



Univerza v Ljubljani
Fakulteta *za elektrotehniko*

Android aplikacija Face remember, REST API storitev in spletni administratorski vmesnik

Projekt pri predmetu: Seminar iz načrtovanja in
razvoja programske opreme v telekomunikacijah
Študijsko leto 2021/2022

Luka Podbregar, 64160122

Kazalo

Kazalo	2
1. Povzetek	3
2. Uvod	4
3. Arhitektura	5
3.1 Zaledje	5
3.2 Čelni del	6
3.2.1 Spletni administratorski vmesnik	6
3.2.2 Mobilna aplikacije	9
4. REST API vmesnik	16
5. Zaključek	18
6. Uporabljene knjižnice	19

1. Povzetek

V zaključnem semestru magistrskega študija elektrotehnike smo pri predmetu Seminar iz načrtovanja in razvoja programске opreme v telekomunikacijah razvijali mobilno Android aplikacijo, ki vključuje spletno REST storitev in vsah dva odjemalca. Za svoja odjemalca sem izbral mobilno aplikacijo, kateri sem posvetil večino svojega časa, in spletni administratorski vmesnik. Uporabniki mobilne aplikacije lahko naložijo slike obrazov oseb, ki jih poznajo in s temi slikami nato igrajo igro ugibanja imen. Igra se imenuje Face remember.

2. Uvod

Po podatkih World health organisation-a (WHO) je na svetu 55 milijonov ljudi z demenco. Vsako leto se to število poveča še za približno 10 milijonov. Znanstveno je dokazano, da lahko s spodbujanjem možganov upočasnimo potek te bolezni ali pa celo pravočasno preprečimo njen razvoj. Zdravniki priporočajo razne miselne igre, ki spodbujajo mišljenje. Najbolj pogoste so kakšne sestavljanke, križanke ali pa razne igre za spomin.



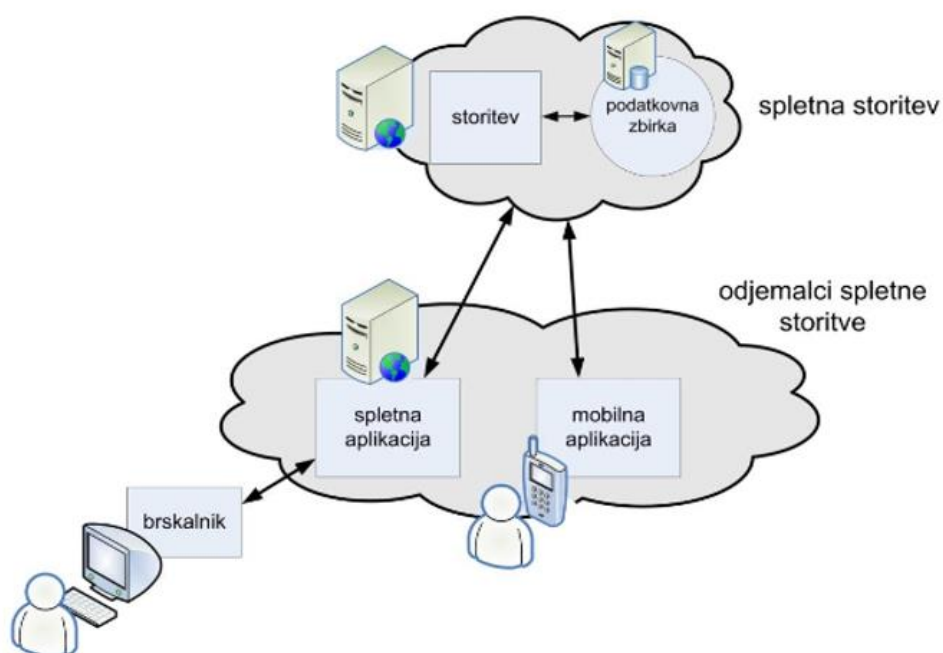
Slika 1: Primer miselne igre

Za temo svoje projektne naloge sem izbral izdelavo aplikacije, ki bi delovala na principu spodbujanja spomina. Uporabniki lahko naložijo slike obrazov svojih bližnjih ali pa samo ljudi, ki jih poznajo. Iz teh slik pa je nato narejena igra, pri kateri mora uporabnik izbrati pravo ime izmed 4 možnosti. Z izbiro pravega imena, uporabnik zbira točke. Pripravljena je tudi že igra z obrazi znanih ljudi, ki bi omogočala kot preizkus aplikacije za nove uporabnike.

