

15006016



SafeZone

Safety & Navigation

Noy Zion · Dana Reshef

Yogev Shani

Cloud Computing Workshop with 



The Problem

Difficulty **locating the nearest shelter** within critical response time



Lack of tools to **notify** loved ones or coordinate with emergency services

Emotional stress and panic due to insufficient real-time guidance

No visibility into **shelter condition** – many are poorly maintained



Target Audience

All residents of Israel

Parents, elderly individuals, and people with **limited mobility**

Tourists unfamiliar with shelter locations

Emergency responders and local authorities

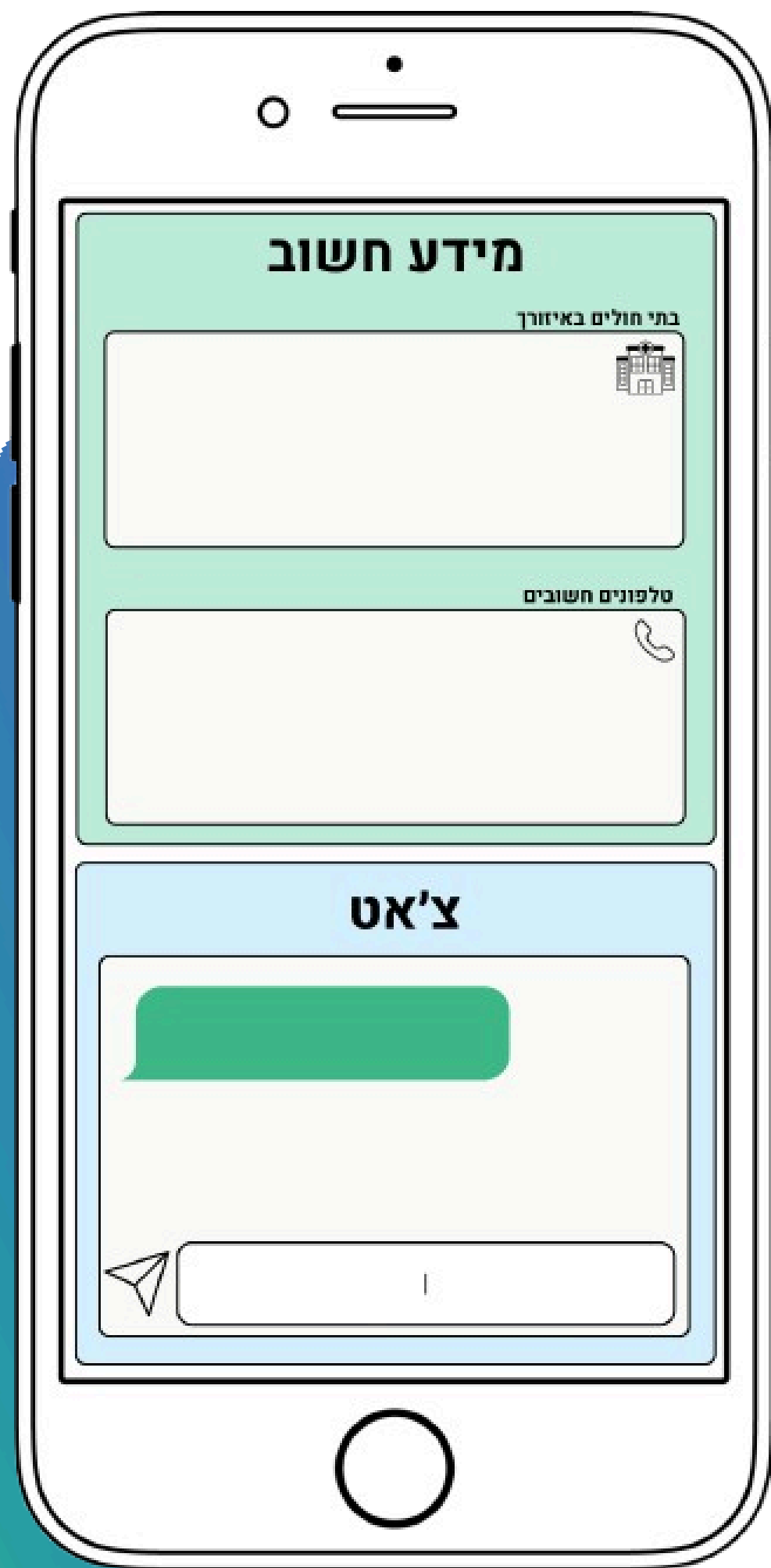
Our Solution –



SafeZone
Safety & Navigation

How Does It Work?

