot vesz, és a küszöbértéket az ablak átlag értéke mínusz vagy plusz, esettől függően, a konstansként megadott C érték adja meg. A Gaussian esetében annyiban változik a módszer, hogy az ablakban található pixelértékek súlyozására Gauss-szűrést használ. Mivel ez a módszerek sem bizonyultak hasznosnak el is vetettük a használatukat és maradtunk az egyszerű threshold-nál.

4.5.2. Szöveg feldolgozása

Az adat felhasználást tekintve a rendszer fő modulja két célt szolgál, a felhasználó profiljának hitelesítése a webes felületen valamint a közúti büntetés kiosztásának a hatékonyabb illetve gyorsabb megoldása a mobilos alkalmazás révén. Egy személy azonosításának legáltalánosabb módszere a személyi igazolvány vagy vezetői engedély használata mivel ezek tartalmazzák minden lényeges adatot a személyről. Mivel a két igazolványon a használt minta más-más, a szövegből való adatok kiszűrése más-más metódust igényel.

Első lépésként a felhasználó által szolgáltatott igazolványt kategorizáltuk. Ez fontos volt, hogy a rendszer ne veszítsen a gyorsaságából az által, hogy minden lehetséges szűrést elvégez az adatok kinyerése érdekében. Ahhoz hogy vezetői engedélyként legyen feldolgozva a szóban forgó okmány, szerepelnie kell két szó közül valamelyiknek a szóban forgó igazolványon, a "PERMIS" vagy a "CONDUCTERE" kifejezés. A személyi esetében, a feldolgozáshoz szükséges fehér háttérű rész beazonosítása a cél. Ebben az esetben a szövegben meg kell jelenjen az "IDROU" vagy egy másik, "[0-9]ROU", reguláris kifejezés.

Amint megtörtént a kérdéses igazolvány besorolása, kezdődhet a specifikus feldolgozás.

4.5.2.1. Személyi igazolvány

A személyi igazolvány esetében sajnos az adatokat nem tudjuk kiszűrni olyan képpen hogy azok azonnal megfeleljenek számunkra a dizájn miatt. A személyi alsó részében viszont megjelenik egy nehezen értelmezhető, fehér háttérű rész. Alaposabb tanulmányozás után rájöttünk, hogy ebben a részben is megjelennek a személyes adataink, viszont más formátumban, például a személyi számot nem az általunk ismert formátumban jelenítik meg, hanem feldarabolva különböző részekre.

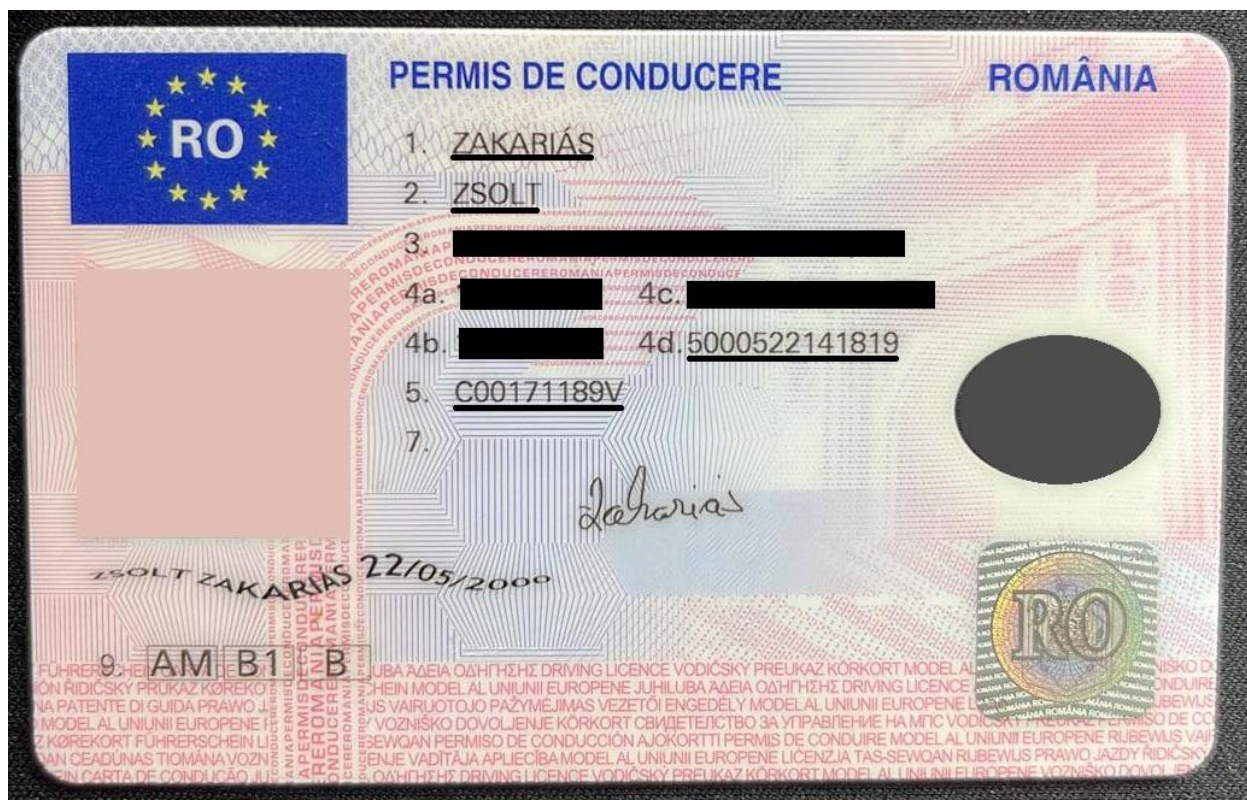
A következőkben bemutatott módszer könnyebb megértéséhez szolgáltatjuk a 4.7-es ábrán látható táblázatot valamint egy igazolvány példát amelyen a módszer levezetését végezzük, lásd 4.8 ábra.

