Kućni video nadzor

Đođ, Luka

Undergraduate thesis / Završni rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:505726

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-19**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Stručni studij

KUĆNI VIDEO NADZOR

Završni rad

Luka Đođ

Osijek, 2020.

SADRŽAJ

1. UVOD 1
1.1. Zadatak završnog rada 1
2. KORIŠTENE TEHNOLOGIJE 2
2.1. Hardver 2
2.1.1. Raspberry Pi 2
2.1.2. Raspberry Pi 2 Model B 3
2.1.3. GPIO 4
2.1.4. Raspberry Pi kamera 5
2.1.5. PIR senzor 5
2.2. Softver 6
2.2.1. Operacijski sustav 6
2.2.2. Pushbullet 6
2.2.3. Python programski jezik 7
2.2.4. Java programski jezik 7
2.2.5. Firebase 8
2.2.6. Android operacijski sustav 8
2.2.7. Android studio 9
2.2.8. Video prijenos u stvarnom vremenu preko Youtube aplikacije 10
3. FUNKCIONALNOST VIDEO NADZORA 11
3.1. Spajanje PIR senzora na Raspberry Pi GPIO 11
3.2. Spajanje kamere na Raspberry Pi CSI 12
3.3. Postavljanje Raspberry Pi za videoprijenos 13
3.4. Konfiguriranje Raspberry Pia 14
3.5. Izrada Android aplikacije 16
4. KORIŠTENJE VIDEO NADZORA 19

4.1. Android aplikacija 19
4.2. Nadzorni sustav 22
ZAKLJUČAK 24
LITERATURA 25
SAŽETAK 26
ABSTRACT 27
ŽIVOTOPIS 28
PRILOZI 29

1. UVOD

Ovim radom se prikazuje funkcioniranje i odnos android aplikacije s IoT (engl. *Internet of Things*) stanicom koja se nalazi na udaljenom mjestu. IoT stanica se sastoji od računala Raspberry Pi 2 model B, te PIR (engl. *Passive infrared sensor*) senzora za detektiranje pokreta, kamere koja je strogo namjenjena za Raspberry Pi, i mikrofon za dohvaćanje zvuka. Spoj navedenoga predstavlja nadzorni sustav koji je spojen na internet i pruža izravan video prijenos sa zvukom na korisnički privatni Youtube kanal te IoT stanica javlja korisniku preko android aplikacije o događanju ispred kamere. Android aplikacija je izrađena u programu Android Studio. Aplikacija sadrži preusmjeravanje na Youtube aplikaciju za gledanje videa u stvarnom vremenu, API (engl. *Application Programming Interface*) od aplikacije PushBullet koji šalje obavijesti o detektiranom pokretu pomoću PIR senzora. Nadzornoj stanici se može pristupiti s bilo kojeg mjesta gdje postoji internet veza. Nadzorna stanica nakon svakog završenog video prijenosa sprema video koji može kasnije poslužiti za daljnju obradu i analizu. Ovakav sigurnosni sustav se može koristiti u osobnoj upotrebi, u kući, garaži, ili za snimanje svog kućnog ljubimca kad nema nikog kod kuće.

U drugom poglavlju opisuju se korištene tehnologije potrebne za razvoj aplikacije, spremanje u bazu i realiziranje nadzornog sustava, dok u trećem poglavlju se prikazuje funkcioniranje aplikacije i nadzornog sustava. U četvrtom poglavlju se daje opis i način korištenja Android aplikacije i nadzornog sustava.

1.1. Zadatak završnog rada

Zadatak je izraditi android aplikaciju koja ima mogućnost gledanja video prijenosa u stvarnom vremenu, te nadzorni sustav koji može snimati u stvarnom vremenu. Nadzorni sustav mora sadržavati kameru, mikrofon i senzor pokreta kako bi mogao poslati obavijest aplikaciji u slučaju detekcije pokreta. Android aplikacija ima mogućnost primanja obavijesti o detekciji pokreta kojeg je poslala nadzorna stanica, video prijenos u stvarnom vremenu i dodatne funkcije kojim se pozivaju hitne službe na broj "112" u slučaju sumnjivih aktivnosti. Prilikom detektiranja pokreta, nadzorni sustav šalje informacije o detektiranom pokretu u bazu podataka, te se te informacije prikazuju u android aplikaciji. U aplikaciji postoji mogućnost uređivanja informacija i nakon uređivanja baza podataka se ažurira s najnovijim informacijama.

2. KORIŠTENE TEHNOLOGIJE

U ovom poglavlju su navedene tehnologije koje su se koristile prilikom izrade kućnog video nadzora. Od hardvera poput Raspberry Pi računala pa sve do softvera kao što je baza podataka Firebase i programski jezici Python i Java.

2.1. Hardver

Za izradu hardverskog dijela potrebno je računalo Raspberry Pi te dodatne komponente kao što su kamera, PIR senzor, USB (engl. *Universal Serial Bus*) mikrofon i WiFi USB adapter.

2.1.1. Raspberry Pi

Raspberry Pi je malo računalo gdje se sve njegove komponente nalaze na pločici veličine jednog pametnog mobitela. Takvo računalno je razvijeno i proizvedeno u Ujedinjenom Kraljevstvu sa svrhom razvijanja programerskih vještina u školama i računalnim organizacijama. Osim razvijanja programerskih vještina, Raspberry Pi ima puno veću primjenu u robotici. Često je korišten kao IoT stanica, meteorološka stanica ili radio stanica, može se konfigurirati kao VPN (engl. *Virtual Private Network*) server, Android STB (engl. *Set-Top box*) televizora postaviti u pametni televizor, to su samo od neki od mnogih primjera. Za korištenje Raspberry Pi potrebno je: tipkovnica, miš, zaslon i utičnica za napajanje računala. Isto tako je moguće pristupiti s udaljenog mjesta na lokalnoj mreži pomoću SSH (engl. *Secure Shell*) protokola ili preko VNC (engl. *Virtual Network Computing*) grafičkog programa gdje za razliku od SSH protokola omogućuje dostupnu udaljenu radnu površinu Raspberry Pi[1].

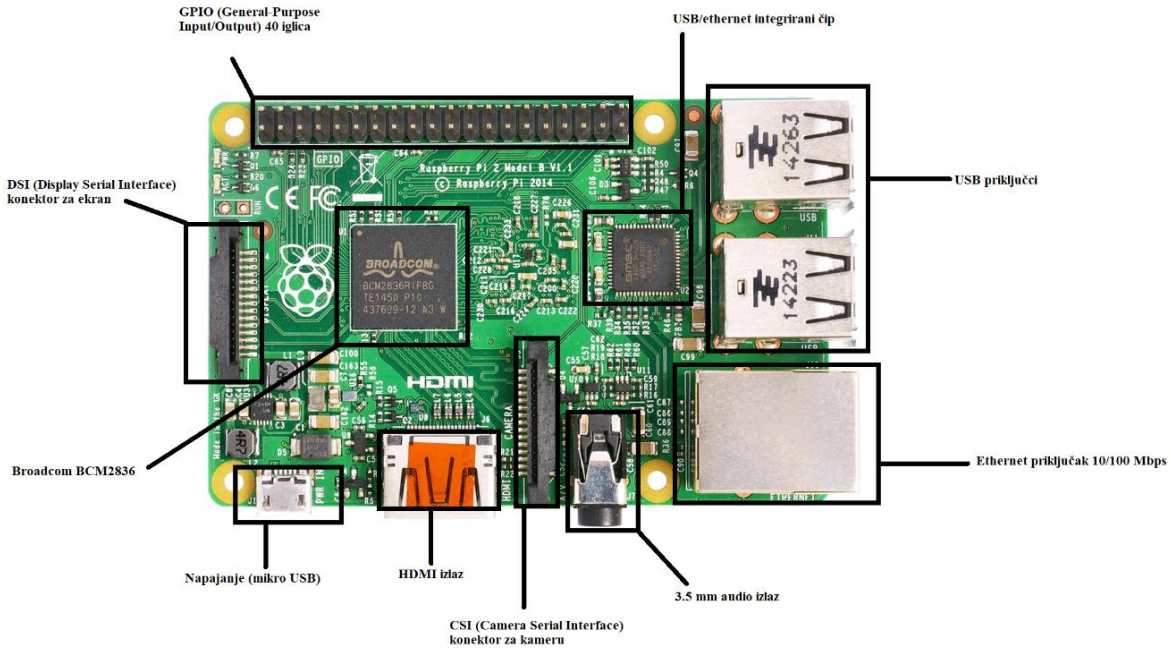
Generacije Raspberry Pi računala tijekom godina:

