

PROIECT DATM – MOBILE CHATROOM IN ANDROID

1. INTRODUCERE

Acest proiect presupune implementarea unui Chat mobil cu ajutorul mediului de dezvoltare Android si limbajul de programare Java. Aplicatie este alcatuita din doua parti principale:

- clientul mobil Android
- server-ul node JS

Pe parcusul proiectului se vor prezenta tehnologiile folosite pentru realizarea aplicatiei si modul in care aceasta a fost realizata.

2. MEDIUL DE DEZVOLTARE ANDROID

Android OS este un sistem de operare open source bazat pe Linux, dezvoltat de catre compania Google. Este folosit in principiu pentru dispozitive mobile cu touchscreen precum smartphone-uri si tablete.

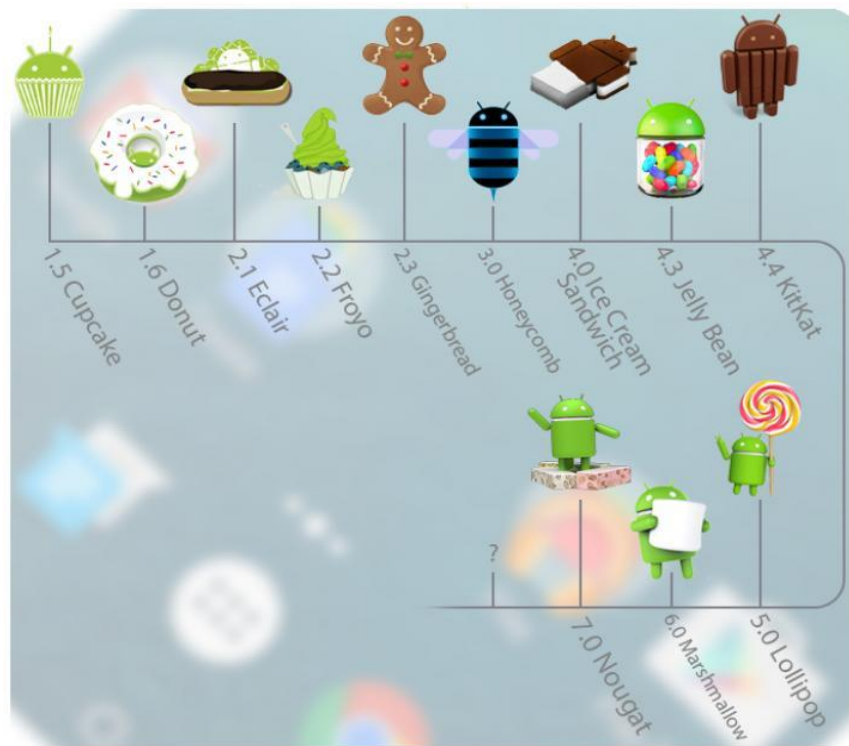


Figure 1 Versiuni Android

Caracteristici ale Android:

- Este open source
- Oricine poate personaliza platforma Android
- Exista o multitudine de aplicatii pe care consumatorii le pot alege
- Oferă suport pentru servicii de mesaje (SMS, MMS), web browser, conectivitate (GSM, CDMA, Bluetooth, Wi-Fi, etc)

Kitul de dezvoltare software Android este oferit de catre Google gratuit și permite developer-ilor sa poata crea aplicatii pentru platforma Android. SDK-ul este format din API-uri Java care ofera dezvoltatorilor de aplicatii acces la functii native pe care dispozitivele Android le suporta, precum: grafica 2D și 3D, codec-uri multimedia, caracteristici de telefonie și servicii de localizare.

Pentru dezvoltarea de aplicatii se poate folosi IDE-ul Android Studio sau Eclipse, unde se adauga plugin-ul de Android. [1]

3. LIMBAJUL DE PROGRAMARE JAVA

Java este un limbaj de programare de nivel înalt pe baza căruia sunt construite o serie de platforme destinate implementării de aplicații pentru toate segmentele industriei software.

