

IoT projekat - Virtual Doorman

Filip Cvetkovic RI 17/2021

Opis sistema

Projekat modeluje integrisani IoT sistem za kontrolu pristupa i nadgledanje prostora pomoću RFID tehnologije i detekcije lica. Sistem omogućava automatsku verifikaciju identiteta koristeći RFID skenere, čime se povećava bezbednost objekata. Implementiran je na Raspberry Pi uređaju koji služi kao centralni server, povezan sa senzorima i uređajima.

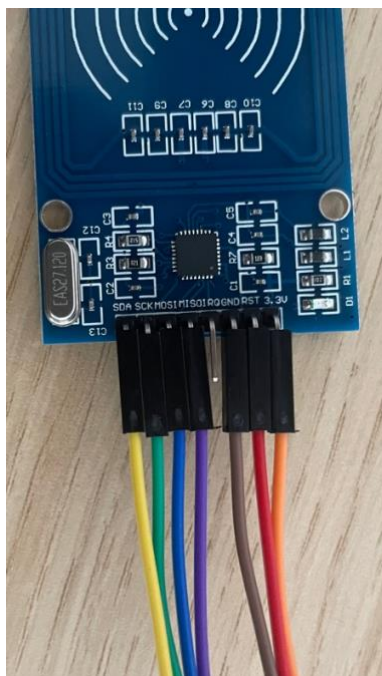
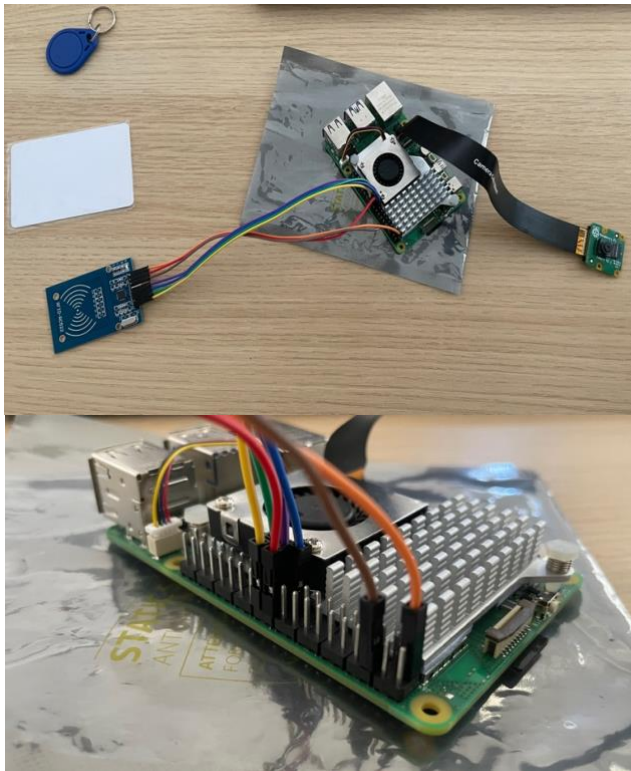
Ovaj sistem koristi hardverske komponente kao što su RFID čitači (model MFRC522) i Raspberry Pi kameru sa softverskim rešenjima za obradu slike. Web aplikacija, razvijena u Flask okruženju, omogućava korisnicima da na intuitivan način upravljaju sistemom, registruju i prate aktivnosti uređaja, te pregledaju statističke podatke i logove pristupa u realnom vremenu. Podaci o pristupima se čuvaju u Firebase bazi podataka, što omogućava brz i efikasan pristup i analizu podataka.

Pored osnovnih funkcija kontrole pristupa, sistem podržava napredne funkcionalnosti kao što su automatska detekcija i mogućnost daljinskog upravljanja i nadzora preko interneta. Implementacija takođe obuhvata razvoj korisničkih interfejsa prilagođenih za desktop i mobilne platforme.

Arhitektura sistema

Sistem se sastoji od Raspberry Pi uređaja, RFID skenera, kamera i web servera. Raspberry Pi služi kao lokalni server koji obrađuje podatke i komunicira sa Firebase bazom podataka. Web servis je implementiran u Python koristeći Flask framework, a korisnička aplikacija je web aplikacija.

Web servis je razvijen kao RESTful API, što omogućava laku integraciju s različitim klijentskim aplikacijama i platformama. Podaci se bezbedno čuvaju u cloud-based Firebase bazi podataka. Nudeći čist i intuitivan korisnički interfejs koji olakšava dnevno upravljanje i nadzor sistema.