3. Arhitektura

Ker sem v bil v tej smeri začetnik, je struktura mojega projekta zasnovana v podobni obliki, kot je bilo predstavljeno na laboratorijskih vajah. Sestavljena je iz zaledja (spletne storitve) in čelnega dela (odjemalci spletne storitve) (slika 2). Odjemalca storitve sta bila spletni administratorski vmesnik in mobilna aplikacija, kateri sem posvetil največ časa.

Za samo strukturo zaledja so bili uporabljeni programski jeziki: PHP, SQL in JavaScript. Za administratorski spletni vmesnik sta bila uporabljena programska jezika: PHP in JavaScript, ter označevalni jezik HTML. Za mobilno aplikacijo sem uporabil programski jezik Java for Android.



Slika 2: Arhitektura sistema

3.1 Zaledje

Zaledje je sestavljeno iz Apache serverja in podatkovne zbirke MySQL, ki teče z uporabo aplikacije XAMPP. Moja podatkovna zbirka, imenovana application, je sestavljena iz 4 različnih tabel, vsaka s svojo nalogo (slika 3).

<input type="checkbox"/> images	★	Browse	Structure	Search	Insert	Empty	Drop	15	InnoDB	utf16_general_ci	32.0 KiB	-
<input type="checkbox"/> imagespreset	★	Browse	Structure	Search	Insert	Empty	Drop	20	InnoDB	utf16_general_ci	16.0 KiB	-
<input type="checkbox"/> randomnames	★	Browse	Structure	Search	Insert	Empty	Drop	47	InnoDB	utf16_general_ci	16.0 KiB	-
<input type="checkbox"/> users	★	Browse	Structure	Search	Insert	Empty	Drop	66	InnoDB	utf16_general_ci	16.0 KiB	-

Slika 3: Podatkovna baza application

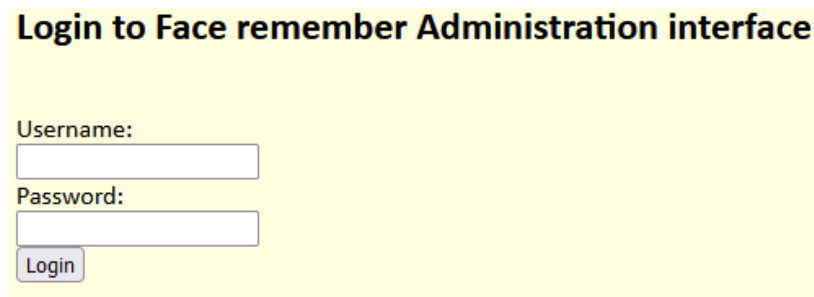
- Tabela **users** vsebuje podatke o uporabnikih
 - username: ime uporabnika, ki ga določi sam pri prvem vpisu
 - password: geslo, ki je v podatkovni bazi zakodirano
 - userID: unikatno število uporabnika, uporabljeno kot primarni ključ
- Tabela **images** vsebuje podatke o slikah, ki so jih naložili uporabniki. Sestavljena je iz:
 - userID: unikatno število uporabnika, ki mu pripada ta slika
 - imageID: unikatno število slike, uporabljeno kot primarni ključ
 - imageName: ime osebe na sliki
 - gender: spol osebe na sliki
 - path: pot do slike na strežniku
- Tabela **randomnames** vsebuje naključna imena, ki so uporabljena v igri
 - name: naključno ime, uporabljeno kot primarni ključ
 - gender: spol tega imena, uporabljeno pri igri, saj je za vsako sliko treba dobiti 3 naključna imena istega spola kot je oseba na sliki
- Tabela **imagespreset** vsebuje imena in slike znanih oseb, ki so uporabljena pri že pripravljeni igri
 - imageID: unikatno število slike, uporabljeno kot primarni ključ
 - imageName: ime osebe na sliki
 - gender: spol osebe na sliki
 - path: pot do slike na strežniku

3.2 Čelni del

3.2.1 Spletni administratorski vmesnik

Spletni administratorski vmesnik je namenjen pregledu nad uporabniki npr. če slučajno nalagajo kakšne neprimerne slike. Administrator ima možnost kreiranja novega uporabnika, pregled uporabnikov, pregled nad slikami uporabnikov, naloži pa lahko tudi sliko določenemu uporabniku.

