

## Structurarea Datelor:

Propunem ca proiectul sa utilizeze o arhitectură Serverless cu o bază de date Non-relațională (NoSQL).

### 1. Tipul de Stocare: Non-relațională (NoSQL)

- BD Relațională / Non-relațională: Datele vor fi stocate într-o bază de date Non-relațională (NoSQL). Aceasta este Firebase Realtime Database.
- NoSQL: Acest model permite citirea rapidă a unor structuri complexe (cum ar fi array-urile de puncte pentru rute) și suportă sincronizarea datelor în timp real (prin WebSocket-uri).
- Fișiere pe disc: Nu se folosește stocare locală pe disc (fișiere) pentru datele persistente ale aplicației. Toate datele sunt stocate și sincronizate prin Cloud (Firebase).

### 2. Formatul Datelor: JSON Tree

În NoSQL, conceptul de tabelă și cheie străină este înlocuit cu noduri (nodes) și referințe (paths). Datele sunt stocate sub forma unui Arbore JSON (JSON Tree).

- Formatul de fișier: Datele sunt manipulate și transferate sub formă de obiecte JSON (JavaScript Object Notation).
- Legături: Legăturile între noduri (similar cu cheile străine) se fac prin referințe la cale (path reference). De exemplu, pentru a lega un traseu de un utilizator, se folosește ID-ul unic al utilizatorului (UID) direct în calea de stocare a rutei.

### 3. Structura Nodurilor

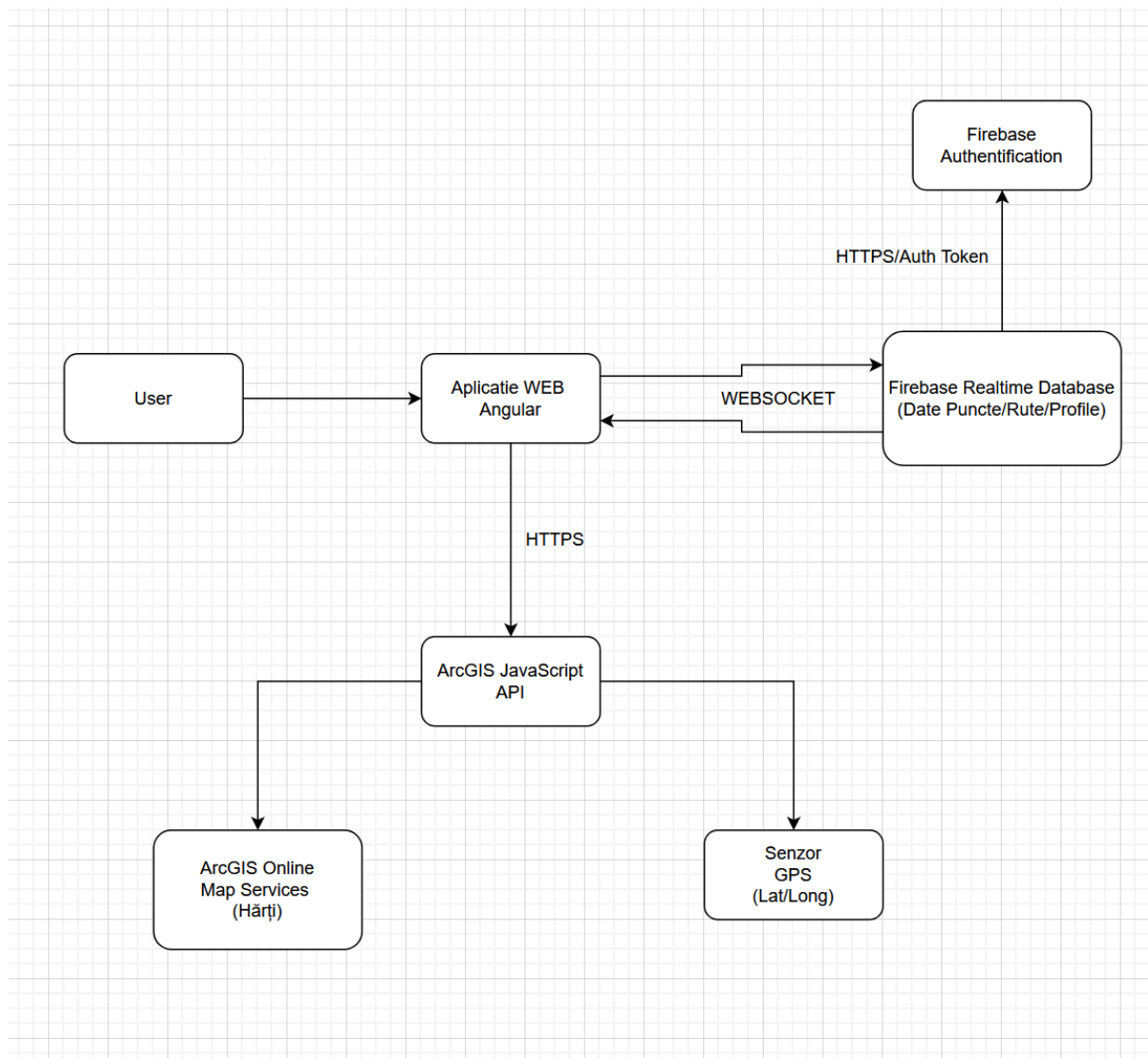
Nod (Cale)	Tip de Dată	Funcția Principală
/puncteColectare	Listă de Obiecte JSON	Stochează locațiile geografice (GeoData) și metadatele (tipuri de deșeuri acceptate) ale punctelor de reciclare.

