



TED UNIVERSITY

CMPE 491 SENIOR PROJECT I

SEARCH AND RESCUE OPERATION PORTAL (SAROP)

High-Level Design Report

Project Name: SAROP (Search and Rescue Operation Portal)

Project Url: sarop.tech

Team Members:

- Arda Gök (10016049214)
- Saliha Nursu Baltacı (48004738666)
- Ceren Özdoğan (13606132136)
- Mert Çıkla (33617203274)

Name of the supervisor: Emin Kuğu

Names of the juries: Tolga Kurtuluş Çapın - Venera Adanova

TABLE OF CONTENTS

1. Introduction	3
1.1 Purpose of the system	3
1.2 Design goals	3
1.3 Definitions, acronyms, and abbreviations	4
1.4 Overview	4
2. Current Software Architecture	5
3. Proposed Software Architecture	5
3.1 Overview	5
3.2 Subsystem decomposition	6
3.3 Hardware/software mapping	8
3.4 Persistent data management	8
3.4.1 Database management	9
3.4.2 Data storage policies	9
3.4.3 Mobile data synchronization	9
3.5 Access control and security	10
3.5.1 Authentication mechanisms	10
3.5.2 Authorization policies	10
3.5.3 Data encryption	10
3.6 Global software control	10
3.6.1 Request handling	11
3.6.2 Synchronization and concurrency	11
3.6.3 System communication	12
3.6.4 Global software control protocols	12
3.7 Boundary conditions	13
3.7.1 Initial behavior	13
3.7.2 Closing behavior	13
3.7.3 Error behavior	13
3.7.4 Emergency protocols	13
3.7.5 System management usage scenarios	14
4. Subsystem Services	14
4.1 Map subsystem	15
4.2 Operation subsystem	15
6. References	16

1. Introduction

Turkey is a country which lies in seismic belt. In recent years, this country experienced a lot of natural disasters. In all of these catastrophes, it is observed that search and rescue teams have some problems managing and organizing their groups. Because of that, this project will be designed as an open-source GIS application which will help search and rescue teams to manage their operations thanks to a map application specialized to their needs.

1.1 Purpose of the system

The project aims to minimize the loss which happened because of disasters. The first goal is to create a GIS application which is specialized to the needs of search and rescue teams so that they can operate in the field more effectively. They can upload their maps to the system and process it according to their needs. The second goal is to create a basic project planning tool for search and rescue teams so they can manage more than one operation at the same time and see their process.

1.2 Design goals

- The first design goal is to create a user-friendly application for the members. Then, it is aimed to create data privacy since the data of the users and some of the maps and operations are private to the foundation. That's why there will be an authentication system and the system will store the data with a hashing algorithm.
- Secondly, it will have a basic and understandable technical implementation because this project will serve as an open source for the sequence projects. Therefore, the project will be well documented, and it will be implementable easily on all kinds of systems.
- Also, the project will be designed as it can be used without internet because there might be some operations in a field which lacks good internet facilities.
- It will also be in communication with third party devices, especially with GPS devices to get the information about the user's location.

1.3 Definitions, acronyms, and abbreviations

- The application will be a specialized GIS application for search and rescue teams. GIS refers to a geographical information system which can analyze and edit spatial data and maps.
- It will use Geoserver to host the maps. Geoserver is an open-source map server which is implemented with Java.
- Leaflet will be used to display the map. Leaflet is a mobile friendly mapping library which is used in JavaScript.
- POSTGIS will be used as an extension of PostgreSQL to analyze and store the data in the database part.

1.4 Overview

In overall, the purpose of the project is to minimize the damage which is caused by natural disasters. The project will serve as an open-source GIS application with its operational features to increase the efficiency of teams in operations.

It will have an authentication system which will allow only the search and rescue team members to access the maps and operations. Once the users enter, they can enter a new operation, update it, and delete it. They can also view one of the current operations.

