

# **CityReach -**

## **Analiza Accesibilității Transportului**

### **Public**

Cozma Andrei - 345C1  
Manea Andrei-Iulian - 342C5  
Tudor Diana-Ioana - 342C5

# 1. Introducere

Acest proiect își propune să analizeze accesibilitatea la transportul public în zonele rezidențiale. Aplicația generează **buffere de câteva sute de metri** în jurul stațiilor de transport, intersectează aceste buffere cu poligoanele cartierelor și afișează **traseele disponibile**. În plus, calculează **procentul populației** care se află în aria acoperită de buffere, oferind utilizatorilor o **vizualizare interactivă și statistică** asupra accesibilității transportului public.

## 2. Structurarea datelor

### 2.1. Tipul sistemului de stocare

Aplicația folosește **Firestore**, o bază de date non-relațională (NoSQL), deoarece permite integrare rapidă cu frontend-ul **React** și sincronizare în timp real.

### 2.2. Colecții Firestore și legături

#### Colecția users

Reprezintă utilizatorii aplicației.

- **id**: identificator unic generat automat de Firestore
- **email**
- **passwordHash**: hash al parolei (gestionat de Firebase Auth)
- **role**: rolul utilizatorului în aplicație (admin sau user)

#### Colecția stations

Reprezintă punctele de acces la transportul public.

- **id**
- **name**
- **type**: (bus, tram, metro)
- **location**: GeoPoint cu coordonatele exacte ale stației
- **bufferRadius**: raza bufferului în metri (configurat de utilizatorii cu rolul de admin; 300-500 m)

#### Colecția neighborhoods

Reprezintă cartierele sau zonele populate.

- **id**
- **name**
- **population**
- **geometry**: poligon GeoJSON al zonei rezidențiale

## Colecția analytics (rezultate analize)

Stochează rezultatele analizelor spațiale.

- **id**
- **neighborhoodId**: referință la neighborhoods.id (legătură între analiza calculată și cartier)
- **coveragePercent**: procentul populației din cartier acoperit de buffere
- **coveredArea**: suprafața efectiv acoperită de buffere (în metri pătrați)
- **totalArea**: suprafața totală a cartierului
- **timestamp**: momentul în care analiza a fost generată

## 2.3. Fișiere GIS

Datele geografice sunt stocate și în format:

- **GeoJSON** pentru stații de transport
- **GeoJSON** pentru cartiere

Structura stațiilor:

