

UNICONNECT: SISTEM DE VIDEOCONFERINȚĂ

Candidat: Petru-Lucian, MOCRIȘ
Coordonator științific: Ș.L. Dr. Ing. Valer BOCAN

Sesiunea: Iunie 2024

Cuprins

1. INTRODUCERE	5
1.1 Context	5
1.2 Stadiul actual al problemei în România	5
1.3 Obiective.....	6
2. PREZENTAREA APLICAȚIEI UNICONNECT	7
2.2 Arhitectura sistemului	7
2.3 Cerințe software.....	8
2.4 Diagrama Use-Case a aplicației UniConnect.....	9
3. FUNDAMENTELE TEORETICE.....	11
3.1 Limbajele de programare HTML și CSS	11
3.2 Limbajul de programare JavaScript	11
3.3 Tehnologia WebRTC	12
4. IMPLEMENTAREA APLICAȚIEI UNICONNECT.....	15
4.1 Back-end-ul aplicației.....	15
4.2 Front-end-ul aplicației	22
5. TESTAREA ȘI VALIDAREA APLICAȚIEI UNICONNECT	27
6. CONCLUZII ȘI DIRECȚII DE DEZVOLTARE	33
6.1 Concluzii	33
6.2 Direcții de dezvoltare	33
7. BIBLIOGRAFIE	35

Lista figurilor

Figură 1 Google Meet ^[5]	6
Figură 2 Microsoft Teams ^[4]	6
Figură 3 ZOOM ^[3]	6
Figură 4 Arhitectura de sistem. Sistemul conține aplicația în sine și cele 2 servere de tip STUN. Server-ul principal este server-ul STUN 1 (Session Traversal Utilities for NAT ^[8]). În cazul în care server-ul STUN 1 nu comunică, atunci se trece pe server-ul STUN 2. De la server merge mai departe la ceilalți participanți.	8
Figură 5 Diagrama Use-Case	9
Figură 6 Web API ^[13]	12
Figură 7 WebRTC ^[18]	13
Figură 8 Conexiunea dintre 2 persoane în WebRTC ^[20]	13
Figură 9 Arhitectura unui server WebRTC ^[21]	14
Figură 10 Agora RTM ^[23]	15
Figură 11 Mesajele trimise în consolă când o cameră este creată iar un utilizator a intrat.	19
Figură 12 Un singur utilizator prezent în cameră	19
Figură 13 Mai mult de un utilizator prezent în cameră	20
Figură 14 Compatibilitatea API-ului AudioContext cu diverse navigatoare de internet ^[25]	22
Figură 15 Avertizarea utilizatorului pentru faptul că a introdus o valoare greșită	24
Figură 16 Aplicația UniConnect rulează în navigatorul web Opera	27
Figură 17 Aplicația UniConnect rulează în navigatorul web Microsoft Edge	28
Figură 18 Aplicația UniConnect rulează în navigatorul web Google Chrome	28
Figură 19 Aplicația UniConnect rulează în navigatorul web Mozilla Firefox	29
Figură 20 Aplicația UniConnect rulează când conexiunea la internet este făcută prin cablu	30
Figură 21 Aplicația UniConnect rulează când conexiunea la internet este făcută prin wireless	30
Figură 22 Aplicația UniConnect nu rulează fără conexiune la internet	31
Figură 23 Orice ID care nu este format din cifre este considerat necorespunzător	31
Figură 24 Bug-uri la nivelul serverelor afectează conexiunea dintre utilizatori	32

REZUMAT

Aplicația a luat naștere din ideea de a satisface nevoia studenților atunci când intervine problema de distanță. Cu toții știm de existența aplicației Zoom, care din păcate poate fi folosită la capacitate maximă doar în schimbul unui abonament plătit. Cu astfel de probleme m-am confruntat și eu în trecut iar din această cauză, aplicația UniConnect este concepută să satisfacă nevoile studenților fără a fi nevoiți să plătească o sumă de bani.

1. INTRODUCERE

1.1 Context

Un sistem de videoconferință este o tehnologie modernă de telecomunicație, prin intermediul căreia mai multe persoane se pot conecta într-un singur loc de la distanță. ^[1]

La nivel global, cel puțin o persoană pe zi are de-a face cu o ședință, ori un training, curs sau un simplu apel în mediul online. Pentru aceste contexte există și sunt folosite diverse aplicații, care de cele mai multe ori te obligă să plătești un abonament lunar pentru a beneficia de toate implementările lor. Sistemele de videoconferință sunt cel mai des utilizate în educație și mediul de afaceri, dar, uneori sunt folosite și pentru a vorbi cu cei dragi care se află la distanță.

Pentru a te putea folosi de aceste aplicații ai nevoie de o cameră web, un microfon, o pereche de căști și un calculator/laptop sau un telefon/tabletă. Acest tip aplicații este foarte benefic deoarece te scutește de timp pierdut și costuri mari, fiind mai ușor și mai accesibil să vorbești sau să te vezi cu o persoană doar printr-un simplu click, decât să mergi până la aceasta, mai ales dacă se află în alt oraș sau în altă țară.

1.2 Stadiul actual al problemei în România

Luând în calcul problemele actuale din acest domeniu, majoritatea problemelor au apărut în domeniul educației. Când țara noastră a fost lovită de pandemie, toate școlile și universitățile au fost închise iar majoritatea au fost luate pe nepregătite. La început, mulți elevi sau studenți au luat asta ca pe o vacanță până s-au găsit soluții la toate problemele. Ulterior s-au găsit soluții precum aplicațiile Zoom, Microsoft Teams sau Google Meet, dar în cazul facultăților unde vorbim de sute de studenți, aceste aplicații nu au fost cele mai eficiente deoarece fără un abonament plătit în cazul aplicației Zoom, timpul unei întâlniri este de maxim 40 de minute cu un număr limită de 100 de persoane. Din această cauză, mulți studenți nu aveau loc sau la fiecare 40 de minute erau nevoiți să creeze o nouă întâlnire.

Aplicația ZOOM (Figura 1) este o platformă care oferă o interfață prietenoasă utilizatorilor și este cunoscută pentru că este ușor de folosit dar și pentru funcțiile sale robuste. Zoom dispune de funcții precum mesagerie, partajarea ecranului sau înregistrarea apelului. ^[2] Această aplicație fiind dezvoltată de o întreagă echipă.

Aplicația Microsoft Teams (Figura 2) este o aplicație care face parte din pachetul Microsoft 365 și este cel mai des utilizată de companii. Pe partea de funcții, Microsoft Teams dispune de gestionarea dispozitivelor, suport pentru ecran dublu, o tablă pe care se poate scrie și de detectarea proximității. ^[2]

Aplicația Google Meet (Figura 3) este o aplicație dezvoltată de gigantul tehnologic Google. Google Meet este o soluție de colaborare securizată care are ca și funcții sincronizarea cu calendarul Google, reprogramarea ședințelor sau crearea ușoară a unei întâlniri. [2]



Figură 3 ZOOM^[3]



Figură 2 Microsoft Teams^[4]



Figură 1 Google Meet^[5]

1.3 Obiective

Obiectivul aplicației UniConnect este acela de a implementa toate funcțiile necesare domeniului educațional și de a beneficia utilizatorii de ele fără a fi nevoiți să plătească.

În final, se dorește ca aplicația UniConnect să fie una stabilă și să satisfacă toate cerințele utilizatorilor.

2. PREZENTAREA APLICAȚIEI UNICONNECT

2.1 Cerințe de sistem

Cerințele de sistem sunt acele specificații care trebuie îndeplinite pentru ca un program să ruleze eficient. Cerințele de sistem pot fi de tip Software sau Hardware. Cerințele de sistem se împart în două categorii: Cerințe de sistem minime și Cerințe de sistem recomandate.^[6]

Cerințele de sistem sunt importante mai ales pentru echipa de dezvoltare deoarece pe baza acestora se pot defini cazuri de testare dar, acestea sunt importante și pentru utilizatori deoarece în funcție de acestea se știe dacă se poate folosi aplicația și dacă pe viitor sunt necesare viitoare îmbunătățiri ale sistemului pentru o mai bună rulare a aplicației.

Cerințele de sistem reflectă ceea ce programul sau aplicația ar trebuie să facă și cum să se comporte.

Cerințele de sistem ale aplicației UniConnect sunt următoarele:

- Aplicația trebuie să permită utilizatorilor să creeze noi camere sau să intre în cele care sunt deja create;
- Aplicația trebuie să permită utilizatorilor să pornească/oprească camera sau microfonul;
- Aplicația trebuie să permită utilizatorilor să își partajeze ecranul;
- Aplicația trebuie să permită utilizatorilor să iasă din cameră oricând doresc.

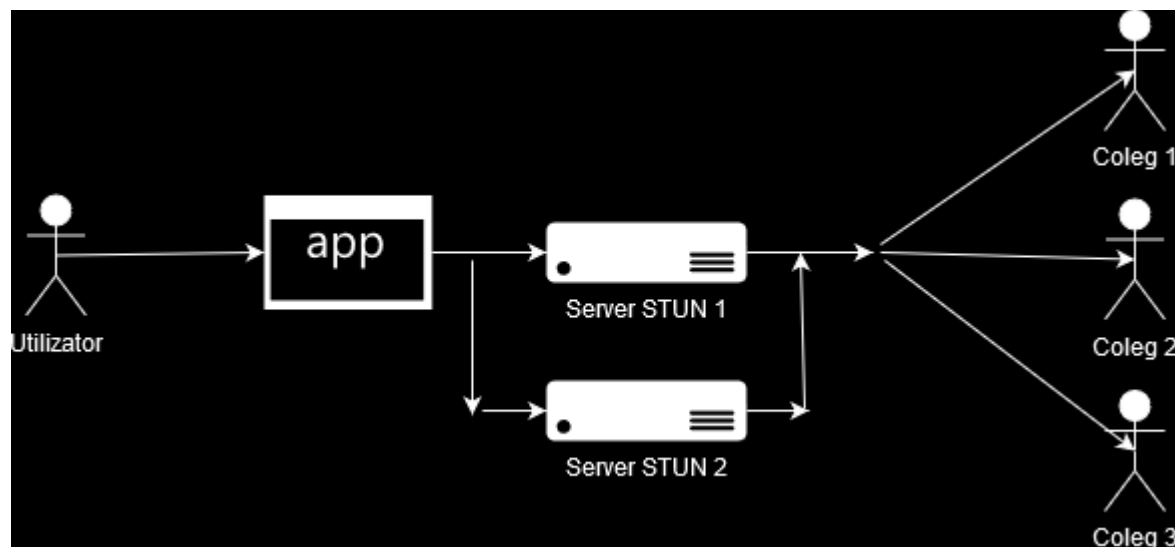
Pentru a folosi aplicația UniConnect aveți nevoie de următoarele:

- Periferice: Căști, cameră web, mouse, tastatură;
- Cerințe hardware: Placă de sunet;
- Conexiune la internet;
- Sistem de operare instalat.