/users/UID	Obiecte JSON	Stochează datele de profil ale utilizatorului, inclusiv rolul (admin/user) și eventualele puncte acumulate.
/rute/routeld	Array de Obiecte JSON	Stochează traseele parcurse sau planificate (datele <i>Polyline</i> ), sub forma unei liste de coordonate (Lat/Long).

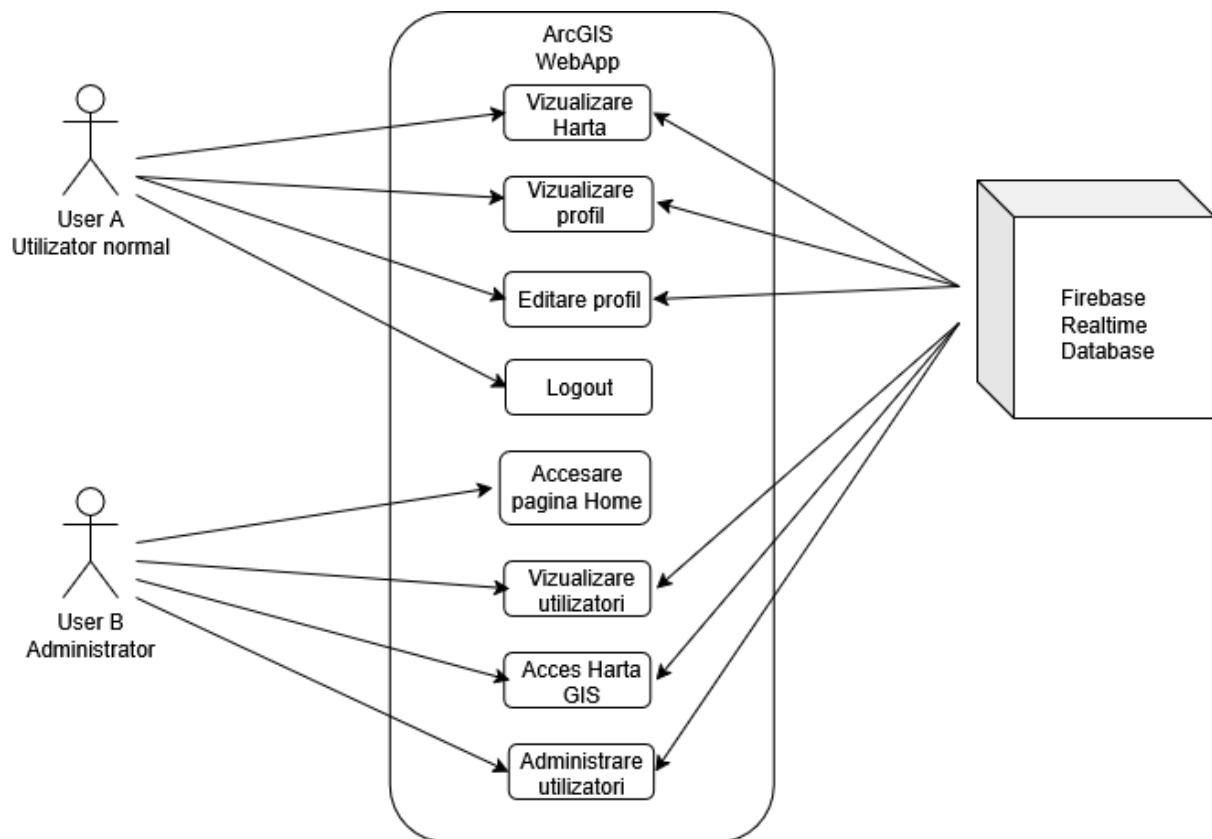
#### 4. Diagrama Bazei de Date

Deoarece nu există tabele și chei străine, diagrama se concentrează pe structura ierarhică (arborele JSON).

## Arhitectura generala a aplicatiei:



## Diagrama cazurilor de utilizare:



## Tehnologii:

### Angular (TypeScript): Framework Frontend

Construiește interfața grafică (UI), gestionează logica Client-Side și rutele aplicației (SPA).

