|  |  |  |
| --- | --- | --- |
| Facultatea de Automatica și Calculatoare |  | Sisteme Distribuite și Tehnologii Web |

Topichat

Aplicatie Mobilă crossplatform

|  |
| --- |
| Coordonator științific  Prof.dr.inf. Mitică CRAUS |
| **Absolvent**  Ciprian Ionut Holbia |

Cuprins

[Introducere 2](#_Toc485203796)

[Protocolul MQTT 3](#_Toc485203797)

[Istoria protocolului 3](#_Toc485203798)

[Cum functionează 3](#_Toc485203799)

[Metoda de comunicare publish/subscribe 3](#_Toc485203800)

[MQTT Client 5](#_Toc485203801)

[MQTT Broker 5](#_Toc485203802)

[Implementarea mesajelor MQTT 5](#_Toc485203803)

[Comunicarea unul la unul 5](#_Toc485203804)

[Comunicarea într-un grup 6](#_Toc485203805)

[Interfață aplicație 7](#_Toc485203806)

[Conversație nouă 8](#_Toc485203807)

[Pagina de conversații 10](#_Toc485203808)

[Codul sursă 10](#_Toc485203809)

# Introducere

Topichat este o aplicație de chat pentru telefoane mobile bazată pe topicuri. Toate conversațiile utilizatorului vor fi grupate pe topicuri în aşa fel încât găsirea unei anumite discuții sa fie foarte simplă.

Am ales acesta tema deoarece am considerat ca un astfel de aplicație mobilă este foarte folositor utilizatorilor și deocamdată nu există ceva asemănător.

Aplicația va folosi un protocol de comunicare numit MQTT foarte performant si rapid, folosit foarte mult în Internet Of Things (IoT). Tehnologia folosita în implementarea aplicației va fi Xamarin (C#). Folosind Xamarin putem avea același cod C# pentru toate platformele.

Serverul de MQTT folosit se numește EMQ, o implementare open source. Cunoscut sub numele de MQTT Broker, serverul va fi instalat pe o instanța AWS (Amazon Web Services). Acea mașina virtuala aflata în AWS va folosi Ubuntu ca și sistem de operare.

Pentru moment nu vom folosi un Backend pentru ca utilizatori vor putea sa folosească aplicația fără a fi nevoiți sa își creeze un cont. Ca și identificator pentru utilizatori vom folosi numărul de telefon, similar cu abordarea aplicației Whatsapp.

Conversațiile utilizatorilor vor fi stocate local, pe device si nu undeva in Cloud. Acesta abordare este mult mai benefică utilizatorului deoarece nu apar probleme de privacy si securitate.

Comunicarea cu Broker-ul MQTT va fi securizata folosind SSL/TLS.

# Protocolul MQTT

MQTT este un protocol de comunicate client/server de tipul publish/subscribe. Acest protocol a fost conceput pentru a fi foarte eficient, rapid și ușor de implementat. Aceste caracteristici îl fac ideal pentru a fi folosit în IoT (Internet of things) datorita dimensiunii reduse a mesajului.

## Istoria protocolului

MQTT a fost inventat de Andy Stanford-Clark (IBM) și Arlen Nipper (Arcom, acum Cirrus Link) în 1999. Atunci, idea a fost de a crea un protocol de comunicare care folosește un minim de baterie și care are o dimensiune redusă pentru conectarea conductelor de petrol prin conexiune satelit. Specificațiile proiectului erau:

* Simplu de implementat
* Verificarea datelor trimise
* Eficient
* Protocol real-time

Aceste obiective sunt în continuare nucleul MQTT, în timp ce accentul sa schimbat de la sistemele embedded sper utilizarea în IoT.

## Cum functionează

MQTT folosește o structură de comunicare de tip publish/subscribe.