RC522	GPIO Pin #	GPIO Pin Name
3.3V	1	3.3v DC Power
RST	22	GPIO 25
GND	25	Ground
IRQ	None	Disconnected
MISO	21	GPIO 9
MOSI	19	GPIO 10
SCK	23	GPIO 11
SDA	24	GPIO 8

Skladište podataka

Skladište podataka je realizovano korišćenjem Firebase baze podataka kao ključne komponente za čuvanje i upravljanje podacima. Ova baza je konfigurisana za skladištenje identifikacionih podataka iz RFID čitača i podataka detekcije lica. Sistem koristi real-time bazu podataka koja omogućava trenutni pristup i ažuriranje podataka.

Firebase baza podataka je odabrana zbog svoje skalabilnosti, pouzdanosti i jednostavnosti upotrebe. Omogućava brzo postavljanje i lako integraciju sa Raspberry Pi uređajem i web aplikacijom kroz API-je.

U bazi podataka se čuvaju sledeći tipovi podataka:

1. **Identifikacioni Podaci:** Uključujući jedinstvene ID brojeve RFID tagova i povezane korisničke informacije koje su kritične za proces identifikacije.
2. **Podaci Detekcije Lica:** Analitički podaci dobijeni iz sistema za detekciju lica, koji se pri prebrojavanju u bazu unose tačan broj ljudi ispred zgrade.
3. **Logovi Prolaza:** Vremenski pečati prolaza osoba kroz kontrolisane tačke, što omogućava praćenje kretanja unutar objekta i analizu pristupnih obrazaca.

Struktura podataka je dizajnirana da podržava efikasno upravljanje velikim volumenima podataka i brzi pristup informacijama koje su potrebne za odlučivanje u realnom vremenu. Pored osnovnih funkcija skladištenja, Firebase nudi i napredne mogućnosti kao što su automatizovane sigurnosne kopije, upravljanje pristupom na nivou podataka i integraciju sa drugim cloud servisima, što dodatno pojačava funkcionalnost i sigurnost sistema.

Web servis

Web servis je implementiran koristeći Flask framework, što omogućava izgradnju robustnog API-a za interakciju sa klijentskim aplikacijama i uređajima. API servis omogućava slanje i prijem podataka između korisničkih interfejsa i centralnog skladišta podataka. Strukturiran je tako da podržava različite HTTP metode, uključujući GET za dohvaćanje podataka i POST za slanje novih podataka ili ažuriranja.

API pruža sledeće funkcionalnosti:

1. **Registracija Uređaja:** Korisnici mogu registrovati nove uređaje u sistem. API prihvata POST zahtev sa neophodnim informacijama o uređaju, uključujući identifikacione podatke i konfiguracijske parametre.
2. **Logovanje Pristupa:** API podržava logovanje svakog pristupa uređaju, pri čemu se zapisuje timestamp i identitet korisnika. Ovo se postiže slanjem POST zahteva nakon svake konfiguracije.

3. **Dohvatanje Podataka:** API omogućava dohvaćanje istorijskih podataka o pristupima uređajima kroz GET zahteve, omogućavajući analizu i monitoring aktivnosti unutar sistema.

4. **Detekcija Lica:** Integracija sa sistemom za detekciju lica omogućava verifikaciju identiteta korisnika putem kamere.

Korisnička aplikacija

Korisnička aplikacija IoT sistema služi kao glavno sredstvo za interakciju između korisnika i uređaja unutar sistema. Aplikacija je razvijena koristeći web tehnologije HTML, CSS i JavaScript, što omogućava dinamički i responzivan korisnički interfejs prilagođen desktop tako i mobilnim uređajima.

Glavne Karakteristike:

1. **Pregled i Upravljanje Uređajima:** Korisnici mogu u realnom vremenu pregledati stanje povezanih uređaja, kao i upravljati njima. Uključuje funkcije poput registracije novih uređaja, ažuriranja njihovih konfiguracija.

2. **Geolokacijske Funkcije:** Aplikacija uključuje geolokacijske funkcije koje omogućavaju da se provere lokacije svih uređaja.

3. **Notifikacije i Upozorenja:** Korisnici dobijaju realno-vremenske notifikacije o važnim događajima unutar sistema, kao što su tehnički problemi sa uređajima.

Tehnologije:

- **HTML i CSS:** Za strukturu i stil aplikacije, omogućavajući moderno i atraktivno korisničko iskustvo.