Karakter Csoport	Adat
1-9	Megye betűkódja és személyi igazolvány száma
10	Érvényesítési kód az 1-9 csoporthoz
11-13	Állampolgárság
14-19	Születési dátum (ÉÉ-HH-NN)
20	Érvényességi kód a 14-19 csoporthoz
21	Nem (F/M)
22-27	A személyi érvényességnek határideje
28	Érvényességi kód a 22-27 csoporthoz
29	Megye betűkódja a személyi számnak
30-35	Személyi szám záró része
36	Érvényességi kód az 1-10, 14-20, 22-35 csoportoknak

4.7 táblázat. A személyi igazolványon megjelenő adatok

A folyamat első lépéseként reguláris kifejezés segítségével azonosítottuk, hogy a képfeldolgozás eredményeként kapott szövegben megjelenik-e a szükséges szövegrészlet. Ha ez megvolt, a szűrést folytattuk a személyi igazolvány megye betűkódja és számának a keresésével. Ezt követte az adott személy születési dátuma illetve az igazolvány megye betűkódja. Utolsó lépésként szükség volt a személyi szám záró részére. Mivel ennyire pontosan meg van határozva az adat szerkezete amire szükségünk van, a kép gyenge minősége miatti hibás karakter felismerés nem fogadható el. Ebben az esetben a felhasználónak szükséges egy új, jobb minőségű kép szolgáltatására.

Visszatérve az adatok szűrésére, a következő lépésben az igazolványt birtokló személy vezeték- illetve keresztnévét kerestük ki. Amint ezek az adatok megvannak az szövegfeldolgozás lezártnak tekinthető és a meglévő adatok felhasználhatóak a kitűzött célra, hitelesíteni az adott személyt vagy adott esetben a formanyomtatvány bizonyos sorainak kitöltése.



4.9 ábra. Vezetői engedély minta.

4.5.3. Kép- illetve szövegfeldolgozás tesztelése

A rendszer kidolgozása során több módszert is kipróbáltunk annak érdekében, hogy a lehető legtöbb esetben sikeres legyen a kép- illetve szövegfeldolgozás. A következőkben bemutatjuk az általunk implementált metódusokat, azok működését és, hogy mennyire voltak eredményesek. Azért ezekre a metódusok esett a választásunk mivel ezek a leggyakrabban használt módszerek ezen a területen, ahogyan az a mellékelt tanulmányban [3] is megjelenik . A módszerek bemutatása során a hangsúlyt maga a képfeldolgozásra szeretnénk helyezni mivel minden esetben a szöveg hasznos részének a szűrése ugyanazzal a módszerrel valósul meg. Az eredmények bemutatására szolgáló képek utolsó sorában

megjelenítsük az általunk felhasználhatónak tartott karakterláncokat egy JSON (JavaScript Object Notation) szöveg-formátumban.

4.5.3.1. 1.módszer - Grayscale, élesítő szűrő illetve Otsu küszöbérték használat

A rendszer kidolgozása során az első módszer amit teszteltünk az a kép szürkeárnyaltos megjelenítése (grayscale) amelyen használtunk egy "laplacian", ismertebb nevén "élesítő" szűrőt. Ezt követően egy bináris küszöbérték szűrést hajtottunk végre a képen ötvözve az Otsu küszöbérték meghatározására használt módszerével. A 4.10-es ábrán látható a módszer implementációja.

```
# # Method testing
# Grayscale, apply sharpening filter, Otsu's threshold
img = cv2.imread(filename, 0)
sharpen_kernel = np.array([[ -1, -1, -1], [ -1, 9, -1], [ -1, -1, -1]])
sharpen = cv2.filter2D(img, -1, sharpen_kernel)
temp, thresh = cv2.threshold(sharpen, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)

# OCR image process
img_method_final = pytesseract.image_to_string(thresh)
texts += img_method_final
ims = cv2.resize(thresh, (1560, 840))
cv2.imshow('Image Display', ims)
cv2.waitKey(0)
```

4.10 ábra. 1.módszer implementációja.

A kódrészletben a sharpen_kernel változóban került deklarálásra az élesítő szűrő melynek célja, hogy az eljárás során az éleket kiemelve és ezáltal kontrasztosabb hatást váltson ki a képen. Amint ez a lépés megvalósult jöhet a küszöbérték szerinti pixel szűrés, threshold. Ennek az eljárásnak az a célja, hogy a képen a pixeleket egy jól meghatározott küszöbérték szerint két csoportba rendezze. Azok a pixelek amelyek kisebbek mint a megadott küszöbérték 0-ás értéket míg a többi pixel 1-es értéket vesznek fel. Ennek kiegészítésére alkalmaztuk az Otsu küszöbérték meghatározás módszerét, amely a kép szerint automatikusan meghatározza a küszöbértéket. A kép hisztogramját használva a két legkimagaslóbb érték mediánját téríti vissza a további használatra. Az eljárás eredménye a következő képen látható, 4.11-es ábra, bal oldalon a személyi igazolvány míg a jobb oldalon

a vezetői engedély esetében. A szövegfeldolgozáshoz használt igazolványok fényképe, filterelés után, a 4.12-es ábrán tekinthető meg.

