ZCAPNet - Sistema de apoio à Proteção Civil

Licenciatura em Engenharia Informática e de Computadores
Projeto e Seminário - Semestre de Verão 2024/25
Março de 2025

Students:

Luis Alves, No. 46974, e-mail: a46974@alunos.isel.pt, tel.: 93 561 40 40

Gonçalo Dimas, No. 48263, e-mail: a48263@alunos.isel.pt, tel.: 92 439 07 14

Supervisors:

André Lourenço, email: andre.lourenco@isel.pt

Artur Ferreira, e-mail: artur.ferreira@isel.pt

1. Introduction

Assistance for the population in the event of a serious accident or catastrophe is one of the concerns of Civil Protection agents and is defined within the scope of the National, Special and Municipal Emergency Plans. This aims to provide safe temporary accommodation to citizens and families displaced due to such events. Support is implemented through the activation, installation and management of Population Concentration and Support Zones (Portuguese: **ZCAP** – **Z**onas de **C**oncentração e **A**poio à **P**opulação) with the cooperation of several entities such as the National Civil Protection Authority, Municipal Councils, the Social Security Institute and others.

The information management in these buildings, although already carried out using computer support, is currently precarious because, in addition to being dispersed and isolated for each building, it fails to have a structured data model, interconnected between the various facilities and the responsible entity. This prevents data from being cross-referenced for the purposes of daily reporting, statistics, and to manage available human resources or even to search for and find missing family relatives.

The objective of this project is to develop a system that enables entities of managing and access information in a structured, simple and fast way through a single, transversal, and centralized application.

2. Problem Analysis

Consider that, in the event of a highly severe accident or catastrophe, the common connection infrastructures between the server and the clients of a Web application may not be guaranteed. Thus, it is vital that the client application keeps locally a minimum set of data to be able to operate and store new data and, when connectivity is reestablished,

it can be provided to the server, which will be responsible for data synchronization data and resolving possible conflicts. The architecture to be implemented will be based on the existence of a backend (Web API and database) that will serve frontend clients (Web App) as well as expose a set of data, namely for reporting, to be provided to other possible services/applications.

3. Mandatory Requirements

As a fully working system, the minimum requirements for the Web App should be:

- **Store data even when offline -** Should have the ability to store data even if off-line, with the help of pre-existing set of local data tables. With this ability, the system will rely on the server synchronization engine when the service is restored.
- **Create and manage Facility/Shelter facilities -** Should be able to create and manage a Facility/Shelter making sure that all the required conditions are satisfied. Should be able to monitor current shelter occupation, capacity, basic needs as well as other details.
- **Create and manage a new Event /Accident/Catastrophe data -** During and after an Event/Accident/Catastrophe, should be able to identify and manage the displaced people to the new ZCAP.
- **Enrol users in two different stages**
    - **Stage I - Quick enrolment and Initial screening -** On this stage, the people are enrolled on the Facility/Shelter with the minimum information about each one. This allows for faster registration, prioritizing everyone's safety.
    - **Stage II - Detailed data -** This second phase serves to gather all the detailed information about each person in the shelter.
- **User profiles for different roles -** Different type of users can use the system for different uses. It should have at least 4 different roles:
    - **Admin -** Users with administrator role can access any table and can add/change system configuration data.
    - **ZCAP managers -** This role is responsible for a given ZCAP, being able to manage all data from a given Event.
    - **ZCAP users -** Should be able to enrol/remove people on the ZCAP in any of the two stages.
    - **Read only users -** Role intended to be given to any entity that needs and is allowed to obtain daily reports from the each or all ZCAP.

Regarding the Web API/Database, the system should be prepared to:

- **Store and provide the following data**
    - Region data (country, region, municipality, and parish)
    - Shelters data (location, facilities, capacity, …)
    - Events data (type of event, start date, end date, …)
    - Shelter users (name, age, needs, special needs, …)
    - Users and user profiles (user, profile, data access)
- **Synchronization engine -** ability to receive and process saved data from clients (ex: conflicts created by service failures).
- **Data share -** Provide endpoints with statistic data for reporting purposes (ex: daily report)

4. Optional Requirements

- Configurable User Profile management.
- Ability to launch a peer-to-peer service to work as local server while connection to API is not available allowing local network instances to synchronize data.
- Ability to cross-reference data between shelters allowing the finding of separated family relatives bringing them together.

5. Architecture

The system's architecture will be composed by a backend with a Web API and Database and Web Application as a frontend, shown in Figure 1.

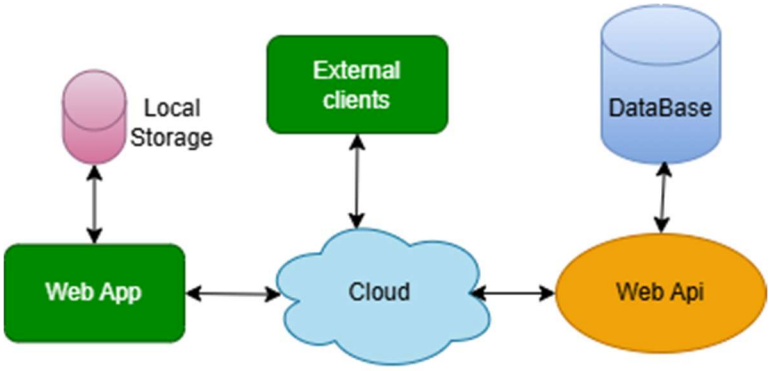

Figure 1- System architecture.