In addition to these features, users can view the maps, mark their places, and can specify a path and apply routing features on it.

At the technical side, Java will be used as backend language and Spring boot will be used as framework. At the front end, React will be used as framework. To process and analyze the spatial data PostgreSQL will be used with POSTGIS extension. Lastly, Geoserver will be used to host the maps while leaflet is used to display them.

2. Current Software Architecture

In the current software architecture, the foundations use more than one application for all of these features. As a GIS application they use SARTOPO to display and process their maps. To manage their operations, they use Notion, Trello, and Jira mostly. To provide their location information, they use some third-party devices which have Bluetooth and can store sim cards to store and share the location information. For instance, in our country Treyki is mostly used for this purpose. As you can see, they use different applications for each of the features and this decreases efficiency. Our goal is to merge all these features in one application and provide a more efficient solution for search and rescue teams.

3. Proposed Software Architecture

3.1 Overview

Users can access the SAROP platform through both a web application and a mobile application. Administrators can control users, manage maps, and efficiently oversee operations with the help of this web application. In contrast, the mobile application is made especially for field users and enables them to use mobile devices to access mapping subsystem features. These users can mark specific locations on the map and view their own locations.

SAROP's main goals are to make it easier for search and rescue teams to manage their operations on a map and to increase team productivity by adding new features.