<pre>Plain Text: 00 00 00 Plain Text End {'lastName': '', 'firstName': '', 'personalCode': '', 'licenseCode': '', 'ciCode': ''}</pre>	<pre>Plain Text: EN (eodelal und (europene jubeubaaal VNAIRUOTOJO RAZYVE IIMAS VIZ 01 ENGEDELY MODE! OZMISKO DOVOLNIENIE: KORKOR ECRMETEIICTES A0 (! AEHAL NOAN PERMISO DE CONDUCCION AJOKORTTPERMIS DE CONDUIRE 1 00 AL UNTER EUR NE ACTA LIECIBANK BEL AL SAV EUROPENE LICENZ 4 Plain Text End {'lastName': '', 'firstName': '', 'personalCode': '', 'licenseCode': '', 'ciCode': ''}</pre>
----------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

4.11 ábra. Az 1.módszer végeredménye (bal oldal - személyi igazolvány, jobb oldal - vezetői engedély).



4.12 ábra. 1.módszer végrehajtása utáni igazolványok (bal oldal - személyi igazolvány, jobb oldal - vezetői engedély).

A személyi igazolvány esetében szemmel kivehető az információ viszont a vezetői engedély esetében ez már elég nehézkesen olvasható számunkra. Kijelenthető, hogy ez a megközelítés nem volt sikeres.

4.5.3.2. 2.módszer - Adaptiv Gauss Thresh Mean C

A következő módszer amit teszteltünk az a már említett Adaptív Gauss Thresh Mean C. A 4.13-as ábrán látható a módszer implementációja.

```
# # Method testing
# ADAPTIVE_THRESH_GAUSSIAN_C
img = cv2.imread(filename, cv2.IMREAD_GRAYSCALE)
thresh = cv2.adaptiveThreshold(img, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY, 11,2)

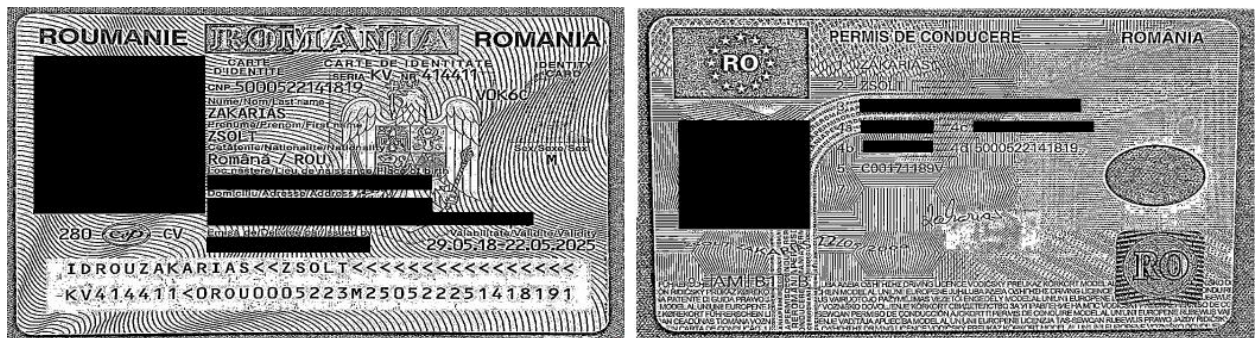
# OCR image process
img_method_final = pytesseract.image_to_string(thresh)
texts += img_method_final
imS = cv2.resize(thresh, (1560, 840))
cv2.imshow('Image Display', imS)
cv2.waitKey(0)
```

4.13 ábra. 2.módszer implementációja.

A képet több régióra osztja, jelen esetben 11*2-es blokkokra, majd az egyes régiókban külön-külön adaptív küszöbérték alkalmazódik. Az eljárás a Gauss-eloszlás alapján becslést használ a küszöbérték kiszámítására és ezáltal lehetővé teszi, hogy jobban kezelje a változó fényviszonyokat, különböző háttereket és kontrasztokat a képen. Az eljárás eredménye a 4.14-es ábrán látható. A szövegfeldolgozáshoz használt igazolványok fényképe a 4.15-ös ábrán tekinthetők meg. Sajnos ezen technika használata sem hozta el a várt sikert.

<pre>Plain Text: SROUANIE J iy) AUS SAN REN a os 'sa Hog ie Plain Text End {'lastName': '', 'firstName': '', 'personalCode': '', 'licenseCode': '', 'ciCode': ''}</pre>	<pre>Plain Text: Pe a ey ae AUN Mes ih aa re ne A AHI HEME ORIVING LICENCE VOIGCS RKC Lee IMODEAL UNIUWI EUROPENSJUHILUBA AAELA OSHHSHE ORIVING LICENCE for IRUOTONO PAZYMEINAS VEZETO) ENGEDELY MODOEAL UNIUWILEUROPENE Itaaerent VIP eat Plain Text End {'lastName': '', 'firstName': '', 'personalCode': '', 'licenseCode': '', 'ciCode': ''}</pre>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

4.14 ábra. A 2.módszer végeredménye (bal oldal - személyi igazolvány, jobb oldal - vezetői engedély).



4.15 ábra. A 2.módszer végrehajtása utáni igazolványok (bal oldal - személyi igazolvány, jobb oldal - vezetői engedély).