Začne se s prijavo (*slika 4*). Za prijavo uporablja php seje, ki onemogočajo dostop uporabnikom, ki nimajo administratorjevega uporabniškega imena in gesla. Uporabniško ime in geslo sta za prikaz oba **admin**.



Login to Face remember Administration interface

Username:

Password:

Login

Slika 4: Prijava v administratorski vmesnik

Po prijavi se prikaže izbirni meni (*slika 5*), ki omogoča dostop do vseh funkcionalnosti administratorskega vmesnika.

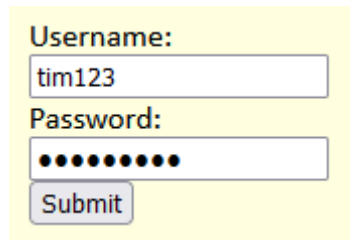
Face remember- administration interface

[Create new User](#) [Upload new Image](#) [Users](#) [User Images](#)

[Logout](#)

Slika 5: Izbirni meni

Create new User (*slika 6*) omogoča kreiranje novega uporabnika.



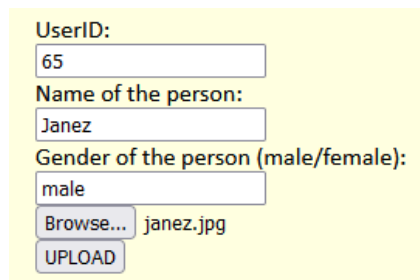
Username:
tim123

Password:
●●●●●●●●

Submit

Slika 6: Kreiranje novega uporabnika

Upload new Image (*slika 7*) omogoča, da administrator naloži sliko za določenega uporabnika.



UserID:
65

Name of the person:
Janez

Gender of the person (male/female):
male

Browse... janez.jpg

UPLOAD

Slika 7: Naloži novo sliko

Users izpiše vsa imena uporabnikov in njihov userID.

User images (slika 8) omogoča, da administrator pridobi uporabnikov žeton (token), napiše le njegov userID, ki ga lahko pridobi pod **Users**. S tem žetonom lahko nato dostopa do njegovih slik. **Pozor:** žeton je v narekovajih, ki jih je za dostop do slik treba ignorirati!

Fetch user token (ignore ""):

UserID:

Fetch

"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2V"

Fetch user Images:

Token:

Search

 Jan

David



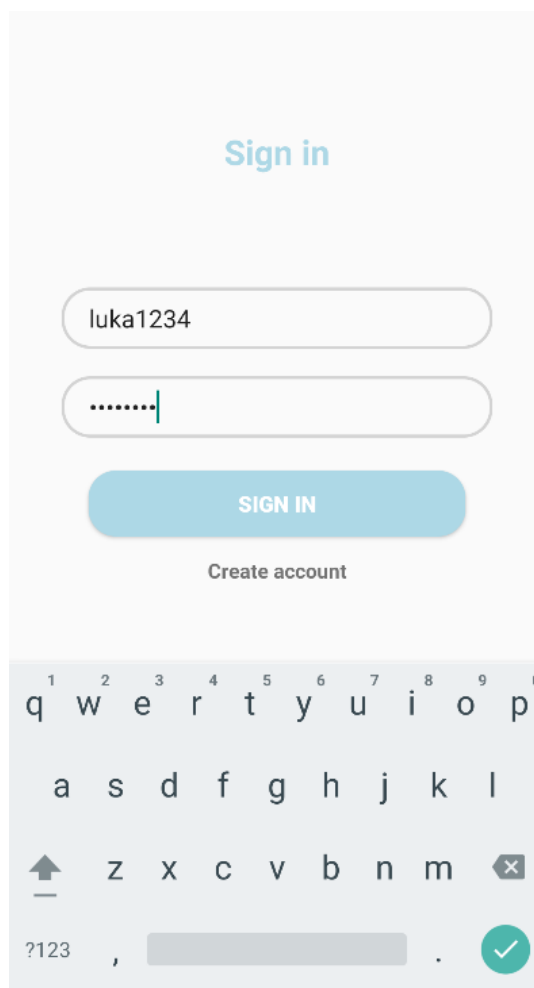
Slika 8: Pregled uporabnikovih slik

Logout zaključi sejo in izpiše administratorja.