2.2 Arhitectura sistemului

Arhitectura sistemului este o reprezentare a structurii și comportamentului unui sistem. Aceasta cuprinde componentele sistemului și ne arată cum acest sistem este folosit.^[7]

Arhitectura sistemului pentru aplicația UniConnect este prezentată în diagrama următoare:



Figură 4 Arhitectura de sistem. Sistemul conține aplicația în sine și cele 2 servere de tip STUN. Server-ul principal este server-ul STUN 1 (Session Traversal Utilities for NAT[®]). În cazul în care server-ul STUN 1 nu comunică, atunci se trece pe server-ul STUN 2. De la server merge mai departe la ceilalți participanți.

2.3 Cerințe software

Cerințele software sunt cele care descriu în detaliu funcționalitatea unui sistem. Fiecare sistem are la rândul său și constrângeri. Cerințele software sunt foarte importante pentru echipa de testare deoarece pe baza acestora sunt create scenariile de testare. Pentru a începe dezvoltarea unui sistem, este necesar să avem un document SRS pus la punct.

Documentul SRS sau Specificația Cerințelor Software este o colecție structurată de informații referitoare la sistemul software care urmează să fie dezvoltat. Cerințele software sunt de două tipuri: Cerințe software funcționale și Cerințe software non-funcționale. Cerințele software reprezintă înțelegerea dintre client și contractor sau furnizor în legătură cu modul de funcționare al sistemului.^[9]

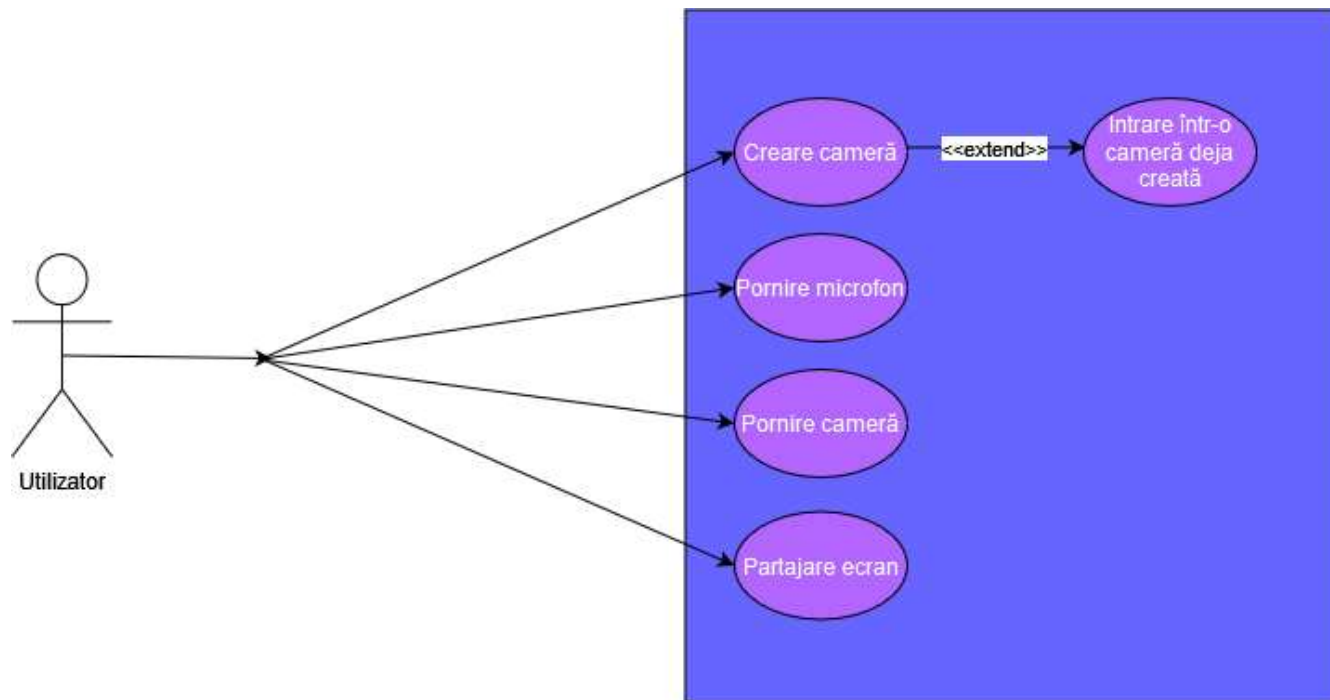
Cerințele software în urma cărora aplicația UniConnect a fost dezvoltată sunt:

- Cerințe funcționale:
 - Utilizatorul trebuie să aibă acces să creeze o cameră sau să intre într-o cameră deja creată;
 - Utilizatorul trebuie să aibă acces să poată partaja ecranul;
 - Utilizatorul trebuie să aibă acces să își folosească microfonul sau camera web.

- Cerințe non-funcționale:
 - Timp de latență minim pentru asigurarea transmisiunii audio și video;
 - Sistemul trebuie să fie compatibil cu orice navigator web;
 - Sistemul trebuie să fie scalabil pentru a asigura o încărcare balansată între servere;
 - Sistemul trebuie să fie scalabil orizontal pentru a suporta un număr cât mai mare de utilizatori și cât mai multe camere create simultan;
 - Utilizatorul trebuie să dispună de o interfață ușor de folosit;
 - În timpul vorbirii, nu trebuie să fie prezent efectul de ecou;
 - Aplicația trebuie să fie capabilă să accepte mai multe tipuri de camere web (camere web cu o claritate ridicată sau camere web cu o claritate mai scăzută).

2.4 Diagrama Use-Case a aplicației UniConnect

Diagrama Use-Case reprezintă cum este folosit un sistem de către unul sau mai mulți utilizatori. Aceasta nu reprezintă cum funcționează sistemul.



Figură 5 Diagrama Use-Case

Utilizatorul poate să creeze o camera nouă sau poate intra deja într-una existentă. Pe lângă acestea, utilizatorul poate să își pornească sau să își oprească camera, să își pornească sau să își oprească microfonul și să își partajeze ecranul.

3. FUNDAMENTELE TEORETICE

3.1 Limbajele de programare HTML și CSS

HyperText Markup Language sau mai des folosit HTML, este un limbaj descriptiv utilizat pentru a crea site-uri. Este cel mai simplu limbaj dintre cele folosite pentru crearea site-urilor.^[10]

Cascading Style Sheets sau CSS, este un limbaj folosit pentru editarea și formatarea elementelor prezente pe site. Majoritatea persoanelor confundă HTML și CSS ca fiind același lucru, dar fiecare este conceput pentru alt scop.^[10]

Cele mai folosite limbaje în domeniul web sunt PHP, Java, JavaScript, ASP.NET, React, Angular sau Python iar având cunoștințe în limbajul HTML reprezintă un avantaj în învățarea acestora.^[10]

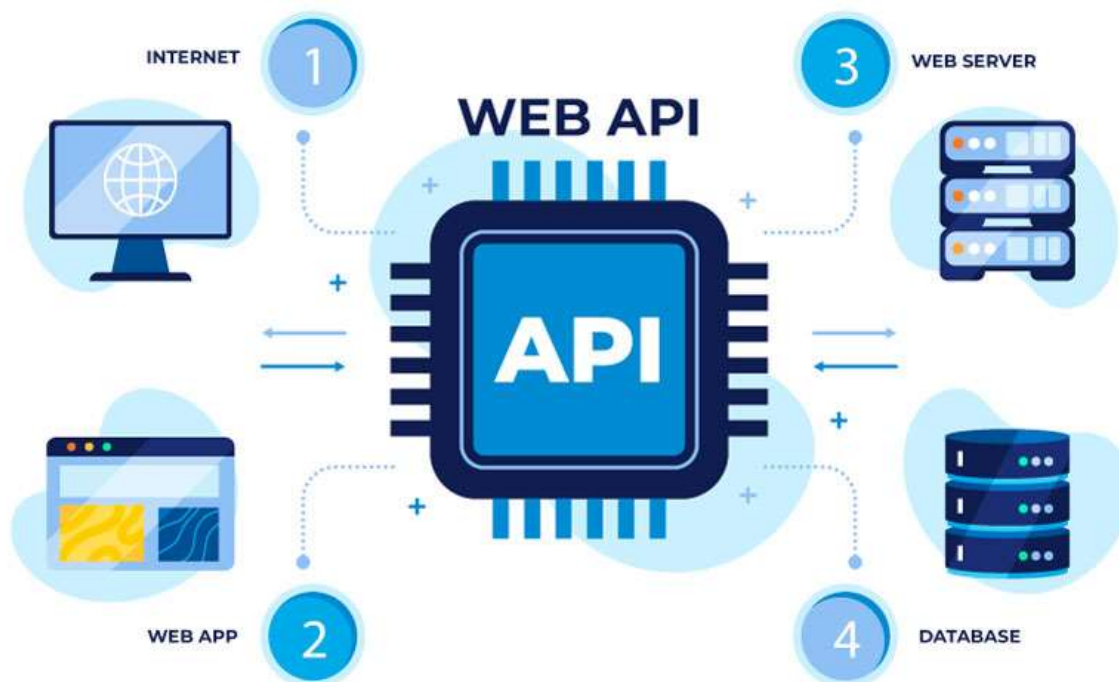
3.2 Limbajul de programare JavaScript

JavaScript sau JS, este un limbaj bazat pe programarea orientate obiecte. Folosit în general pentru implementarea funcționalităților care stau la baza unui site web, codul JS este rulat de către browser. La începuturile sale, în anul 1995, JS a apărut sub numele de Mocha, ulterior LiveScript, iar în final primind numele de JavaScript. Multe persoane consideră că JavaScript și Java vin din aceeași “familie”, dar ele sunt concepute pentru chestii diferite.^[11]

Reprezentarea unei pagini web este reținută de către browsere sub forma unui arbore de obiecte, aceste obiecte fiind puse la dispoziția script-urilor de JS. Acest arbore este cunoscut sub denumirea de DOM sau Document Object Model. Pentru acest DOM, există un standard numit W3C, care oferă baza scrierii de script-uri portabile, pentru ca acestea să poată fi rulate pe toate navigatoarele de internet.^[11] W3C sau cunoscut ca și Consorțiul Web, are ca scop dezvoltarea de standarde pentru World Wide Web.^[12]