With this architecture, the solution can be accessed from multiple platforms or applications beyond the WebApp to be developed. It also allows for easier expansion of functionality without requiring updates on every or any client as long as backward compatibility is maintained. Additionally, if traffic to the API increases, the system can scale horizontally to handle the load, ensuring reliable performance.

6. Deliverables/Milestones

- 10 March - Project Proposal Document delivery
- 19 March - Requirement analysis
- 17 April - Data model, database design, and implementation
- 28 April – Progress Presentation delivery
- 21 May - Back-end prototype covering a subset of total functionality
- 21 May - Front-end prototype covering a subset of total functionality
- 02 June – Beta Version delivery
- 21 July – Final Version delivery, Final Report, User Manual

7. Risks

We identified some risks that may impact the project development and performance. The main risks are:

- **Backend unavailability –** The system must be able to operate without a connection to the backend during a service outage. It will be necessary to test

this loss of connection and evaluate the system's behaviour, as well as its state when the connection is restored.

- **Unfamiliar Technology –** The technology required to enable offline functionality has not yet been used by any of the students, which may represent an additional challenge in development and troubleshooting.

8. Project planning

Figure 2 depicts the Gantt chart with the project planning.

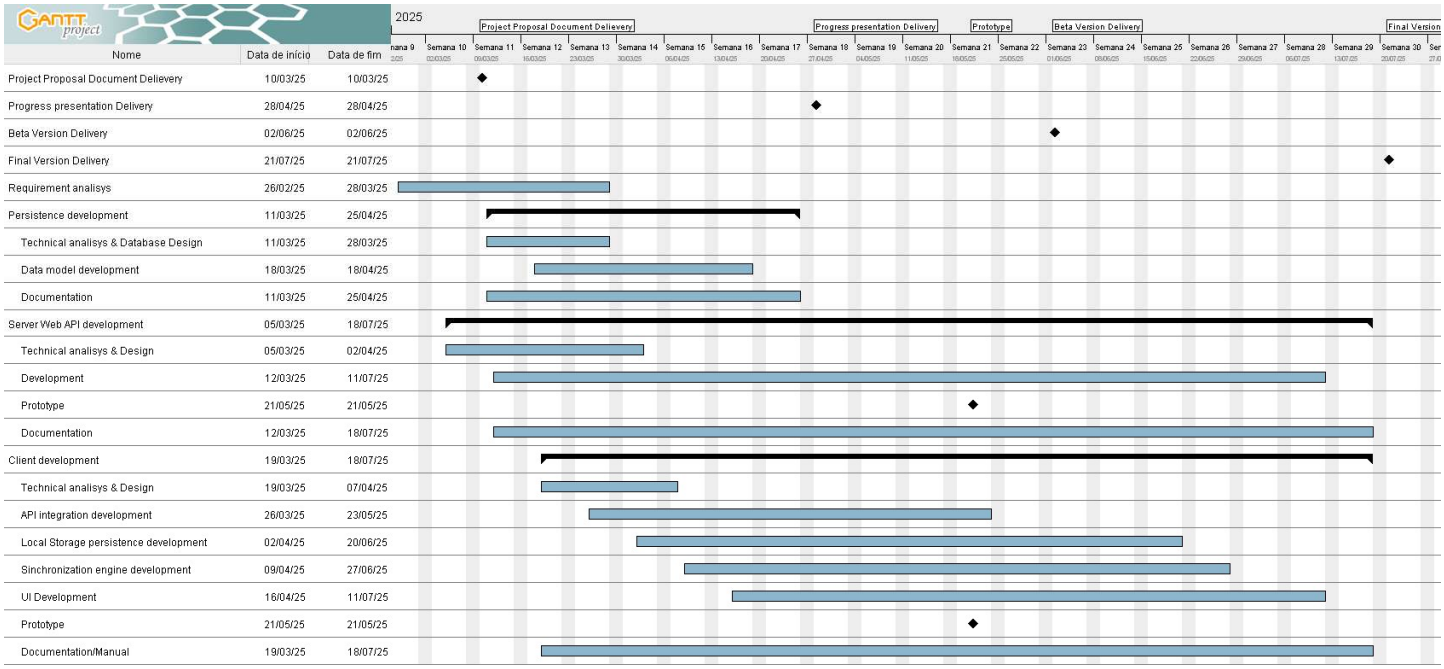

Figure 2 - Project planning Gantt chart.

The system development will be initiated by design the data base for a better understanding of the table relationships to help define which and how the backend endpoints should be as well as the frontend user interface forms. The tasks will be distributed evenly between both students along each stage to allow both students to experience the same challenges and enhance teamwork.

Expected task distribution plan:

- Luis Alves
  - Database – Entity relationship creation
  - Backend – API design and implementation
  - Frontend – User interface test and feedback, user guide
- Gonçalo Dimas
  - Database – Data storage logic Implementation
  - Backend – API testing and documentation
  - Frontend – UI Design and implementation