3.2.2 Mobilna aplikacija

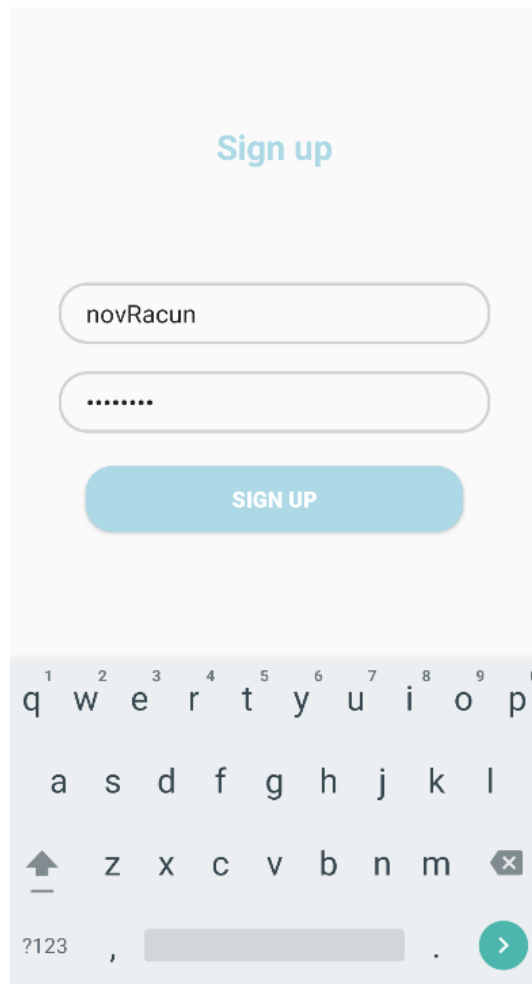
Največ časa sem posvetil razvoju aplikacije, ki bi jo uporabniki uporabljali za igranje igre spomina z raznimi slikami. Pri zagonu aplikacije nas pričaka vpisno okno, ki omogoča vpis že registriranim uporabnikom (*slika 9*). To se zgodi z uporabo razreda SigninAPI, ki pošlje zahtevo z uporabniškim imenom in zakodiranim geslom na strežnik (v obliki JSON objekta). Nazaj dobi enega iz treh različnih odgovorov: vpis uspešen, narobno geslo ali pa uporabnik ne obstaja. Če je vpis uspešen uporabniški terminal prejme tudi svoj žeton, ki ga uporablja dokler je prijavljen v aplikacijo. Žeton se shrani v podatkovno bazo TinyDB, ki sem jo v aplikacijo vključil z uporabo knjižnice TinyDB.

Za svojo aplikacijo sem izbral žetone JWT (JSON Web Token, ki sem jih vključil s pomočjo knjižnice ReallySimpleJWT), ki omogočajo preprosto generiranje novih in validacijo uporabljenih žetonov. Knjižnica TinyDB omogoča preprosto shranjevanje in dostopanje do podatkov, ki morajo biti shranjeni tekom aplikacije.



Slika 9: Vpisno okno aplikacije

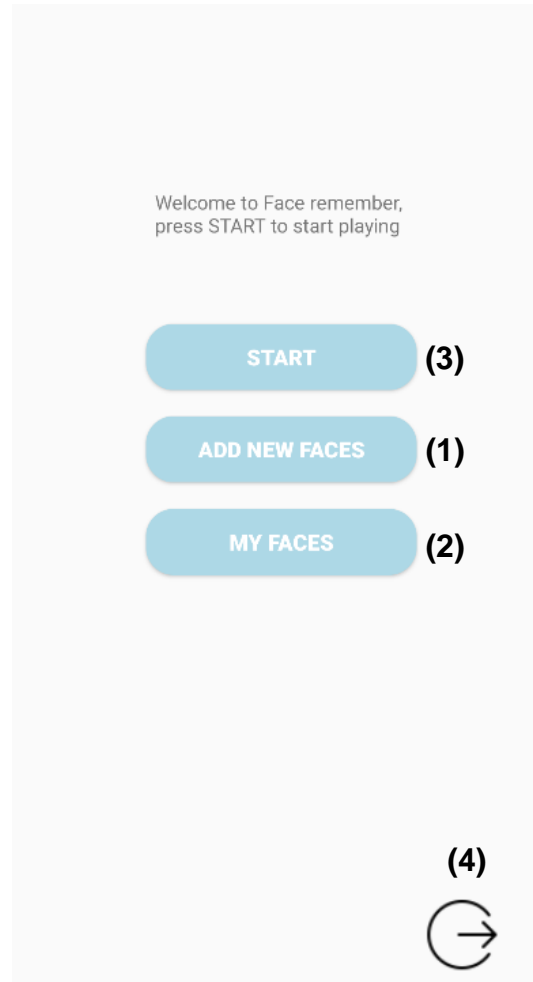
Če uporabnik še nima svojega računa, ga lahko ustvari s klikom na "Create account" (*slika 10*). Zahteva se pošlje z uporabo razreda SignupAPI, ki na podoben način kot SigninAPI pošlje novo uporabniško ime in zakodirano geslo. V odgovor dobi ali je uporabniško ime dovoljeno (še ni uporabljeno) in če je bilo kreiranje novega uporabnika uspešno. Uporabnik je po kreaciji računa preusmerjen na vpisno okno.