În ultima perioadă, cea mai întâlnită tehnică de dezvoltare a paginilor web este AJAX sau “Asynchronous JavaScript and XML”. Tehnica constă în executarea cererilor de tip HTTP în fundal fără a fi necesară reîncărcarea în totalitate a paginii web.^[11]

Majoritatea aplicațiilor web sunt dezvoltate folosind API-uri. API sau Application Programming Interface este un set de protocoale, sub-programe definite care ajută la programarea aplicațiilor.^[14] API-urile trimit date în mod bidirecțional.^[15]

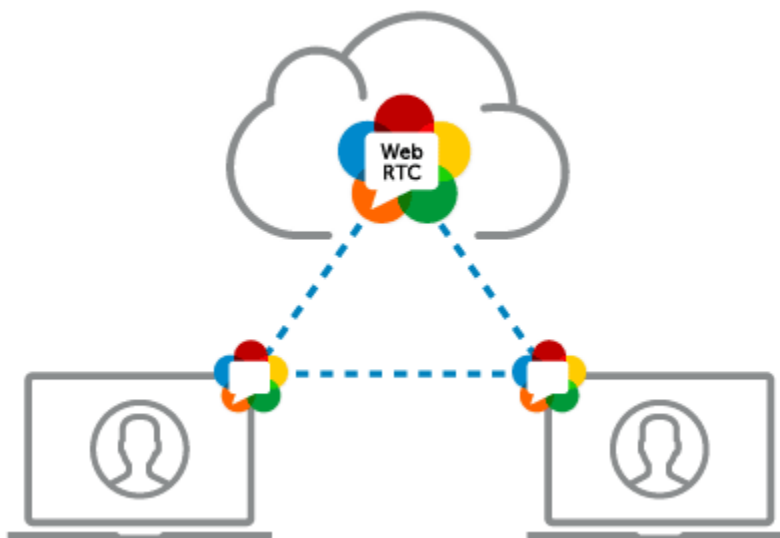


Figură 6 Web API^[13]

3.3 Tehnologia WebRTC

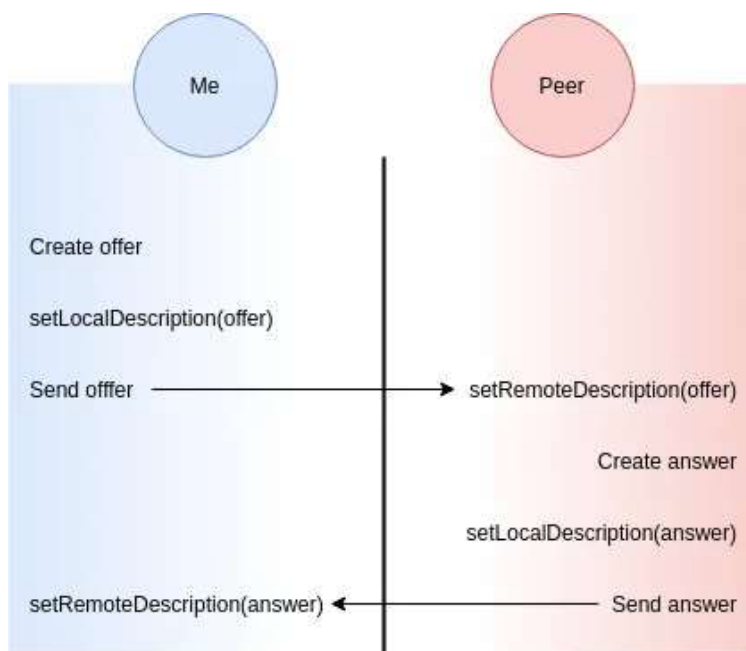
WebRTC sau Web Real-Time Communications, este o tehnologie open-source care a fost promovată de Google. Această tehnologie permite comunicarea în timp real prin intermediul JavaScript API. Codec-ul folosit de WebRTC care convertește semnalele analogice în semnale digitale pentru a le trimite pe o rețea de date este VP8.^[16]

VP8 sau Video Compression Format, este specificație folosită la codarea și decodarea video de înaltă calitate. Codec-ul VP8 este în contradicție cu codec-ul H.264 care este oferit gratuit de către Google.^[17]



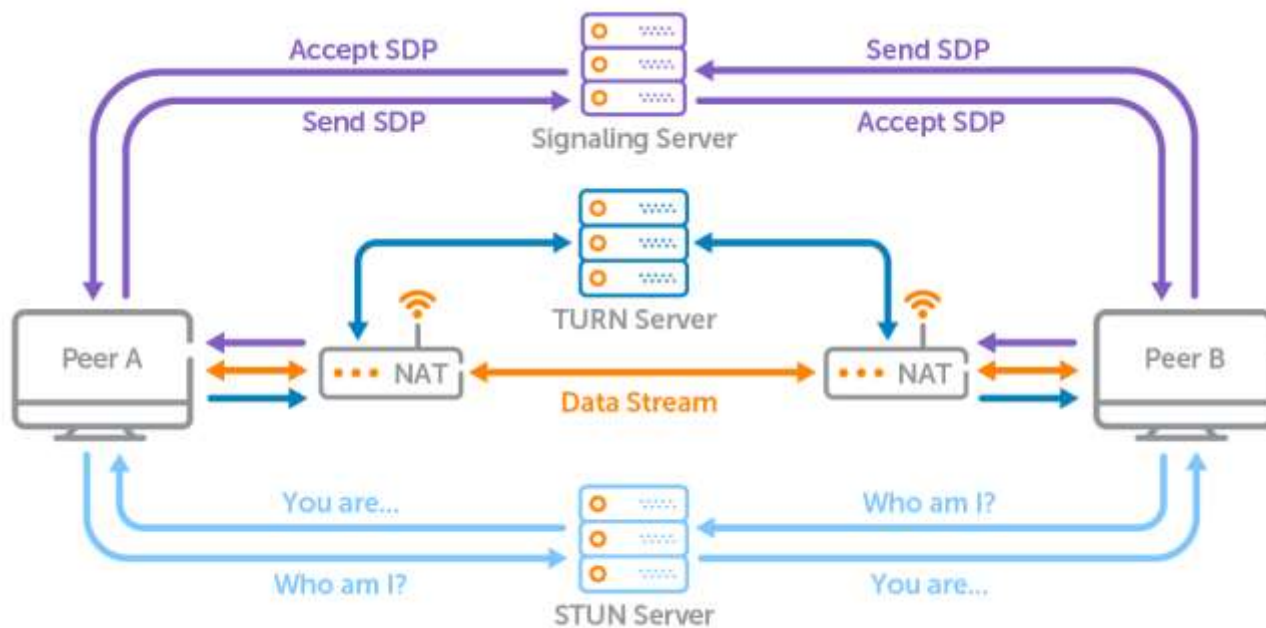
Figură 7 WebRTC^[18]

AgoraRTM folosit în combinație cu WebRTC poate oferi o bază solidă când vine vorba de dezvoltarea aplicațiilor de tip Real-Time Communications.^[19]



Figură 8 Conexiunea dintre 2 persoane în WebRTC^[20]

Arhitectura unui server WebRTC:



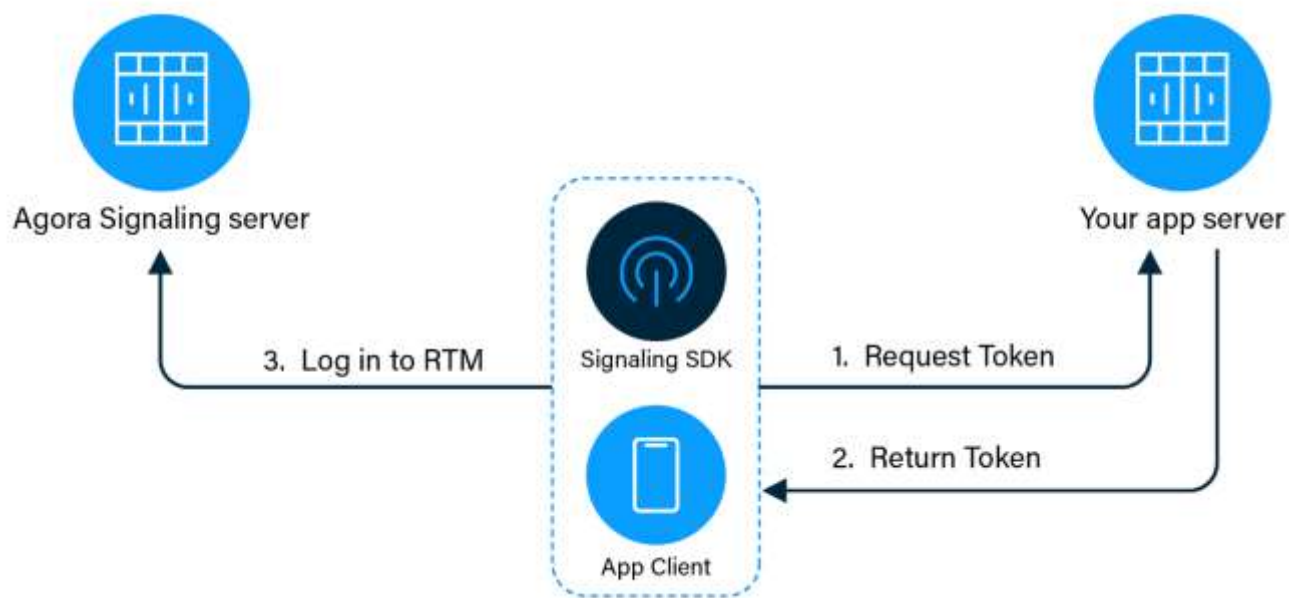
Figură 9 Arhitectura unui server WebRTC^[21]

4. IMPLEMENTAREA APLICAȚIEI UNICONNECT

4.1 Back-end-ul aplicației

Pentru a dezvolta partea de back-end a aplicației m-am folosit de tehnologia WebRTC împreună cu SDK-ul Agora RTM.