3.2 Subsystem decomposition

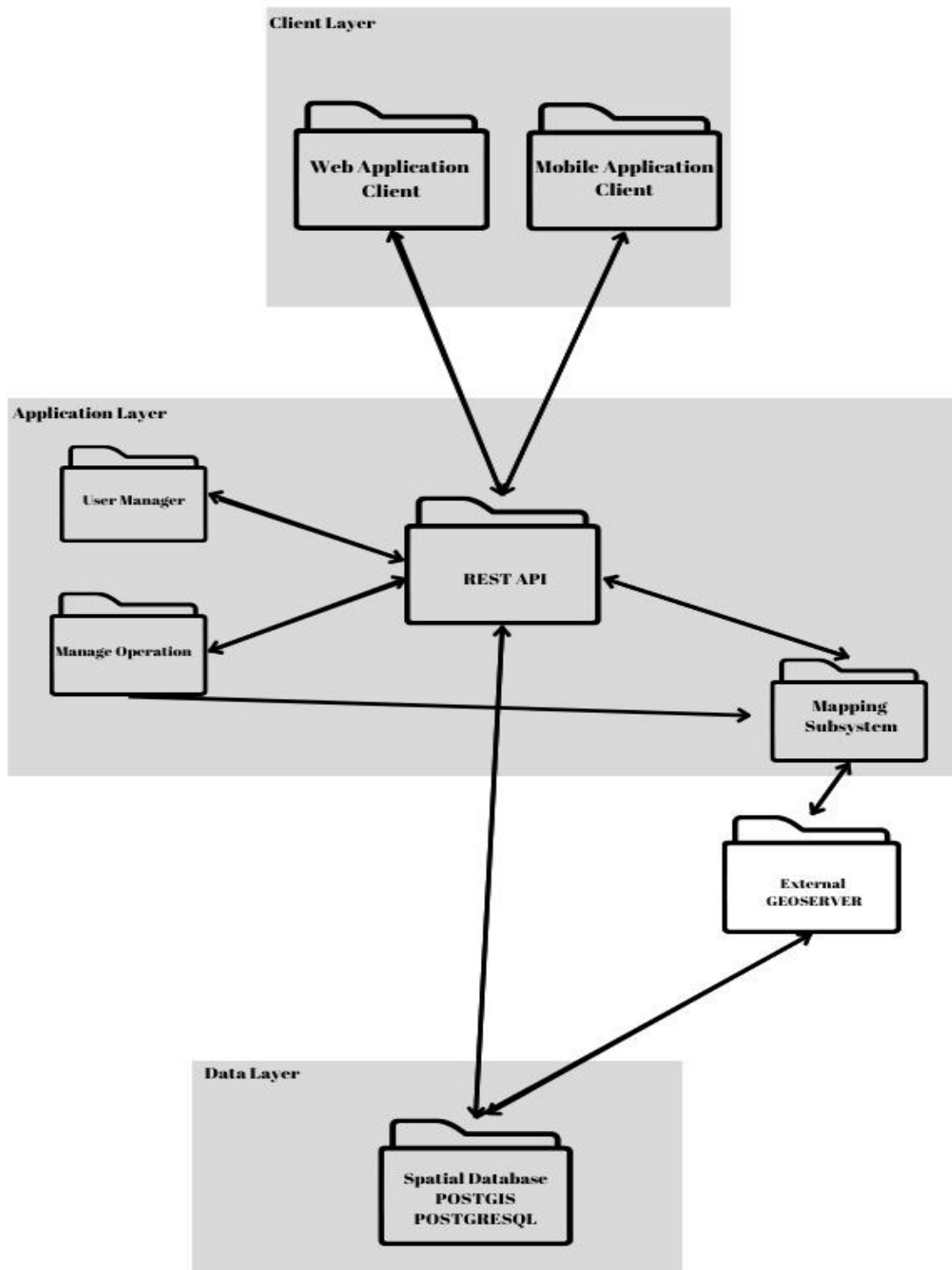


Figure 1: Subsystem Decomposition

- **Web application client:** Stands in for admin users when they use the program. The web app allows administrators to assign tasks to users and manage their operations. They can view, manipulate, and upload maps.
- **Mobile App Client:** Stands in for all users who use the application in general. On the mobile app, these users can view their location. Users can observe their routes. able to view other users. On the map, they can annotate.
- **REST API:** At the client layer, communicates with the clients of mobile and web applications. It uses the HTTP protocol to receive and handle requests. Also, Api supply, other subsystem and integrate with database server.
- **User Manager:** Carries out tasks related to user management. It oversees CRUD operations, or the creation, updating, and deletion of users.
- **Manage Operation Subsystem:** Oversees activities. It facilitates operations management for users.
- **Mapping Subsystem:** Manages the processing of geographic data and map operations. This system can communicate with this geodatabase and is linked to the Geoserver.
- **Geoserver:** Geoserver is a server for maps. Maps are also created and managed with its help. The Mapping Subsystem can retrieve and display geographic data by establishing a connection with this map server.
- **PostgreSQL spatial database:** It is the location where geographic information is kept. Location-based data can be managed and stored using PostgreSQL. In addition, we use it for operations management, user management, and the handling of admin and user data.

