

IMPLEMENTACIJA PROGRAMSKE PODRŠKE ZA PREPOZNAVANJE GOVORA I UPRAVLJANJE PRISTUPOM ZAŠTIĆENOJ ZONI**IMPLEMENTATION OF NATURAL SPEECH RECOGNITION SOLUTION AND PROTECTED AREA ACCESS CONTROL**

Duško Ožegović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – MEHATRONIKA

Kratak sadržaj – Bezbednost predstavlja jednu od osnovnih potreba čoveka. Zahvaljujući razvoju tehnologije došlo i do napredka u polju bezbednosti. U okviru ovog rada vršena je implementacija modernih tehnologija u svrhu povećanja sigurnost zaštićene zone. Izvršena je implementacija programske podrške za prepoznavanje izgovorene kodne fraze od strane korisnika koji želi da uđe u zaštićenu zonu, kao i web stranice za unos i manipulaciju podacima, pri čemu web stranice imaju komunikaciju sa blockchainom, koji predstavlja jednu od najmodernijih tehnologija za decentralizovano čuvanje podataka.

Ključne reči: *Prepoznavanje govora, JavaScript Decentralizovana Aplikacija, Sigurnosni Sistemi, Blockchain.*

Abstract – *Safety is one of the basic human needs. Technology development has also brought advancements in field of security. This thesis aims to implement modern technologies in purpose of increasing safe zone security. A user spoken code phrase recognition software has been developed for access permission to a safe zone alongside a series of web pages for data entry and manipulation where web pages are linked with blockchain, a state of art technology for decentralised data storage.*

Keywords: *Speech recognition, JavaScript, Decentralised Application, Security Systems, Blockchain.*

1. UVOD

Sigurnost predstavlja jednu od osnovnih potreba čoveka. U svrhu obezbeđivanja sigurnosti, angažuju se ljudi, zaduženi za kontrolu pristupa zgradama, firmama, kao i nekim posebno značajnim prostorijama. Zahvaljujući napretku tehnologije, danas je moguće zameniti čoveka, koji je podložan greškama i umoru, nekim sigurnosnim sistemom.

Ovakvi sistemi su bezbedniji, a i finansijski isplativiji, tako da se sve više koriste. Pored kontrole ulaska, takođe nude i mogućnost beleženja informacija o tome ko je i kada ušao u prostoriju.

Ova funkcionalnost može biti od koristi firmama, kako bi imali evidenciju o tome kada su zaposleni došli na posao ili imali pristup nekoj prostoriji.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Bojan Mrazovac, docent.

Ovakvi sistemi takođe omogućavaju i vrlo jednostavnu zabranu pristupa određenim korisnicima, jer administrator sistema može da kontroliše dozvole pristupa. U ovom radu predstavljena je realizacija sigurnosnog sistema, baziranog na konceptu pametnih vrata koja koriste digitalni ključ, formiran na osnovu kodne fraze, jedinstvene za svakog korisnika i biometrijskog podatka, geometrije lica. Pri formiranju sistema, korišćeni su *blockchain* [1] i *IPFS* [2], koje predstavljaju najmodernija rešenja za decentralizovano čuvanje podataka. Fokus ovog rada je na izradi programske podrške za prepoznavanje kodne fraze, aplikacije za unošenje podataka korisnika i aplikacije za kontrolu pristupa korisnika, od strane administratora sistema.

Pametna vrata su u velikoj meri zastupljena, kako u stambenim zgradama, tako i u poslovnim objektima. Korisnici uglavnom koriste poseban privezak, *smartphone* aplikaciju ili *QR* kod za otvaranje ovakvih vrata. Prednost virtualnog ključa je što je lako podeliti ključ korisnicima van sistema, korišćenjem *e-maila*, *sms-a*, itd. Ili zabraniti pristup postojećim korisnicima. Neka pametna vrata imaju ugrađenu *Wi-Fi* konekciju koja omogućuje monitoring pristupa, zahvaljujući obaveštenjima o pristupu vratima, kao i korišćenje kamera kako bi se proverilo ko pokušava da otvori vrata, pogotovo kada se pametna vrata koriste u kombinaciji sa pametnim zvonom. U tom slučaju zvono se aktivira kada neko priđe vratima i administrator sistema ili neko ko ima mogućnost otvaranja vrata može pogledati ko hoće da otvori vrata. Trend u industriji je težnja ka tome da se koriste biometrijski podaci, odnosno prepoznavanje lica i glasa, kako bi se povećala sigurnost i praktičnost sistema. U tom slučaju pristup ima samo osoba kojoj je to dozvoljeno, a ne osoba koja je došla do digitalnog ključa, a takođe je moguće otvoriti vrata bez korišćenja *smartphone-a*, priveška [3].