SDK sau Software Development Kit, este un set de instrumente care ajută la dezvoltarea software deoarece totul este predefinit deja.^[22]



Figură 10 Agora RTM^[23]

Back-end-ul aplicației este împărțit în două fișiere. Unul pentru toată funcționalitatea aplicației, numit "main.js" și unul pentru funcționalitatea paginii principale din care se pornesc apelurile numit "lobby.js".

```
document.addEventListener('DOMContentLoaded', function() {  
  const form = document.getElementById('joinForm');  
  if (form) {  
    form.addEventListener('submit', function(event) {  
      event.preventDefault();  
      if (event && event.target) {  
        let invite = event.target.inviteLink.value;  
        if (invite) {  
          window.location = `main.html?room=${invite}`;  
        }  
      }  
    });  
  }  
});
```

În codul de mai sus are loc partea de validare a butonului submit și se verifică dacă este trimis către server atunci când utilizatorul dorește să creeze o cameră nouă sau să intre într-o cameră deja creată. Dacă utilizatorul introduce în "joinForm" id-ul unei camere iar acesta este unul valid, mai departe se trimite o cerere la server pentru a-i permite utilizatorului să intre. Dacă utilizatorul are permisiunea de a intra, acesta este redirecționat către pagina camerei prin linia de cod "window.location = `main.html?room=\${invite}`". În fișierul main.html sunt implementate butoanele pe care utilizatorul le poate folosi și chenarul în care este arătat ce captează camera web.

```
navigator.mediaDevices.getSupportedConstraints();  
navigator.mediaDevices.getUserMedia({  
  audio: {  
    echoCancellation: true,  
    noiseSuppression: true,  
    autoGainControl: true  
  }  
});
```

Această bucată de cod reprezintă constrângerile pentru partea de audio, pentru a reduce efectul de ecou și pentru a anula zgomotul din fundal.


```
let APP_ID = "9aebd21fd5b24c7692b448e0f38df752"

//user
let token = null;
let uid = String(Math.floor(Math.random() * 10000))

let client;
let channel;

//room
let queryString = window.location.search
let url = new URLSearchParams(queryString)
let roomid = url.get('room')

if(!roomid) {
  window.location = 'lobby.html'
}

let localStream;
let remoteStream;
let peerConnection;
```

Mai sus au fost declarate variabilele globale de care ne vom folosi pe parcursul dezvoltării aplicației. Variabila „APP_ID” reprezintă identificatorul aplicației din consola SDK-ului Agora RTM. Pentru indentificatorul utilizatorilor „uid” am folosit o formulă pentru a genera un indentificator aleatoriu de fiecare dată când un utilizator intră. Dacă același utilizator intră și reintră în conferință, acesta va avea un alt identificator. Identificatorul camerei este preluat de comanda “url.get(‘room’)”. Variabilele “localStream”, “remoteStream” și “peerConnection” sunt folosite la realizarea conexiunii dintre utilizator și ceilalți utilizatori virtuali.

```
const constraints = {
  video: {
    width: {min:854, max:4096},
    height: {min:480, max:2160},
  },
  audio: {
    echoCancellation:true,
  }
};
```

Aici avem constrângerile pentru captarea video. Minimul acceptat pentru claritatea unei camere web este de 480p, iar maximul acceptat este o rezoluție de 4K. Dacă nu sunt respectate aceste valori, camera nu poate fi pornită.

```
const servers = {
  iceServers: [
    {
      urls: ['stun:stun1.l.google.com:19302', 'stun:stun2.l.google.com:19302']
    }
  ]
}
```

Serverele de tip ICE sau “Interactive Connectivity Establishment server”, sunt importante în stabilirea conexiunii de tip peer-to-peer. Serverele ajută la localizarea interfețelor de rețea și a IP-urilor care permit conexiunea peer-to-peer.^[24]

Personal am folosit serverele de la Google deoarece sunt gratuite și ajută ca aplicația să fie dezvoltată cu un cost cât mai mic. Primul server este cel principal, prin care se realizează conexiunea, iar al doilea server este cel de rezervă, în cazul în care primul server pică sau nu poate stabili conexiunea.

```
let init = async () => {
  client = await AgoraRTM.createInstance(APP_ID)
  await client.login({uid, token})

  channel = client.createChannel('roomid')

  await channel.join()

  channel.on('Member-joined', handleUserJoined)
  channel.on('Member-left', handleUserLeft)

  client.on('Message-from-peer', handleMessageFromPeer)

  localStream = await navigator.mediaDevices.getUserMedia(constraints)
  document.getElementById('user1').srcObject = localStream
}
```

Pentru a crea instanțele, am creat o funcție asincronă care trimite un mesaj în consola navigatorului web atunci când o persoană a intrat sau a ieșit din cameră.

```
RTM:INFO The Agora RTM Web SDK version is 1.4.4. +0ms
RTM:INFO The process ID is 1df6d1e5-a098-4276-b826-35d802e672d8. +1ms
RTM:INFO RTM Client logging in as 14**. +7ms
RTM:INFO A channel created, id room**. +1s
RTM:INFO The channel room** joined successfully. +139ms
RTM:INFO Channel MemberJoined - room**, memberId: 14**. +311ms
```

Figură 11 Mesajele trimise în consolă când o cameră este creată iar un utilizator a intrat.

În cazul în care un utilizator nu reușește să se conecteze, va apărea un mesaj de eroare în consolă.

```
let handleUserLeft = (MemberId) => {
  document.getElementById('user2').style.display = 'none'
  document.getElementById('user1').classList.remove('smallFrame')
}
```

Dacă este un singur utilizator în cameră, acesta se va vedea pe tot ecranul. Însă, dacă este o situație în care sunt prezenți mai mulți utilizatori, utilizatorul prezent în fața monitorului se va vedea într-un chenar mic amplasat în partea de sus a paginii iar pe suprafața cea mare vor apărea restul utilizatorilor. Dacă utilizatorul iese, acel chenar este scos.



Figură 12 Un singur utilizator prezent în cameră



Figură 13 Mai mult de un utilizator prezent în cameră

```
let createPeerConnection = async (MemberId) => {
  peerConnection = new RTCPeerConnection(servers)

  remoteStream = new MediaStream()
  document.getElementById('user2').srcObject = remoteStream
  document.getElementById('user2').style.display = 'block'

  document.getElementById('user1').classList.add('smallFrame')

  if(!localStream){
    localStream = await navigator.mediaDevices.getUserMedia({video:true, audio:false})
    document.getElementById('user1').srcObject = localStream
  }

  localStream.getTracks().forEach((track) => {
    peerConnection.addTrack(track, localStream)
  })

  peerConnection.ontrack = (event) => {
    event.streams[0].getTracks().forEach((track) => {
      remoteStream.addTrack(track)
    })
  }

  peerConnection.onicecandidate = async (event) => {
    if(event.candidate){
      client.sendMessageToPeer({text:JSON.stringify({'type':'candidate', 'candidate':event.candidate}}), MemberId)
    }
  }
}
```

În bucata de cod de mai sus are loc conexiunea de tip peer-to-peer. La începutul funcției a fost declarată variabila “remoteStream” care reține fluxul media al utilizatorului conectat de la distanță. Utilizatorul 1 sau cel prezent “fizic” este trimis către acel chenar mic unde se va vedea ce capturează camera web. După aceea, este verificat dacă “localStream” este definit. Dacă “localStream” nu este definit, acesta poate reține doar fluxul video, fără cel audio. Ulterior, fiecare urmă din fluxul local este adăugată la “peerConnection” pentru a permite conexiunea cu celălalt utilizator. Evenimentul de tip ICECandidate este trimis între utilizatori pentru a stabili conexiunea.

```
let MicON = async () => {
  console.log(localStream);
  let audio = localStream.getTracks().find(track => track.kind === 'audio')
  console.log(audio);

  if(audio.enabled) {
    audio.enabled = false
    document.getElementById('mic-button').style.backgroundColor = 'rgb(240, 50, 60)'
  }
  else {
    audio.enabled = true
    document.getElementById('mic-button').style.backgroundColor = 'rgb(179, 102, 249, .10)'
  }
}

let CameraON = async () => {
  let video = localStream.getTracks().find(track => track.kind === 'video')

  if(video.enabled) {
    video.enabled = false;
    document.getElementById('camera-button').style.backgroundColor = 'rgb(240, 50, 60)';
  }
  else {
    video.enabled = true;
    document.getElementById('camera-button').style.backgroundColor = 'rgb(179, 102, 249, .10)';
  }
}
```

Aici sunt definite funcțiile de control pentru partea audio și partea video. Totodată, aici sunt aplicate și culorile pentru fiecare buton. Fiecare funcție are butonul ei. Utilizatorul are posibilitatea de a-și opri sau porni microfonul sau camera web. Pe lângă acestea, utilizatorul poate să vadă partea video în modul “image-în-image”.

```
function shareScreen() {
  navigator.mediaDevices.getDisplayMedia({cursor: true}).then(stream => {
    let screenTrack = stream.getTracks()[0];
    senders.current.find(sender => sender.track.kind === 'video').replaceTrack(screenTrack);
    screenTrack.onended = function () {
      senders.current.find(sender => sender.track.kind === 'video').replaceTrack(userStream.current.getTracks()[1]);
    }
  })
}
```

Utilizatorul are posibilitatea de a-și partaja ecranul cu ceilalți utilizatori. Funcția trebuie să solicite permisiunea utilizatorului pentru a partaja ecranul, totodată este vizibil și cursorul mouse-ului.

Pentru o mai bună claritate a conținutului audio, am ales să aplic câteva metode de procesare și decodare cu ajutorul API-ului AudioContext. AudioContext este o interfață care are capacitatea de a face procesare audio.^[25]

	Desktop					Mobile					
	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android
AudioContext	✓ 35 ...	✓ 12 ...	✓ 25 ...	✓ 22 ...	✓ 14.1 ...	✓ 35 ...	✓ 25 ...	✓ 22 ...	✓ 14.5 ...	✓ 3.0 ...	✓ 37 ...