3.3 Hardware/software mapping

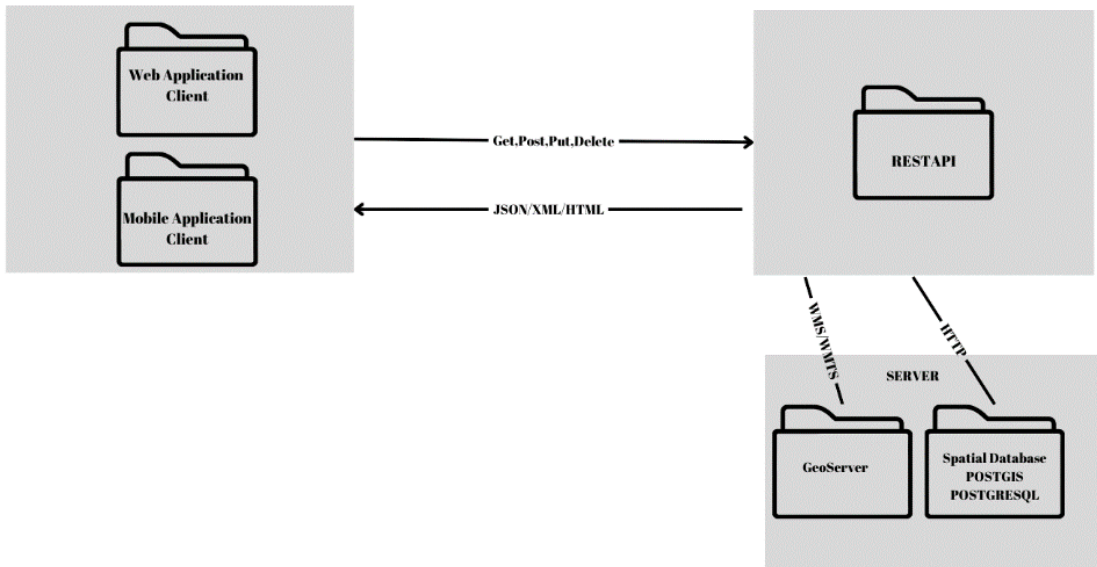


Figure 2: Hardware/Software Mapping

We are developing our server-side modules with the Java Spring Boot framework. With the addition of POSTGIS for geospatial functionality, PostgreSQL is the database of choice. A REST API is used by the web application and mobile clients to communicate with the server. The HTTP protocol is the only one used for client-server communication. Mobile phones are used by users to access the system, and an authenticated web browser is used for administrative tasks.

In essence, the HTTP protocol is used to facilitate communication between the server and clients. Furthermore, the WMS/WMTS protocols are used to establish communication between the API and Geoserver.

3.4 Persistent data management

The proposed software architecture incorporates a persistent data management subsystem designed to facilitate the effective storage and retrieval of data for search and rescue teams' operations.

3.4.1 Database management

The persistent data management subsystem will utilize a relational database management system (RDBMS) to efficiently store structured data. Determining the database schema, defining tables, relationships, and constraints is crucial to establishing a well-organized and normalized data structure.

3.4.2 Data storage policies

Policies governing data storage must be defined, encompassing data retention periods, archiving processes, and mechanisms for data cleansing or deletion. These policies should align with legal and regulatory requirements that govern data management.

3.4.3 Mobile data synchronization

To address the storage of field workers' locations in the database and ensure secure backup in scenarios without internet access, the following additional information can be included:

3.4.1.1 Mobile location backup

During the execution of tasks via the mobile application, location information designated by field workers should be regularly recorded in a local database by the application. This enables the secure storage of location data even when the user lacks internet access.

3.4.1.2 Offline mode and movement analysis

The mobile application should securely store location data from field workers even when there is no internet access. This functionality allows for the creation of a path representing the area where the worker has moved. Upon regaining internet access, the system can transfer this data to the central database, providing real-time **monitoring capabilities**.

3.4.1.3 Synchronization upon Internet access

The system should securely store location data from field workers, synchronizing this data with the central database when internet access is restored. The synchronization process ensures that the path depicting the user's

movement and other relevant activities is accurately displayed in real-time within the central database.

3.5 Access control and security

The access control and security subsystem define mechanisms to safeguard the integrity, confidentiality, and availability of the system's resources and data.

3.5.1 Authentication mechanisms

Authentication mechanisms for users should be specified, encompassing elements such as username/password combinations, multi-factor authentication, or biometric identity verification. User roles and privileges must be clearly defined.

3.5.2 Authorization policies

Access control policies should be specified, defining which actions users or user roles can perform within the system. Permissions, limits, and role-based access control (RBAC) techniques are all part of this.

3.5.3 Data encryption

Encryption protocols and algorithms used for data transmission and storage should be defined. To improve security, sensitive information is encrypted.

3.6 Global software control

Global software control in the SAROP project includes how user requests and application responds with data flow and the synchronization of both activities between the different components of the application. The following describes the aspects that illustrate how global control is implemented in the SAROP project.