- **JavaScript:** Za dinamičke interakcije unutar aplikacije, obradu podataka u realnom vremenu i komunikaciju sa backend servisima preko AJAX-a i Fetch API-a.

- **Responsive Design:** Koristeći CSS Frameworks poput Bootstrap-a, aplikacija je optimizovana za rad na različitim uređajima i veličinama ekrana, što je čini dostupnom i na mobilnim telefonima i tabletima.

Implementacija:

Web aplikacija je hostovana na lokalnom serveru koristeći raspberry pi i **ngrok** kros-platformske mogućnosti za tunelovanje lokalnog server, osiguravajući visoku dostupnost i pouzdanost sistema. Za razvoj je korišćen razvojni stack koji je lako skalabilan, omogućavajući lako dodavanje novih funkcionalnosti i integracija u budućnosti. Takođe je kroz **ngrok** omogućen statičli domain name kojem se pristupa preko QR koda.

12:12

f755-24-135-92-95.ngrok-free.app

Register Your Device

Device ID:

Location:

Description:

Register Device



Virtual DoormanRFID LogsFace Detection Logs

Face Detection Logs

0 faces detected

Call the Police

Virtual DoormanRFID LogsFace Detection Logs

RFID Entry and Exit Log

ID	Name	Status	Timestamp	Duration Inside
288070244888		Entered	2024-05-21 22:12:18	---
756107451172	Filip Cvetkovic	Entered	2024-05-21 22:12:12	---

Uređaj

IoT sistem koristi Raspberry Pi 5 kao centralnu jedinicu za obradu podataka i koordinaciju uređaja unutar sistema. Raspberry Pi 5 je odabran zbog svoje moćne procesorske sposobnosti, podrške za različite vrste komunikacija i pristupačnosti.

Konfiguracija i Funkcionalnosti:

1. **RFID Čitač:** Uređaj je opremljen RFID čitačem koji omogućava brzu i pouzdanu identifikaciju korisnika putem RFID kartica. Čitač je povezan sa Raspberry Pi preko SPI interfejsa, što osigurava efikasnu i sigurnu komunikaciju.
2. **Kamera za Detekciju Lica:** Integrirana kamera se koristi za implementaciju funkcionalnosti prepoznavanja lica. Kamera snima video u realnom vremenu i koristi algoritme za prepoznavanje lica.
3. **Automatizacija:** Uređaj automatski obrađuje podatke sa RFID čitača i kamere, i na njih, omogućava ili onemogućava pristup. Svi događaji se zapisuju u lokalnu bazu podataka, a relevantni podaci se šalju na centralni server za dalju analizu.

Softverska Implementacija:

- **Python:** Glavni programski jezik korišćen za razvoj softvera uređaja. Python je odabran zbog svoje jednostavnosti, velike podrške za biblioteke i zajednicu koja nudi razne module za rad sa hardverom.
- **GPIO Povezivanje:** GPIO portovi na Raspberry Pi se koriste za povezivanje i kontrolu RFID čitača i drugih senzora. Korišćenje GPIO portova omogućava direktan nadzor i kontrolu fizičkih interakcija uređaja.
- **Automatizacija Proces:** Skripte pisane u Pythonu automatski upravljaju procesima čitanja RFID kartica i analize video snimaka, te na osnovu definisanih pravila, upravljaju pristupom korisnika.

Prilozi

Projekat je organizovan u folderima kako bi se osigurala jasna struktura i lako održavanje koda. Ova organizacija takođe pomaže u navigaciji i razumevanju projekta za sve koji s njim rade. Detaljan opis strukture foldera i sadržaja je sledeći:

1. **web_server:**
 - Sadrži Flask aplikaciju koja funkcioniše kao web server. Ovaj folder uključuje sve Python skripte koje upravljaju logikom web servera, kao i konfiguracione fajlove potrebne za njegovo pokretanje i održavanje.
2. **device_code:**

- Ovaj folder sadrži sve Python skripte koje se izvršavaju na Raspberry Pi uređaju. Skripte obuhvataju interakciju sa hardverom, kao što su RFID čitači i kamere, i kontrolu GPIO portova. Takođe, ovde se nalaze skripte za obradu i slanje podataka na web server.

3. **html_templates:**

- Sadrži HTML fajlove koji čine korisnički interfejs projekta. Ovi fajlovi su temelj za izgradnju frontend dela aplikacije, omogućavajući korisnicima interakciju sa sistemom kroz preglednik.