Figură 14 Compatibilitatea API-ului AudioContext cu diverse navigatoare de internet^[25]

4.2 Front-end-ul aplicației

Pentru realizarea interfeței aplicației au fost folosite limbajele de programare HTML și CSS.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>UniChat</title>

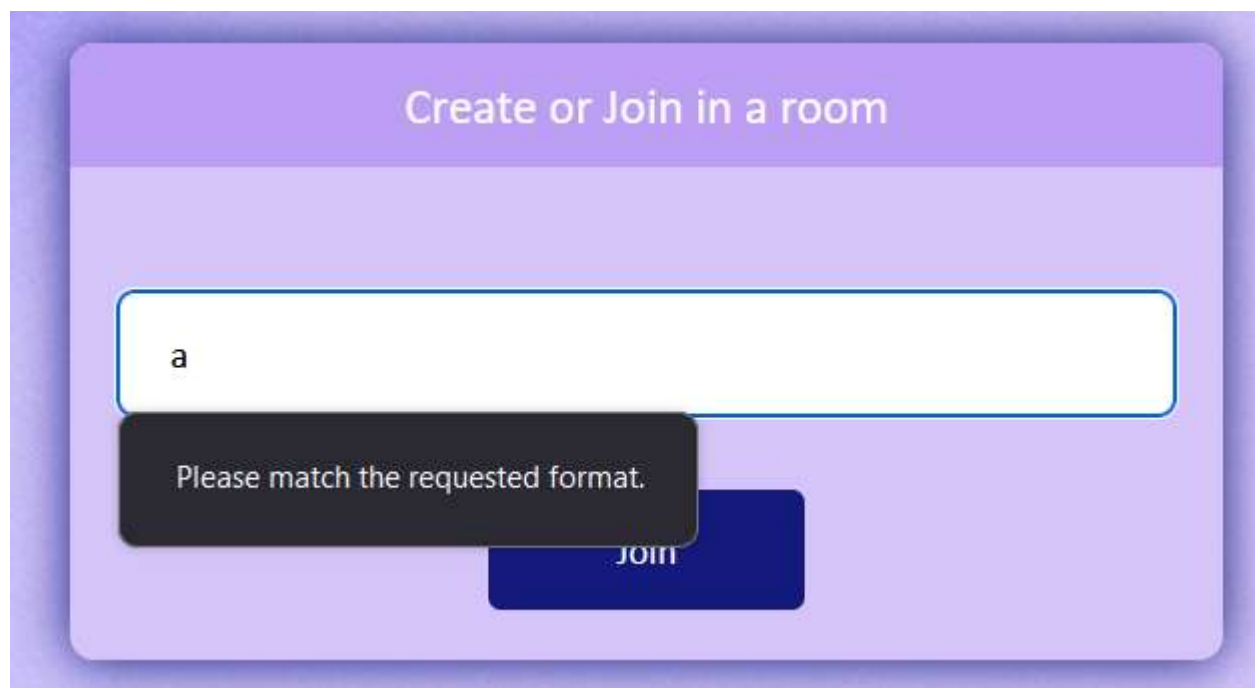
  <!-- JavaScript -->
  <script src="lobby.js"></script>
  <!-- Design -->
  <link rel="stylesheet" type="text/css" media="screen" href="lobby.css">
</head>

<body>
<center></center>
<main id="lobbyContainer">
  <div id="formContainer">
    <div id="formContainer-header">
      <p>Create or Join in a room</p>
    </div>

    <div id="formContent">
      <form id="joinForm">
        <input type="text" pattern="[0-9]" placeholder="Room ID" name="inviteLink" required>
        <center><input type="submit" value="Join"></center>
      </form>
    </div>
  </div>
</main>
</body>
</html>
```

O prezentare scurtă a acestei bucăți de cod ar fi că, aici a fost implementată pagina de creare sau de intrare într-o cameră existentă.

Pentru introducerea id-ului camerei a fost folosit un element de tip input, asupra căruia a fost aplicată constrângerea care permite doar introducerea de cifre. Dacă este introdusă altă valoare decât o cifră atunci utilizatorul va fi atenționat că a introdus ceva gresit.



Figură 15 Avertizarea utilizatorului pentru faptul că a introdus o valoare greșită

```
<body>
  <div id="video">
    <video class="video-player" id="user1" autoplay playsinline></video>
    <video class="video-player" id="user2" autoplay playsinline></video>
  </div>

  <div id="buttons">
    <div class="button-container" id="mic-button">
      
    </div>

    <div class="button-container" id="camera-button">
      
    </div>

    <div class="button-container" id="share-button">
      
    </div>

    <a href="lobby.html">
      <div class="button-container" id="leave-button">
        
      </div>
    </a>
  </div>
</body>
</html>
```


Celălalt fișier HTML conține butoanele cu ajutorul cărora utilizatorul acționează asupra părții audio și video. La butonul pentru ieșirea din cameră a fost adăugat link-ul de la pagina principală pentru a te putea întoarce la ea. Pe partea de aspect, toate modificările au fost făcute în fișierele CSS.

Câteva implementări în limbajul CSS vor fi prezentate mai jos:

- Chenarul în care este poziționată camera locală

```
@media screen and (max-width:600px) {  
  .button-container img{  
    height: 20px;  
    width: 20px;  
  }  
  
  .smallFrame{  
    height: 80px;  
    width: 120px;  
  }  
}
```

```
.smallFrame{  
  position: fixed;  
  top: 20px;  
  left: 20px;  
  height: 170px;  
  width: 300px;  
  border-radius: 5px;  
  border: 2px solid #b366f9;  
  box-shadow: 3px 3px 15px -1px rgba(0,0,0,0.77);  
  -webkit-box-shadow: 3px 3px 15px -1px rgba(0,0,0,0.77);  
  z-index: 999;  
}
```

- Fereastra în care este încadrată captura video a utilizatorului remote

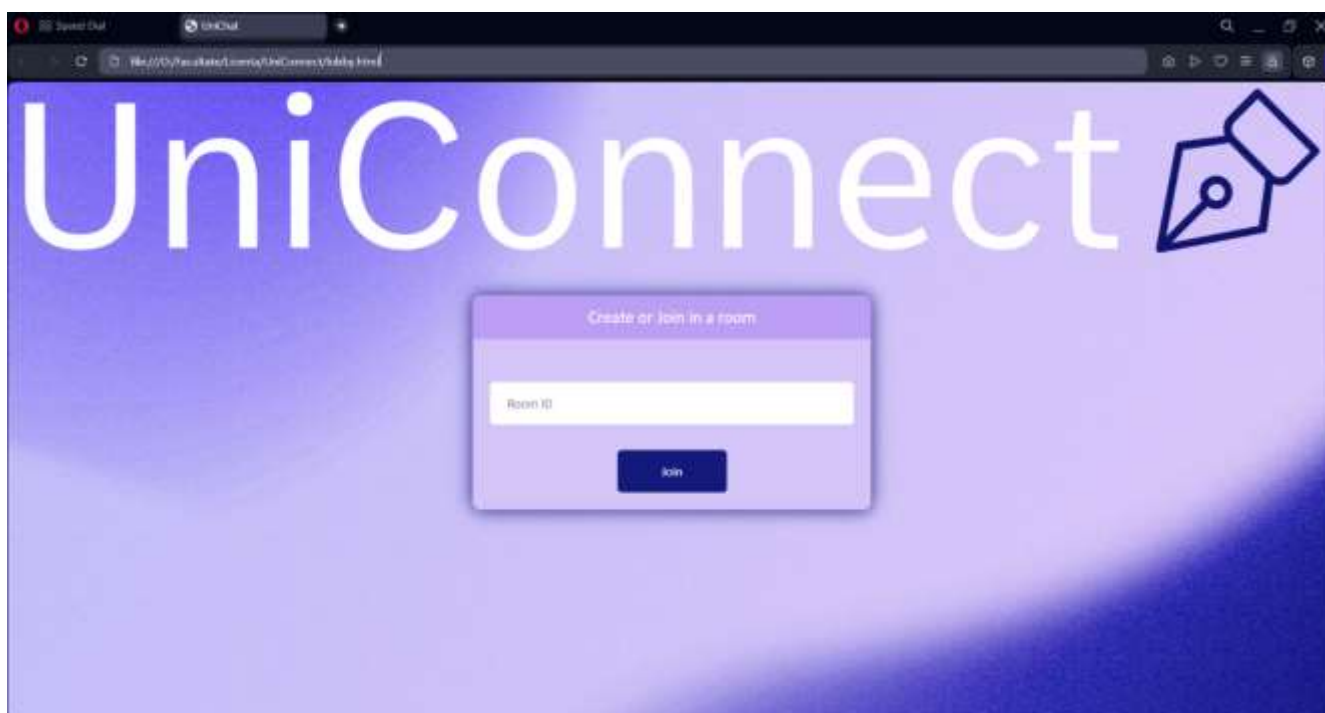
```
#video {  
  display: grid;  
  grid-template-columns: 1fr;  
  height: 100vh;  
  overflow: hidden;  
}  
  