2. KORIŠĆENE TEHNOLOGIJE

Za razvoj programske podrške za prepoznavanje izgovorene fraze i *web* stranica za unošenje i manipulaciju podacima projektovanog sigurnosnog sistema koriste se sledeće tehnologije:

2.1. Python

Python je programski jezik visokog nivoa opšte namene. Podržava, u prvom redu imperativni, objektno-orijentisan i funkcionalni stil programiranja. Sintaksa jezika *Python* omogućava pisanje veoma preglednih programa. Programi pisani u *Python* jeziku se najčešće interpretiraju. Uz

interpreter se obično isporučuje i veoma razvijena standardna biblioteka modula [4].

Interpreteri i standardne biblioteke modula se stalno razvijaju i prenose na veliki broj različitih platformi. Postoje i kompajleri, koji *Python* kod prevode u mašinski jezik. Oni omogućavaju da se stvore izvršni moduli nezavisni od interpretera, te da se programi izvršavaju brže. Ono što se gubi ovakvim prevođenjem je prenosivost ovako prevedenog programa na različite hardverske i softverske platforme.

Ukoliko je potrebno razviti novi modul, to je moguće izvesti u *Python*-u ili u nekom drugom podržanom jeziku. Obično je to *C* jezik, kada je potrebno sistemsko programiranje ili kada je brzina izvršavanja kritična. Primeri drugih programskih jezika, koji se koriste u ove svrhe, su *Java* ili *Pyrex* jezik koji predstavlja mešavinu programskog jezika *Python* i *C* jezika.

U ovom radu korišćeno je i višenitno programiranje. Nit (eng. *Thread*) je odvojeni tok izvršavanja. To znači da će se u programu istovremeno događati dve stvari. Korisno je da u toku izvršavanja programa rade dva ili više različitih procesora, i pri tome svaki od njih nezavisno obavlja svoj zadatak, ali to nije u potpunosti moguće. *Thread*-ovi se mogu izvoditi na različitim procesorima, ali se moraju pokretati jedan po jedan.

2.2. Javascript

JavaScript je interpreterski skriptni programski jezik koji se uklapa u *ECMAScript* specifikaciju. Objektno je orijentisana, koristi sintaksu sa vitičastim zagradama i podržava dinamičko kucanje. Uz *HTML* i *CSS*, *JavaScript* je jedna od glavnih tehnologija za "World Wide Web". *JavaScript* omogućava interaktivne *web* stranice i esencijalni je deo *web* aplikacija. Većina *web* stranica je koristi i poseduje „engine” namenjene za izvršavanje *JavaScripta* [5].

Najčešće korišćenje *JavaScript*-a prilikom izrade *web* stranica je da se dodaku "client-side" mogućnosti *HTML* stranicama, takođe poznatim kao *Dynamic HTML* (*DHTML*). Skripte su ugrađene ili uključene sa *HTML* stranica i vrše interakciju sa *Document Object Model*-om (*DOM*) stranice.

2.3. PHP

PHP je jedan od najpopularnijih jezika koji se koriste za razvoj *web* aplikacija. On omogućava programeru da brzo razvije dobro formirane programe bez greške, koristeći tehnike proceduralnog i objektno-orijentisanog programiranja. Obezbeđuje mogućnost upotrebe mnogih postojećih biblioteka koda koje su uključene u osnovnu instalaciju ili mogu da se instaliraju unutar *PHP* okruženja [6].

Iako se *PHP* može koristiti za programiranje konzolnih aplikacija i grafičkih interfejsa (biblioteka *PHP-GTK*), njegova osnovna i glavna upotreba je u programiranju dinamičnih stranica na internetu.

Program koji se napiše u *PHP*-u ne zahteva prevođenje (kompajliranje), nego se interpretira pri svakom izvršavanju. *PHP* interpretator može raditi po *PHP CGI* principu, odnosno tako što će interpretator postojati kao eksterna aplikacija, koja se poziva da izvrši datu skriptu svaki put kad bude zahtevana od nekog korisnika, a može biti instaliran i kao modul *web*-servisa.

