



Smart and secure chat solutions

Arhitectura sistemului

Nume proiect	ZKYBE
Nume document	Arhitectura sistemului / Architectural Design Document
Autori	Toma Alexandru-Ionuț, Cornea Radu-Valentin, Lungu Stefania-Paraschiva, Păuleț Faustina, Bordea Bianca-Andreea, Iorga Elena
Reviewed by	-
Status	În lucru

Versiuni

Data	Versiune	Modificări
25.10.2022	0.1	Începerea documentului
03.11.2022	0.2	Completarea documentului cu descrierile componentelor

1. Introducere

1.1 Scopul documentului

Documentul “Arhitectura Sistemului” are ca scop principal descrierea arhitecturii sistemului și a modului în care a fost gândită să funcționeze, în ceea ce privește aplicația pentru socializat Zkybe.

De aceea, pe parcursul documentului, vor apărea diagrame ce cuprind arhitectura sistemului.

Acest document este necesar în vederea dezvoltării arhitecturii sistemului.

1.2 Scurtă descriere a proiectului

Proiectul Zkybe este o soluție web și mobile prin care utilizatorii pot comunica cu ușurință, își pot crea un profil minimal, pot crea grupuri, totul într-o soluție “user-friendly” și securizată.

1.3 Definiții și abrevieri

- Tag = eticheta care descrie minimal preferințe ale utilizatorului/grupului
- Scrum = metodologia folosită în dezvoltarea proiectului
- UAC = user access control (folosit la controlul utilizatorilor)
- API = application programming interface, folosit pentru ca aplicațiile să comunice între ele
- SQL = structured query language (limbajul de programare folosit pentru bazele de date)
- ASP.NET = framework folosit pentru încărcarea dinamică a paginilor web
- DB = Database
- HTTP = Hypertext Transfer Protocol (protocol la nivel de aplicație folosit să încarce pagini web prin hypertext)
- JWT = JSON Web Tokens (este un standard folosit să descrie un mod compact de a transmite informații sub formă de JSON într-o manieră sigură)

2. Prezentarea sistemului

Principalele componente sunt: Server API, Server UAC, Pagina Web și Aplicația Mobile.

Serverul API: strâns legat de baza de date principală (SQL MariaDB / noSQL MongoDB) și constă într-o aplicație Spring Boot în Java.

Serverul UAC: strâns legat de baza de date UAC (MS SQL Server) și constă într-o aplicație ASP.NET care folosește IdentityServer4 pentru gestionarea utilizatorilor și drepturilor acestora.