.video-player {  
  background-color: #000000;  
  width: 100%;  
  height: 100%;  
  object-fit: cover;  
}
```

5. TESTAREA ȘI VALIDAREA APLICAȚIEI UNICONNECT

Pentru a testa aplicația UniConnect am decis să mă folosesc de testarea manuală. Testarea manuală este un tip de testare prin care software-ul este testat de către developer sau echipa de testare fără ajutorul unor scenarii de testare automatizate. Uneori, testarea automată poate să dea greș din cauza anumitor factori. Personal am decis să mă axez pe testarea de scenarii pozitive și scenarii negative.

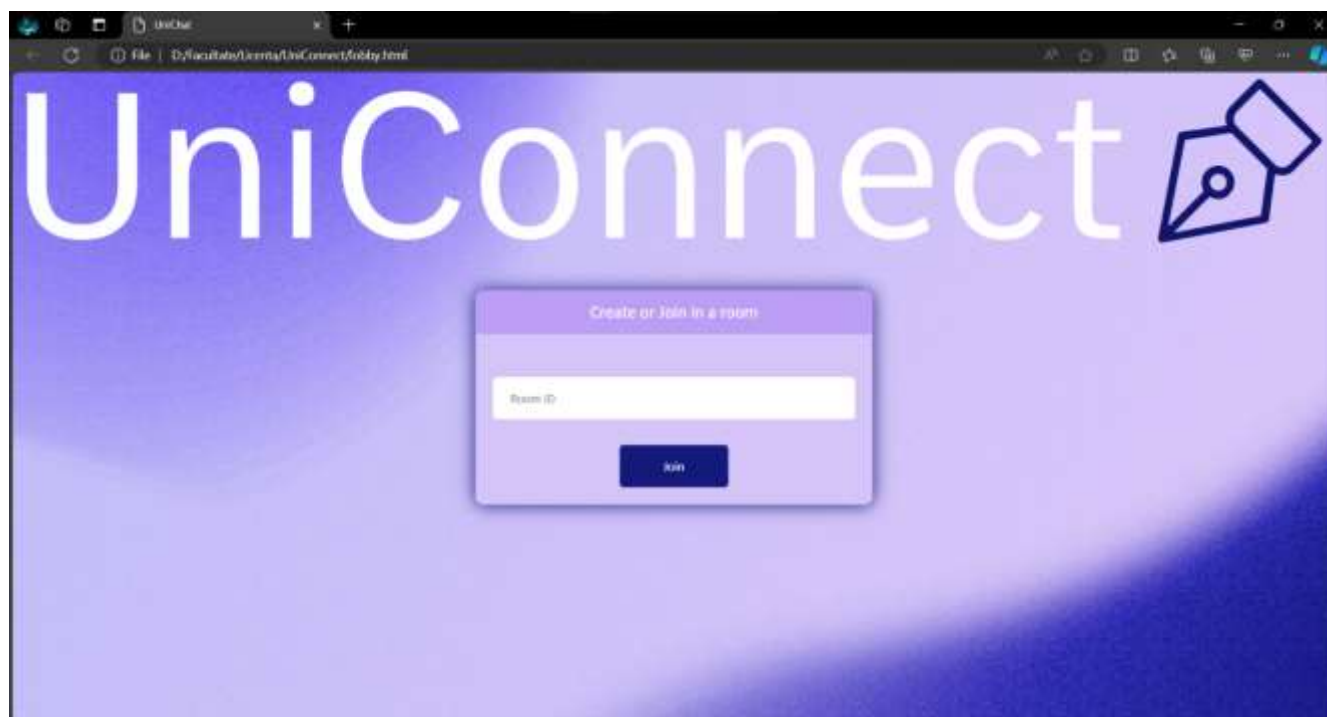
Pentru prima data, o să încep prin a prezenta scenarii pozitive pentru aplicația UniConnect:

- ✓ Aplicația rulează pe navigatorul web Opera



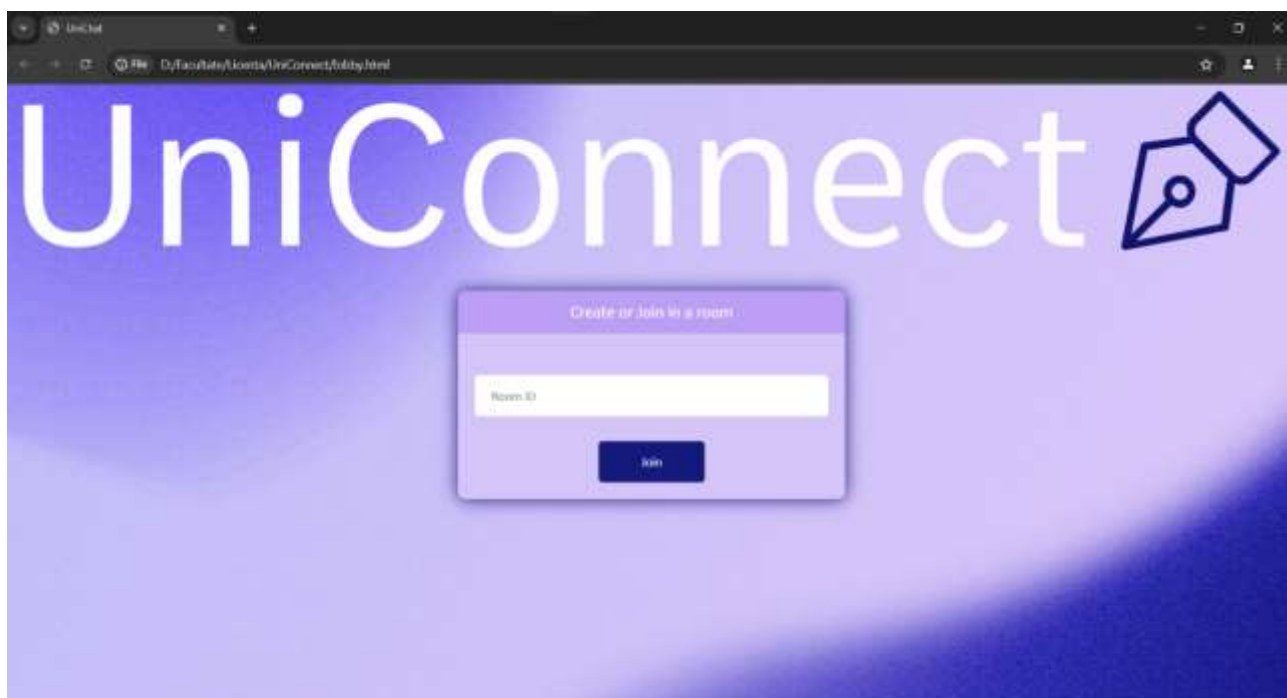
Figură 16 Aplicația UniConnect rulează în navigatorul web Opera

- ✓ Aplicația rulează pe navigatorul web Microsoft Edge



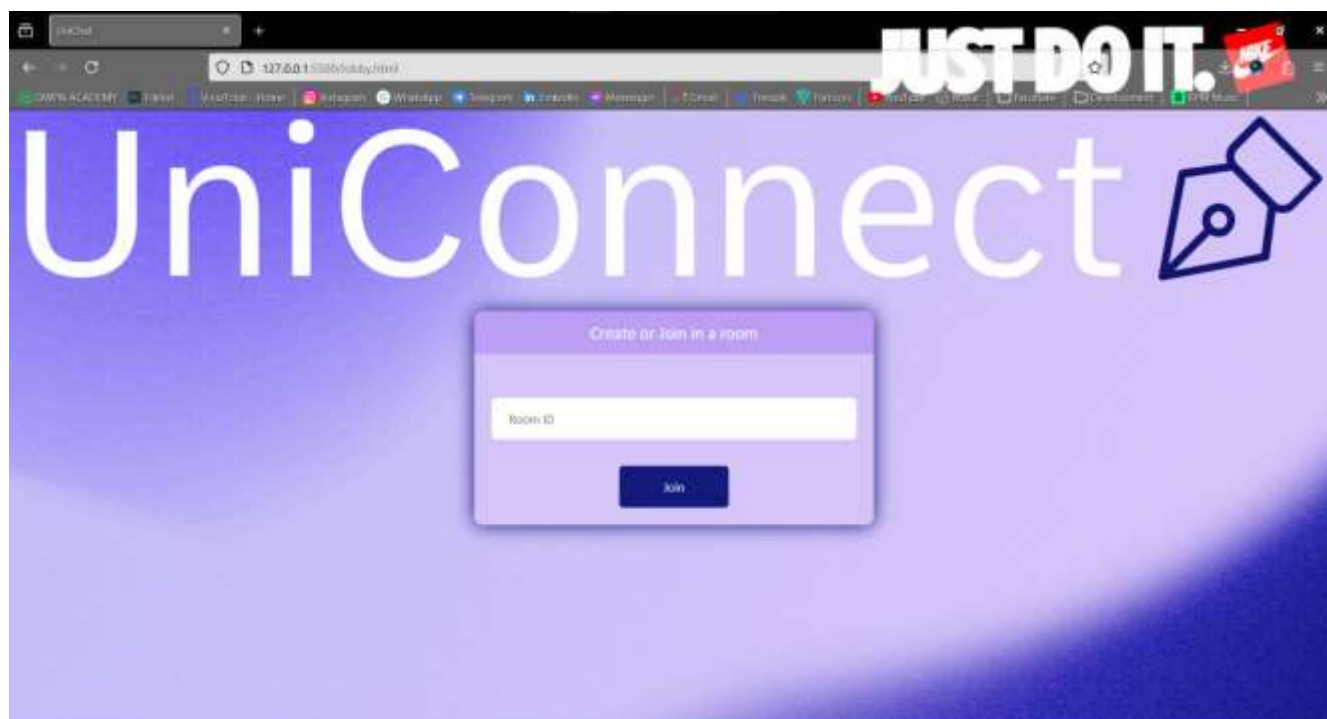
Figură 17 Aplicația UniConnect rulează în navigatorul web Microsoft Edge

✓ Aplicația rulează pe navigatorul web Google Chrome



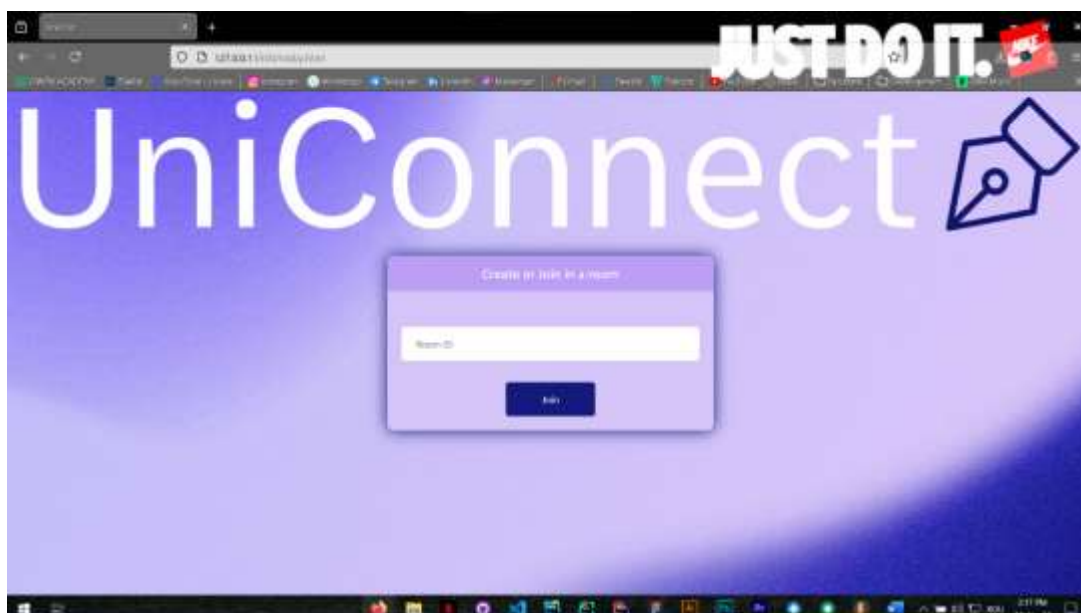
Figură 18 Aplicația UniConnect rulează în navigatorul web Google Chrome

- ✓ Aplicația rulează pe navigatorul web Mozilla Firefox



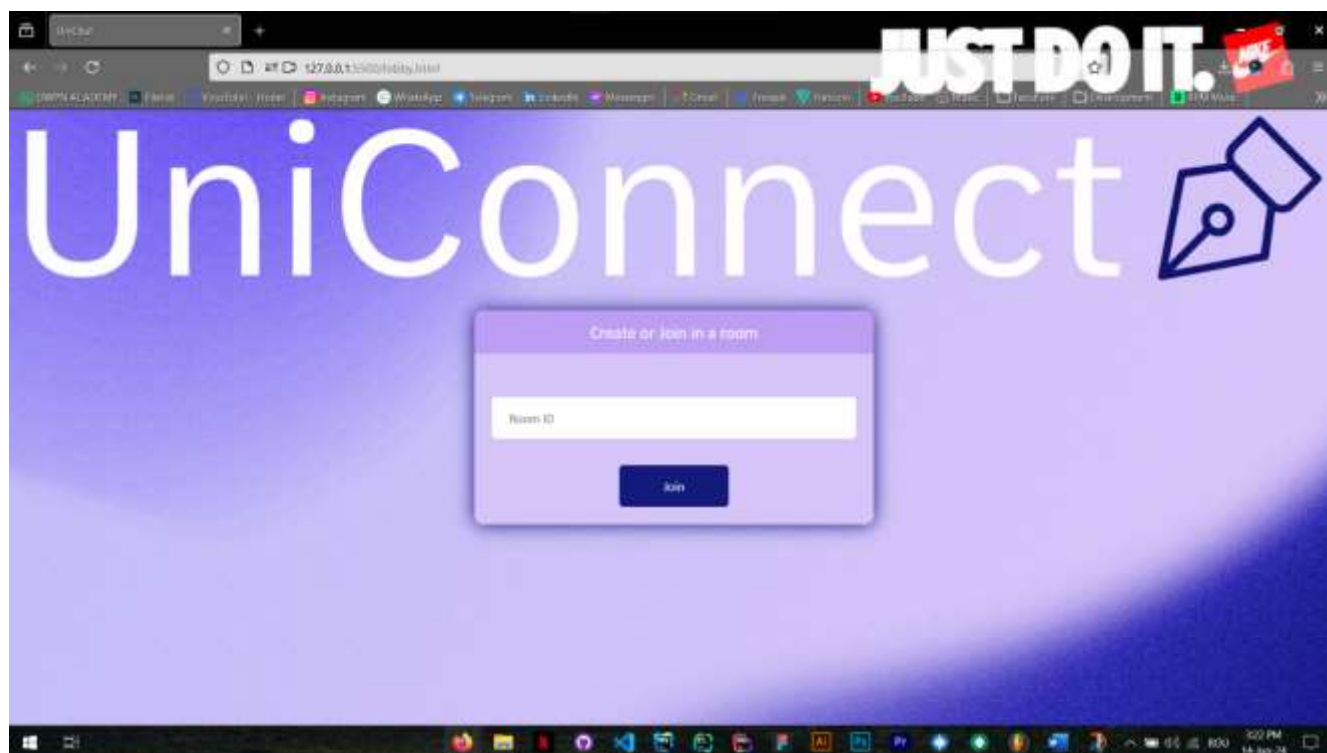
Figură 19 Aplicația UniConnect rulează în navigatorul web Mozilla Firefox

- ✓ Aplicația rulează când dispozitivul este conectat la internet prin cablu



Figură 20 Aplicația UniConnect rulează când conexiunea la internet este făcută prin cablu

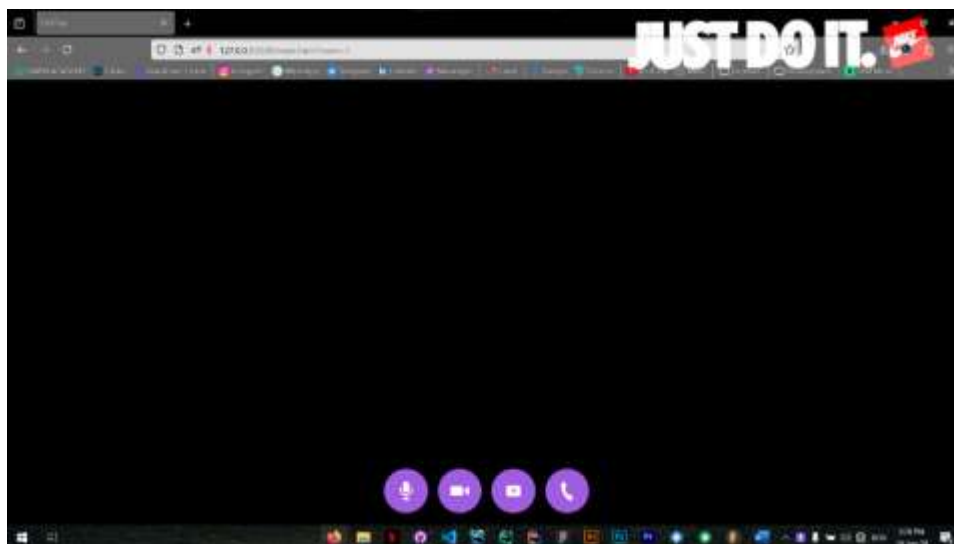
- ✓ Aplicația rulează când dispozitivul este conectat la internet prin wireless



Figură 21 Aplicația UniConnect rulează când conexiunea la internet este făcută prin wireless

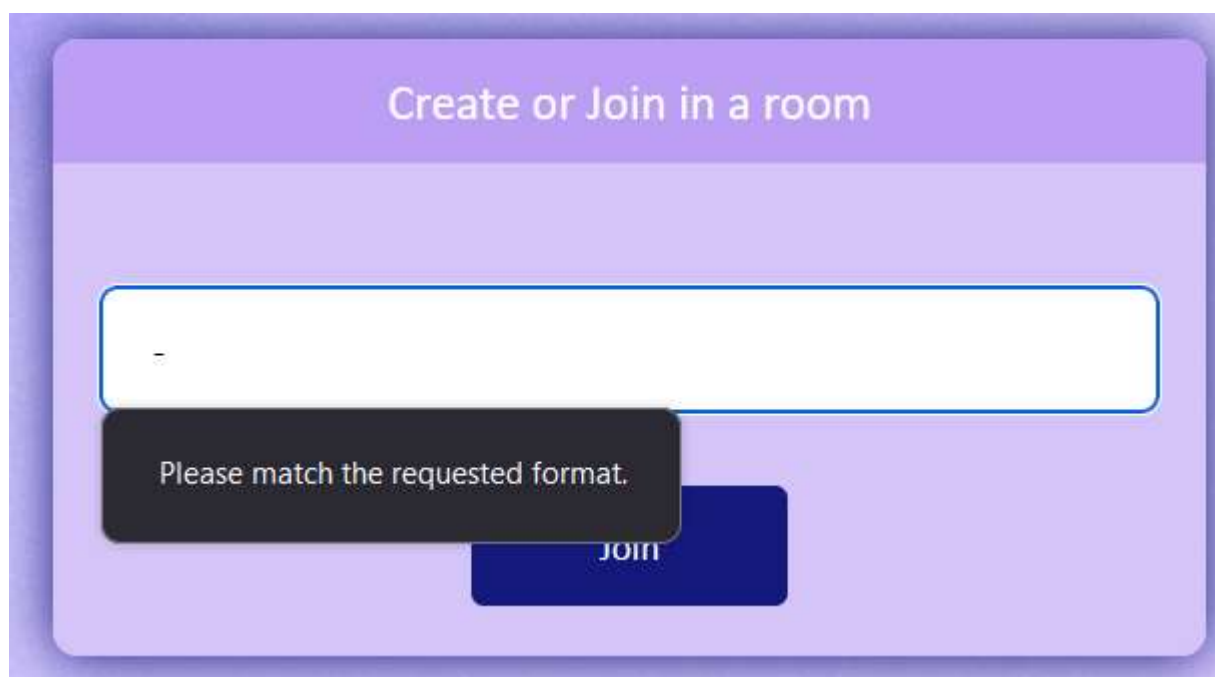
Acum urmează să fie prezentate câteva dintre scenariile negative:

- ✗ Aplicația nu rulează fără a avea o conexiune la internet



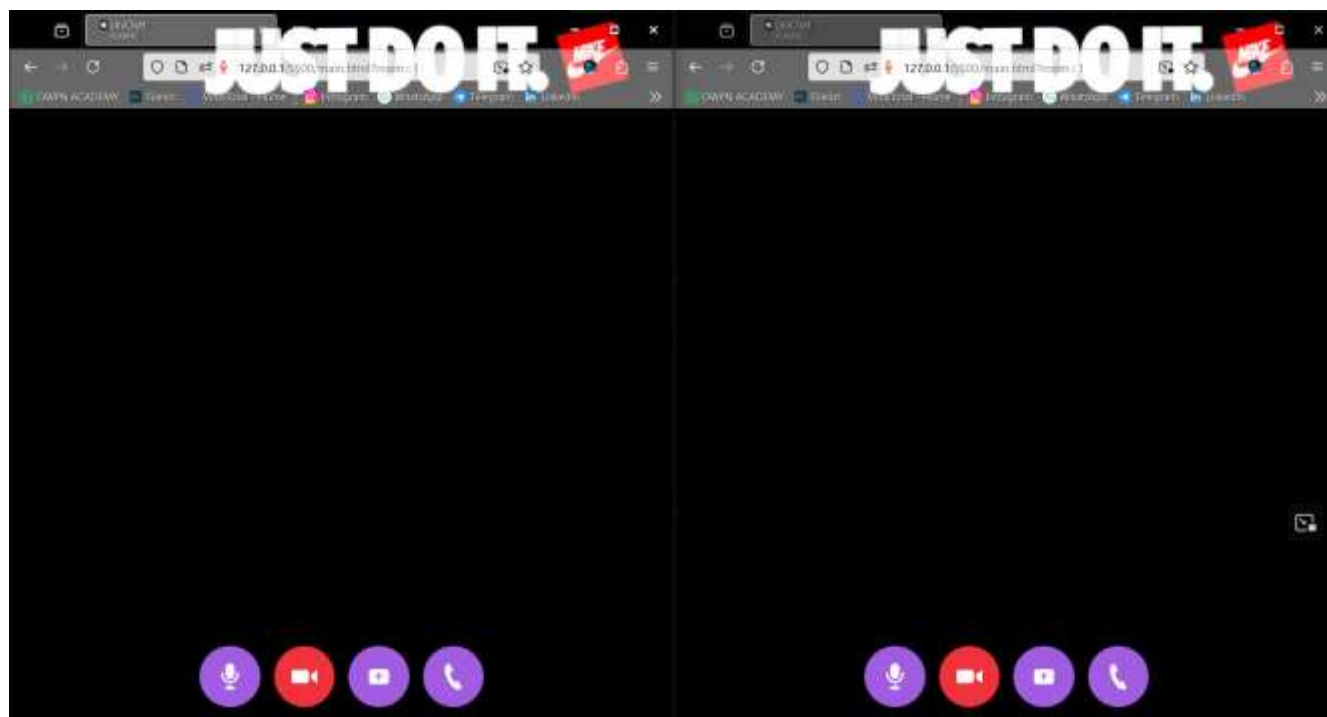
Figură 22 Aplicația UniConnect nu rulează fără conexiune la internet

- ✗ Pentru a putea crea o cameră sau a intra într-una deja creată, utilizatorul trebuie să introducă un ID valid. Un ID valid este acela care este compus doar din cifre. Introducerea unui alt tip de ID duce la avertizarea utilizatorului pentru faptul că ID-ul introdus de el nu este valid.



Figură 23 Orice ID care nu este format din cifre este considerat necorespunzător

- ✖ Uneori sunt prezente anumite bug-uri la nivel de servere iar în momentul în care sunt prezenți mai mulți utilizatori, chiar dacă aceștia sunt în aceeași cameră, aceștia nu se văd din cauza conexiunii.



Figură 24 Bug-uri la nivelul serverelor afectează conexiunea dintre utilizatori

6. CONCLUZII ȘI DIRECȚII DE DEZVOLTARE

6.1 Concluzii

Scopul principal al aplicației este acela de a satisface toate problemele întâlnite în acest domeniu. Ca și obiective inițiale, s-a dorit implementarea minimului de funcționalități necesare pentru acest tip de aplicație pentru a fi sigur că este o aplicație stabilă. Dacă ne gândim doar la faptul că a fost implementat doar numărul minim de funcționalități necesare, putem spune că este o aplicație simplă, dar putem să ne gândim și la faptul că această aplicație a fost dezvoltată de o singură persoană și fără vreun cost, pe când, aplicațiile de acest tip care se află pe piață au fost dezvoltate de o întreagă echipă și cu costuri enorme. Deoarece costul aplicației a fost 0, se poate întâmpla ca la nivelul serverelor să apară bug-uri care pot impacta conexiunea dintre utilizatori.