- Raspberry Pi 1 Model B, 2012
- Raspberry Pi 1 Model A, 2013
- Raspberry Pi 1 Model B+, 2014
- Raspberry Pi 1 Model A+, 2014
- Raspberry Pi 2 Model B, 2015

- Raspberry Pi Zero, 2015
- Raspberry Pi 3 Model B, 2016
- Raspberry Pi 3 Model A+, 2018
- Raspberry Pi 3 Model B+, 2018
- Raspberry Pi 4 Model B (1 GiB), 2019
- Raspberry Pi 4 Model B (2 GiB), 2019
- Raspberry Pi 4 Model B (4 GiB), 2019
- Raspberry Pi 4 Model B (8 GiB), 2020

2.1.2. Raspberry Pi 2 Model B

Raspberry Pi Model B je druga generacija Raspbbery Pi koji je proizveden 2015. godine te je navedena generacija korištena u izradi nadzorne stanice. Na slici 2.1. su prikazani glavni dijelovi Raspberry Pi računala[2]. Osnovni priključci korišteni u završnom radu su: GPIO, USB priključci, CSI (engl. *Camera Serial Interface*) i napajanje[3].



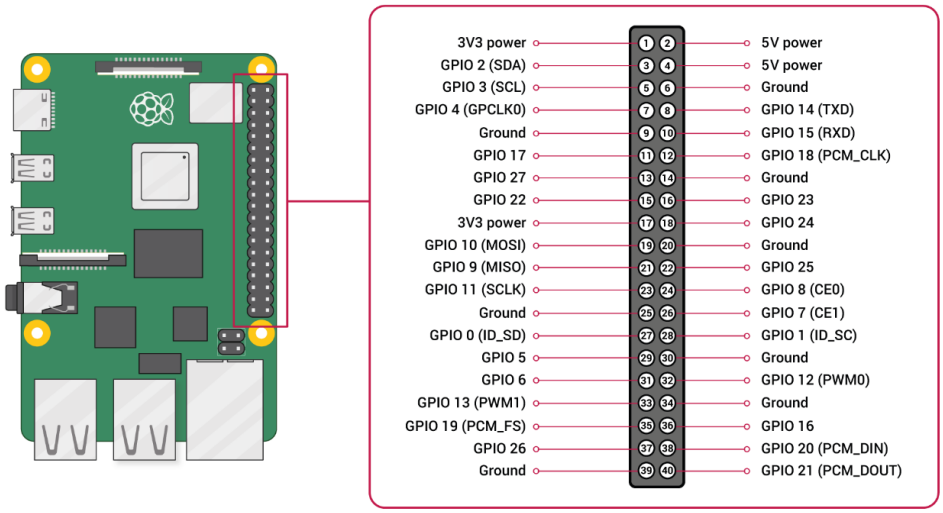
Slika 2.1. *Raspberry Pi 2 Model B* ***s*** *prikazanim dijelovima.*

Pojedinosti Raspberry Pi 2 model B:

- Procesor: Broadcom BCM2836 900MHz četri jezgre
- Grafički procesor: Broadcom VideoCore IV 250MHz
- Memorija: 1Gb
- USB ulazi: 4
- Video ulaz: 15-pin MIPI CSI konektor
- Video izlaz: HDMI
- Internet ulaz: Ethernet 10/100 Mbps
- Audio izlaz: Analog 3.5 mm
- Pohrana: MicroSD kartica

2.1.3. GPIO

GPIO (engl. *General-Purpose Input/Output*) predstavljeni su pinovima koji su poredani sa strane Raspberry Pi pločice, i na primjeru Raspberry Pi 2 Model B ih ima 40. Dok ih je u prethodnim generacijama bilo samo 26. GPIO pinovi se koriste za spajanje raznih senzora, LED dioda, tranzistora i drugih elektroničkih komponenata. Slikom 2.2. su prikazani raspored pinova gdje su najvažniji naponi i uzemljenje[4]. Potrebno je pripaziti na koji se napon spajaju komponente, prijedlog je uvijek korištenje otpora zbog zaštite komponenata.

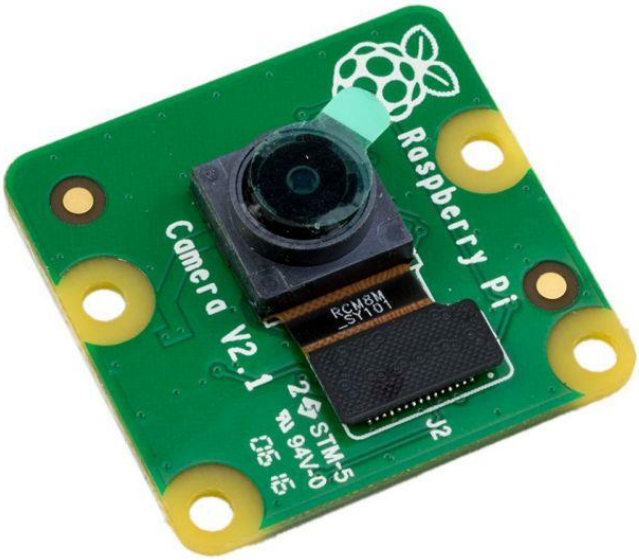


Slika 2.2. *Shematski prikaz Raspberry Pi GPIO pinova.*

2.1.4. Raspberry Pi kamera

Raspberry Pi kamera modul je službeni proizvod od Raspberry Pi tvrtke. Slikom 2.3. prikazana je kamera modul V2 koja je iskorištena u ovom radu. Postoje tri vrste Raspberry Pi kamere[5, 6]:

- Kamera Modul V1 objavljena je 2015. godine
- Kamera Modul V2 objavljena je 2016. godine
- HQ Kamera objavljena je 2020. godine



Slika 2.3. *Raspberry Pi kamera Modul V2.*

2.1.5. PIR senzor

PIR (engl. *Passive Infrared Sensor*) je senzor koji detektira pokret. Odnosno to je elektronički senzor koji mjeri infracrveno svjetlo koje zrači iz objekata u vidnom polju. Najčešće se koristi u sigurnosnim alarmima ili automatskim rasvjetama. Slika 2.4. prikazuje PIR senzor[7].

Slika 2.4. *PIR senzor koji se koristio u projektu.*

2.2. Softver

Za izradu softverskog dijela potrebne su aplikacije Pushbullet za primanje obavijesti, Firebase baza podatka, Android studio za kreiranje Android aplikacije, te programski jezik za kreiranje Android aplikacije je Java, a za izradu skripte na hardverskom dijelu je Python programski jezik.

2.2.1. Operacijski sustav

Operacijski susta koji se koristi i koji je preporučen za normalno funkcioniranje je Raspbian ili Raspberry Pi OS. Raspberry PI OS je besplatan operacijski sustav baziran na Debian OS, te optimiziran da radi sa Raspberry Pi hardverom bez poteškoća. Operacijski sustav sadrži preko 35,000 paketa. Operacijski sustav se instalira isključivo preko MicroSD kartice jer je to jedino mjesto pohrane. Preporuča se veličina kartice od 16Gb [8].