Slika 10: Ustvari nov račun

Po prijavi nas pričaka osnovni meni, ki nam omogoča izbiro različnih akcij. V nadaljevanju bom opisal postopek uporabe aplikacije za prvega uporabnika.

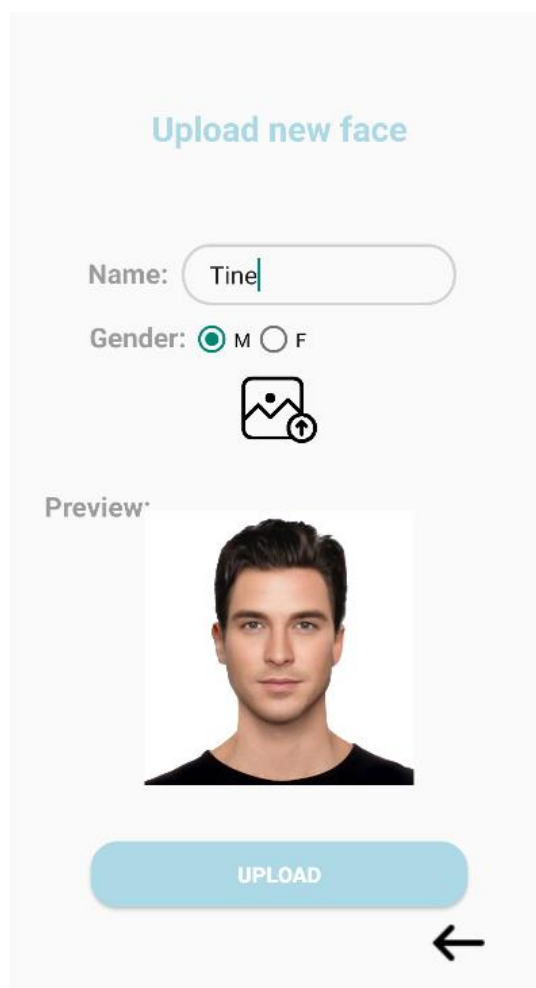
- **Dodajanje novega obraza (slike) (1)**
- **Pregled že dodanih obrazov (knjižnica obrazov) (2)**
- **Začetek nove igre (3)**
- **Izpis (4)**



Slika 11: Izbirni meni aplikacije

1. Dodajanje novega obraza

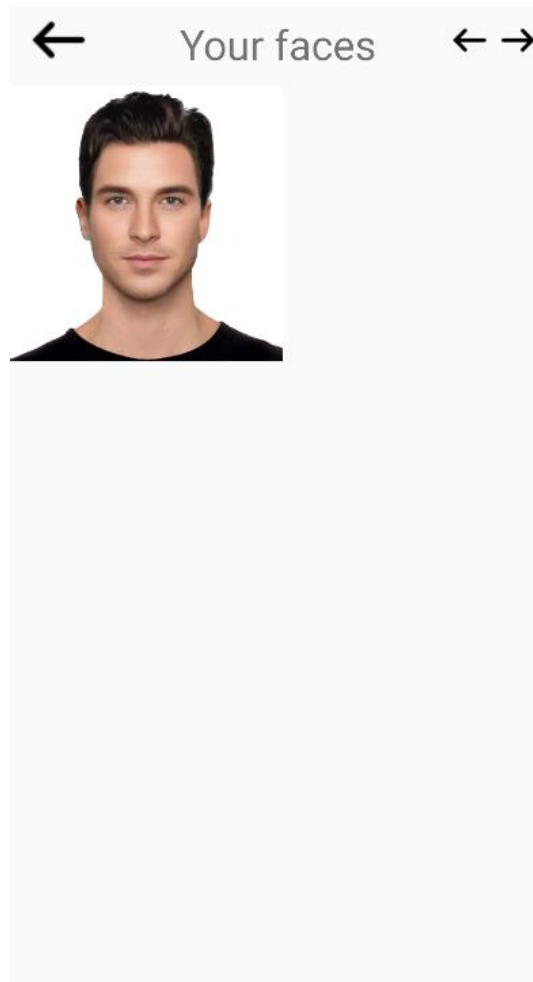
S klikom na "Add new face" smo preusmerjeni na okno, kjer lahko vpišemo ime obraza, spol in iz galerije na svojem mobilnem terminalu izberemo sliko, ki jo želimo naložiti. Ko je slika izbrana, se nam prikaže pod "Preview". Izbrani morajo biti vsi vhodi, drugače akcija ne bo dovoljena. Z gumbom na puščico lahko akcijo prekličemo. Podatki so poslani s pomočjo razreda ImageUploadAPI, ki pošlje podatke in sliko, zakodirano v base64 obliki. Ta se nato dekodira in shrani na strežniku. "Preview" slike (in vsi ostali prikazi slik) je prikazan z uporabo knjižnice Glide.