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "properties": { "name": "Stație A", "type": "bus" },
      "geometry": { "type": "Point", "coordinates": [...] }
    }
  ]
}
```

Structura cartierelor:

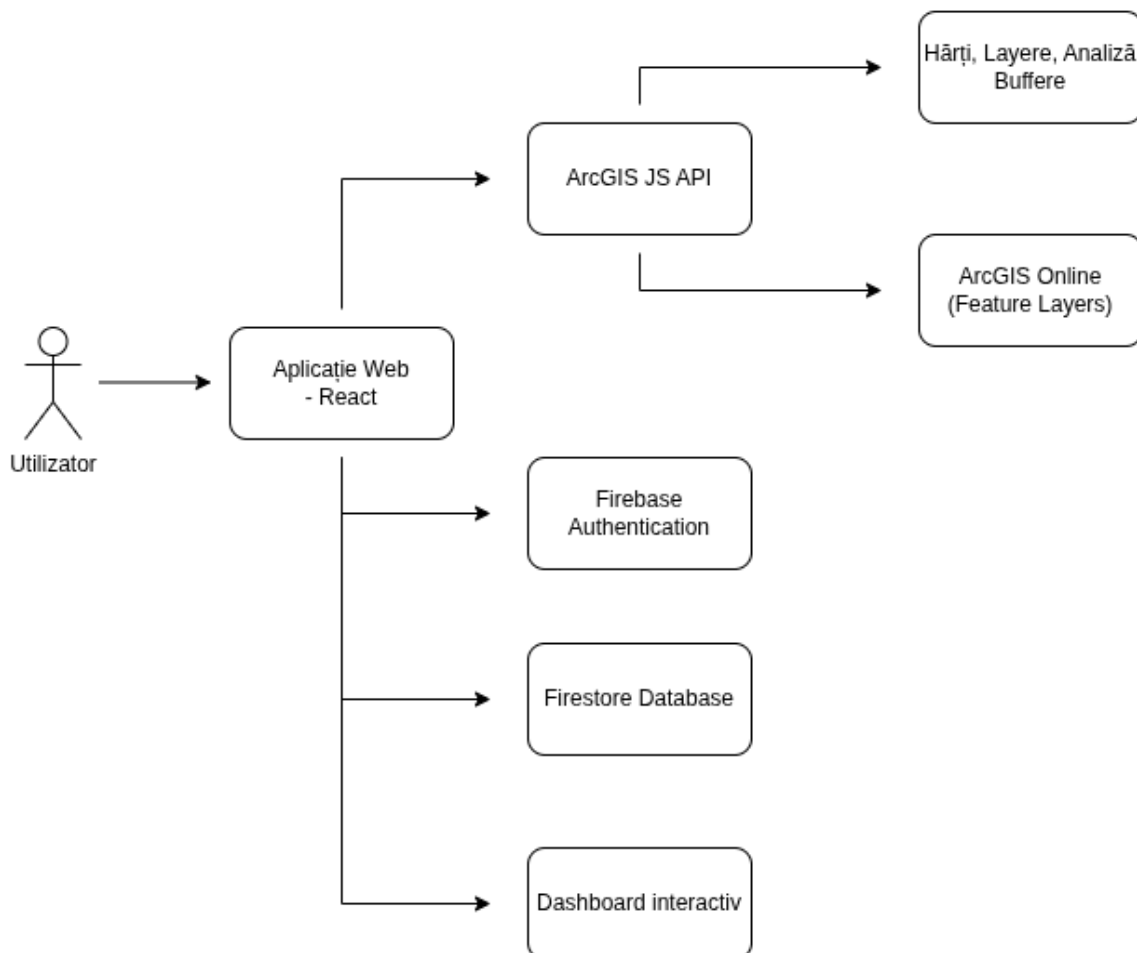
```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "properties": { "name": "Cartier 1", "population": 12000 },
      "geometry": { "type": "Polygon", "coordinates": [ ... ] }
    }
  ]
}
```

## 3. Arhitectura generală a aplicației

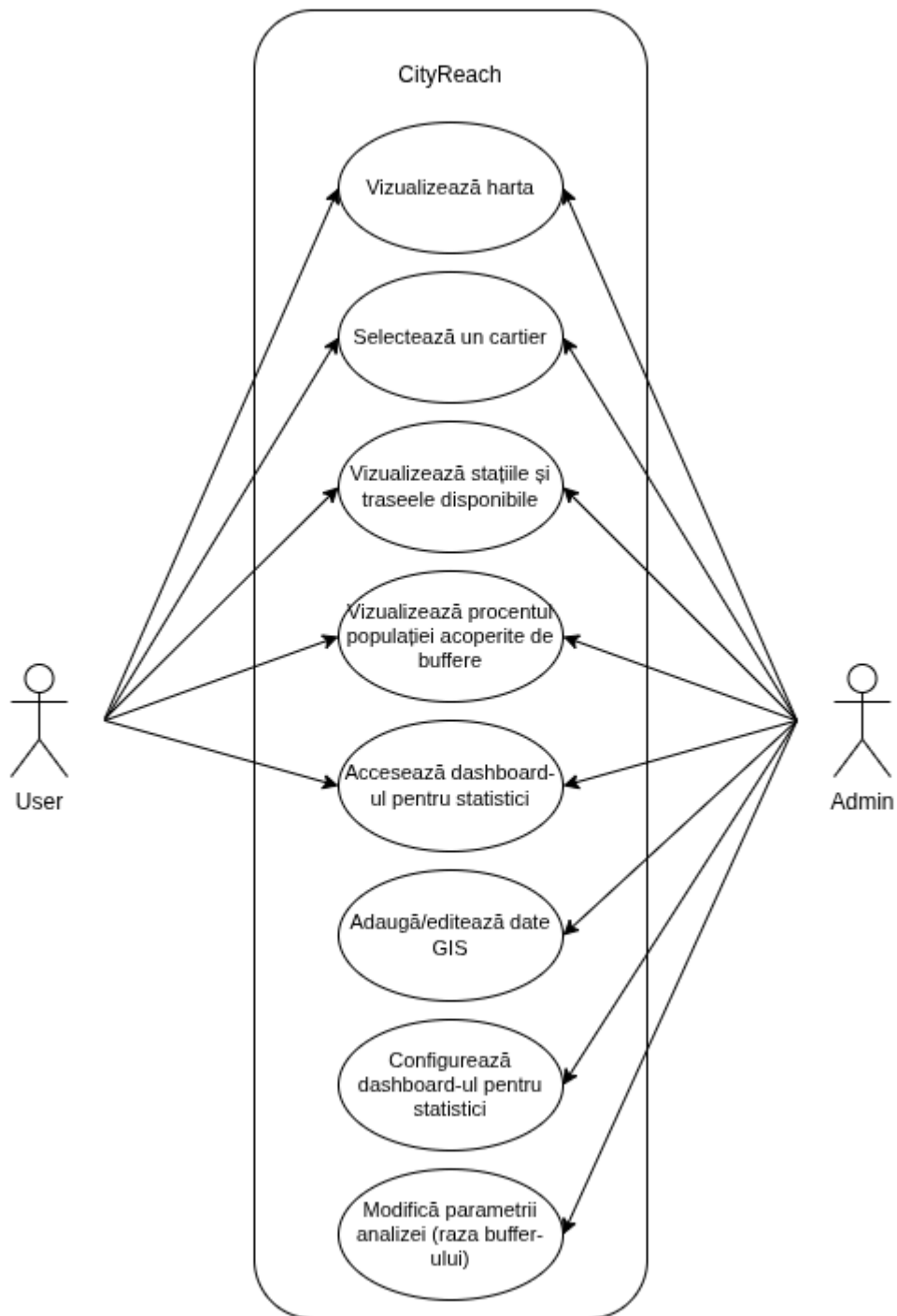
### 3.1. Componente principale

- Aplicație Web - React
- Firebase Authentication
- Firestore Database
- ArcGIS Online Feature Layers
- **Module de analiză spațială realizate în browser** (generare buffere, intersecție cu cartiere, calcul procent populație acoperită)
- **Dashboard interactiv** pentru vizualizarea statisticilor de acoperire și a procentului populației

### 3.2. Diagramă



## 4. Diagrama cazurilor de utilizare



## 5. Tehnologii folosite

### 5.1. Tehnologii generale

- React
- ArcGIS JavaScript API
- Firebase (Auth + Firestore)
- HTML/CSS/JS
- GeoJSON

### 5.2. Tehnologii specifice

- **Feature Layers** create în ArcGIS Online pe baza GeoJSON-urilor
- **Configurare popup** folosind modulul esri/widgets/Popup
- **Analiză spațială** folosind geometryEngine:
  - buffer()
  - intersect()
  - geodesicArea()
- **Dashboard statistic** construit în React cu grafice și tabele pentru acoperirea populației și statistici
- **Autentificare:** Firebase Auth
- **Baza de date:** Firestore

## 6. Lista task-urilor

Cozma Andrei - Task 3, 4, 8, 9, 12, 13, 19, 20

Manea Andrei-Iulian - Task 1, 5, 6, 10, 14, 15, 19, 20

Tudor Diana-Ioana - Task 2, 7, 11, 16, 17, 18, 19, 20

Task	Responsabil	S10	S11	S12	S13
1) Import date OSM și conversie în GeoJSON	Manea Andrei-Iulian				
2) Validare și curățare date GeoJSON	Tudor Diana-Ioana				