Utilizarea acestui limbaj de programare în dezvoltarea proiectului este susținută de următoarele caracteristici:



- **Orientat pe obiecte (OOP–Object Oriented Programming)** – elimină stilul de programare procedural. În OOP programele sunt reprezentate sub forma unor colecții de obiecte care interacționează între ele prin intermediul unor mesaje.
- **Portabilitatea** – este un limbaj independent de platformă ceea ce înseamnă că o aplicație poate rula pe diferite sisteme de operare fără nicio modificare.
- **Neutralitate arhitecturală** – aplicațiile Java nu vor depinde de arhitectura fizică a mașinii pe care vor rula.
- **Modelat după C și C++** - astfel trecerea de la C, C++ la **Java** fiind un pas foarte ușor pentru orice software developer.

4. NODE.JS

Node JS este mediu de dezvoltare open source JavaScript independent de platforma dedicat dezvoltării unor multitudini de tool-uri și aplicații server-side. Deși nu este un framework JavaScript, multe dintre modulele de bază ale acestuia sunt scrise în JavaScript. Aceste module simplifică modul de dezvoltare al aplicațiilor.

Aplicațiile Node.JS sunt scrise în JavaScript și pot fi rulate pe orice mașină Windows, Linux sau OS X.



Caracteristici Node.JS:

- asincron – toate API-urile bibliotecilor Node sunt asincrone, nu blochează aplicația, putând executa mai multe operații simultan
- foarte rapid – este construit pe motorul JavaScript al Google Chrome, fiind foarte rapid în execuția codului
- fără buffering – aplicațiile Node.JS nu fac buffer datelor ci trimite datele la ieșire pe bucăți [2]

5. SOCKET.IO

Socket.IO este o bibliotecă JavaScript destinată aplicațiilor în timp real. Oferă comunicare bi-direcțională și în timp real între clienți și server. Este compusă din două părți:

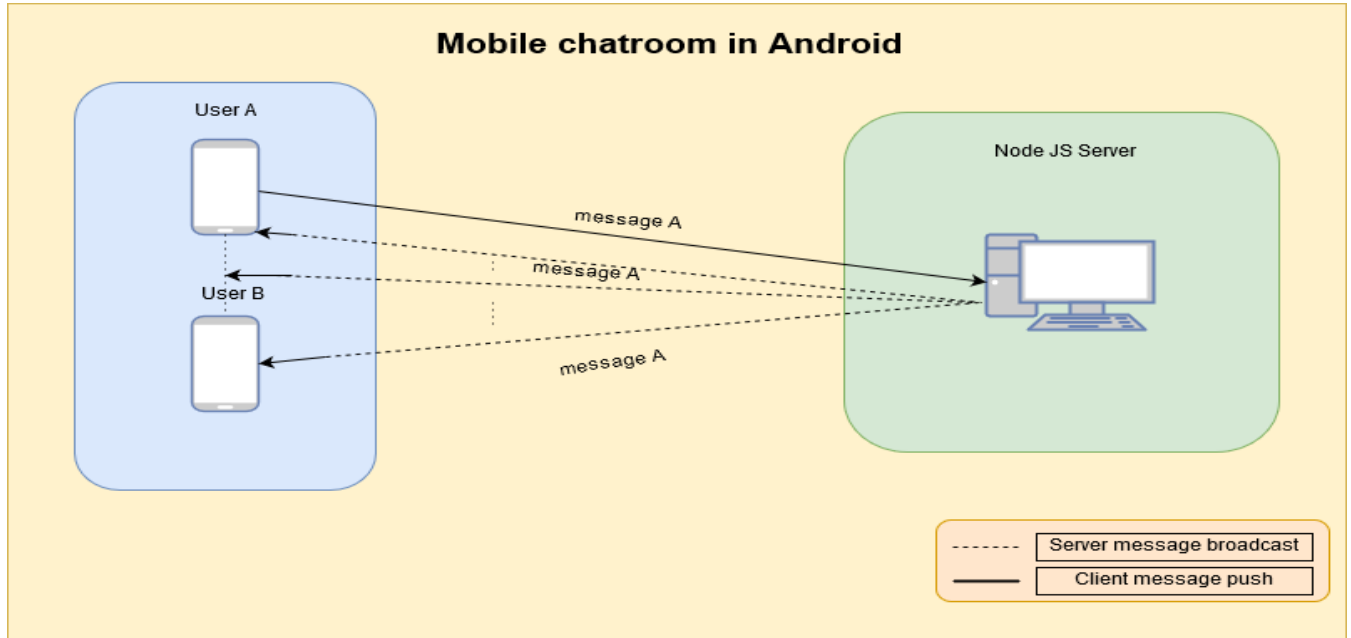
- bibliotecă client – care rulează pe aplicația client, în cazul nostru Android
- bibliotecă server – bibliotecă care trebuie instalată pe un server Node.JS