Druga varijanta je danas u najvećoj upotrebi jer pruža znatno veću brzinu izvršavanja - interpretator je na taj način uvek učitao u memoriju te se ne mora pozivati spoljašnji program.

2.4. web3.js

Digitalna imovina, kao i "pametni" ugovori su glavne komponente *blockchain*-a. Kako bi se došlo u kontakt sa komponentama u "lancu", transakcije se moraju kreirati *blockchain*-u. Da bi korsinik ili softver van lanca treirali transakciju na *blockchain*-u, čvor mora da prenese transakciju na "peer to peer" mrežu. *Web3.js* je kolekcija biblioteka koja omogućava programerima da vrše interakciju sa ostalim komponentama u lancu, tako što ima olakšava konekciju sa *Ethereum* čvorovima.

Web3.js je popularna biblioteka koja omogućava programerima interakciju sa *Ethereum blockchain*-om. Povezuje *JavaScript* jezik sa *Ethereum*-ovim *JSON RPC* interfejsom, što je čini direktno upotrebljivom u *web* tehnologijama. *Web3.js* se takođe često koristi i na serverskoj strani unutar *Node.js* aplikacija i nekih desktop aplikacija baziranih na *Electron*-u [7].

3. POVEZIVANJE SISTEMA

3.1. Povezivanje programske podrške

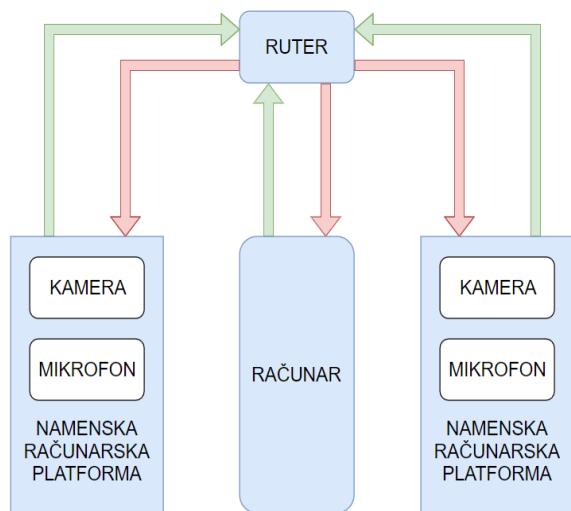
Programska podrška za prepoznavanje lica je razvijen na bazi neuronskih mreža *FaceNet*. Nakon obuke modela na privatnom skupu slika sistem za prepoznavanje lica je sposoban za validaciju korisnika. Kada je očitano lice, ono se pretvara u odgovarajući oblik podataka, da bi se poredilo sa podacima koji se nalaze u modelu. Ova programska podrška izrađena je u vidu *Python* skripte.

Kako bi radili programska podrška za prepoznavanje lica i programska podrška za prepoznavanje izgovorene kodne fraze zajedno u realnom vremenu, koristi se *multithreading*. *Multithreading* je organizovan tako da se *thread*-ovi pozivaju iz *main* funkcije, pri čemu je jedan *thread* glavni, a drugi se poziva po potrebi (nalazi se u *sleep* modu). U glavnom *thread*-u izvršava se kod za prepoznavanje lica, a u spodenom kod za prepoznavanje govorne fraze.

Korsinčke podatke je potrebno skladištiti u decentralizovanoj bazi podataka, kako bi u svakom momentu bili dostupni svim uređajima koji su povezani na mrežu. U tu svrhu koristi se *Etherium blockchain*, dok se za skladištenje većih podataka, u ovom slučaju koristi *IPFS*. Za povezivanje *web* aplikacija sa *blockchain*-om korišćena je *JavaScript* biblioteka *web3.js*. *Web3.js* je popularna biblioteka koja omogućava programerima interakciju sa *Ethereum blockchain*-om. Povezuje *JavaScript* jezik sa *Ethereum*-ovim *JSON RPC* interfejsom, što je čini direktno upotrebljivom u ovom slučaju.

3.2. Povezivanje fizičke arhitekture

Komponente fizičke arhitekture sistema su ruter, računar i razvojna okruženja (u ovom projektru korišćeni su *MinnowBoard* i *Raspberry Pi*) na koji putem *USB* priključka priključeni kamera i mikrofon. Na slici 1. Prikazana fizička arhitektura kompletnog sistema povezana u celinu, pri čemu je crvenim i zelenim linijama ilustrovana komunikacija između pojedinačnih komponenti.