Serverul MQTT cunoscut sub numele de MQTT Broker este o aplicație server foarte eficienta ce poate fi instalată pe aproape orice sistem de operare (Mac, Linux, Windows). Fiind o aplicație mica este ideală pentru device-uri embedded sau device-uri cu resurse mai puține gen Raspberry Pi.

Clienții poate fi orice dispozitiv care implementează comunicarea de tipul publish/subscribe.

### Metoda de comunicare publish/subscribe

Metoda de comunicare folosita de protocolul MQTT este o alternativă a metodei clasice client-server unde un client comunica direct cu un endpoint. Folosind MQTT, clienții sunt decuplați unul de celălalt. Asta înseamnă că un client care trimite un mesaj (publish) nu va ști destinatarul mesajului care poate fi unul sau mai mulți clienți care fac subscribe la acel tip de mesaj.



În diagrama de mai sus putem vedea trei clienți conectați la Broker. Un senzor de temperatură, un telefon mobil și un laptop. Telefonul mobil și laptopul subscriu la mesaje de tip "temperatură". Atunci când intervine o schimbare de temperatură, senzorul de temperatură va publica un mesaj de tip "temperatură" care va fi redirecționat de Broker către toți clienții, în cazul nostru telefonul mobil și laptopul.

## MQTT Client

Un client MQTT este orice dispozitiv de la un microcontroler până la un server, care folosește o librărie MQTT și se conectează la un broker MQTT pe orice tip de rețea. Acesta ar putea fi un dispozitiv foarte mic și limitat în resurse, care este conectat printr-o rețea WiFi și care folosește o libărie MQTT optimizata la maxim sau un computer tipic care rulează un client grafic. Practic orice dispozitiv care suporta TCP/IP și implementează comunicarea MQTT.

Librării de clienți MQTT sunt disponibile pentru o mare varietate de limbaje de programare, de exemplu Android, Arduino, C, C ++, C #, Go, iOS, Java, JavaScript, .NET. O listă completă poate fi găsită pe wiki-ul MQTT.

## MQTT Broker

Serverul numit și MQTT Broker, este configurat pe o mașina virtuala aflata în Amazon Web Services. Brokerul pe care ales se numeşte EMQ si este o implementare open source.

EMQ MQTT Broker:

http://emqtt.io/

# Implementarea mesajelor MQTT

Autentificarea clienţilor va fi bazată pe numărul de telefon. Mesajele vor fi trimise pe portul 8883, SSH end to end encryption va fi activat.

## Comunicarea unul la unul

message/[ReceiverId]/[SenderID]/[TopicName]

Folosind acest model fiecare client va face subscribe la **message/[MyId]. În cazul unui utilizator cu numărul de telefon 0744360800, acesta va face subscribe la message/0040744360800/#. Un expeditor cu numarul de telefon 0040740660810 va face publish la:**

**message/0040744360800/0040740660810/VacantaLaMunte**

**atunci când vrea sa comunice cu 0744360800 despre topicul „Vacanta la munte”.**

## Comunicarea într-un grup

message/[ReceiverId]/[GroupId]/[TopicName]/[MembersIDs]/[SenderID]

Administratorul grupului va asigna automat un ID unui grup nou creat. Acest ID va fi un GUID unic. Acest grup ID va fi folositor atunci când un utilizator vrea sa facă unsubscribe la un anumit grup. De asemenea pentru că avem și numele topicului în structura mesajului un anumit utilizator poate sa facă unsubscribe pe un topic, dar rămâne în grup pentru celelalte topicuri sau pentru alte topicuri noi.