2.2.2. Pushbullet

Pushbullet je besplatna aplikacija koja spaja računala, mobitele i tablete i omogućava im da rade zajedno. Preko Pushbullet aplikacije šalju se obavijesti s jednog uređaja na drugi. Na računalu

Raspberry Pi korišten je API koji šalje obavijest na drugi uređaj prilikom detektiranja pokreta. Kako bi se mogao iskoristi API prvo se treba napraviti korisnički račun na njihovoj web stranici, te se dodjeli jedinstveni API ključ. Za iskorištavanje API ključa mora se instalirati na Raspberry Pi preko terminala navedenom naredbom:

```
$ sudo pip3 install pushbullet.py
```

Prva riječ "sudo" označava superkorisnika koji ima sve ovlasti u Raspberry Pi OS, "pip3" predstavlja python paket za instaliranje programa, "install" označava da želimo instalirati servis "pushbullet.py"[9].

2.2.3. Python programski jezik

Python je objektno orijentirani programski jezik visoke razine s dinamičkom semantikom. Uključuje strukture podataka visoke razine, dinamičko pisanje, dinamičko povezivanje i još mnogo raznih značajki koja pomažu kod razvoja kompleksnih aplikacija. Python sintaksa je jednostavna za naučiti i omogućava laku čitljivosti stoga smanjuje troškove održavanja programa. Python podržava razne module i pakete, što daje modularnost programima i ponovnu upotrebu koda[10].

2.2.4. Java programski jezik

Java je programski jezik koji je objavljen 1995. godine od tvrtke Sun Microsystems. Velik broj aplikacija i stranica neće moći raditi ako nije instaliran java programski jezik. Java je brz, siguran, pouzdan i objektno orijentirani programski jezik. Poznat je po riječi WORA (engl. *Write Once Run Anywhere*) što znači da programeru nije bitno hoće li se program izvoditi na 16, 32 ili 64 bitnoj arhitekturi. Prednost je što se ne koriste pokazivači, ima ugrađena svojstva za zaštitu od ubacivanja neželjenog kôda, te nema potrebe za oslobađanjem memorije jer Java provodi automatsko oslobađanje memorije. Koristi se u izradi android aplikacija, gdje je aplikacija za kućni video nadzor i izrađena[11].

2.2.5. Firebase

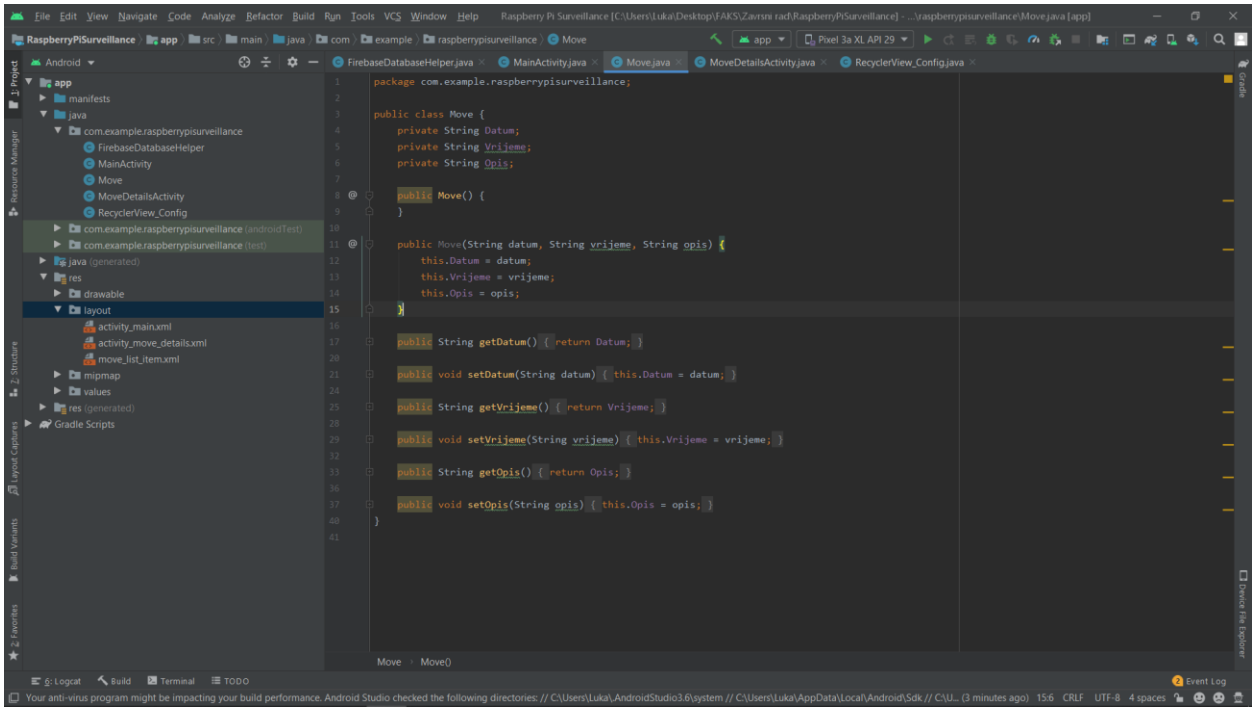
Firebase je BaaS (engl. *Backend-as-a-Service*), platforma za razvoj aplikacija sljedeće generacije na Google Cloud platformi. Firebase omogućuje programerima da se usredotoče na stvaranje jednostavnog korisničkog iskustva. Ne mora se upravljati poslužiteljima niti pisati API. Firebase predstavlja poslužitelj, API i pohranu podataka, a predstavljen je tako da se može izmijeniti prema potrebi. Može se koristiti za bazu podataka, autentifikaciju korisnika, spremanje slika i raznih podataka. Firebase je najčešće korišten kao baza podataka u stvarnom vremenu. Povezuje se putem WebSocketa a oni su puno brži u odnosu na zastarjeli HTTP (engl. *Hyper Text Transfer Protocol*) protokol za prijenos informacija. Firebase prima i šalje podatke čim se ažuriraju i zbog te prednosti Firebase je izabran kao baza podataka za navedenu Android aplikaciju i za nadzorni sustav. Kako bi se koristila Firebase aplikacija potrebno je napraviti Google korisnički račun i Firebase projekt. Nakon postavljenog projekta potrebno ga je spojiti sa android aplikacijom. To se može izvesti u Android studio okruženju, pomoću "Firebase Assistant" koji automatski spaja bazu podataka s Android projektom. "Firebase Assistant" dodaje potrebne dodatke/ovisnosti (engl. *Dependency*) te sinkronizira projekt s navedenim dodacima[12].

2.2.6. Android operacijski sustav

Android operacijski sustav je jedan od najrasprostranjenijih operacijskih sustava za pametne uređaje. Koristi se u uređajima poput pametnih mobitela, satova, televizora, tableta, hladnjaka i mnogo raznih uređaja. Prema statistici provedene u svibnju 2019. godine, Android ima najviše korisnika, više od 2.5 milijarde mjesečno aktivnih korisnika. Android ima razne inačice, najnovija inačica je trenutno 11 i datum izlaska je 8. rujan 2020. godine. Android je u početku bio u vlasništvu Android Inc. tvrtke, no 2005. godine kupio ju je Google. Osnovni i temeljni kôd Androida poznat je kao Android projekt otvorenog kôda (engl. *Android Open Source Project*) te je dostupan pod licencom Apache. Google Play Protect skenira sve aplikacije, softver dobiva redovita sigurnosna ažuriranja i platforma se stalno poboljšava. Android ima mogućnost dijeljenje svojih podataka poput web pretraživanja i dijeljenja lokacije. To se može onemogućiti u postavkama privatnosti. S Android operacijskim sustavom dobije se i aplikacije "Digital Wellbeing" koja prikazuje detalje o korištenju sustava i performansa cijelog mobitela. Prednost takve aplikacije omogućava uvid u sve aplikacije na mobitelu, moguće je vidjeti koliko aplikacije troše mobilnog prometa, ili potrošnju baterije[13].

