

Rezumatul proiectului de diplomă al studentului: George Burlăcel, grupa: 443D

Programul de studiu: Rețele și Software de Telecomunicații, 2023

Cameră video de bord pentru autoturisme, cu interfață web și pentru terminale mobile

Conducător științific: Ș. L. Dr. Ing. Laurențiu Boicescu

Obiectivele proiectului: S-a proiectat o cameră de bord capabilă să înregistreze video și să transmită în timp real către un server. Camera reține datele utilizatorului (ID-ul acestuia) pentru utilizarea dispozitivului în viitor. Aceasta salvează local înregistrările video realizate de către utilizator. Înregistrările și transmisiunea video în timp real pot fi pornite sau oprite prin intermediul comenzilor Bluetooth transmise prin intermediul unei aplicații mobile.

S-a proiectat serverul și acesta este alcătuit din mai multe componente.

- Baza de date conține datele utilizatorilor, adresa de email și parola criptată dar și date despre înregistrările stocate în server.
- Serverul de stocare și distribuire este cel care recepționează înregistrările video de la cameră și permite utilizatorilor să vizualizeze aceste înregistrări.
- Serverul de transmisiune în timp real recepționează transmisiunea video de la cameră. Transmisiunea este accesată de către utilizator prin interfețele aplicațiilor.

S-a proiectat aplicația mobilă ce permite utilizatorului să transmită comenzi către cameră, să vizualizeze înregistrările stocate pe server, să vizualizeze transmisiunea în timp real și să își modifice datele contului.

S-a proiectat aplicația web ce permite utilizatorului să vizualizeze înregistrările video de pe server, să vizualizeze transmisiunea în timp real și să își modifice datele contului.

Realizarea proiectului și rezultate obținute: Aplicația camerei (funcțiile de înregistrare video, comunicația cu serverul) a fost proiectată și realizată în Python. Pentru microcalculator s-a folosit un Raspberry Pi Zero W 2. Pentru a asigura comunicația dintre cameră și server în afara unui LAN se poate folosi fie un modem USB alături de o cartelă SIM sau un modul HAT 4G alături de o cartelă SIM.

O înregistrare video fără compresie ocupă foarte multă memorie din spațiul de stocare. Din acest motiv pentru a putea stoca cât mai multe înregistrări video am ales rezoluția înregistrării, numărul de cadre pe secundă și rata de bit astfel încât să minimizăm memoria ocupată de o înregistrare fără a pierde din lizibilitate.

S-a obținut experimental o rată de bit de 4 Mb/s pentru a asigura o cantitate de memorie ocupată de aproximativ 60 Mo pentru o înregistrare video de 2 minute.

Înregistrările video s-au realizat cu ajutorul comenzii "ffmpeg" și bibliotecii Python "subprocess" pentru a putea controla parametrii înregistrării (rezoluția, codec-ul folosit, numele înregistrării, dacă se înregistrează sau nu cu sunet, etc). Pentru a asigura că aplicația poate primi comenzi Bluetooth în orice moment, funcțiile de înregistrare video, transmisiune în timp real, încărcare pe server, log in utilizator și închide dispozitiv vor fi realizate prin fire de execuție.

Serverul este realizat pe o mașină virtuală Ubuntu Server. Pentru virtualizarea serverului s-a folosit soluția de virtualizare VirtualBox, un hipervizor de tip 2.

Baza de date a serverului s-a realizat prin intermediul serverului MySQL. Baza de date este alcătuită din două tabele conectate printr-o relație de tip 1:N: tabela "utilizatori" și tabela "videoUtilizator". Relația dintre tabele este definită de ID-ul utilizatorului, o cheie străină folosită pentru a stabili legătura dintre tabele, legătură ce definește cărui utilizator îi aparțin înregistrările din tabel.

