

CivicAlert

Platforma Smart City pentru Managementul Sesizarilor Urbane

Echipa: GoSky

Membri:

Alexandru Grecu

Andrei-Alexandru Girleanu

30 noiembrie 2025

Cuprins

1 Definirea Temei	2
1.1 Tema generala si obiectivele propuse	2
2 Structurarea Datelor	2
2.1 Modelul de Date (Colectii Firestore)	2
2.1.1 1. Colectia: <code>users</code>	2
2.1.2 2. Colectia: <code>tickets</code>	3
2.1.3 3. Sub-colectia: <code>tickets/{ticketId}/comments</code>	4
2.1.4 4. Colectia: <code>notifications</code>	4
3 Arhitectura Generala a Aplicatiei	5
3.1 Schema Bloc	5
4 Diagrama Cazurilor de Utilizare	5
5 Tehnologii	6
5.1 Generale	6
6 Organizare Activitati	7
7 Identificarea Riscurilor	8

1 Definirea Temei

1.1 Tema generala si obiectivele propuse

Proiectul propune dezvoltarea unei platforme web de tip *Smart City*, destinata digitalizarii interactiunii dintre cetateni si administratia locala. Solutia se bazeaza pe trei piloni fundamentali: raportarea spatiala a incidentelor, implicarea comunitara (prin voturi si comentarii) si un sistem proactiv de notificari in timp real, care asigura transparenta totala a procesului de solutionare a problemelor.

Obiective principale:

- **Sistem de Notificari:** Alertarea automata a utilizatorilor la modificarea statusului tichetelor sau la aparitia unor noutati in comunitate, mentionand utilizatorul conectat permanent.
- Implementarea unui sistem de **Ticketing Geospatial**: transformarea sesizarilor in ti-chete gestionabile direct pe harta.
- Vizualizarea interactiva folosind **ArcGIS API**.
- **Sistem de Votare (Prioritizare):** Cetatenii pot sustine sesizarile existente prin "Upvote".
- **Identitate si Personalizare:** Profiluri de utilizator cu avatar (poza profil) si setarea *Orasului de Interes* pentru centrarea automata a hartii.
- Analiza spatiala prin vizualizare de tip **Heatmap**, utila pentru analizarea problemelor.

2 Structurarea Datelor

Infrastructura se bazeaza pe Google Firebase, asigurand stocarea datelor, a fisierelor media si gestionarea utilizatorilor.

2.1 Modelul de Date (Colectii Firestore)

2.1.1 1. Colectia: users

Contine setarile de profil si preferintele de navigare.

- `uid` (String): ID unic.
- `email` (String).
- `displayName` (String).
- `photoUrl` (String): URL catre avatarul stocat in Firebase Storage.
- `role` (String): '*citizen*' sau '*admin*'.
- `homeCity` (Object): Preferinta de localizare (evita geolocatia repetitiva).
 - `name` (String): Ex: "Craiova".
 - `latitude` (Number): Coordonata centrala a orasului.
 - `longitude` (Number).
- `createdAt` (Timestamp).



Figura 1: Schema bazei de date Firestore

2.1.2 2. Colectia: tickets

Incluseaza logica de votare si galerie foto.

- **id** (String): ID unic.
- **userId** (String): Referinta user.
- **category** (String): Ex: 'Gropi', 'Iluminat'.
- **description** (String).
- **status** (String): '*pending*', '*approved*', '*rejected*', '*resolved*'.
- **location** (Object): lat/long.
- **imageUrls** (Array of Strings): Galerie foto (initiala).
- **votes** (Array of Strings): Lista UID-urilor utilizatorilor care au dat "Upvote".

- *Nota:* Numarul de voturi se calculeaza ca `votes.length`.
 - `createdAt`, `updatedAt` (Timestamp).
- 2.1.3 3. Sub-colectia: `tickets/{ticketId}/comments`**
- `id`, `userId`, `userName` (String).
 - `userPhotoUrl` (String): Avatarul autorului (pentru afisare rapida in lista de discutii).
 - `message` (String).
 - `attachmentUrls` (Array of Strings): Galerie foto (comentarii).
 - `timestamp` (Timestamp).