2.2.7. Android studio

Android studio je službeno integrirano razvojno okruženje za Google operacijski sustav Android, izgrađen na JetBrains IntelliJ razvojnom okruženju i dizajniran posebno za razvijanje Android aplikacije. Dosupan je za Windows, Mac i Linux. Android studio predstavlja zamjenu za Eclipse Android Development Tools koji je isto služio za razvoj android aplikacija. Ovakvo razvojno okruženje ima veliku prednost što nudi korisnicima android emulator, tako da se kod pravljenja aplikacije može odmah testirati na računalu, emulatoru određenog android mobitela. Isto tako se može spojiti mobitel na računalo, pa prilikom pokretanja aplikacije, Android sustav, omogućuje instaliranje aplikacije automatski na mobilni uređaj. Android sustav ima mogućnost izrade kôda u dva programska jezika: Java i Kotlin. Svaki projekt sadrži jedan ili više modula sa izvornim kôdom. Slika 2.5. prikazuje Android studio[14].



Slika 2.5. *Izgled Android Studia prilikom kreiranja aplikacije.*

2.2.8. Video prijenos u stvarnom vremenu preko Youtube aplikacije

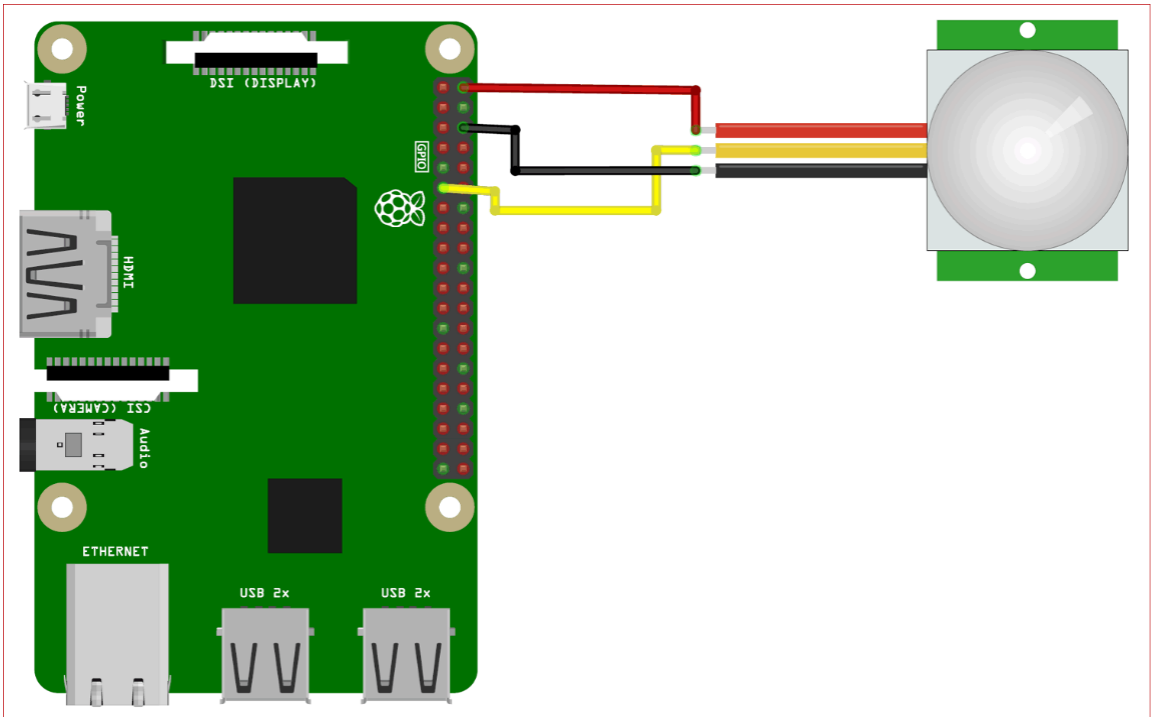
Prilikom snimanja i postavljanja video prijenosa u stvarnom vremenu, odabran je jedan od najpoznatijih platformi za dijeljenje videa po imenu Youtube. Youtube je besplatna platforma za objavljivanje videa i snimanje videa u stvarnom vremenu. Youtube dozvoljava postavljanje videa u stvarnom vremenu u privatnom načinu rada, što nam omogućuje jednostavno gledanje nadzornog sustava bez straha od neovlaštenog pristupa. Naime, da bi se privatni video mogao vidjeti na mobilnom uređaju, potrebno je dijeliti isti korisnički račun koji se koristi i za snimanje preko nadzorne kamere. Kod postavljanja videa u stvarnom vremenu, potrebnu je uzeti tajni ključ koji ima svaki korisnički račun na Youtube platformi. Taj ključ omogućuje snimanje preko bilo kojeg drugog uređaja. Takav tajni ključ se koristio prilikom realiziranja video prijenosa u stvarnom vremenu.

3. FUNKCIONALNOST VIDEO NADZORA

Za ispravan rad nadzornog sustava potrebno je spojiti sve komponente na odgovarajuća mjesta. Kamera ima posebno mjesto zvan CLI dok se PIR senzor spaja na GPIO pinove. Mikrofon i Wifi se spajaju preko USB porta. Nakon postavljenog nadzornog sustava, potrebno je omogućiti detektiranje pokreta i slanje obavijesti na mobilnu aplikaciju i spremanje podataka o pokretu u Firebase bazu podataka. Mobilna aplikacija ima mogućnost gledanja svih podatak u bazi te manipuliranje istim podacima.

3.1. Spajanje PIR senzora na Raspberry Pi GPIO

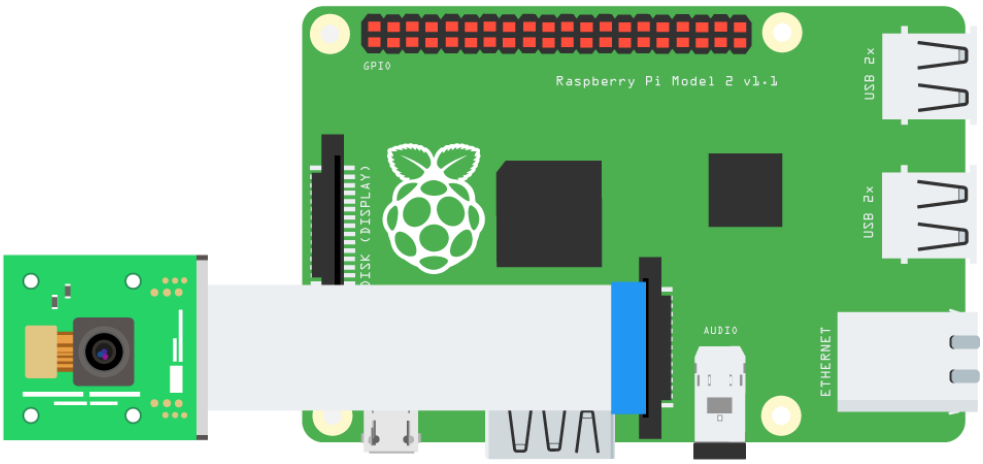
PIR senzor se povezuje sa Raspberry Pi preko GPIO pinova. PIR je direktno povezan sa Raspberry Pi, gdje crvena žica prikazuje napajanje, crna uzemljenje, i žuta broj pina na koji se povezuje, te na koji šalje podatke o detektiranju podataka. Spajanje se vidi na slici 3.1[15].