4.5.3.3. 3.módszer - Adaptiv Thresh Mean C

A harmadik módszer amit teszteltünk a már említett Adaptiv Thresh Mean C volt. A 4.16-os ábrán látható a módszer implementációja. A fentiekben leírt 2.módszertől annyiban különbözik ez, hogy a küszöbérték meghatározásához más, középértéken alapuló becslést használ.

```
# # Method testing
# ADAPTIVE_THRESH_MEAN_C
img = cv2.imread(filename, cv2.IMREAD_GRAYSCALE)
thresh = cv2.adaptiveThreshold(img, 255, cv2.ADAPTIVE_THRESH_MEAN_C, cv2.THRESH_BINARY, 11,2)

# OCR image process
img_method_final = pytesseract.image_to_string(thresh)
texts += img_method_final
imS = cv2.resize(thresh, (1560, 840))
cv2.imshow('Image Display', imS)
cv2.waitKey(0)
```

4.16 ábra. A 3.módszer implementációja.

Az eljárás eredménye a következő képen látható, 4.17-es ábra. A szövegfeldolgozáshoz használt fényképek a 4.18-as ábrán tekinthető meg.


```

Plain Text: MAN - -
oo " bE
Romana
ae 7 " - ' es 1)
fe: nS ae t i i ve < es 1
y Se ae Se = 2:05 ee
-i EUN

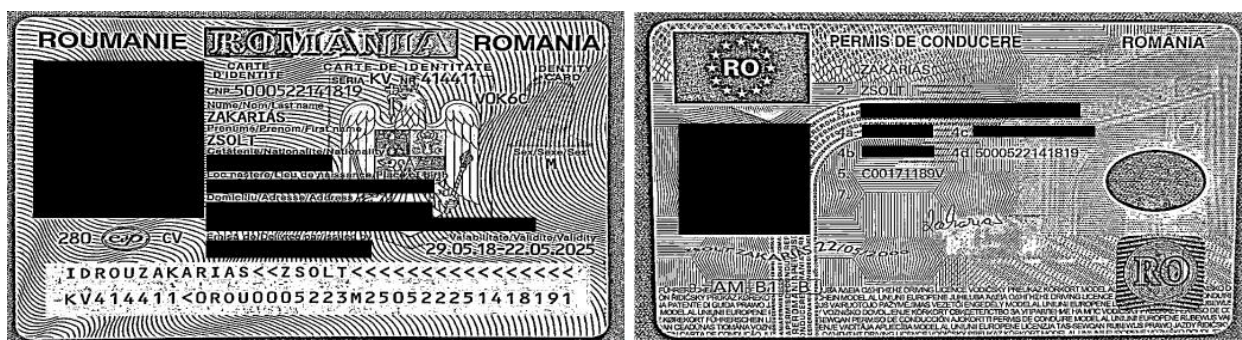
Plain Text End
{'lastName': '', 'firstName': '', 'personalCode': '', 'licenseCode': '', 'ciCode': ''}

Plain Text: SENG aol
DE-COND
Tan Te
1

Plain Text End
{'lastName': '', 'firstName': '', 'personalCode': '', 'licenseCode': '', 'ciCode': ''}

```

4.17 ábra. A 3.módszer végeredménye (bal oldal - személyi igazolvány, jobb oldal - vezetői engedély).



4.18 ábra. A 3.módszer végrehajtása utáni igazolványok (bal oldal - személyi igazolvány, jobb oldal - vezetői engedély).

4.5.3.4. 4.módszer - Simple threshold

Ez a módszer a már említett küszöbérték szerinti pixel szűrést, threshold-ot, alkalmazza viszont ebben az esetben egy előre jól meghatározott küszöbértéket használtunk, 127-et, az előzőekkel ellentétben ahol ennek meghatározása adaptív módon történt. A metódus

implementációja a 4.19-es ábrán látható.

```
# # Method testing
# SIMPLE_THRESH

img = np.array(Image.open(filename))
temp, thresh = cv2.threshold(img, 127, 255, cv2.THRESH_BINARY)

# OCR image process
img_method_final = pytesseract.image_to_string(thresh)
texts += img_method_final
imS = cv2.resize(thresh, (1560, 840))
cv2.imshow('Image Display', imS)
cv2.waitKey(0)
```

4.19 ábra. A 4.módszer implementációja.

A tesztelések beváltották a módszerhez fűzött reményeket, ahogy az a 4.20-as ábrán látható, mivel az eredményhalmazban megjelentek azok az adatok amelyek a rendszer további működéséhez szükségesek. Ennek a megvalósításához használt fényképek a 4.21-es ábrán tekinthető meg.

<pre>Plain Text: ROUMANIE itely /CZ\ A Romania 280 Ep) cv IDROUZAKARIAS<<ZSOLT<<<<<<<< KKK < <<< KV414411<OROU0005223M25052251418191 CARTE DE IDENTITATE IDENTITY DIDENTITE seria KV ona 414411; \ ARO enp 500052214181 NNN VP Name/Non/Last name = ZAKARIAS ape Non Prenume/Prenon/First S L Pass /) 1 ZSOLT ' MH aillttiet Ah CetatenieiNationalite/Natignatiy 1 fl Sex/Sexe/Sex Romana / ROU, Loc nastere/Lieu de nals angele [REDACTED] [REDACTED] - yN. [REDACTED] [REDACTED] 29.05. 18-22. 05. 2028 Plain Text End {'lastName': 'ZAKARIAS', 'firstName': 'ZSOLT', 'personalCode': '5000522141819', 'licenseCode': '', 'ciCode': 'KV414411'}</pre>	<pre>Plain Text: PERMIS DE CONDUCERE ROMANIA 1. ZAKARIAS 2. ZSOLT 3. [REDACTED] 4a. [REDACTED] 4c. [REDACTED] 4b. [REDACTED] 4d.5000522141819 5. C00171189V 7. rwN Sbhaios cer Za ARN 2210s 260% 3. AM: 01.. 0 Plain Text End {'lastName': 'ZAKARIAS', 'firstName': 'ZSOLT', 'personalCode': '5000522141819', 'licenseCode': 'C00171189V', 'ciCode': ''}</pre>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

4.20 ábra. A 4.módszer végeredménye (bal oldal - személyi igazolvány, jobb oldal - vezetői engedély).