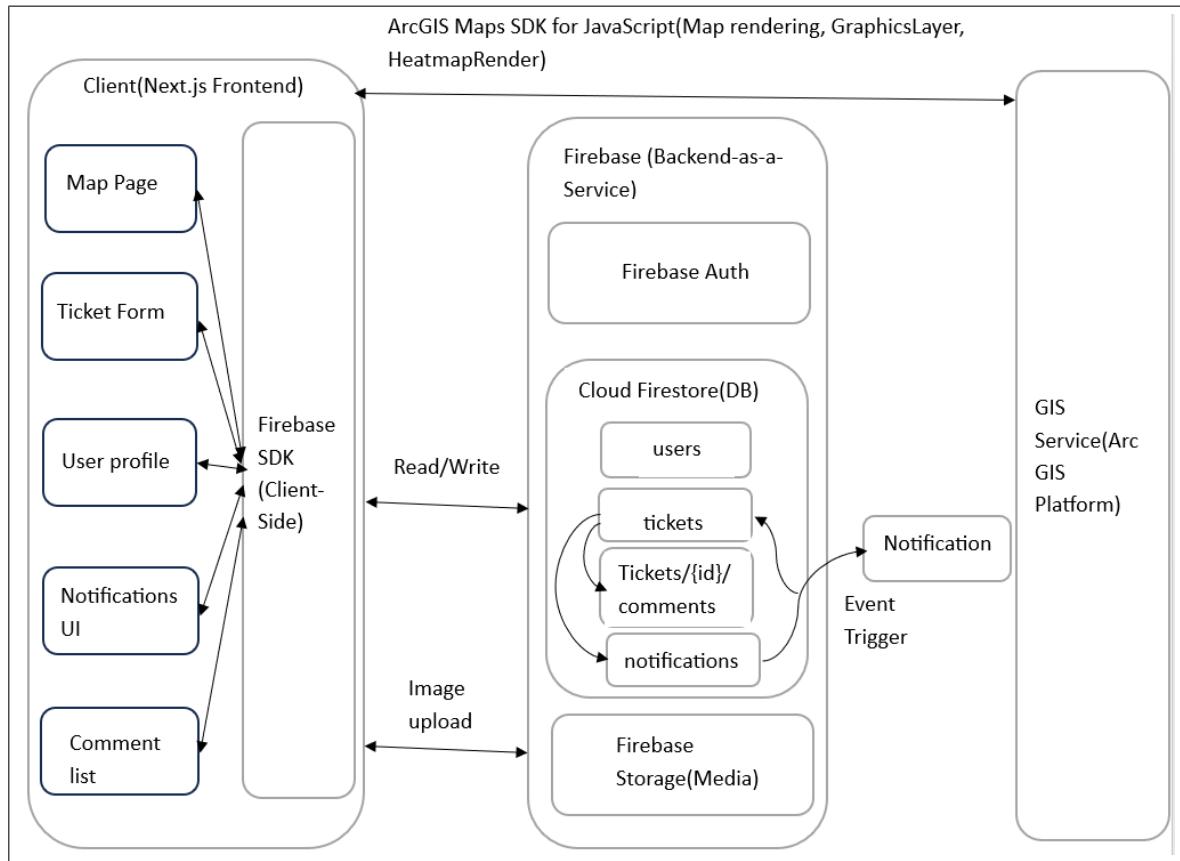
2.1.4 4. Colectia: `notifications`

Stocheaza alertele pentru utilizatori.

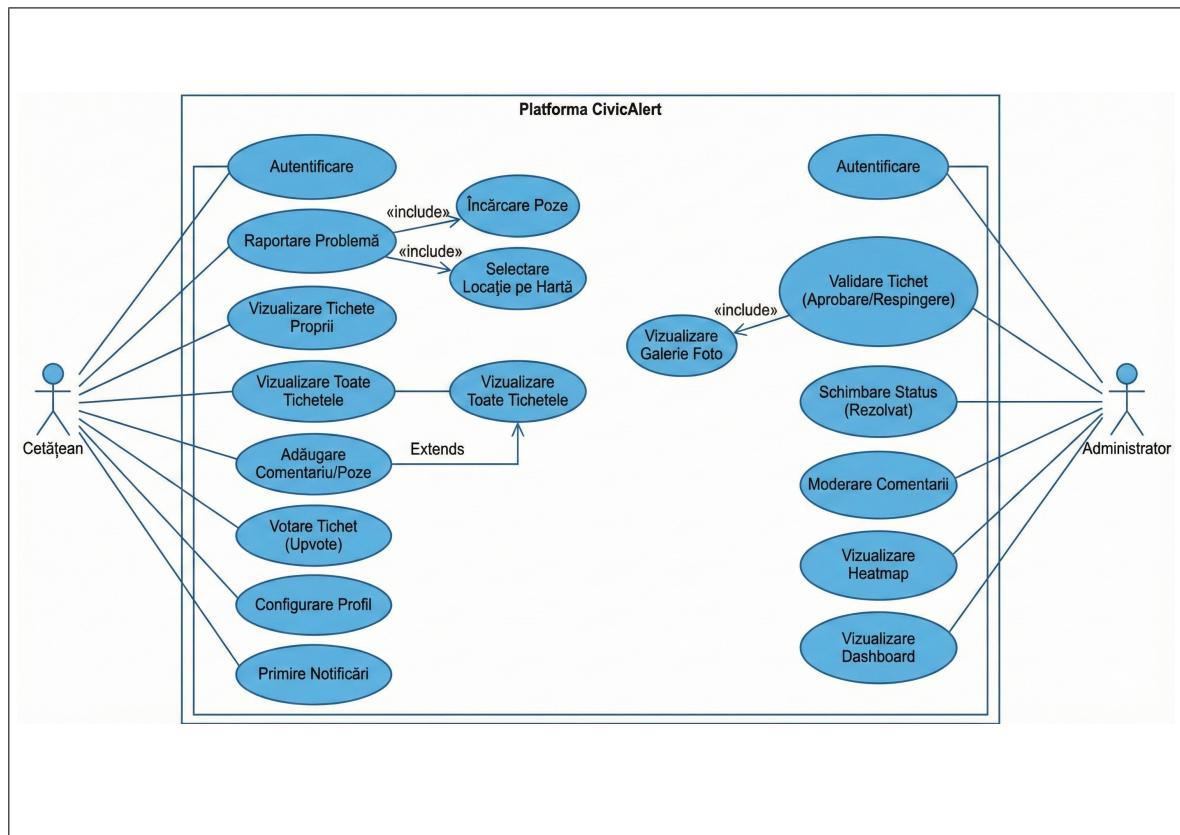
- `id` (String): ID unic.
- `recipientId` (String): Destinatarul notificarii.
- `type` (String): `'status_change'`, `'new_comment'`, `'vote_milestone'`.
- `ticketId` (String): Referinta catre ticket.
- `message` (String): Textul afisat (ex: "Sesizarea ta a fost rezolvata!").
- `isRead` (Boolean): Indicator citit/necitit.
- `createdAt` (Timestamp).