Slika 3.1. *Shematski prikaz spajanja PIR senzora na Raspberry Pi GPIO pinove..*

3.2. Spajanje kamere na Raspberry Pi CSI

Kako bi se kamera mogla koristi za video prijenos potrebno ju je spojiti na Raspberry Pi te omogućiti u operacijskom sustavu. Nakon što je kamera omogućena i spremna za upotrebu potrebno je napraviti testiranje kako bi se vidjelo da je sve ispravno. Kamera se spaja na CSI konektor kako je prikazano na slici 3.2[16].



Slika 3.2. *Spajanje kamere na Raspberry Pi.*

Spajanjem svih dijelova u konačnu verziju, te postavljanje u kutiju koja se zove "SPI-BOX" radi ljepšeg izgleda dobiva se konačni rezultat hardverskog dijela prikazanog na slici 3.3.

Slika 3.3. *Gotova izvedba nadzornog sustava baziranog na Raspberry Pi računalu.*

3.3. Postavljanje Raspberry Pi za videoprijenos

Da bi se ostvario video u stvarnom vremenu potrebno je instalirati Docker. Docker pojednostavljuje pakiranje, distribuciju, instalaciju i pokretanje kompleksnih aplikacija.

```
$ curl -sSL https://get.docker.com | sh
$ sudo usermod pi -aG docker
$ reboot
```

Navedenim naredbama u terminalu se instalira Docker na Raspberry Pi i time omogućuje instaliranje Docker paketa koji je potreban za pokretanje video prijenosa.

```
$ docker pull alexerllis2/streaming:17-5-2017
```

Kad smo "povukli" paket, moramo još pozvati jednu naredbu sa tajnim ključem koji smo spremili sa YouTube korisničkog računa. Slova "x" predstavljaju tajni ključ u naredbi[17]:

```
$ docker run --privileged --name cam -ti alexellis2/streaming:17-5-2017 xxxx-xxxx-xxxx-xxxx
```

3.4. Konfiguriranje Raspberry Pia

Prema slici 3.4. u kôdu se vidi uključivanje biblioteka koje omogućuju korištenje GPIO pinova na Raspberry Pi računalu, "*pyrebase*" omogućuje povezivanje sa Firebase bazom podataka, "*pushbullet*" omogućuje korištenje Pushbullet aplikacije za slanje besplatnih obavijesti na mobilni uređaj. Isto tako iskorištavaju se biblioteke za određivanje trenutnog vremena u satima, minutama i sekundama kao i o datumu dan, mjesec i godina. Nakon uključivanja biblioteka definira se brojanje pinova prema onome što piše na pločici na Raspberry Pi računalu. Sljedeća linija uklanja sva upozorenja koja program može izbaciti prije samog pokretanja. Nakon toga "*GPIO.setup(7,GPIO.IN)*" govori o "*slušanju*" događanja na pinu 7.

```
import RPi.GPIO as GPIO
import pyrebase
from pushbullet import Pushbullet
from time import sleep
from datetime import *

GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)
GPIO.setup(7, GPIO.IN)
```

Slika 3.4. *Učitavanja biblioteka za PIR senzor i Pushbullet aplikaciju.*

Da bi se poslala obavijest na mobilni uređaj preko Pushbullet aplikacije treba se napraviti korisnički račun te preuzeti API ključ kojeg je aplikacija dodijelila. Kako bi se koristila Firebase aplikacija potrebno je preuzeti informacije o projektu koji je namijenjen za mobilnu aplikaciju i za spajanje s bazom podataka te je potrebno dodijeliti podatke u objekt "*config*". Prilikom detektiranja pokreta želi se nastaviti dodavati podatke na već postojeće podatke u bazi, stoga se provjerava preko ugrađene funkcije "*len()*" koliko ima podataka u bazi. Ukoliko se već nalaze neki podatci, "*id_num*" će se povećati za jedan, inače ukoliko nema podataka u bazi "*id_num*" će preuzeti vrijednost nula. Slika 3.5. prikazuje sve navedeno.

```
pb = Pushbullet("PUSHBULLET_API_KEY")
print(pb.devices)

config = {
  "apiKey": "FIREBASE_API_KEY",
  "authDomain": "rpi-security-e10ab.firebaseapp.com",
  "databaseURL": "https://rpi-security-e10ab.firebaseio.com",
  "storageBucket": "rpi-security-e10ab.appspot.com"
}

firebase = pyrebase.initialize_app(config)
id_num=len(firebase.database().child("Pokret").get().val())
if id_num!=0:
    id_num+=1;
else:
    id_num=0;
```

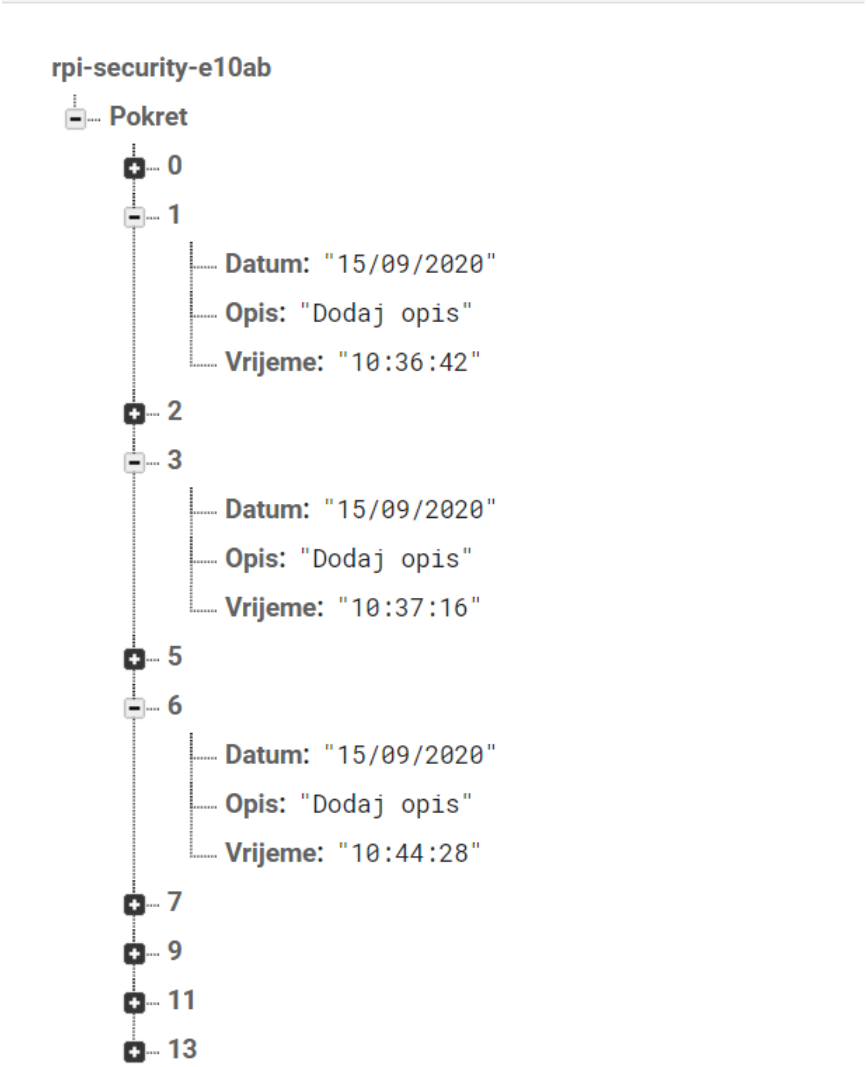
Slika 3.5. *Prikaz spajanja Raspberry Pia sa Firebase bazom i Pushbullet aplikacijom.*