3) Creare FeatureLayer pentru stații	Cozma Andrei				
4) Creare FeatureLayer pentru	Cozma Andrei				
5) Configurare popup pentru stații	Manea Andrei-Iulian				

6) Configurare popup pentru cartiere	Manea Andrei-Iulian				
7) Generare buffere 300-500 m în jurul stațiilor	Tudor Diana-Ioana				
8) Intersecție buffere cu cartiere	Cozma Andrei				
9) Calcul procent populație acoperită	Cozma Andrei				
10) Calcul suprafață acoperită	Manea Andrei-Iulian				
11) Implementare heatmap accesibilitate	Tudor Diana-Ioana				
12) Autentificare Firebase (login/register)	Cozma Andrei				
13) Sistem roluri admin/user	Cozma Andrei				
14) Dashboard admin cu statistici și grafice	Manea Andrei-Iulian				
15) Salvare rezultate analiză în Firestore	Manea Andrei-Iulian				
16) Optimizare performanță analize GIS	Tudor Diana-Ioana				
17) Pagină admin pentru adăugare/editare stații	Tudor Diana-Ioana				
18) Pagină admin pentru import GeoJSON cartiere	Tudor Diana-Ioana				
19) Testare funcționalități GIS și debugging	Toți				
20) Realizare prezentare finală	Toți				

## 7. Identificarea riscurilor

Risc	Probabilitate	Impact	Mitigare
Date OSM incomplete sau eronate	medie	scăzut	folosirea unor dataset-uri suplimentare
GeoJSON cu erori de geometrie	medie	scăzut	validare și reparare cu tool-uri GIS
Analize GIS lente pe browser	medie	ridicat	optimizare geometrie, reducere nr. de entități
Conexiune instabilă Firebase	mică	ridicat	retry automat / cache local
Bug-uri greu de reparat	mare	moderat	testare incrementală
Discuții în contradicție în echipă	medie	moderat	stabilirea unui lider tehnic și reguli clare
Întârzierea unor membri ai echipei	medie	ridicat	redistribuire task-uri
Neînțelegerea cerințelor din barem	mică	ridicat	verificare periodică cu asistentul de laborator
Risc ca proiectul să nu fie gata la timp	mare	ridicat	MVP rapid + adăugare treptată feature-uri