4.21 ábra. A 4.módszer végrehajtása utáni igazolványok (bal oldal - személyi igazolvány, jobb oldal - vezetői engedély).

4.5.3.5. 5.módszer - Simple Grayscale Image Usage

Legvégső módszerként alkalmaztuk a kép sima szürkeárnyaltos használatát, vagyis nem végeztünk rajta semmilyen képfeldolgozási eljárást a minőség javulását illetően. Az implementáció a 4.22-es ábrán tekinthető meg.

```
# # Method testing
# OCR process without image preprocess
img = np.array(Image.open(filename))

# OCR image process
img_method_final = pytesseract.image_to_string(img)
texts += img_method_final
imS = cv2.resize(img, (1560, 840))
cv2.imshow('Image Display', imS)
cv2.waitKey(0)
```

4.22 ábra. Az 5.módszer implementációja.

Az eredményességet tekintve mindkét esetben, személyi igazolvány illetve vezetői engedély, az alkalmazott módszer megtalálta a legtöbb hasznos információt viszont itt is akadt eset amikor az eredményhalmaz nem volt teljes. A módszer végeredménye a 4.23-as ábrán található meg. A használt személyi igazolvány illetve vezetői engedély a 4.24-es ábrán látható.

Módszer	Keresett adatok					
	Vezetéknév		Keresztnév		Személyi szám	
	1	2	1	2	1	2
Grayscale, élesítő szűrő illetve Otsu	✗	✗	✗	✗	✗	✗
Adaptiv Gauss Thresh Mean C	✗	✗	✗	✗	✗	✗
Adaptiv Thresh Mean C	✗	✗	✗	✗	✗	✗
Simple threshold	✓	✓	✓	✓	✓	✓
Simple Grayscale Image Usage	✓	✓	✓	✗	✓	✓

4.25 tábla. A módszerek eredményessége a (1-személyi igazolvány, 2-vezetői engedély esetében.)

5. Felhasznált technológiák

A projekt kivitelezése során több technológiát is használtunk:

5.1. PyCharm

A PyCharm egy népszerű integrált fejlesztői környezet (IDE) a Python alapú alkalmazásfejlesztéshez. A PyCharm több olyan funkcióval is el van látva amelyek nagyban növelik a programfejlesztés hatékonyságát. Ilyen például a kényelmes fejlesztői felület amely szintakszis kiemelés révén segít a kód szerkesztésében illetve a hibák könnyű azonosításában. Ezen kívül igénybe vehetjük a kódkiegészítést illetve a az automatikus befejezést amely nem csak felgyorsítja a kód írást de minimalizálja a gépelési hibákat is. Ezeken felül rendelkezik integrált vezérlőkezelővel, például Git, lehetővé téve a hatékony változásokövetést, ág- valamint verziókezelési műveletek elvégzését az IDE-ből. Ezeken kívül még számos hasznos eszközt is igénybe vehetünk a használata során [4]. Maga a Python, mint programozási nyelv tűnt a legjobb választásnak a Backend fejlesztéshez mivel széleskörű palettával rendelkezik könyvtárak és modulok szintjén, platformfüggetlen illetve olvasható és egyszerű szintaxisokból épül fel a kód. A Backend fejlesztéséhez

felhasználtunk több könyvtárat is, ezek közül is kiemelten fontos a FastAPI, OpenCV, mysql illetve a Tesseract.

A FastAPI [5] egy modern, aszinkron webes keretrendszer amely kiváló teljesítményt nyújt, lehetővé téve a gyors válaszidejű és nagy terhelésű alkalmazások létrehozását.

Az OpenCV (Open Source Computer Vision) egy nyílt forráskódú könyvtár. A C++-ban implementált algoritmusok nagyfokú hatékonyságot biztosít illetve támogatja a többszálú feldolgozást. Tartalmaz alapvető műveleteket, például képvágás, átméretezés és kontrasztjavítás, valamint összetettebb funkciókat is például alakfelismerés, mozgásérzékelés, stb [6].

5.2. Android Studio

Az Android alkalmazás fejlesztéshez az IntelliJ IDEA alapú Android Studio integrált fejlesztői környezetet alkalmaztuk. A rugalmas Gradle-alapú build rendszer, a letisztult környezet, a funkciókban gazdag emulátor, stb [7] növelik a hatékony Android-alkalmazások fejlesztését. Az alkalmazás megvalósításához Kotlin nyelvet használtunk. Azért döntöttünk emellett a technológia mellett mivel a célközönség legtöbbször Android operációs rendszert támogató okostelefont használ.

5.3. Firebase

A Firebase [8] az egy Google által fejlesztett fejlesztői platform és eszközkészlet amely olyan szolgáltatásokat kínál amelyek segítenek az alkalmazások háttérrendszerének kialakításában és kezelésében. Könnyen integrálható a kliensoldali alkalmazásokkal és számos elkészített szolgáltatást és modult kínál. A mi esetünkben a képek tárolására a Firebase által kínált Storage, felhőalapú tárolót vettük igénybe a képek Backend-hez való eljuttatásához.