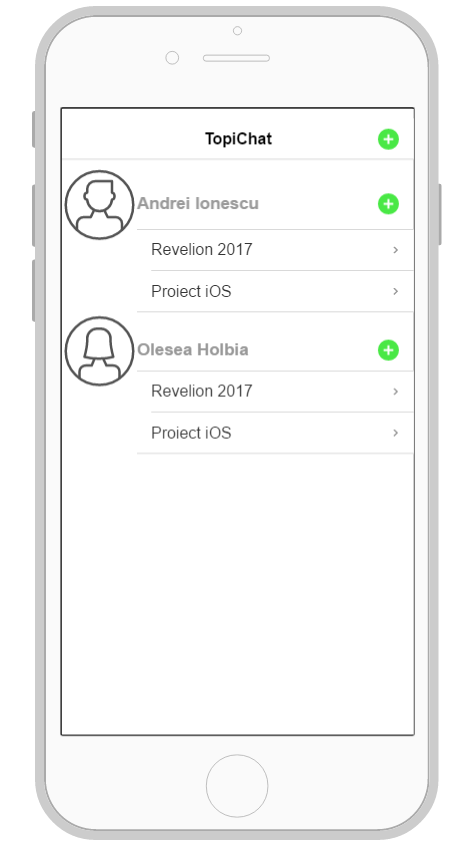
De exemplu într-un grup de trei persoane, când un utilizator cu numărul de telefon 0744360800 vrea sa trimită un mesaj nou, el va genera trei mesaje pentru fiecare destinatar.

message/0040744555555/d428a487-9932-471c-b701-74aef7889b4e/Revelion/0040740000000-0040740111111-0040740222222/0040740000000  
  
message/0040740111111/d428a487-9932-471c-b701-74aef7889b4e/Revelion/0040740000000-0040744555555-0040740222222/0040740000000  
  
message/0040740222222/d428a487-9932-471c-b701-74aef7889b4e/Revelion/0040740000000-0040740111111-0040744555555/0040740000000

Folosind aceasta structura un client va şti din structura mesajului MQTT ca el comunica cu un grup şi câte persoane sunt în acel grup.

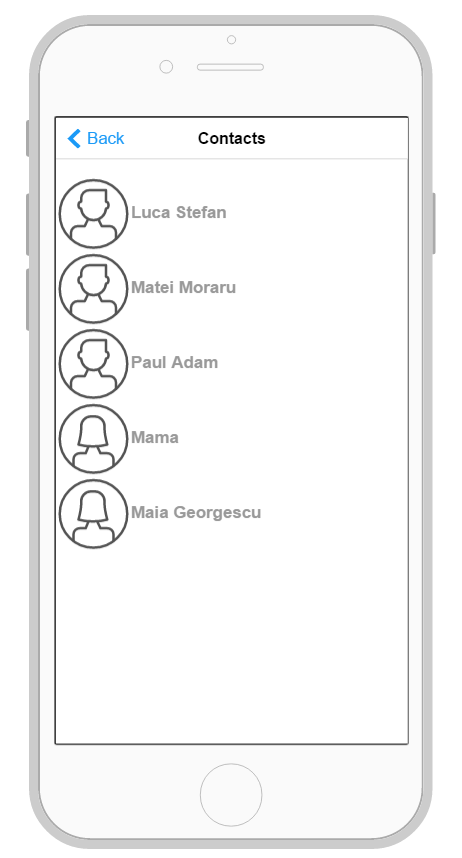
# Interfață aplicație

Pagina principală va conține o lista cu toate conversațiile user-ului. Toate conversațiile vor fi grupate pe topicuri.



## Conversație nouă

Atunci când userul va dori sa înceapă o noua conversație cu un nou contact va apăsa pe plusul din header și va fi trimis în lista de contacte.

După ce va selecta contactul va fi imediat redirecționat la pagina de conversație și numele topicului va fi automat generat (Default topic). Mai apoi user-ul poate schimba numele topicului pentru o organizare buna a conversațiilor.

## Pagina de conversații

Pagina de conversații va fi una simplă și va conține ca titlu numele topicului și restul paginii va fi folosit pentru conversația efectivă.

Codul sursă  
  
Codul sursa se gaseste pe Github:  
https://github.com/ionutholbia/topichat  