Kako bi se program mogao stalno izvoditi, postavljen je uvjet da se program izvede kad god je "*True*". Unutar petlje provjerava se PIR senzor, odnosno PIR senzor provjerava ima li pokreta. Ukoliko detektira pokret u Firebase bazu podataka se sprema trenutni datum i vrijeme te opis što se može vidjeti na slici 3.6.

```
while True:
    today = datetime.now()
    i = GPIO.input(7)
    if i == 0:
        print ("Nema pokreta!")
        sleep(2)
    elif i == 1:
        print ("Pokret!")
        dev = pb.get_device('Xiaomi Redmi Note 7')
        push = dev.push_note("Upozorenje!", "Pokret detektiran!")
        db = firebase.database()
        today_date = today.strftime("%d/%m/%Y")
        today_time = today.strftime("%H:%M:%S")
        data = {"Datum": today_date, "Vrijeme": today_time,"Opis": "Dodaj opis"}
        db.child('Pokret',id_num).set(data)
        id_num+=1
        sleep(10)
```

Slika 3.6. *Petlja koja čeka pokret te slanje u bazu pojedinosti detektiranja pokreta.*

Nakon poslane obavijesti, određuje se trenutni datum i vrijeme i te informacije se šalju u bazu podataka. U bazi su podatci predstavljeni s jedinstvenim identifikacijskim brojem, i pod tim brojem su spremljeni "*datum*", "*opis*" i "*vrijeme*". Opis je postavljen tako da se može promijeniti u Android aplikaciji proizvoljno. Slika 3.7. prikazuje Firebase bazu podataka u stvarnom vremenu. Prikazuje glavno dijete "*Pokret*" i na njega se dodaju identifikacijski brojevi.



Slika 3.7. *Firebase baza podataka u stvarnom vremenu s detektiranim pokretima i njihovim informacijama.*

3.5. Izrada Android aplikacije

Programski kôd na slici sadrži tri bitne funkcije. Funkcija za čitanje iz baze podataka "*readMoves*", funkcija za mijenjanje podataka u bazi "*updateMove*", i funkcija "*deleteMove*" za

brisanje podataka iz baze. Svaki put kad se dogodi promjena na bazi izvest će se funkcija "*readMoves*" koja će pročitati podatke koji se nalaze u bazi. Funkcija prolazi kroz svaki identifikacijski broj te svi podaci koje se nalaze pod tim brojem su spremljeni u objekt. Prolazi se kroz objekt i prikazuju se svi dohvaćeni podaci iz baze. Za ažuriranje baze podataka koristi se ključ, objekt "*move*". Ključ služi za dohvaćanje podataka sa odgovarajućim identifikacijskim brojem te objekt koji sadrži podatke spremljene pod tim brojem/ključem. Kod brisanja podataka potreban je samo ključ koji će dohvatiti odgovarajući podatak za brisanje. Slikom 3.8. su prikazane navedene funkcije.

```
public void readMoves(final DataStatus dataStatus){
    mReferenceMoves.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            moves.clear();
            List<String> keys = new ArrayList<>();
            for(DataSnapshot keyNode : snapshot.getChildren()){
               keys.add(keyNode.getKey());
               Move move=keyNode.getValue(Move.class);
               moves.add(move);
            }
            dataStatus.DataIsLoaded(moves,keys);
        }

        @Override
        public void onCancelled(@NonNull DatabaseError error) {

        }
    });
}

public void updateMove(String key, Move move, final DataStatus dataStatus){
    mReferenceMoves.child(key).setValue(move).addOnSuccessListener((OnSuccessListener) (aVoid) -> {
            dataStatus.DataIsUpdated();
    });
}

public void deleteMove(String key, final DataStatus dataStatus){
    mReferenceMoves.child(key).setValue(null).addOnSuccessListener((OnSuccessListener) (aVoid) -> {
            dataStatus.DataIsDeleted();
    });
}
```

Slika 3.8. *Prikaz tri glavne funkcije za rad s bazom podataka.*

Da se omogući uređivanje postojećih podataka potrebno je dohvatiti točan naziv svakog podataka kojeg se želi izmijeniti, isto tako i njegov odgovarajući ključ. Kad se odabere neki podatak za

izmjenu otvorit će se novi "*activity*" s detaljnim podacima i mogućnosti za izmjenu datuma, vremena i dodavanja opisa, te mogućnosti brisanja cijelog podatka. Podaci iz baze su već upisana u poljima za izmjenu tako da se vidi što se izmjenjuje. U slici 3.9. se vidi i dohvaćanje elemenata i gumba za izmjenu, brisanje i vraćanje na početnu pozadinu aplikacije.

```
key = getIntent().getStringExtra( name: "key");
datum = getIntent().getStringExtra( name: "Datum");
vrijeme = getIntent().getStringExtra( name: "Vrijeme");
opis = getIntent().getStringExtra( name: "Opis");

mDatum_editTxt = (EditText) findViewById(R.id.datum_editTxt);
mDatum_editTxt.setText(datum);
mVrijeme_editTxt = (EditText) findViewById(R.id.vrijeme_editTxt);
mVrijeme_editTxt.setText(vrijeme);
mOpis_editTxt = (EditText) findViewById(R.id.opis_editTxt);
mOpis_editTxt.setText(opis);

mEdit_btn = (Button) findViewById(R.id.edit_btn);
mDelete_btn = (Button) findViewById(R.id.delete_btn);
mBack_btn = (Button) findViewById(R.id.back_btn);
```

Slika 3.9. *Dohvaćanje podataka koji su spremljeni u bazi, te dohvaćanje ID elemenata.*

Na sljedećoj slici 3.10. prikazuje se gumb za preusmjeravanje na odgovarajuću aplikaciju. Ukoliko se odabere "*GO LIVE*" aplikacija će otvorit Youtube aplikaciju na odredištu gdje se nalazi video u stvarnom vremenu. Ukoliko se odabere gumb s ikonom telefona, aplikacija će preusmjeriti na poziv hitne službe.

```
FloatingActionButton phoneCallFAB = (FloatingActionButton) findViewById(R.id.callFAB);
phoneCallFAB.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        startActivity(new Intent(Intent.ACTION_VIEW,Uri.parse("tel://112")));
    }
});
Button buttonLive = (Button) findViewById(R.id.button_live);
buttonLive.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        startActivity(new Intent(Intent.ACTION_VIEW, Uri.parse("https://www.youtube.com/channel/UC6UUbju5FZTYtYuShS748rw/live")));
    }
});
```

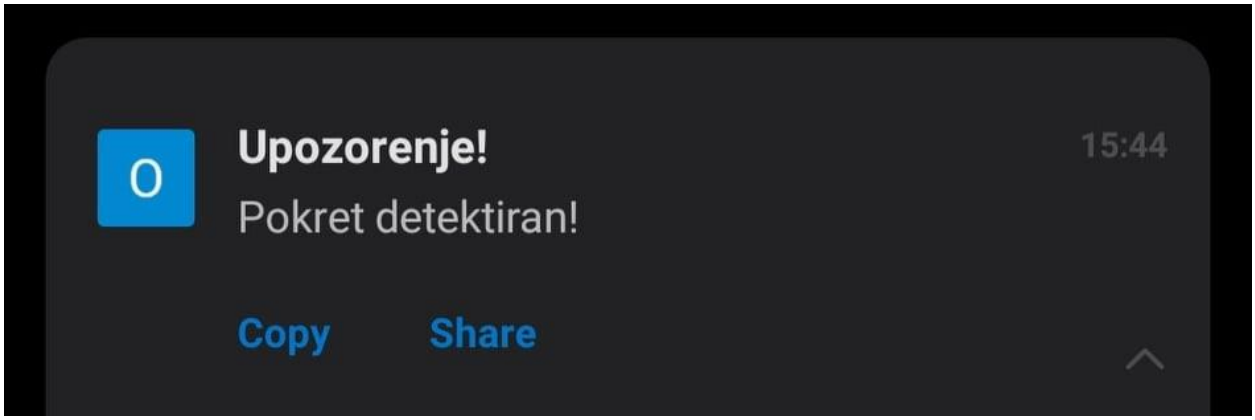
Slika 3.10. *Kôd prikazuje funkcije za dva gumba.*