Serverul de stocare și distribuție a fost proiectat să accepte conexini de la cameră și de la utilizator. Serverul recepționează de la cameră numele înregistrării și ID-ul utilizatorului căruia îi aparține înregistrarea. Introducerea în baza de date a datelor înregistrării se realizează înaintea transmisiunii acestora către server (Figura 1). Conexiunile ce provin de la utilizatori sunt pentru a vizualiza înregistrările în interfața grafică a aplicației mobile sau web. Comunicația cu serverul se realizează la nivel de socket TCP pentru a asigura transmisiunea datelor fără pierdere de pachete sau recepția de pachete duplicate.

```
mysql> select * from videoUtilizator;
+-----+-----+-----+-----+
| idVideo | numeVideo      | idUtilizator | caleVideo |
+-----+-----+-----+-----+
| 7       | 2024_05_29_20_10 | 7            | /7        |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select * from videoUtilizator;
+-----+-----+-----+-----+
| idVideo | numeVideo      | idUtilizator | caleVideo |
+-----+-----+-----+-----+
| 7       | 2024_05_29_20_10 | 7            | /7        |
| 9       | 2024_05_30_20_10 | 7            | /7        |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

Figura 1 Interogarea bazei de date înainte și după realizarea unei înregistrări

Serverul transmisiunii în timp real a fost proiectat în funcție de protocolul WebRTC. Pentru a vizualiza transmisiunea în timp real se realizează o conexiune de tip punct la punct între cameră și server. Apoi în funcție de ce interfață grafică folosește utilizatorul se realizează o conexiune de tip punct la punct între server și interfața utilizatorului.

Aplicația mobilă a fost proiectată și dezvoltată în limbajul Kotlin. Un punct cheie în dezvoltarea aplicației a fost metoda în care se criptează parola utilizatorului. Metoda trebuie să întoarcă același rezultat atât pentru aplicația mobilă dar și pentru aplicația web. În figura de mai jos datele utilizatorilor de test din baza de date au fost adăugate prin intermediul aplicației web, aceștia au aceeași parolă. În terminalul de jos se observă rezultatul verificării dintre hashurile criptografice generate de aplicația web (obținute în acest test prin interogarea bazei de date) și datele generate cu algoritmul aplicației mobile. Astfel nu mai avem nevoie de o altă metodă de a valida autentificarea utilizatorului.

```
mysql> select * from utilizatori;
+-----+-----+-----+-----+
| idUtilizator | eMailUtilizator | passUtil | passSalt |
+-----+-----+-----+-----+
| 7            | abc@gmail.com   | 653f4500be48960a76b14117e9d7e76ec5568f6caab51fb2f7e802131a06a405 |
| W+IRGEk1    | 11             | f63e6aeadafe6ba1beefb4bf765fc9e74c311bac0a49dbf8fc272d967f75f8cc |
| U|hu@@@G    |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

```
653f4500be48960a76b14117e9d7e76ec5568f6caab51fb2f7e802131a06a405
653f4500be48960a76b14117e9d7e76ec5568f6caab51fb2f7e802131a06a405
true
f63e6aeadafe6ba1beefb4bf765fc9e74c311bac0a49dbf8fc272d967f75f8cc
f63e6aeadafe6ba1beefb4bf765fc9e74c311bac0a49dbf8fc272d967f75f8cc
true

Process finished with exit code 0
```

Figura 2 Rezultatul testului algoritmilor de hashing al aplicațiilor web și mobile

Aplicația mobilă a fost proiectată să poată interacționa cu baza de date, serverul de stocare și distribuție și cel de transmisiune în timp real și să poată să trimită comenzi Bluetooth către cameră. Înainte de a trimite comenzi aplicația verifică dacă utilizatorul a asociat telefonul mobil cu dispozitivul. Dacă da, se realizează conexiunea aplicației mobile cu aplicația camerei la nivel de socket Bluetooth pentru a putea trimite comenzi. Altfel utilizatorul trebuie să își asocieze telefonul la dispozitiv sau să pornească serviciile Bluetooth pe telefon.

Aplicația web a fost proiectată să fie o alternativă limitată a aplicației mobile. Limita impusă este lipsa de comenzi Bluetooth datorită suportului limitat pentru API-uri Bluetooth în browserele web. Posibilitatea de a vizualiza înregistrări, transmisiunea în timp real, de a modifica sau crea contul utilizatorului sunt dezvoltate pentru aplicația web.