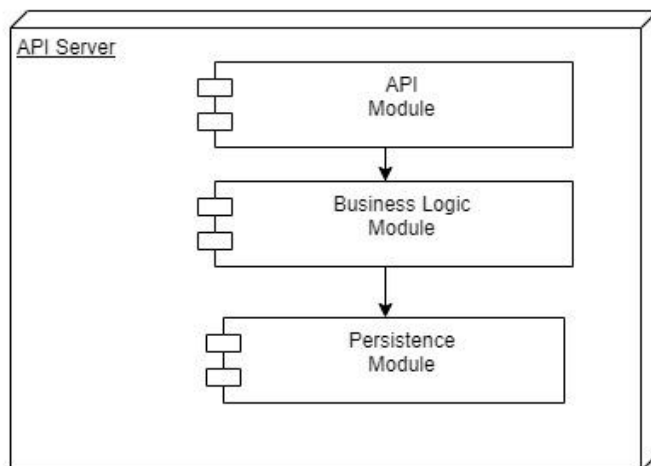
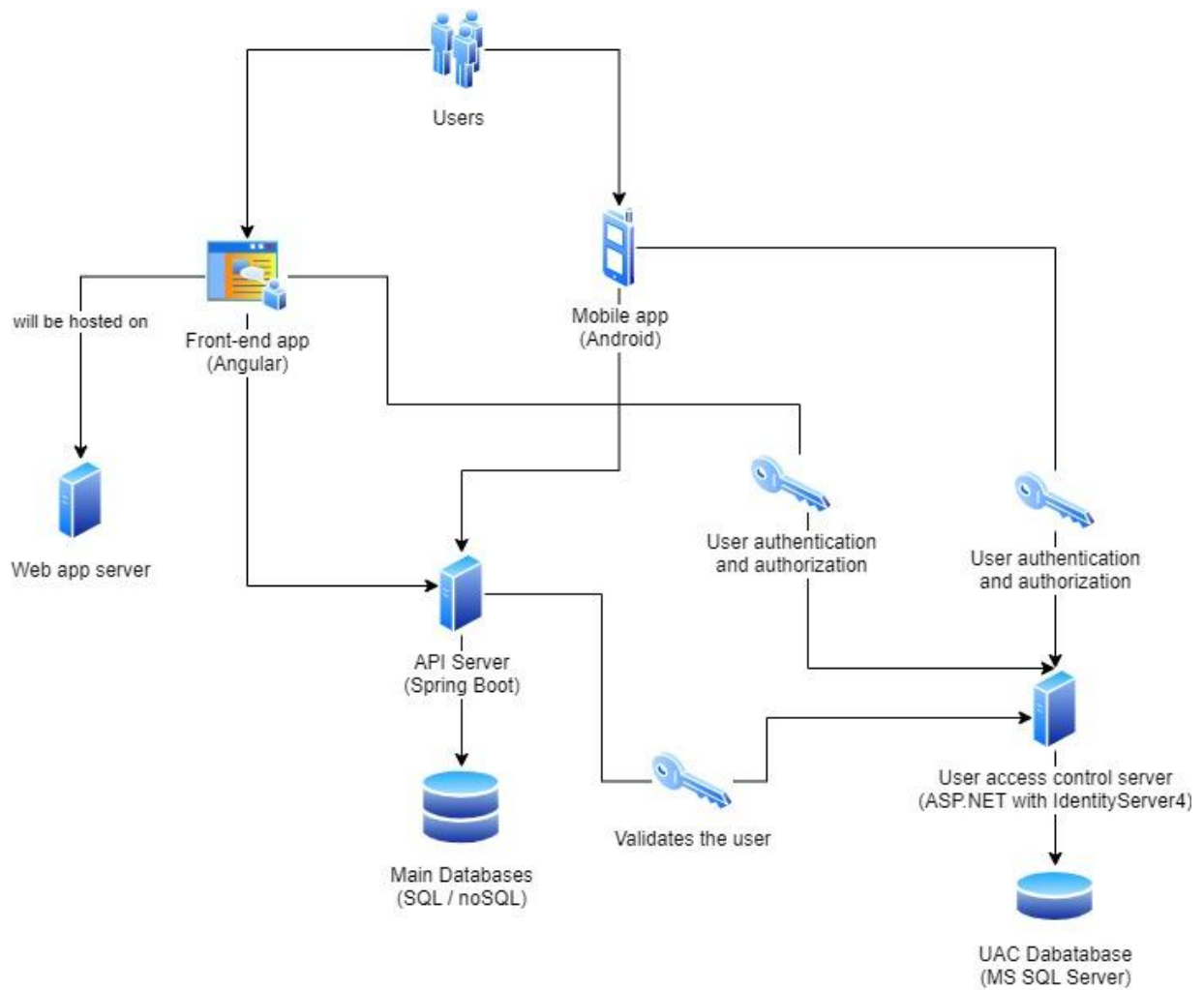
Pagina Web: hostata pe un server simplu care răspunde doar cu resursele html, css, js și consta într-o aplicație web folosind framework-ul Angular care comunică cu Serverul API prin apeluri HTTP.