4. KORIŠTENJE VIDEO NADZORA

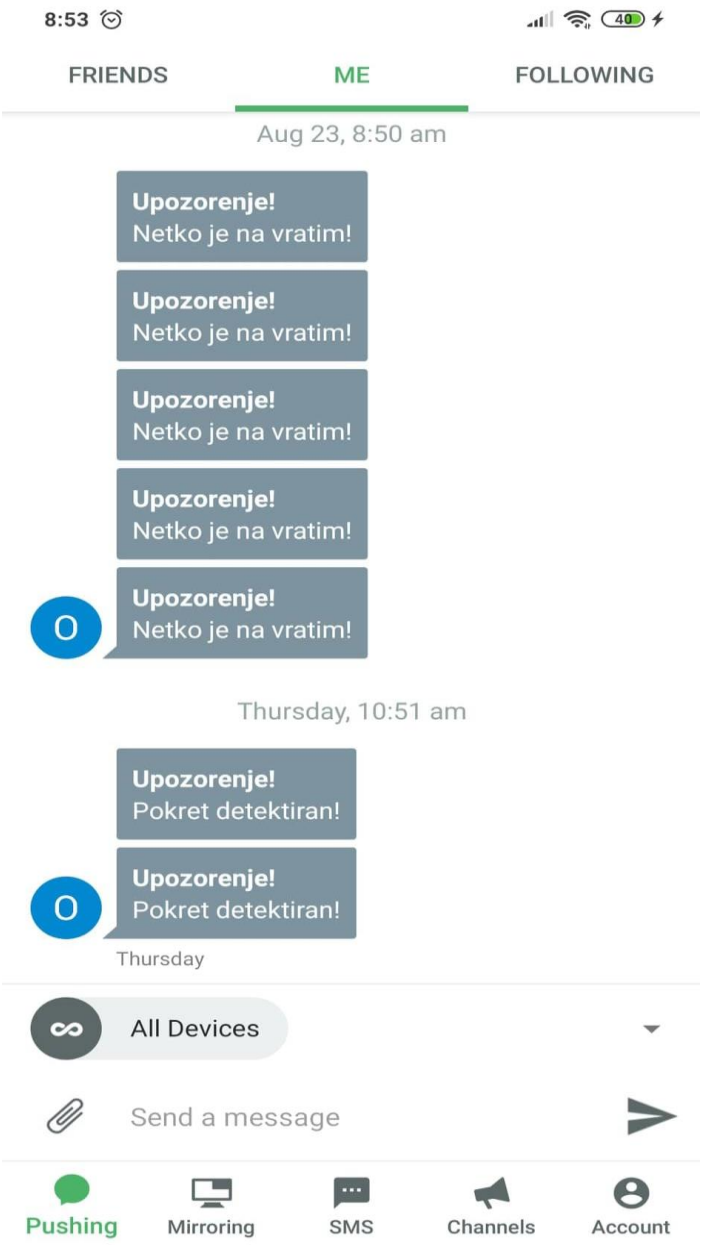
Android aplikacija se temelji na Firebase bazi podataka. Aplikacija čita sve što se nalazi u bazi te prikazuje sve podatke i daje mogućnost za uređivanje pojedinog podatka ili brisanje podataka. Također aplikacija ima dva gumba koji preusmjeravaju na odgovarajuću aplikaciju.

4.1. Android aplikacija

Kad Raspberry Pi detektira pokret šalje se obavijest preko Pushbullet aplikacije, kako ta obavijest izgleda vidi se na sljedećoj slici 4.1. i 4.2.



Slika 4.1. *Obavijest na mobilnom uređaju prilkom detektiranja pokreta*



Slika 4.2. *Slika prikazuje obavijesti poslane preko Raspberry Pia na Pushbullet aplikaciju.*

Sljedećom slikom 4.3. prikazani su podaci iz baze podataka na aplikaciji s gumbom za pristup Youtube aplikaciji za gledanje videa u stvarnom vremenu i gumb za pozivanje "112" koji preusmjerava na aplikaciju za pozive.

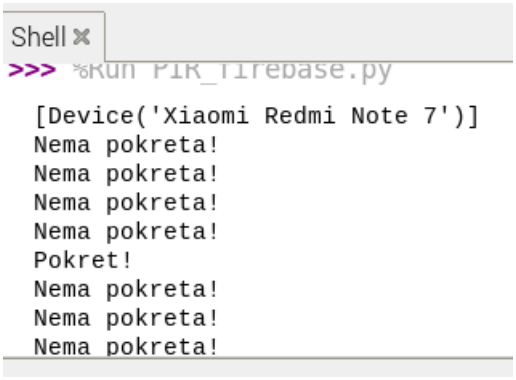
Slika 4.3. *Aplikacija s podacima iz baze podataka.*

Kad se odabere gumb za uređivanje ili brisanje, ukoliko je uspješno pojavit će se poruka da je uspješno obavljena radnja i preusmjeriti na početnu pozadinu aplikacije. Kad se odabere gumb za brisanje, svi podaci koji su upisani za taj pokret biti će obrisani. Isto tako sadrži i gumb za vraćanje na početnu pozadinu aplikacije ukoliko se ne želi ništa izmijeniti. Slika 4.4. prikazuje podatke iz baze koji se mogu izmijeniti ili obrisati.

Slika 4.4. *Detalji podataka kojeg se može uređivati, brisati ili vratiti nazad na naslovnu pozadinu.*

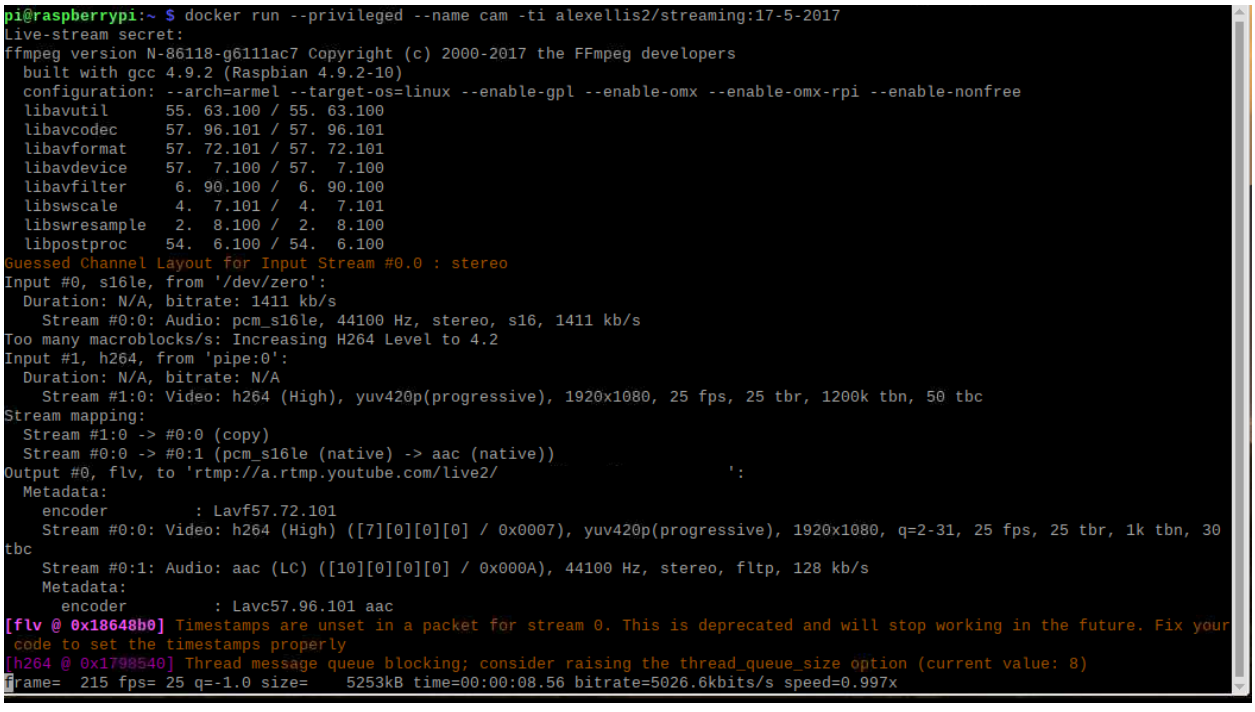
4.2. Nadzorni sustav

Na Raspberry Pi računalu se pokreće skripta koja se ne zaustavlja. Ta skripta detektira pokret i šalje podatke u bazu, te obavijest korisniku o detektiranom pokretu. Skripta u konzoli prikazuje ukoliko dođe do detektiranja pokreta što se može vidjeti na slici 4.5.