- Real-time alerts based on user location 
- Step-by-step shelter navigation 
- Notify loved ones with location updates 
- Emotional support powered by GenAI 
- Real-time shelter condition reports 
- Post-emergency recovery tools 
- “Need Help” mode alerts nearby users and responders. 



Live Demo



How does enable SafeZone to operate efficiently?

To deliver a fast, secure, and reliable emergency
response,

SafeZone leverages AWS cloud services for
real-time data management,
communication, and alert delivery.



How SafeZone Uses AWS – Services Overview



Lambda

Runs all backend logic serverlessly:

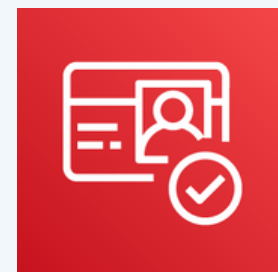
saving shelter reports, sending alerts, updating shelters, fetching user location, and more – all in real time without managing servers.



Dynamo DB

Main database for all system data:

shelters, users, reports, contacts, and alert zones.



Cognito

Manages user authentication and identity:

handles sign-up, login, and links user data (e.g., reports, contacts) to their identity via phone/email

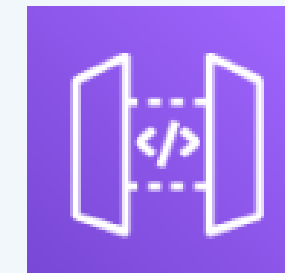


S3 Bucket

Stores all uploaded images –

including shelter photos and report attachments.

Uses signed upload URLs for secure, direct uploads from the app.



API Gateway

Exposes public HTTP APIs for the mobile app to communicate with the backend.



EventBridge

Triggers scheduled background tasks:

a Lambda function runs every minute to check for new alerts and sends push notifications to relevant users based on their location.



Nearest Shelter Detection – How It Works

Live Location Access

Using expo-location, we get accurate GPS coordinates from the device.

Coordinate Conversion

Since our shelter data is in ITM format, we convert it to WGS84 using proj4 to align with GPS.

Distance Calculation

We loop through all shelters, calculate distances in real time, and select the closest one

Data Caching with AsyncStorage

Once the closest shelter is found, we store it locally to improve performance and provide fallback access.

Navigation Integration

A single tap opens Google or Apple Maps, preloaded with the shelter's exact location.

Technologies Used:

- expo-location – for precise GPS access
- proj4 – for ITM ↔ WGS84 coordinate conversion
- TypeScript + Haversine formula – for distance calculation
- AsyncStorage – to store and reuse the selected shelter



Third-Party Services

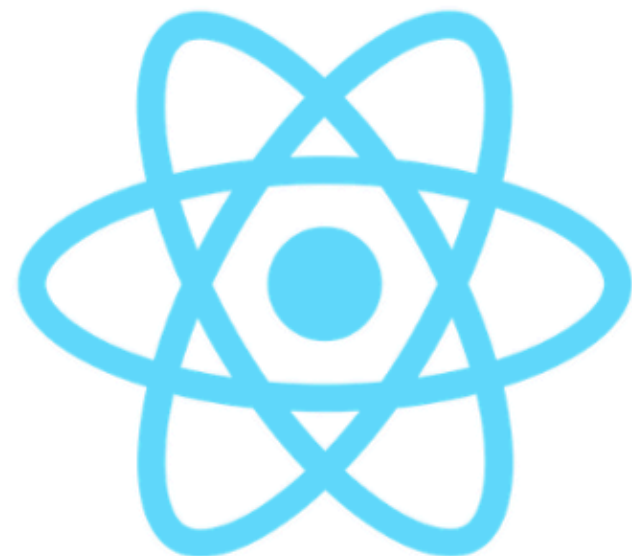


Google Maps



OpenAI

Frontend



+



Limitations of Existing Emergency Apps



01

Shelter Navigation Missing

These apps send alerts — but users have no way to quickly locate or navigate to the nearest shelter

02

No Communication Tools

Users can't notify loved ones or update contacts during emergencies

03

No Emotional Support

None of the existing apps help users deal with panic or anxiety — which are often more dangerous than the alert itself

Over the past few months, we built a system that brings together technology,
real-world needs, and human impact.
This journey challenged us – from solving technical challenges to designing for
real users under pressure.
We also gained valuable experience in creating a real-time, serverless system
using AWS services.

Thank you for listening.
We'd love to hear your thoughts
and we're happy to answer any questions



SafeZone
Safety & Navigation