Pe parcursul implementării aplicației, problemele și-au făcut apariția. Lucrul care a dat cele mai multe bătăi de cap a fost realizarea unei conexiuni stabile între utilizatori, care însă, uneori are bug-uri prezente. Ca o soluție pentru rezolvarea acestor probleme ar fi utilizarea unor servere plătite care cu siguranță sunt stabile.

Pe plan personal, aplicația m-a ajutat să gestionez problemele în situații de stres și să înțeleg ce înseamnă dezvoltarea unei aplicații complexe cu WebRTC de la 0. De altfel, acest lucru care m-a învățat să îmi stabilesc de la bun început direcția în care doresc să mergă aplicația și felul în care să fie implementate funcționalitățile. Pentru a stabili o direcție a aplicației m-am gândit la publicul țintă, acesta fiind format din studenți și elevi.

În final, în procesul de dezvoltare al aplicației s-a dorit ca aplicația să ruleze fără probleme, iar utilizatorii să poată beneficia de aceasta în întregime.

6.2 Direcții de dezvoltare

După etapele de testare și după analiza tuturor problemelor întâmpinate, aplicația poate beneficia pe viitor de multe optimizări și de noi funcționalități.

În primul rând, folosirea unor servere mult mai stabile ar fi un punct forte pentru aplicație. Pentru acest lucru, se poate folosi tehnologia Socket.IO^[26].

În al doilea rând, funcționalități precum un chat prin intermediul căruia utilizatorii să comunice în timp real ar fi foarte util deoarece nu toți participanții beneficiază de un microfon. Pe lângă acest chat, aplicația ar putea folosi un codec mai avansat pentru partea de audio. Codec-ul VP9 este mai avansat decât VP8 care a fost folosit în dezvoltarea aplicației.

În al treilea rând, aplicația ar putea beneficia de implementări precum înregistrarea apelului, tablă pentru scrierea notițelor, opțiunea de a pune pe cineva pe mute când este necesar și opțiunea de a da permisiune în apel doar anumitor persoane.

În al patrulea rând, ar fi utilă dezvoltarea unei aplicații care să fie compatibilă și pe telefoane sau tablete, care să conțină un mediu de execuție.

În ultimul rând, folosirea unei baze de date și crearea unei pagini de autentificare și creare de cont ar fi foarte utilă deoarece utilizatori s-ar conecta cu același cont, iar securitatea ar fi mai ridicată.

7. BIBLIOGRAFIE

- [1] - claudiad (2007) 'videoconferință', MDA2. <https://dexonline.ro/definitie/videoconferin%C8%9B%C4%83/1252506>. [Accesat în 12 iunie 2024]
- [2] - <https://vivo.tech/blogs/news/zoom-vs-google-meet-vs-microsoft-teams-how-to-choose>. [Accesat în 13 iunie 2024]
- [3] - <https://teach.cbs.dk/tools-for-teaching/zoom/>. [Accesat în 13 iunie 2024]
- [4] - <https://its.northeastern.edu/2023/10/19/make-the-switch-today-to-the-new-microsoft-teams/>. [Accesat în 13 iunie 2024]
- [5] - <https://bignet.in/blog/2158/what-is-a-google-meet-in-english>. [Accesat în 13 iunie 2024]
- [6] - <https://pctroubleshooting.ro/topic/16962-ce-sunt-cerintele-de-sistem/>. [Accesat în 13 iunie 2024]
- [7] - <https://arhitectura980.wordpress.com/arhitectura-sistemului/>. [Accesat în 13 iunie 2024]
- [8] - <https://en.wikipedia.org/wiki/STUN>. [Accesat în 13 iunie 2024]
- [9] - https://en.wikipedia.org/wiki/Software_requirements_specification. [Accesat în 13 iunie 2024]
- [10] - <https://www.link-academy.com/curs-introducere-html-css>. [Accesat în 13 iunie 2024]
- [11] - <https://ro.wikipedia.org/wiki/JavaScript>. [Accesat în 13 iunie 2024]
- [12] - https://ro.wikipedia.org/wiki/World_Wide_Web_Consortium. [Accesat în 13 iunie 2024]
- [13] - <https://dev.addinn-group.com/2024/05/20/api-management-a-catalyst-for-innovation-in-the-banking-world/>. [Accesat în 13 iunie 2024]