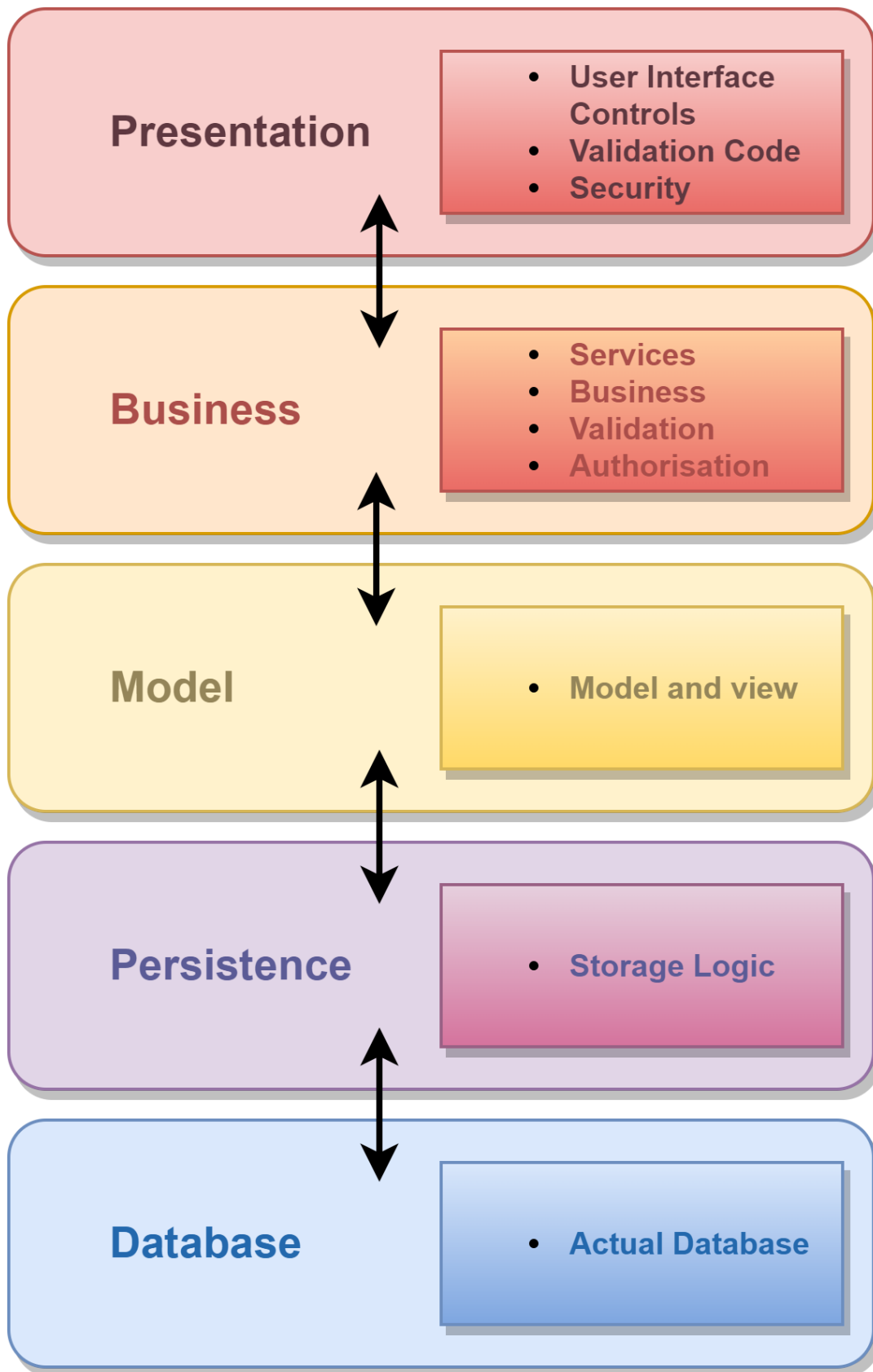
Aplicația Mobile: constă dintr-o aplicație Android care comunică cu Serverul API prin apeluri HTTP.