3 Arhitectura Generala a Aplicatiei

3.1 Schema Bloc



4 Diagrama Cazurilor de Utilizare



5 Tehnologii

Aceasta sectiune detaliaza tehnologiile alese pentru implementarea platformei CivicAlert, justificand alegerile facute in functie de cerintele de performanta, scalabilitate si functionalitatatile geospatiale necesare.

5.1 Generale

Arhitectura aplicatiei este de tip *Serverless* si *Single Page Application (SPA)* (cu elemente de Server-Side Rendering pentru performanta initiala), bazata pe ecosistemul JavaScript/TypeScript.

Tehnologii si Framework-uri Principale:

- **Frontend Framework: Next.js (React)**

- Folosit pentru construirea interfetei utilizator, gestionarea rutarii si optimizarea performantei prin incarcarea hibrida a paginilor.
- Limbajul de programare utilizat este **TypeScript**, pentru a asigura robustetea codului prin tipizare statica.

- **Backend-as-a-Service (BaaS): Google Firebase**

- Inlocuieste necesitatea unui server backend traditional.
- **Cloud Firestore:** Baza de date NoSQL, orientata pe documente, folosita pentru stocarea tuturor datelor structurate (utilizatori, tichete, comentarii, notificari, voturi).
- **Firebase Authentication:** Gestioneaza intregul flux de autentificare si securitatea conturilor.
- **Firebase Storage:** Stocare de obiecte (bucket) pentru fisierele media (imagini incarcate de utilizatori).

- **Platforma GIS: ArcGIS**

- **ArcGIS Maps SDK for JavaScript:** Biblioteca principala utilizata pentru toate functionalitatatile geospatiale: randarea hartii interactive, afisarea incidentelor si analiza spatiala (heatmap).
- **ArcGIS Online basemaps:** Hartile de fundal (ex: strazile, imaginile satelitare) sunt consumate ca servicii de la Esri.

Unelte de Dezvoltare si UI:

- **Tailwind CSS:** Framework CSS *utility-first* pentru stilizarea rapida si responsive a componentelor UI.
- **Node.js & npm:** Mediul de executie si gestionarul de pachete pentru dependentele proiectului.
- **Git & GitHub:** Pentru controlul versiunilor si colaborarea in echipa.

		L1	L2	L3	L4
Student 1	S1-T1				
	S1-T2				
	S1-T3				
	S1-T4				
	S1-T5				
	S1-T6				

		L1	L2	L3	L4
Student 2	S2-T1				
	S2-T2				
	S2-T3				
	S2-T4				
	S2-T5				
	S2-T6				

Figura 2: Planificarea task-urilor pe laboratoare

6 Organizare Activitati

Legenda Task-uri

Student 1:

- S1-T1** Configurare Mediu + Autentificare (Login/Register)
- S1-T2** Creare Structura Baza de Date + Pagina Profil
- S1-T3** Integrare Harta Base View (ArcGIS)
- S1-T4** Formular Tichet + Upload Multiplu Poze
- S1-T5** Implementare Centrare Harta pe Orasul Userului
- S1-T6** Creare Componenta Sidebar Listare Tichete

Student 2:

- S2-T1** Conversie Date Firestore in ArcGIS Graphics
- S2-T2** Implementare si Configurare Pop-ups (cu Galerie Foto)
- S2-T3** Creare Dashboard Administrator
- S2-T4** Implementare Heatmap Renderer
- S2-T5** Implementare Sistem de Votare (Upvote)
- S2-T6** Dezvoltare Modul Comentarii si Notificari

7 Identificarea Riscurilor

Analiza riscurilor a fost efectuata pe baza unei matrici de tip *Probabilitate vs. Impact*, pentru a prioritiza masurile de preventie.