Pentru a pune în funcțiune aplicația web s-a folosit serverul apache tomcat pentru a pune în funcțiune și a containeriza aplicația web. În figura de mai jos putem observa că s-a realizat conexiunea către server de pe browserul web al mașinii gazdă.

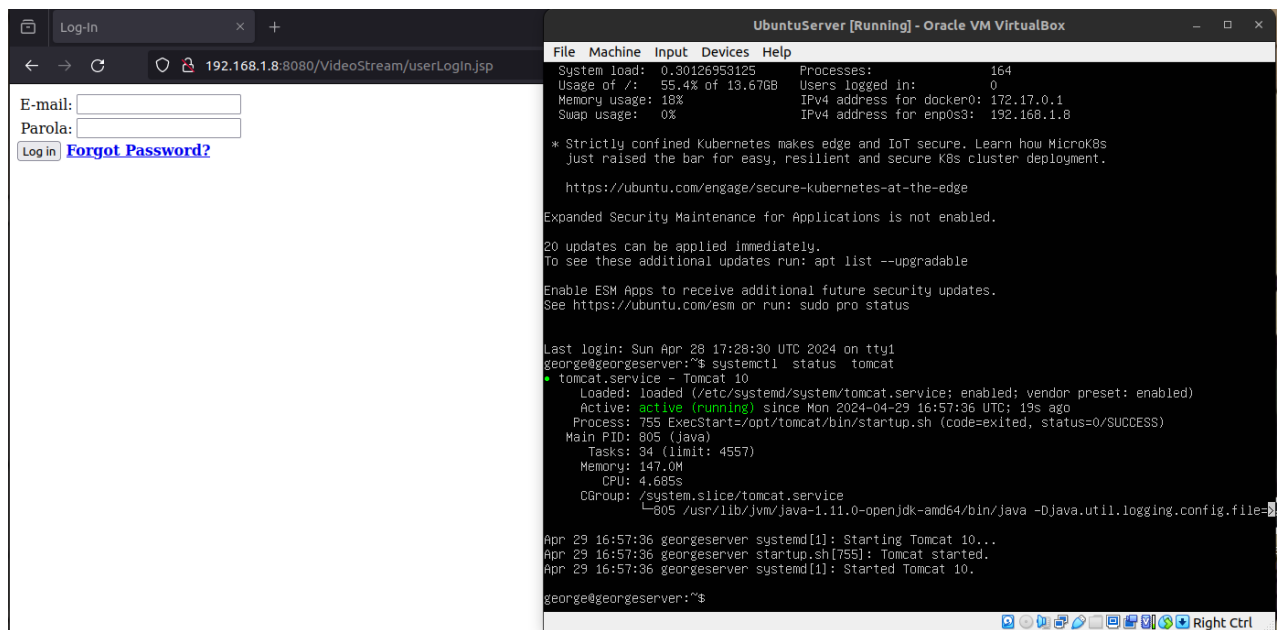


Figura 3 Aplicația web containerizată prin Apache Tomcat

Proiectarea și dezvoltarea acestui proiect a dus la următoarele concluzii:

- Este facil pentru un utilizator să transmită comenzi Bluetooth către un dispozitiv aflat într-o locație sau poziție greu de accesat de către acesta.
- Utilizatorul trebuie să se asigure că oprește camera înainte de a întrerupe alimentarea, acest fapt reprezintă un dezavantaj al dispozitivului.
- Virtualizarea serverului este avantajoasă deoarece permite unui utilizator să folosească servicii fără întreruperi. Metoda în care s-a implementat serverul este dezavantajoasă deoarece trebuie să existe mai multe versiuni ale bazei de date cu aceleași intrări pentru a asigura posibilitatea de conexiune la server a unui utilizator.
- Utilizarea dispozitivelor mobile și a comunicațiilor prin Bluetooth către un dispozitiv este un avantaj al proiectului. Nu este viabil să avem un dispozitiv separat pentru a trimite comenzi dispozitivului când majoritatea telefoanelor mobile pot folosi Bluetooth.
- Constrângerea utilizatorului la browsere web care au suport pentru API-uri Bluetooth nu este o opțiune fiabilă.