3. Arhitectura top-level

3.1 Identificarea componentelor

- Server DB SQL
- Proiect Angular
- Proiect Mobile
- Server simplu pentru Frontend
- Server pentru Backend (API)
- Server User Access Control





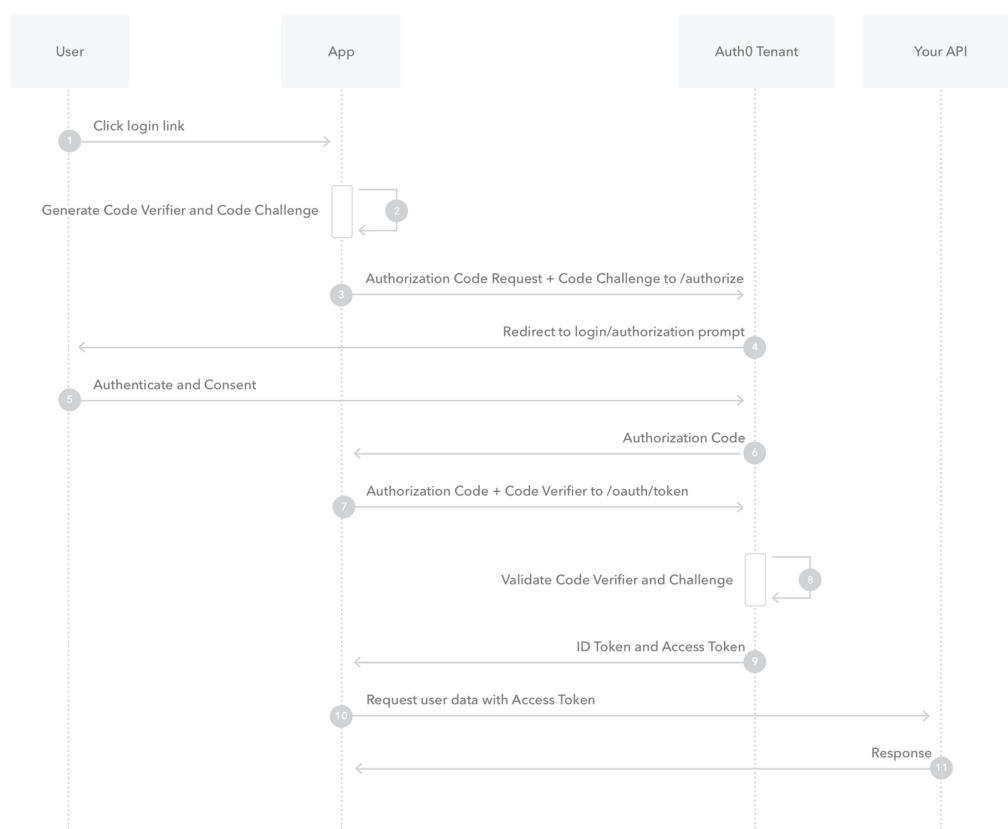
3.2 Relațiile și comunicarea dintre componente

Aplicația Web comunică cu Serverul UAC pentru a facilita înregistrarea și autentificarea user-ului, care primește token-ul (JWT) cu care poate face apeluri la serverul API.

Pe baza token-ului se face autorizarea la anumite resurse de pe Serverul API (claim-uri pentru a determina rolul din grupuri).

Serverul API comunică cu Serverul UAC pentru a valida token-uri prin cereri HTTP.

Folosim un standard OAuth2.0 pentru autentificare numit Authorization Code Flow with Proof Key for Code Exchange (PKCE)