- [14] - <https://www.google.com/search?client=firefox-b-d&q=ce+este+un+API>. [Accesat în 13 iunie 2024]
- [15] - <https://support.apple.com/ro-ro/guide/shortcuts-mac/apd2e30c9d45/mac>. [Accesat în 13 iunie 2024]
- [16] - <https://www.3cx.com/global/ro/voip-sip-webrtc/ce-este-webrtc/>. [Accesat în 13 iunie 2024]
- [17] - <https://www.3cx.com/global/ro/voip-sip-webrtc/vp8/>. [Accesat în 13 iunie 2024]
- [18] - <https://dyte.io/blog/webrtc/>. [Accesat în 13 iunie 2024]
- [19] - <https://docs.agora.io/en/signaling/overview/product-overview?platform=android>. [Accesat în 13 iunie 2024]
- [20] - <https://eytanmanor.medium.com/an-architectural-overview-for-web-rtc-a-protocol-for-implementing-video-conferencing-e2a914628d0e>. [Accesat în 13 iunie 2024]

[21] - <https://www.wowza.com/blog/webrtc-server-what-it-is-and-why-you-need-one>. [Accesat în 13 iunie 2024]

[22] - <https://www.adjust.com/glossary/sdk/>. [Accesat în 14 iunie 2024]

[23] - <https://docs.agora.io/en/1.x/signaling/get-started/get-started-sdk?platform=android>. [Accesat în 14 iunie 2024]

[24] - <https://medium.com/@mshuecodev/what-should-i-know-about-ice-server-ebed04f54369>. [Accesat în 14 iunie 2024]

[25] - <https://developer.mozilla.org/en-US/docs/Web/API/AudioContext>. [Accesat în 14 iunie 2024]

[26] - <https://devanshupadhyay01.medium.com/implementing-real-time-communication-in-full-stack-applications-8af79fbc6ad4>. [Accesat în 15 iunie 2024]

DECLARAȚIE DE AUTENTICITATE A
LUCRĂRII DE FINALIZARE A STUDIILOR*

Subsemnatul MOCRIS PETRU-LUCIAN
legitimă cu CI seria AR nr. 977289
CNP 503030314395
autorul lucrării UNICONNECT: SISTEM DE VIDEOCONFERINȚĂ

elaborată în vederea susținerii examenului de finalizare a studiilor de
LICENȚĂ organizat de către Facultatea
AUTOMATICĂ ȘI CALCULATOARE din cadrul Universității
Politehnica Timișoara, sesiunea Iunie 2024 a anului universitar
2023-2024, coordonator S.L. DR. ING. VALER BOGAN, luând în
considerare art. 34 din Regulamentul privind organizarea și desfășurarea examenelor de
licență/diplomă și disertație, aprobat prin HS nr. 109/14.05.2020 și cunoscând faptul că în
cazul constatării ulterioare a unor declarații false, voi suporta sancțiunea administrativă
prevăzută de art. 146 din Legea nr. 1/2011 – legea educației naționale și anume anularea
diplomei de studii, declar pe proprie răspundere, că:

- această lucrare este rezultatul propriei activități intelectuale;
- lucrarea nu conține texte, date sau elemente de grafică din alte lucrări sau din alte surse fără ca acestea să nu fie citate, inclusiv situația în care sursa o reprezintă o altă lucrare/alte lucrări ale subsemnatului;
- sursele bibliografice au fost folosite cu respectarea legislației române și a convențiilor internaționale privind drepturile de autor;
- această lucrare nu a mai fost prezentată în fața unei alte comisii de examen/prezentată public/publicată de licență/diplomă/disertație;
- În elaborarea lucrării ~~am utilizat~~ instrumente specifice inteligenței artificiale (IA) și anume _____ (denumirea) _____ (sursa), pe care le-am citat în conținutul lucrării/nu am utilizat instrumente specifice inteligenței artificiale (IA)¹.

Declar că sunt de acord ca lucrarea să fie verificată prin orice modalitate legală pentru confirmarea originalității, consimțind inclusiv la introducerea conținutului său într-o bază de date în acest scop.

Timișoara,

Data

14.06.2024

Semnătura

[Semnătură]

*Declarația se completează de student, se semnează olograf de acesta și se inserează în lucrarea de finalizare a studiilor, la sfârșitul lucrării, ca parte integrantă.

¹ Se va păstra una dintre variante: 1 - s-a utilizat IA și se menționează sursa 2 – nu s-a utilizat IA.