Slika 12: Dodajanje novega obraza

2. Pregled že dodanih obrazov (knjižnica obrazov)


S klikom na "My faces" smo preusmerjeni v knjižnico, kjer so prikazani naši naloženi obrazi. Če bi imeli več naloženih obrazov, bi lahko s puščicami v desnem zgornjem kotu listali med stranmi (prikazanih je 10 slik na stran). S klikom na puščico v levem zgornjem kotu se vrnemo nazaj na glavni meni. Slike pridobimo s strežnika z uporabo razreda imagesAPI.



Slika 13: Knjižnica obrazov


Knjižnica omogoča tudi urejanje podatkov posamezne slike in brisanje. S klikom na sliko smo preusmerjeni na okno, ki dovoli spreminjanje in brisanje s pomočjo razreda `imageUpdateAPI` ter `imageDeleteAPI`.


Change the name and gender of the person!



Name:

Gender: ☒ M ☐ F



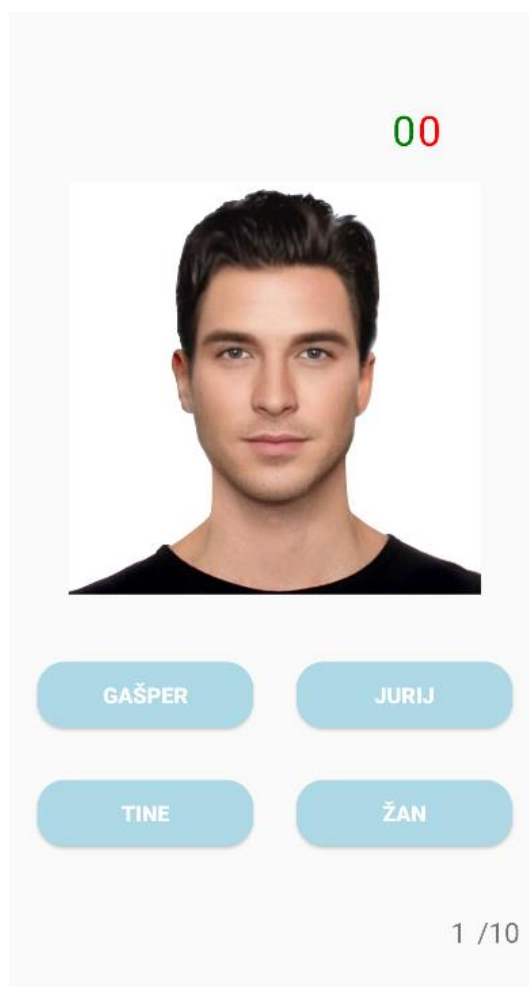


Slika 14: Urejanje slike v knjižnici

3. Začetek nove igre

S klikom na "Start" smo preusmerjeni na okno, kjer mammo na izbiro 2 načina igre: igra z uporabnikovimi obrazi (Custom game) ali pa igra z znanimi obrazi (Preset game). Če uporabnik še ni naložil nobenega obraza, "Custom game" ne bo dovoljen.