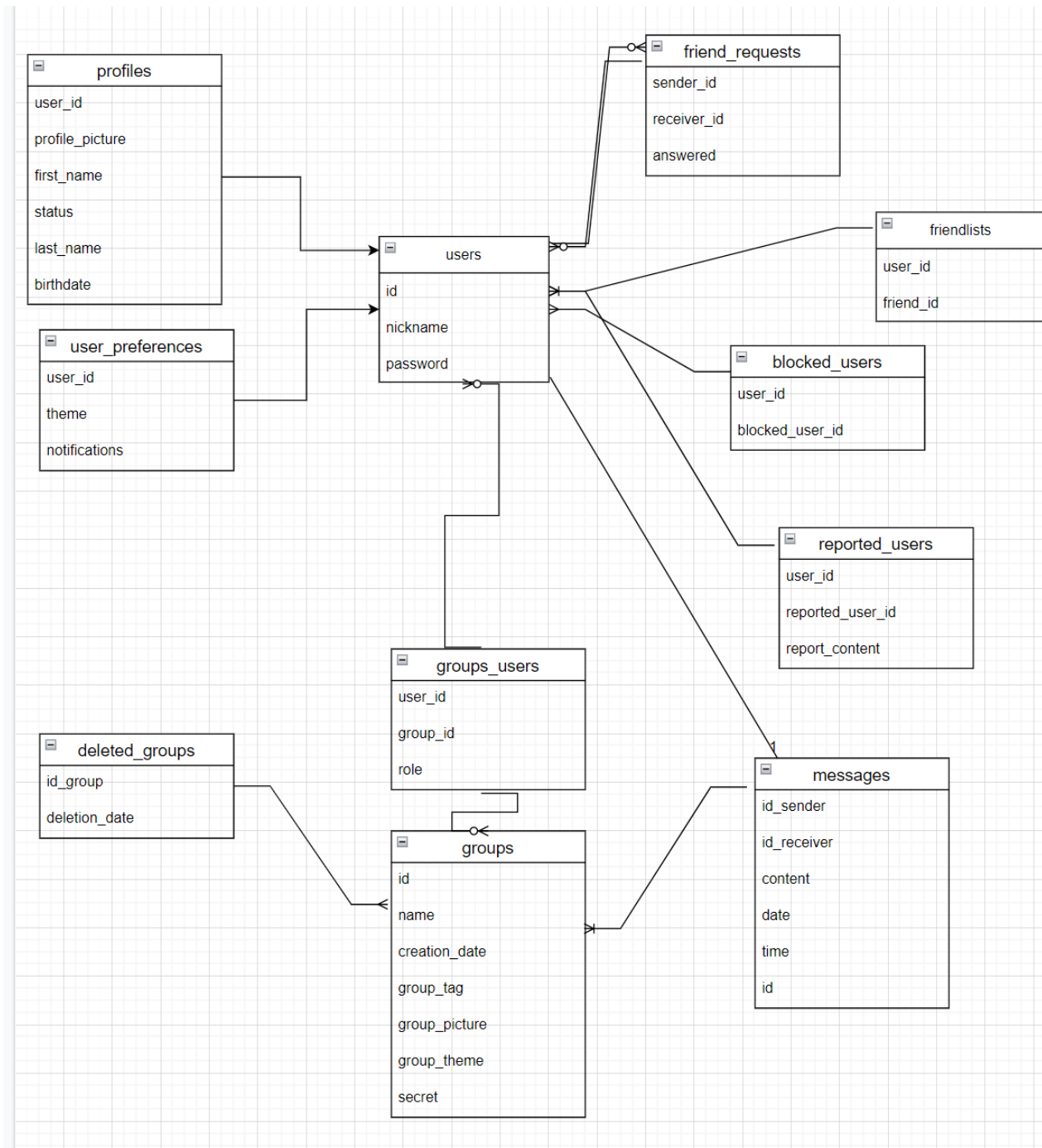


4. Descrierea componentelor

Aplicația WEB este un program care este stocat pe serverul web și este accesat prin intermediul unui browser. Astfel, clientul poate interacționa cu interfața (partea de front-end care se va ocupa de culori, așezare în pagină, practic tot ce va vedea utilizatorul) și folosi aplicația fără a ști ce este în spatele ei.

API server-ul implementat cu Spring Boot va avea conexiunea cu baza de date centrală (unde vor fi stocate informațiile despre conversații, grupuri, utilizatori etc), având totodată și funcțiile de procesare și

accesare ale acestor date. La acesta se vor face apeluri API de către aplicația web (și de cea mobile) pentru a solicita datele de care are nevoie.