5.4. HTML

A HTML (HyperText Markup Language) egy olyan jelölő nyelv amelyet a weboldalak strukturálására és formázására használnak. Strukturált és szemantikus kódolást tesz lehetővé, a helyes elemek és címkék használata elősegíti a tartalom logikus elhelyezését, csoportosítását valamint a böngészők és keresőmotorok számára érthetőbbé teszi az oldalt és elősegíti a keresőmotoroptimalizációját.

A weboldal kidolgozásához felhasználtunk egy általunk megvett weboldal dizájnt amelyet átdolgozás után a saját tetszésünkre igazítottunk.

5.5. PHP

A PHP (HyperText Preprocessor) egy szerveroldali szkriptnyelv amelyet a dinamikus oldalak fejlesztésére használnak. Lehetővé teszi a tartalom és az adatok szerver oldalon történő dinamikus generálását. A PHP egy ismert bővítménye, a cURL (Client URL), révén valósítottuk meg a kommunikációt a webes applikáció és a BackEnd közt. A cURL kiterjedt és erőteljes funkcióhalmazzt kínál a hálózati kommunikációhoz.

5.6. Postman

A Postman [9] egy olyan fejlesztői eszköz és kliens alkalmazás amely az API-k fejlesztésében. Lehetőséget nyújt API konfigurációra amely révén beállíthatjuk a hívási paraméterek, a fejléc illetve a testadatokat valamint maga a kérés típusát is. Ezen kívül rendelkezésre áll automatizált tesztelés illetve a csoportos fejlesztés esetében használatos megosztás amely lehetőséget biztosít az API-konfigurációk illetve dokumentációk megosztásában.

5.7. GitHub

A GitHub [10] egy felhő alapú verziókezelő platform illetve fejlesztői közösség, amely segítségével a fejlesztők könnyebben együttműködhetnek, megoszthatják és nyomon követhetik a kódjaikat. Ezen felül a GitHub integrálható más fejlesztői eszközökkel mint például az általunk is használt PyCharm. A projekt táblák révén a felület lehetőséget nyújt a munkafolyamatok rögzítésére, követésére és rendszerezésére.

6. A rendszer által kínált további lehetőségek

6.1. Webes felület

Kijelenthetjük, hogy a rendszer alapját a képfeldolgozás adja, viszont annak érdekében, hogy egy pontosabb képet adjunk az általunk kigondolt rendszerre, megvalósítottunk további funkciókat is; habár ezek csak szimulációként működnek. A 4.9-es ábrán látható a webes felület oldalsó navigációs konzolja, amely kiválóan összefoglalja a rendszer összes funkcionálisát. A "Dashboard" fül alatt a felhasználó talál egy áttekintő részt a

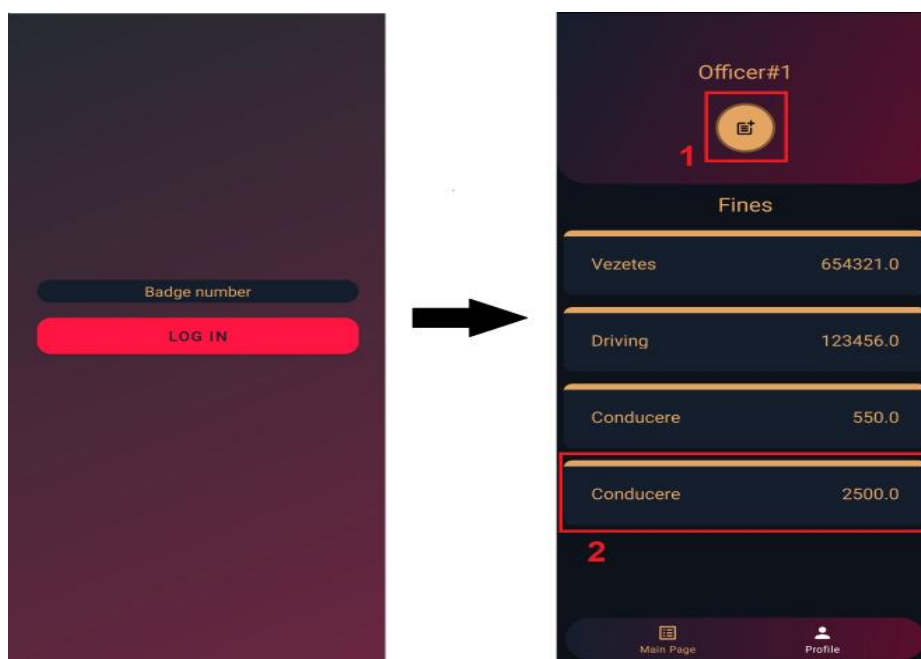
profilját illetően: jelenlegi egyenlege, a profilhoz csatolt bankkártyák illetve az eddigi tranzkciók listáját láthatja. A “Transactions” fülre kattintva megjelenik egy táblázat amely az eddigi összes banki utalást tartalmazza időrendi sorrendbe. A “Sanctions” résznél ugyancsak egy listát kapunk a meglévő büntetésekkel, illetve mindegyik esetében megtekinthető a bővebb leírás is. A “Deposit Money” illetve a “Withdraw Money” fülek alatt kezdeményezhet banki utalásokat. Az “Account” gomb révén megjelenik a felhasználó profil. Itt módosíthatja a megadott adatait, jelszót válthat illetve kezelheti a profiljához csatolt bankkártyákat valamint adhat hozzá újakat. Végző soron pedig, a “Quit” menüpontra kattintva kijelentkezhet a fiókjából.