Socket.IO folosește protocolul de WebSocket, aceasta permițând broadcast-ul către clienți. [3]



6. DEZVOLTAREA APLICATIEI

Aplicatia este realizata din doua parti, aplicatia client, care este aplicatia mobila in sine si aplicatia server:



Scenariul de functionare este urmatorul:

- utilizatorul A se conecteaza la serverul Node.JS si trimite un mesaj catre acesta
- serverul receptioneaza mesajul utilizatorului A
- server-ul face broadcast mesajului catre toti utilizatorii conectati
- mesajul este afisat in parte, pentru fiecare utilizator conectat

Figure 2 Arhitectura aplicatie

6.1 CONFIGURARE SERVER-ULUI

Aplicatie server a fost realizata cu ajutorului mediului de dezvoltare Node.JS si librariiei Socket.IO. Pentru obtinerea unui server functional, urmasorii pasi au fost urmati:

- instalare Node.JS de pe [site-ul oficial](#)
- creare unei aplicatii NodeJS din linia de comanda utilizant comanda: **npm init**
- instalarea modului express: **npm install express --save**
- instalarea modului socket.io: **npm install socket.io --save**

Intr-un final, dupa parcurgerea pasilor specificati vom obtine server-ul aplicatiei, care v-a asculta mesajele trimise de catre utilizatori pe portul 3000 al retelei locale:

```
E:\>cd Programming\socketServer  
E:\Programming\socketServer>node index.js  
server listening on port 3000
```

Figure 3 Pornire server

```
var app = require('express')();  
var http = require('http').Server(app);  
var io = require('socket.io')(http);  
app.get('/',function(req,res){  
    res.sendFile(__dirname + '/index.html');  
})  
  
io.on('connection',function(socket)  
{  
    console.log('user connected: ' + socket.id);  
  
    socket.on('message',function(data){  
  
        io.emit('message',data);  
  
    })  
  
    socket.on('disconnect', function(){  
        console.log( socket.name + ' has disconnected from the chat.' +  
socket.id);  
    });  
}  
)  
  
http.listen(3000,function(){  
    console.log('server listening on port 3000');  
})
```

Figure 4 Socket server code

6.2 APLICATIA MOBILA

Aplicatia este alcatuita din doua ecrane:

- Ecranul de start, unde are loc setarea numele utilizatorului si conectare catre server-ul aplicatiei
- Ecranul de chatroom, unde, dupa conectare utilizatorul va putea face schimb de mesaje cu alti utilizatori conectati la server

```
E:\Programming\socketServer>node index.js  
server listening on port 3000  
one user connected: 4MjF6CFnIwdCNTfEAAAA
```