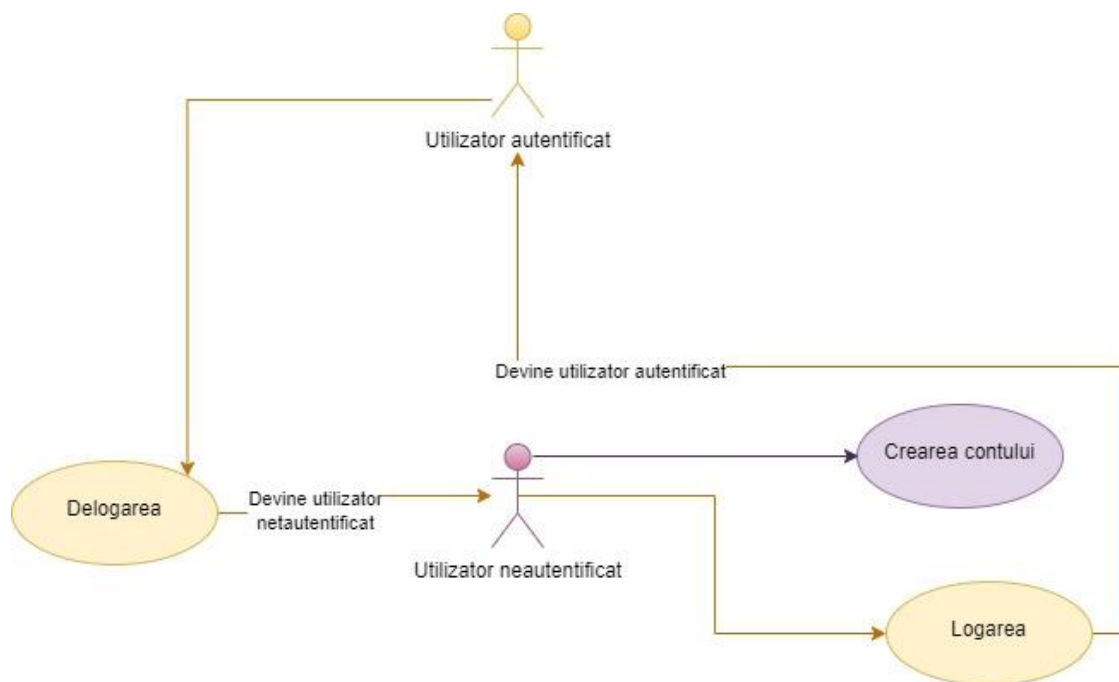
Acest server va fi format dintr-un modul API, un modul de logică de business și unul de persistență. Cel din urmă este cel care se va ocupa de stocarea datelor astfel încât ele să rămână în sistem. Cel de logică este responsabil pentru procesările care au loc asupra datelor și pentru manipularea lor. Modulul de API permite primirea și trimiterea de cereri de la o instanță la alta, fiind practic cel care face comunicarea cu mediul exterior.