Matricea de Evaluare a Riscurilor(Probabilitate vs Impact)

	Scazut	Impact moderat	Ridicat
Probabilitate Mica	Risc 1: Incompatibilitate browser vechi	Risc 4: Esec temporar servicii externe	Risc 8: Exponere chei API(Security Leak)
Probabilitate medie	Risc 3: Acuratete GPS redusa	Risc 5: Integrare React-ArcGIS	Risc 7: Depasire cote Firebase
Mare	Risc 2: Tichete duplicate	Risc 6: Continut inadecvat(spam)	Risc 9: Intarzieri functionalitati complexe

Figura 3: Matricea de evaluare a riscurilor (Probabilitate vs. Impact)

Catalogul Riscurilor si Planul de Mitigare

Tabelul urmator detaliaza cele 9 riscuri identificate in matricea de mai sus si solutiile tehnice propuse pentru fiecare.

Cod Matrice	Descriere Risc	Masuri de Contracarare
Risc 1	Incompatibilitate Browser Vechi (Probabilitate Mica / Impact Scazut) Un numar mic de utilizatori ar putea accesa platforma de pe browsere care nu suporta WebGL (necessar ArcGIS) sau ES6+.	Afisarea unui mesaj proactiv ("Banner") pentru utilizatorii detectati cu browsere incompatibile, recomandand actualizarea la una moderna (Chrome, Firefox, Edge).
Risc 2	Tichete Duplicate (Probabilitate Mare / Impact Scazut) Probabilitate mare ca vecinii sa raporteze independent aceeasi problema (ex: o groapa).	Implementarea functiei de " Upvote " (Sustinere). Utilizatorul vede pin-ul existent si apasa "Sustin" in loc sa creeze un tichet nou.
Continua pe pagina urmatoare...		

Tabela 1 – continuare din pagina anterioara

Cod Matrice	Descriere Risc	Masuri de Contracarare
Risc 3	Acuratete GPS Redusa (Probabilitate Medie / Impact Scazut) Semnalul GPS al telefonului/browserului poate avea abateri de 10-50m.	Possibilitatea de ajustare manuala a pozitiei marker-ului pe harta (<i>Drag & Drop</i>) inainte de trimitera formularului.
Risc 4	Esec Temporar Servicii Externe (Probabilitate Mica / Impact Moderate) Intreruperi rare ale API-urilor Esri (ArcGIS Online) sau Google Firebase.	Implementarea unei tratari elegante a erorilor (Graceful Degradation) in UI (ex: afisarea unui mesaj "Harta este momentan indisponibila" in loc de blocarea aplicatiei).
Risc 5	Integrare React-ArcGIS (Probabilitate Medie / Impact Moderate) Dificultati in sincronizarea starii React cu DOM-ul hartii ArcGIS (risc de memory leaks).	Utilizarea <i>React Hooks</i> ('useEffect', 'useRef') pentru a gestiona strict ciclul de viata al hartii si distrugerea instantei la unmount.
Risc 6	Continut Inadecvat (Spam) (Probabilitate Mare / Impact Moderate) Incarcarea de imagini irelevante sau limbaj ofensator in comentarii.	Implementarea unui buton de "Raporteaza" si a unui panou de moderare pentru Admin (stergere continut).
Risc 7	Depasire Cote Firebase (Probabilitate Medie / Impact Ridicat) Atingerea limitelor planului gratuit (Spark) din cauza numarului mare de citiri.	Optimizarea interogarilor (paginare, limitare rezultate) si stocarea datelor statice (ex: lista orase) in cache local.
Risc 8	Expunere Chei API (Security Leak) (Probabilitate Mica / Impact Ridicat) Includerea accidentală a cheilor private (ex: Firebase Service Account) in codul client-side.	Utilizarea strictă a variabilelor de mediu ('.env.local') si asigurarea ca cheile sensibile sunt accesate doar din API Routes (server-side), nu din browser.
Risc 9	Intarzieri Functionalitati Complex (Probabilitate Mare / Impact Ridicat) Riscul de a nu finaliza Heatmap-ul sau Votarea la timp.	Prioritizarea modulelor critice (MVP) si simplificarea cerintelor (ex: Heatmap static in loc de dinamic) daca este necesar.