### Firestore Backend-as-a-Service (BaaS):

Platforma Cloud care asigură serviciile de backend, eliminând nevoia unui server dedicat (Node/Java/etc.).

### Firestore Realtime DB: Bază de Date NoSQL

Stochează și sincronizează rapid datele (puncte de colectare cu Lat/Long, trasee, profile utilizatori) folosind structuri JSON.

### Firestore Authentication: Serviciu de Securitate

Gestionează securizat procesul de login/logout și token-urile utilizatorilor, inclusiv verificarea rolurilor (admin/user).

### Esri ArcGIS API for JS / Hărți:

Permite afișarea hărților, vizualizarea punctelor de colectare, randarea traseelor (Polyline) și configurarea elementelor interactive (PopupTemplate).

### RxJS Librărie Angular:

Gestionează asincronismul și asigură sincronizarea datelor în timp real, în special pentru preluarea datelor din Firebase (Observable).

**SCSS (Sass):** Limbaj de Stilizare

Folosit pentru pre-procesarea și structurarea modulară a codului CSS.

### Organizare activitati:

NR	Task Specifi	Tehnologie Principală	Responsabil	Laborator
I. CLOUD & DATE (Firebase)				
1	Configurare inițială Firebase și instalare module AngularFire (compat).	Firebase, Angular	Student 1	L7
2	Implementare funcții de scriere în BD: addListObject2 (adăugare puncte colectare cu lat, long) și setUserPosition (poziția curentă a utilizatorului).	Firebase Realtime DB	Student 1	L7

3	Definire reguli de securitate (Security Rules) în Firebase pentru a proteja datele: citire permisă tuturor, scriere permisă doar adminilor și utilizatorului propriu (pentru poziție).	Firestore Security Rules	Student 1	L7
4.	Creare structură BD pentru stocarea traseelor (logging de coordonate - <i>Polyline</i> ).	Firestore Realtime DB	Realtime DB Student 1	L8
II. FRONTEND & LOGICĂ (Angular)				
5.	Implementare Serviciu de Autentificare (Login/Logout) folosind Firebase Authentication.	Angular, Firebase Auth	Student 2	L8
6.	Implementare Auth Guard și Logică de Roluri (admin vs. user) pentru a proteja rutele și butoanele (*ngIf="user?.role !== 'admin'").	Angular, Firebase Auth	Student 2	L8

7.	Creare și rutare componente de bază: Login, Profile, Home (pentru admini) și Map.	Angular Router	Student 2	L8
8.	Implementare afișare dinamică a "Tabelei Puncte de Colectare" în interfața de administrare.	Angular Components, RxJS	Student 2	L8
9.	Integrare componenta de hartă Esri ArcGIS (inițializarea MapView) în modulul esri-map.	ArcGIS JS API	Student 1	L9
10.	Afișarea punctelor de colectare (din Firebase) pe hartă folosind un ArcGIS GraphicsLayer.	ArcGIS JS API, RxJS	Student 1	L10
11.	Implementare funcționalitate de geolocalizare pentru a trimite poziția utilizatorului către Firebase.	Browser GPS API, Firebase DB	Student 2	L11

12.	Configurarea șablonului de PopupTemplate pentru a afișa detaliile punctelor de colectare la click (Nume, materiale: Plastic, Hârtie, etc.).	ArcGIS JS API	Student 2	L12
13.	Implementare funcționalitate de citire și randare a traseelor (Polyline) din baza de date pe hartă.	ArcGIS JS API	Student 1	L13

### Identificarea riscurilor:

1. Esec la autentificarea Firebase poate bloca accesul utilizatorilor; rezolvare: verificare conexiune si tratare erori.
2. Bug-uri in integrarea hartii Esri pot impiedica incarcarea acesteia; rezolvare: folosirea versiunilor stabile si testare pe multiple browsere.
3. Acces neautorizat la date sensibile poate expune informatii; rezolvare: reguli stricte Firebase Security.
4. Lipsa de colaborare in echipa. rezolvare: comunicare constanta, impartire task-uri in mod adecvat, moduri de lucru colaborative (ex: github)
5. Nefamiliaritate cu tehnologiile folosite. rezolvare: se lucreaza din timp, ne ajutam in echipa cu resurse si debugging.



6. Documentatia insuficienta ingreuneaza mentenanta proiectului; rezolvare: actualizare continua a documentatiei.
7. Lipsa testarii adecvate poate introduce bug-uri in productie; rezolvare: testare unitara si testare manuala.
8. Datele introduse gresit de utilizatori afecteaza acuratetea sistemului; rezolvare: validari in front-end si backend.
9. Utilizatorii care uita parola pot solicita suport suplimentar; rezolvare: functionalitate reset password.