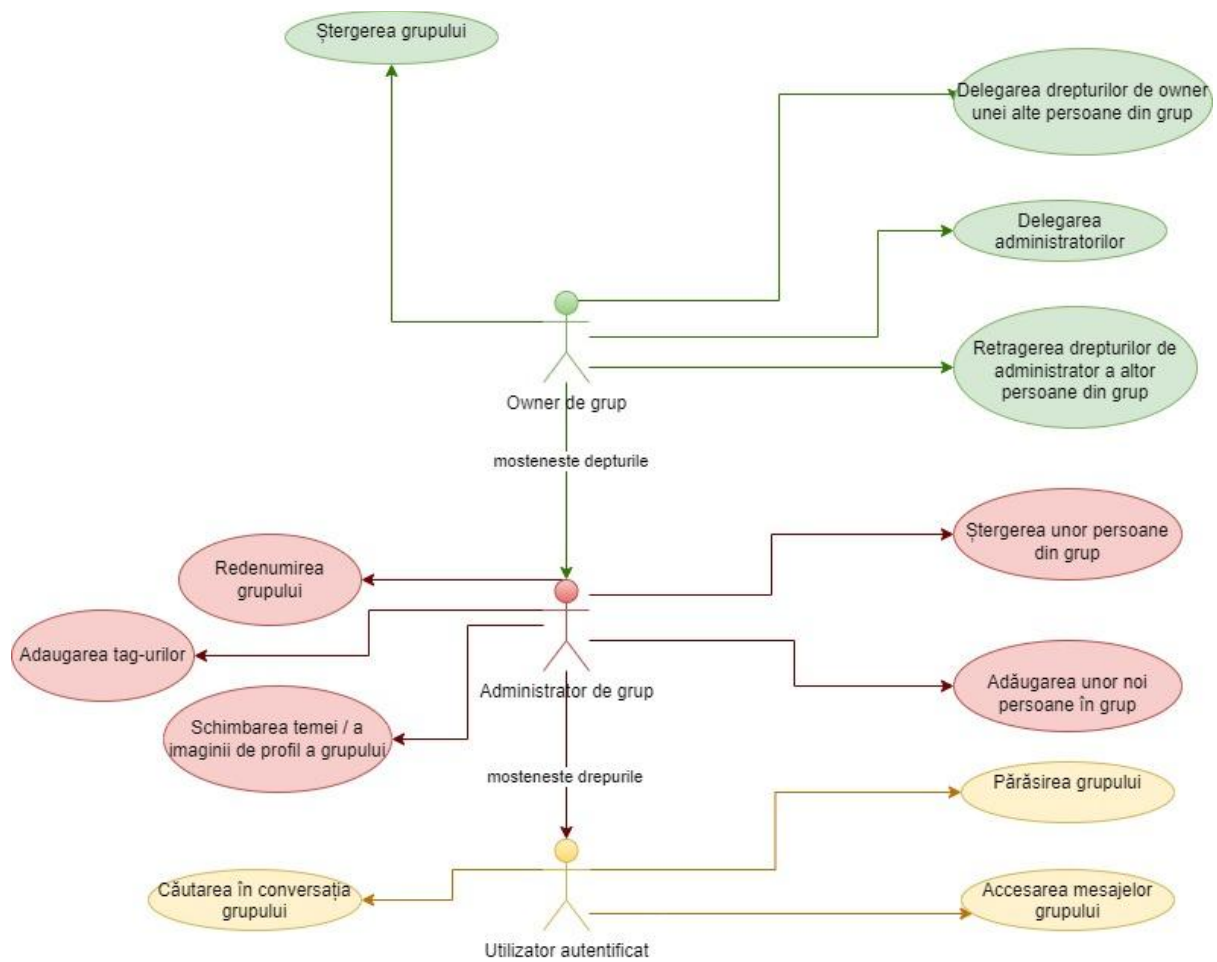
Aplicația mobile va fi separată de aplicația web, fiind o extensie pentru dispozitive Android a programului nostru care va comunica cu API server-ul în același fel ca aplicația web pentru a extrage datele de care are nevoie.