Figure 5 Informare server client conectat

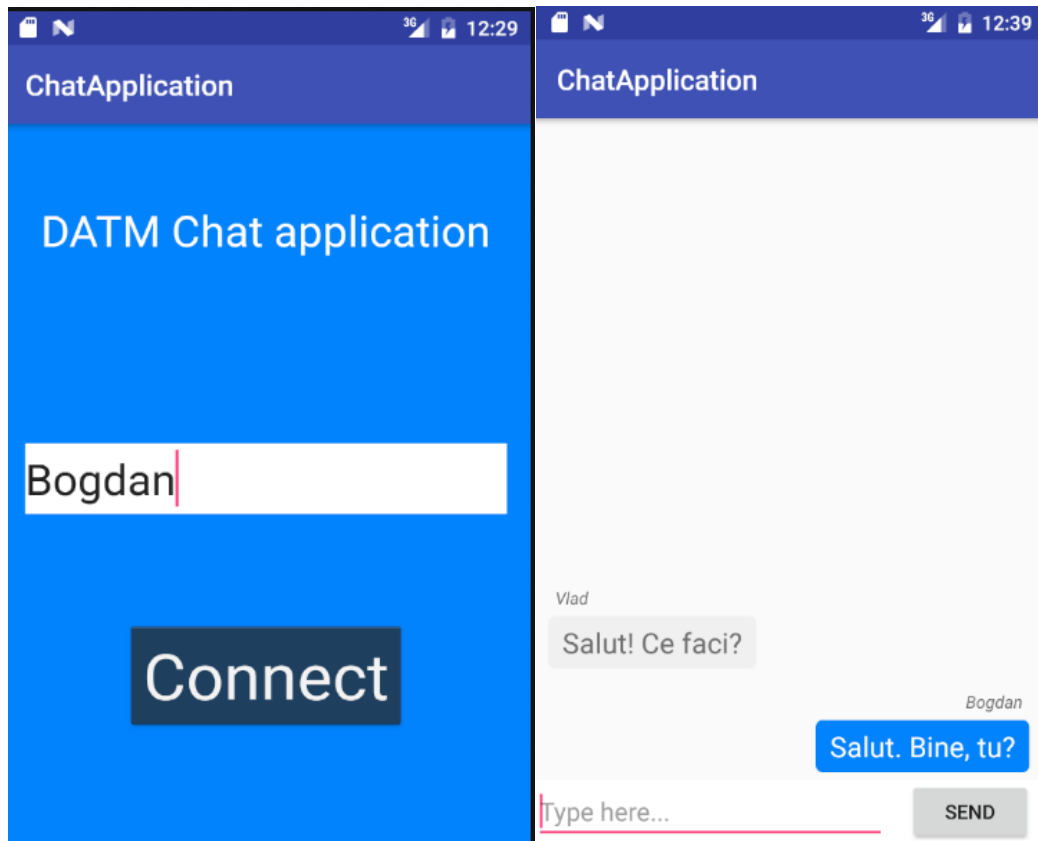


Figure 6 Ecran logare si ekran chat room

7. CONCLUZII

In acesta lucrare au fost descrise tehnologiile necesare implementarii proiectului cu tema Mobile chatroom in Android. Aplicatia a fost dezvoltata cu ajutorul mediului de dezvoltare Android si limbajul de programare Java pe parte de client, iar partea de server a fost realizata cu ajutorul mediului de dezvoltare Node.JS si JavaScript.

Aplicatia ofera un chat room in timp real pentru utilizatorii conectati la server, avand un scenariu de functionare simplu: Un client se conecteaza la server, trimite mesajul dorit care acesta, iar in final mesajul este emis catre toti utilizatorii conectati la server, prin broadcast.

Pe parcursul implementarii problemele intampinate au fost in zona:

- configurarii server-ului Node.JS
- layout-ul chat room-ului, care decide care mesaj este propriu si care este de la alti utilizatori

BIBILIGRAFIE

- [1] Android SDK http://www.webopedia.com/TERM/A/Android_SDK.htm
- [2] Node.JS https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm
- [3] Socket.IO <https://en.wikipedia.org/wiki/Socket.IO>