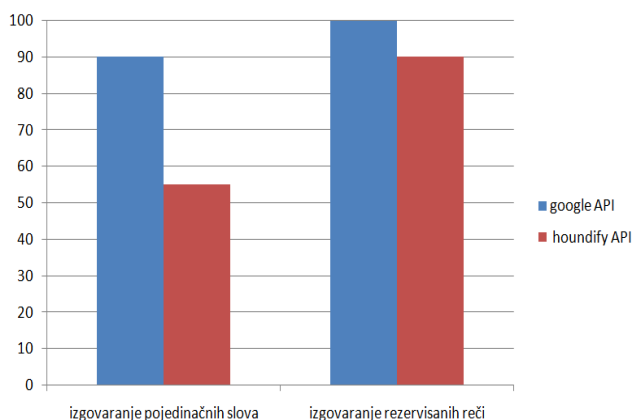
Slika 1. Prikaz načina povezivanja fizičke arhitekture

4. REALIZACIJA UPRAVLJANJA

4.1. Realizacija programske podrške za prepoznavanje izgovorene kodne fraze

Za realizaciju softvera za prepoznavanje izgovorene kodne fraze korišćena je *Python* biblioteka *speechrecognition* [8] koja ima mogućnost prepoznavanja govora i konvertovanja izgovorenih reči u tekst. Postoje mnoge alternative: *apiai*, *pocketsphinx*, *wit* i druge. Neke od njih, kao što su *wit* i *apiai* imaju ugrađene dodatne mogućnosti koje idu i dalje od običnog prepoznavanja govora, na primer mogućnost određivanja namera korisnik. Ipak, odlučeno je da se koristi biblioteka *speechrecognition*, zbog jednostavnosti korišćenja.

U okviru rada testirane su dva *API*-ja za prepoznavanje izgovorenih reči. U pitanju su *google API* i *houndify API*. Testiranje je rađeno u dva slučaja: prilikom izgovaranja pojedinačnih slova i prilikom izgovaranja rezervisanih reči za slova. Rezultati su prikazani na slici 1.



Slika 2. Rezultati testa

Rezultati testiranja pokazuju da *google API* postiže bolje rezultate, pa je odlučeno da se on koristi prilikom realizacije finalne aplikacije.

4.2. Realizacija web aplikacije za unošenje podataka

Aplikacija za unošenje podataka korisnika je vrlo jednostavna. Njena namena je da novi korisnik unese korisničko ime i lozinku koji će se kasnije koristiti u sistemu.

Da bi se ova aplikacija povezala sa *blockchain*-om, korišćena je biblioteka *web3.js*. Da bi se povezivanje izvršilo potrebno je u *JavaScript* kod stranice uneti *ABI* (*Application Binary Interface*) koji služi da poveže *web* aplikaciju sa *blockchain* aplikacijom.

Kako bi se slike koje će se uneti na ovoj *web* aplikaciji uspešno poslale na *IPFS* potrebno je formirati folder na osnovu korisničkog imena u kojoj će one biti sačuvane. Ta operacija ne može biti izvršena korišćenjem *JavaScript*-a jer iz bezbednosnih razloga *JavaScript* ne može da radi sa podacima sa korisničkog računara. U tu svrhu korišćena je *PHP* skripta.

4.3. Web aplikacija za manipulaciju podacima

Prilikom izrade ove aplikacije korišćeni su *JavaScript* i *HTML*, a za komunikaciju sa *blockchain*-om, koristi se, kao i u ranijim slučajevima, biblioteka *web3.js*.

Kada korisnik, u ovom slučaju administrator sistema, klikne na dugme nekih vrata biraće između dve opcije: da dozvoli/izmeni mogućnost pristupa nekim vratima ili da zabrani pristup. Ukoliko izabere opciju da dozvoli/imeni mogućnost pristupa, unosi vreme od kad do kad je omogućen pristup. Takođe, administrator ima uvid u kodne fraze korisnika. Sve funkcije na ovoj stranici implementirane su *JavaScript* kodu.

5. ANALIZA RADA APLIKACIJE

U realizaciji sistema pametnih vrata korišćena su dve različita namenske računarske platforme za upravljanje vratima. Rad aplikacije se analizirao sa *Raspberry Pi 3+* [9] kao uređajem koji upravlja vratima, a zatim *Minnowboard Turbot B* [10].