6.1.1 ábra. Webes felületen megjelenő navigációs konzol.

6.2. Android applikáció

Az applikáció indítását követően bejelentkezésre van szükség. Ahogy azt a weboldal esetében is kifejtettük, egyfajta funkcionálisok csak demonstrációként vannak megvalósítva, hogy egy teljes képet adjon miként is képzeltük el a rendszert, ez a mobilos applikáció esetében is érvényes. A bejelentkezéshez egy ideiglenesen statikusra állított jelvéyszámra van szükség, amelyet a továbbfejlesztés során valódi jelvéyszámok cserélnénk le, ez által az alkalmazást használó hivatali személy a saját számát használva tudná az applikációt kezelni. A 6.2.1-es ábrán látható, hogy a bejelentkezést követően egy leegyszerűsített kezdőképernyő fogadja a felhasználót, ahol megtekintheti a legfrissebb kiosztott büntetéseket valamint kezdeményezhet új büntetés kiosztást a gombra kattintva, lásd 6.2.1-es ábra 1-es bekeretezett gombja. Amennyiben több információt szeretne megtudni az eddig a rendszerben lévő büntetéseket illetően a kiválasztott büntetésre koppintva, példa a 6.2.1-es ábra 2-es bekeretezett büntetés, ez egy új képernyőre navigálja ahol több információt szolgáltatunk a büntetést illetően, lásd 6.2.2-es ábra. Innen természetes bármikor a visszanavigálhat a kezdőképernyőre.

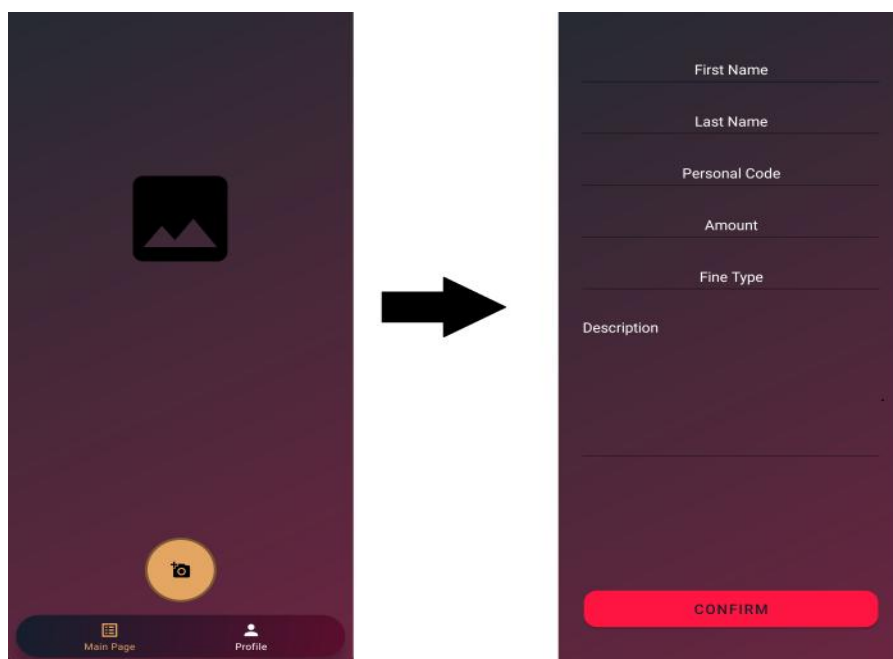


6.2.1 ábra. Android Applikáció bejelentkező felülete illetve az követő fogadó felülete (Home Screen) .



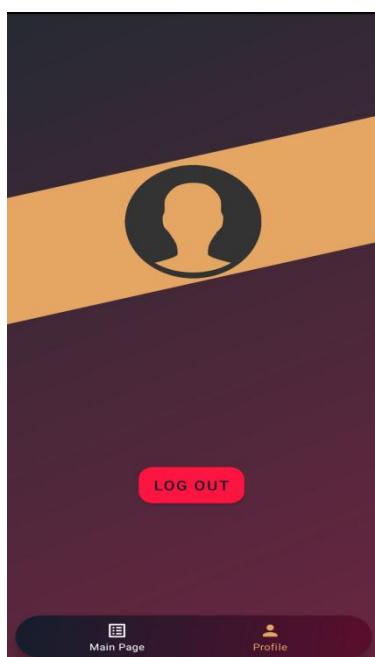
6.2.2 ábra. A kiválasztott büntetés bővebben való megtekintése.

Amennyiben egy új büntetést szeretne a felhasználó kiosztani abban az esetben átirányítjuk a következő képernyő nézetre, lásd 6.2.2-es ábra, ahol lehetősége adódik egy igazolványkép készítésére. A gombra kattintva megnyílik a hátlapi kamera és máris készítheti a fényképet. Amennyiben a készített fénykép minősége megfelelő a kamera nézetben ezt elfogadja az arra kialakított ikon vagy gomb megnyomásával és máris továbbítja azt a képet a rendszernek. Amint a rendszer megkapta és sikerült a feldolgozása átirányítjuk a felhasználót a következő illetve végső képernyőnézetre, az űrlap kitöltéséhez. A rendszer által üresen hagyott mezőket kitöltve illetve az általa megadottakat leellenőrizve és esetleges hibajavítást követően a “Confirm” gomb lenyomásával feltölti a büntetést és az bekerül a rendszerbe. Ezt követően visszaírányul a kezdőképernyőre.