Ko se igra začne, se uporabniku prikaže naključno izbrana slika, z naključno pomešanimi 4 imeni (pridobljeni z uporabo razreda namesAPI), eno izmed njih je pravilno. Če uporabnik zbere pravilno ime, dobi točko za pravilen odgovor (zeleno). Če pa ne, pa dobi točko za napačen odgovor (rdečo). Igra traja največ 10 različnih obrazov. Če ima uporabnik naloženih manj kot 10 obrazov, bo igra trajala toliko dokler ne zmanjka njegovih obrazov.



Slika 15: Primer igre

4. REST API vmesnik

Za izvajanje REST API klicev sem, kot na vajah, uporabil PHP. Vsi klici, njihov opis in poti se nahajajo v tabeli spodaj.

Pot	Metoda	Medijski tip	Vsebina zahtevka	Opis
/application/users	GET	JSON	/	Pridobi podatke o vseh uporabnikih
/application/users/{userID}	GET	JSON	/	Vrne uporabniško ime in zakodirano geslo uporabnika z {userID}
/application/users	POST	JSON	{"username": "____", "password": "____"}	Doda novega uporabnika v zbirko uporabnikov
/application/images/{token}	GET	JSON	/	Vrne imena, spol in pot do slik uporabnika z žetonom {token} (preveri tudi, če je žeton veljaven)
/application/images	GET	JSON	/	Vrne imena, spol in pot do slik znanih oseb (že naloženih)
/application /imagesRandom/{token}	GET	JSON	/	Vrne imena, spol in pot do slik uporabnika z žetonom {token} (tudi preveri, če je žeton veljaven) v naključnem vrstnem redu
/application/login	GET	JSON	/	Vrne imena in userID od vseh uporabnikov
/application/login /{username}/{password}	GET	JSON	/	Preveri, če se vpisni podatki ujemajo s tistimi v podatkovni bazi. Če se, vrne žeton
/application/names	GET	JSON	/	Vrne vsa imena v podatkovni bazi
/application/names/{gender}	GET	JSON	/	Vrne 10 imen spola {gender}

/application /imageUploadAndroid/{token}	POST	JSON	{ "image": "____", "name": "____", "gender": "____", "extension": "____"} }	Vnese ime in spol osebe na sliki v podatkovno bazo in sprejme sliko v base64 formatu. To sliko nato shrani v "extension" formatu v mapo userImages na strežniku. Uporabnik je validiran in določen z uporabo žetona {token}
/application /imageUpdate/{token}	GET	JSON	{ "imageName": "____", "gender": "____", "path": "____"} }	Validira žeton {token} in nato posodobi podatke v podatkovni bazi z novo prejetimi
/application/imageDelete /{token}&{imagePath}	DELETE	JSON	/	Validira žeton {token} in nato zbríše podatke slike s potjo {imagePath} ter jo zbríše iz strežnika
/application /fetchUserTokenAdmin /{userID}	GET	JSON	/	Vrne žeton od uporabnika z {userID}. Deluje samo za administratorja v veljavni seji v administratorskem vmesniku

5. Zaključek

V projektni nalogi sem se naučil precej novih tehnologij. Bil sem namreč čisti začetnik v razvoju aplikacij in zalednih sistemov. Največ težav sem imel z shranjevanjem in nalaganjem slik na in iz strežnika. Na koncu sem se odločil za transformacijo v base64 format, za lažji prenos do strežnika, in nato shranjevanje iz base64 nazaj v datoteko slike. Za prenos slik iz Android aplikacije sem uporabil knjižnico okhttp3.

S svojim projektom sem le opraskal vrh potenciala, ki ga ima ideja te aplikacije. Razširil bi jo namreč lahko na slike znanih mest, znamenitosti, slik znanih slikarjev in mnogo drugih. Mogoče tudi na kratke gif odseke iz znanih filmov ali pa celo zvočne posnetke. Ciljnih uporabnikov mi verjetno tudi nebi primanjkovalo, saj je, kot sem omenil v uvodu, na svetu približno 50 milijonov ljudi z demenco. Večino je seveda starejših, vendar ima vedno več starejših ljudi pametne telefone, ki jih vedno bolje znajo uporabljati.

6. Uporabljene knjižnice

- Glide (<https://bumptech.github.io/glide/>)
- Okhttp3 (<https://square.github.io/okhttp/4.x/okhttp/okhttp3/>)
- TinyDB (<https://tinydb.readthedocs.io/en/latest/index.html>)
- Really Simple JSON Web Tokens (<https://github.com/RobDWaller/ReallySimpleJWT>)