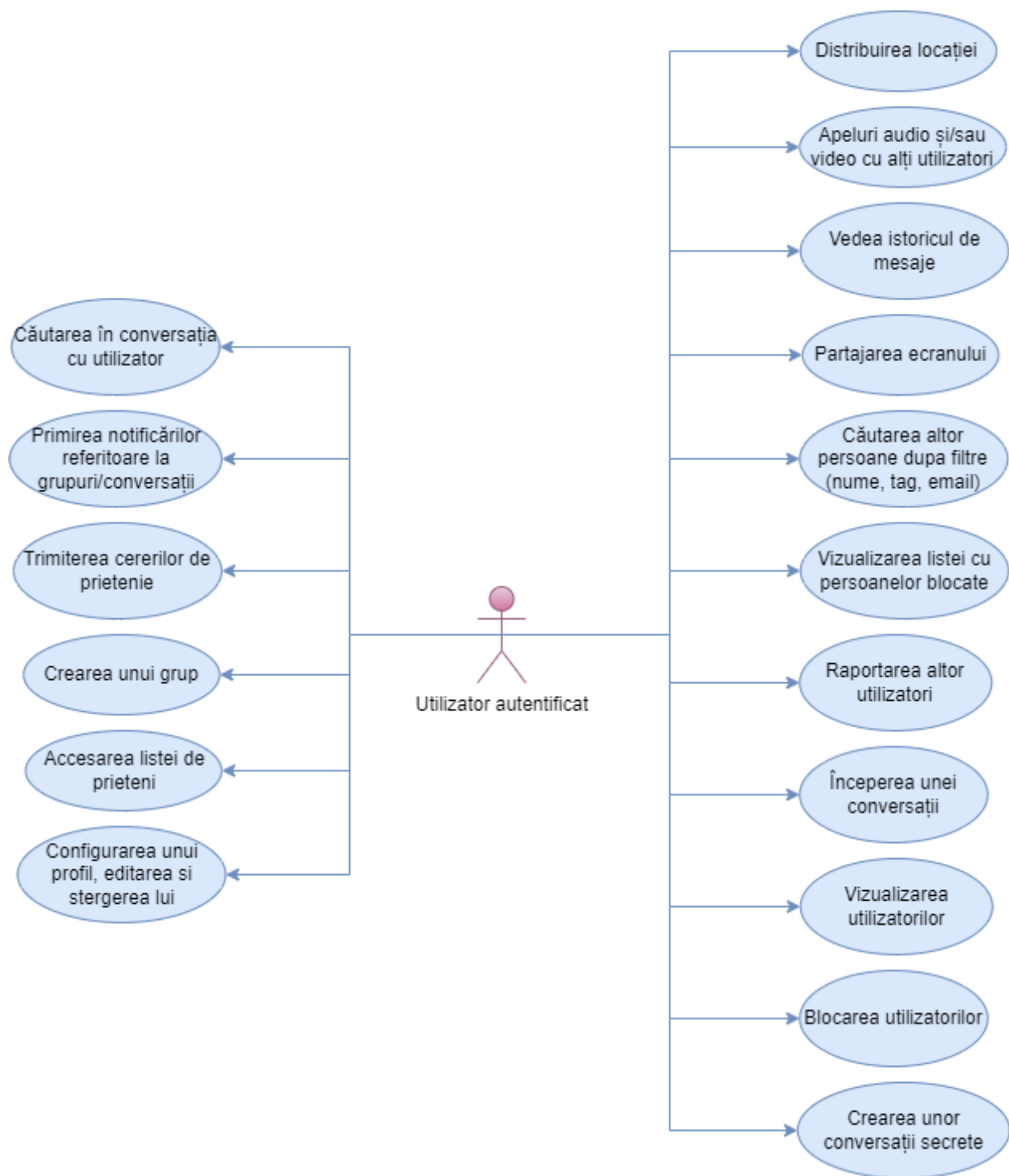
Serverul UAC va fi folosit la înregistrarea și autentificarea utilizatorului. Pentru aceasta avem nevoie de o baza de date UAC care va conține datele și tipul fiecărui client. În funcție de acestea, se va trimite un token (JWT) către aplicația web pentru ca aceasta să știe ce tipuri de acțiuni sunt permise. Pentru a valida request-urile trimise către API server, se va folosi același mecanism în care se verifică token-ul. Pentru autentificarea utilizatorilor se va folosi flow-ul de autentificare și autorizare bazat pe standardul OAuth 2.0: authorization code flow with PKCE.

Modul de autentificare a utilizatorilor sau de stabilire a ce operații au voie să facă aceștia pe aplicația mobile va fi același cu cel al aplicației web, folosindu-se de același server și de aceeași bază de date.

Input-ul va veni de la client prin intermediul interfeței și va comunica după cu sistemul de control al utilizatorilor și cu serverul care acces la baza de date centrală. Output-ul va fi oferit de aceste mecanisme interne și va fi comunicat utilizatorului tot la nivel de front-end.







5. Concluzii

Aplicația Zkybe se adresează tuturor persoanelor doritoare de socializare, dar și de împărtășire a hobby-urilor. Utilizatorii se vor bucura de o experiență plăcută și sigură, oferind o interfață pe înțelesul oricui.

În document s-a discutat despre sistem, prezentarea acestuia, arhitectura aplicației și componentele ei mai în detaliu, dar și despre comunicarea dintre componente.

// todo: trebuie pusa diagrama la baze de date... trb sa ne gandim putin mai mult cum facem, ce componente avem, cum le legam si sa le si normalizam putin sa n-avem 14000 de tabele. Mai trb sa vedem si cu jira ce facem