Prilikom testiranja rada aplikacije na namenskoj računarskoj platformi *Raspberry Pi 3+* utvrđeno je da prepoznavanje lica radi uspešno, pri čemu programska podrška za prepoznavanje lica radi na oko 3.4 FPS, što je dovoljno za stabilan rad. Sistem je takođe uspešno obavio prepoznavanje izgovorene kodne fraze.

Prilikom testiranja rada aplikacije na namenskoj računarskoj platformi *Minnowboard Turbot B* system je bez ikakvih poteškoća uspešno izvršio sve zadatke, pri čemu je programska podrška za prepoznavanje lica radila na 6.4 FPS, što je znatno bolje u odnosu na test sa *Raspberry Pi 3+*.

6. ZAKLJUČAK

U ovom radu realizovan je sigurnosni sistem pametnih vrata koji na osnovu biometrijskih karakteristika lica i kodne fraze jedinstvene za svakog korisnika određuje da li korisnik ima pravo pristupa. Kada korisnik priđe vratima, vrši se prepoznavanje lica, a zatim ukoliko je lice prepoznato vrši se prepoznavanje izgovorene kodne fraze. Ovi podaci se šalju decentralizovanoj bazi podataka gde se vrši poređenje podataka na osnovu čega se utvrđuje da li korisnik ima pravo pristupa u datom momentu. Decentralizovana baza podataka realizovana je korišćenjem *Ethereum blockchain*-a.

Programska podrška za prepoznavanje izgovorene kodne fraze radi na zadovoljavajućem nivou u uslovima u kojima su vršena testiranja, na obe namenske računarske platforme. Da bi se uspešno koristila u drugim uslovima potrebno je ponovo izvršiti podešavanja filtracije buke u prostoriji.

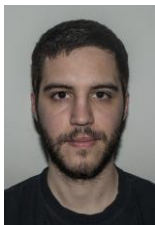
Moguće unapređenje sistema bilo bi da se na osnovu dodatnih testiranja utvrdi spisak rezervisanih reči za koje bi se minimalizovala nastajanje grešaka prilikom prepoznavanja ili formiranje sopstvenog sistema za prepoznavanje izgovorene fraze koji bi radio na osnovu [11] pomoću koga bi se sistem mogao realizovati za srpski jezik formiranjem modela za svako slovo pojedinačno.

Postoji i mogućnost formiranja sistema koji prepoznaje govornika na osnovu *GMM*-ova (Gaussian Mixture Model) [12]. Takođe, jedno od mogućih proširenja jeste i da se omogući naknadno unošenje slika korisnicima, kako bi mogli da dopune bazu podataka i posle kreiranja svog naloga na sistemu.

6. LITERATURA

- [1] <https://en.wikipedia.org/wiki/Blockchain>, (pristupano u oktobru 2019.)
- [2] [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language)) (pristupano u novembru 2019.)
- [3] https://en.wikipedia.org/wiki/Smart_lock (pristupano u oktobru 2019.)
- [4] [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language)) (pristupano u novembru 2019.)
- [5] <https://en.wikipedia.org/wiki/JavaScript> (pristupano u oktobru 2019.)
- [6] Matt Doyle: Beginning PHP 5.3
- [7] <https://www.mycryptopedia.com/what-is-web3-js-a-detailed-guide/> (pristupano u novembru 2019.)
- [8] https://github.com/Uberi/speech_recognition/blob/master/reference/library-reference.rst, (pristupano u novembru 2019.)
- [9] https://en.wikipedia.org/wiki/Raspberry_Pi (pristupano u novembru 2019.)
- [10] <https://minnowboard.org/minnowboard-turbot/technical-specs> (pristupano u oktobru 2019.)
- [11] Wendel, Tyler James, M.S. Feature Extraction and Feature Reduction for Spoken Letter Recognition
- [12] <https://appliedmachinelearning.blog/2017/11/14/-spoken-speaker-identification-based-on-gaussian-mixture-models-python-implementation/> (pristupano u novembru 2019.)

Kratka biografija:



Duško Ožegović rođen je u Novom Sadu 1995. god. Master rad na Fakultetu tehničkih nauka iz oblasti mehatronika odbranio je 2020.god.

kontakt: d.ozegovic1995 @gmail.com