Slika 4.5. *Prikaz u Python shellu kad se dogodi pokret.*

Kako bi se pokrenuo video u stvarnom vremenu potrebno je pokrenuti naredbu u terminalu sa jedinstvenim ključem koji ima svaki Youtube korisnički račun. Na slici 4.6. prikazuje se izvedba naredbe i podaci o video prijenosu. Poput koliko okvira se poslalo, koliko se potrošili kilobajta, koji je bit/rate te proteklo vrijeme video prijenosa.



Slika 4.6. *Terminal na Raspberry Pi prilikom pokretanja naredbe za video prijenos.*

ZAKLJUČAK

Izradom ovog završnog rada cilj je razviti jednostavni kućni video nadzor primjenom detektiranja pokreta sa PIR senzorom koji je spojen s Raspberry Pi računalom koji je povezan na internet preko WiFi adaptera te sadrži kameru. Takav video nadzor ima prednost što je mobilnost, može se postaviti bilo gdje i ne zauzima prostora. Glavni nedostatak je što mora imati pristup internetu i izvoru napajanja kako bi mogao neprestano raditi. Uz to javlja se i problem zagrijavanja, Raspberry Pi se može preopteretiti pokretanjem skripti i naredbi za snimanje videa u stvarnom vremenu. Ako bi se takav nadzorni sustav koristi u svakodnevnoj upotrebi trebalo bi instalirati hladnjak koji bi držao temperaturu na optimalnoj za rad.

Osim nadzornog sustava, cilj je bio izrada android aplikacije koja ima mogućnost primanja obavijesti o detektiranom pokretu i pozivanja hitne službe prilikom detekcije neovlaštenog pristupa. Aplikacija ima mogućnost uvid detektiranih pokreta koji su spremljeni u bazu, te mogućnost izmjene opisa kako bi se dao proizvoljni opis o detektiranom pokretu što je velika prednost. Također može se i obrisati detektirani pokret ukoliko se smatra nepotrebnim. Aplikacija ima dva gumba, jedan je za otvaranje Youtube aplikacije kako bi se mogao vidjeti video u stvarnom vremenu, te drugi gumb koji služi za poziv hitne službe u slučaju vidljivih neprilagođenih aktivnosti na kameri.

U konačnici, za izradu ovakvog nadzornog sustava je potrebno puno vremena i novaca. Za istu cijenu moguće je pronaći gotov nadzorni sustav s kamerom visoke kvalitete i jako malim kašnjenjem između okvira. Android aplikacija je obećavajuća i može se iskoristiti za razne upotrebe uz male preinake. Spoj aplikacije i baze podataka omogućuje veliku fleksibilnost kod čitanja podataka i izmjene podataka u stvarnom vremenu.

LITERATURA

[1]. https://www.raspberrypi.org/

[2]. https://upload.wikimedia.org/wikipedia/commons/3/31/Raspberry_Pi_2_Model_B_v1.1_top_new_%28bg_cut_out%29.jpg

[3]. https://opensource.com/resources/raspberry-pi

[4]. https://www.raspberrypi.org/documentation/usage/gpio/

[5]. https://www.raspberrypi.org/documentation/hardware/camera/

[6]. https://auvidea.eu/wp-content/uploads/2017/03/rpi_cam_v2.jpg

[7]. https://www.seeedstudio.com/PIR-Motion-sensor-module-p-74.html

[8]. https://www.raspberrypi.org/documentation/raspbian/

[9]. https://iotdesignpro.com/projects/home-security-system-using-raspberry-pi-and-pir-sensor-with-push-notification-alert

[10]. https://opensource.com/resources/python

[11]. https://www.java.com/en/download/help/whatis_java.html

[12]. https://howtofirebase.com/what-is-firebase-fcb8614ba442

[13]. https://www.android.com/what-is-android/

[14]. https://developer.android.com/studio/intro

[15]. https://iotdesignpro.com/projects/home-security-system-using-raspberry-pi-and-pir-sensor-with-push-notification-alert

[16]. https://www.allaboutcircuits.com/projects/how-to-build-a-picamera/

[17]. https://github.com/alexellis/raspberrypi-youtube-streaming

SAŽETAK

Naslov: Kućni video nadzor

U sklopu ovog završnog rada izrađen je nadzorni sustav pomoću računala Raspberry Pi koji sadrži kameru, USB mikrofon, USB WiFi te PIR senzor za detekciju pokreta. Izrađenom skriptom za detekciju pokreta u Python programskom jeziku, šalje se obavijest korisniku te se spremaju detalji o detektiranom pokretu u Firebase bazu podataka. U bazu se sprema datum, vrijeme i opis. Ono što se sprema u bazu to se prikazuje u android aplikaciji koja je izrađena u Java programskom jeziku u Android studio razvojnom okruženju. Aplikacija ima mogućnost čitanja podataka iz baze, te izborom korisnika podaci se mogu izbrisati ili urediti prema želji. Isto tako postoji mogućnost gledanja videa u stvarnom vremenu tako da se pomoću Android aplikacije preusmjeri u Youtube aplikaciju gdje se nalazi privatan video. Ukoliko se preko kamere vidi neka ne željena aktivnosti postoji gumb koji daje mogućnost pozivanja hitnih službi.

Ključne riječi: Android aplikacija, Firebase, Raspberry Pi, video nadzor

ABSTRACT

Title: Home video surveillance

As part of this final paper, a video surveillance system was created using a Raspberry Pi computer that contains a camera, USB microphone, USB WiFi and a PIR sensor for motion detection. With the Python motion detection script created, a notification is sent to the user and the details of the detected motion is saved to the Firebase database. The date, time and description are stored in the database. All information that is stored in the database is displayed in an android application created in the Java programming language in an Android studio development environment. The application has the ability to read data from the database, and you can delete or modify the selected data as desired. There is also the possibility of watching videos in real time by using Android app to redirect to the Youtube app where the private video is located. If some unwanted activity is seen through the camera, there is a button that gives the possibility to call emergency services.

Keywords: Android application, Firebase, Raspberry Pi, video survelliance

ŽIVOTOPIS

Luka Đođ, rođen 28. kolovoza 1998. godine u Koprivnici. Dolazi iz Podravine malog sela Ferdinandovac. Pohađao je srednju strukovnu školu u susjednom gradu Đurđevac za zvanje tehničar za računarstvo. Nakon srednje škole upisao Fakultet elektrotehnike, računarstva i informacijskih tehnologija u Osijeku smjer stručni studij informatike. Bavio se programiranjem od srednje škole te je u to vrijeme naučio razne tehnologije. Odlučio je ići u smjeru web developmenta kao frontend developer, gdje bi se kasnije želio prebaciti na fullstack development. Tehnologije kojima najviše teži su Javascript, React i NodeJs.

PRILOZI

Projektna mapa s izvornim kôdom i priloženim word i pdf dokumentom nalazi se na optičkom disku koji je priložen uz printanu verziju završnog rada.