6.2.3 ábra. A kiválasztott büntetés bővebben való megtekintése.

Pár képernyőnézetnél megfigyelhető a képernyő alján megjelenő navigációs pult amely könnyed elérést biztosít a kezdőképernyőhöz illetve a "Profil" képernyőnézethez melynek kezdetleges kialakítása a 6.2.3-as ábrán látható. Itt lehetősége nyílik a felhasználónak a kijelentkezésre.



6.2.4 ábra. A felhasználó profiljának megtekintése.

7. Fejlesztési lehetőségek

A projekt során kidolgozott rendszer esetében törekedtünk a lehető legtöbb funkcionalitást megvalósítani de így is maradtak olyan részek amelyek fejlesztésre szorulnak. A következőkben kisebb alfejezetekre bontva megpróbáljuk összegezni, hogy milyen fejlesztések azok amelyeket kigondoltunk de sajnos nem sikerült kivitelezni vagy amelyek lényegesen javíthatnak a rendszer minőségén és teljesítményén.

7.1. Webes felület

A jelenleg megvalósított webes felület tervezése és kivitelezése során próbáltuk a lehető legtöbb funkcionalitást beépíteni amely egy hatékony és letisztult legyen. Ennek kiegészítése egy valós idejű fizetési rendszer beépítésével sokban hozzájárulna egy teljes rendszer eléréséhez. Ezen kívül a biztonsági részt tekintve egy 2-lépcsős azonosító beépítése előnyt jelentene mivel így biztosítható hogy az adott fiókhoz csak a megfelelő személy tud hozzáférni.

7.2. Android Applikáció

Az android applikáció fejlesztéséhez nélkülözhetetlen egy adatbázis kiépítése a rendőrségi alkalmazottak tárolására. Ezt követően megvalósítható lenne a regisztráció valamint a belépés módszerének egy biztonságosabb formája.

Ezen kívül egy információgazdagabb "Profil" képernyőnézet kivitelezése is megvalósíthatóvá válna, hiszen meglenének a szükséges adatok az adatbázisban. Ezt kiegészítve egy a felhasználó teljeskörű szabadságot kaphatna a profiljának a módosításához.

Az android applikáció fejlesztéséhez nélkülözhetetlen egy adatbázis kiépítése a rendőrségi alkalmazottak tárolására. Ezt követően megvalósítható lenne a regisztráció valamint a belépés módszerének egy biztonságosabb formája.

Ezen kívül egy információgazdagabb "Profil" képernyőnézet kivitelezése is megvalósíthatóvá válna, hiszen meglennének a szükséges adatok az adatbázisban. Ezt kiegészítve egy a felhasználó teljeskörű szabadságot kaphatna a profiljának a módosításához.

Hivatkozások

- [1] Patel, C. I., Patel, D., & Patel, D.A. (2012, October). *Optical Character Recognition by Open source OCR Tool Tesseract: A Case Study*. ResearchGate. https://www.researchgate.net/profile/Chirag-Patel-12/publication/235956427_Optical_Character_Recognition_by_Open_source_OCR_Tool_Tesseract_A_Case_Study/links/00463516fa43a64739000000/Optical-Character-Recognition-by-Open-source-OCR-Tool-Tesseract-A-Case-St
- [2] Song, C., & Shmatikov, V. (2018). Fooling OCR Systems with Adversarial Text Images. <https://arxiv.org/pdf/1802.05385.pdf>
- [3] Jyothi, S. (2014, November). A Survey on Threshold Based Segmentation Technique in Image Processing.
- [4] JetBrains. (n.d.). *PyCharm: the Python IDE for Professional Developers by JetBrains*. JetBrains. Retrieved May 24, 2023, from <https://www.jetbrains.com/pycharm/>
- [5] Sebastián Ramírez. (n.d.). FastAPI. Retrieved May 24, 2023, from <https://fastapi.tiangolo.com/>
- [6] OpenCV. (n.d.). *OpenCV Documentation*. OpenCV Documentation. Retrieved May 24, 2023, from https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html
- [7] *Meet Android Studio*. (n.d.). Android Developers. Retrieved June 18, 2023, from <https://developer.android.com/studio/intro>
- [8] Google. (2019, May 7). . Firebase Documentation. Retrieved May 24, 2023, from https://firebase.google.com/docs?gad=1&gclid=CjwKCAjw67ajBhAVEiwa2g_jEIYaCHLBGI_X_DsJUEaTpJNWW8uv1heaFbkYMAbLjocY7sFXgciZgMBoCXfEQAvD_BwE&gclsrc=aw.ds
- [9] Google. (2019, May 7). . Firebase Documentation. Retrieved May 24, 2023, from https://firebase.google.com/docs?gad=1&gclid=CjwKCAjw67ajBhAVEiwa2g_jEIYaCHLBGI_X_DsJUEaTpJNWW8uv1heaFbkYMAbLjocY7sFXgciZgMBoCXfEQAvD_BwE&gclsrc=aw.ds

[10] <https://www.hostinger.com/tutorials/what-is-github>

UNIVERSITATEA SAPIENTIA DIN CLUJ-NAPOCA
FACULTATEA DE ȘTIINȚE TEHNICE ȘI UMANISTE, TÎRGU-MUREȘ
SPECIALIZAREA CALCULATOARE

Vizat decan
Conf. dr. ing. Domokos József

Vizat director departament
Ș.l. dr. ing. Szabó László Zsolt