3.6.1 Request handling

User request

- Users interact with the SAROP system primarily through a web interface or an application interface.
- This includes applications such as login entries, map interactions and operations or user management actions.
- When a user performs an action, a request is sent to a Rest API interface. The API interface in turn routes the request to an appropriate service.
- Requests can be synchronous (data retrieval, etc.) or asynchronous (background operations involving long processes, etc.).

System request

The system automatically initiates its own requests in certain situations:

- **Data Synchronization:** During times of internet disconnection, when a device comes back online, the system synchronizes data collected offline.
- **Regular Maintenance Operations:** The system is programmed to perform automatic backups or updates at specified times.

3.6.2 Synchronization and concurrency

- **Database transaction**

Database transactions use ACID (Atomicity, Consistency, Isolation, Durability) to maintain data integrity even when transactions can be performed by multiple users at the same time. This property means that each transaction is executed on its own and in isolation from other transactions.

- **Operation locking**

When a user, especially an admin, is working on an operation, they can enable the operation lockout feature to prevent others from editing it at the same time. Once the operation is complete, it will be made available again for other users to view and edit.

- **Session management**

The system manages the user's authentication status and session information to ensure continuity and security across different parts of the application.

3.6.3 System communication

- **API gateway:** Acts as a single-entry point for all customer communications, routes requests to the appropriate services and ensures that all subsystems are synchronized.
- **Service network:** Enables transparent and reliable communication between services, addressing network and security issues.

3.6.4 Global software control protocols

- **HTTPS:** For secure communication over the Internet, the system secures all data transmission by encrypting it.
- **RESTful API:** The RESTful API, a standard protocol for client-server communication, is used to facilitate development and maintenance.

3.6.5 Error management and recording

- **Centralized recording:** SAROP collects and evaluates error logs, enabling early detection of potential problems.
- **Exception management:** A standardized method for handling errors that occur at different layers of the system.

3.6.6 Concurrency Issues

- **Race conditions:** In situations requiring simultaneous operation, the system is equipped with appropriate locking mechanisms.
- **Deadlocks:** The system uses specific strategies and timeout policies to avoid deadlocks.

3.7 Boundary conditions

3.7.1 Initial behavior

- **Initialization:** When the system is initialized, the SAROP application core subsystems are initialized, including authentication, mapping, and operations management.
- **Self-Check:** The system performs a series of self-checks to ensure that all components are working correctly before being made available to users.

3.7.2 Closing behavior

- **Graceful termination:** Ensures that all ongoing processes are completed or safely undone before the application closes.
- **User sessions:** When Graceful Termination is complete, it effectively notifies connected users that a shutdown will occur and manages the termination of the user session.
- **Data integrity:** Ensures that no data is lost or left in an inconsistent state during the shutdown process.

3.7.3 Error behavior

- **Exception management:** A systematic approach to catching, logging, and recovering from exceptions.
- **User notification:** In the event of a critical error, the system provides feedback to the user, where appropriate, and logs the event for further investigation.
- **Failover mechanisms:** Implements redundant options for subsystems to maintain functionality in the face of partial system failures.

3.7.4 Emergency protocols

- Emergency protocols are in place for data breaches, hardware failures or other scenarios.
- Automated scenarios are in place to ensure that critical systems are quickly brought back online in a predetermined sequence.

- Continuous backup systems and data recovery procedures are regularly tested to guarantee minimal downtime and data loss in the event of a failure.

3.7.5 System management usage scenarios

- If new use cases related to system administration are defined, such as backup restore, maintenance mode activation or system health monitoring, they will be integrated into the detailed requirements and appropriate sections of the project documentation. Such use cases will not only be addressed technically, but also reflected in the user interface and access controls to ensure that only authorized personnel can perform system administration tasks.
- These global controls and boundary conditions are designed to ensure that the SAROP platform remains robust, responsive, and reliable, delivering the level of service required by critical search and rescue operations.

4. Subsystem Services

Subsystems include components of our system. Our system has two subsystem which will run in an environment on Linux. To simulate Linux Environment, we will use Docker Containers. The two subsystems are below.

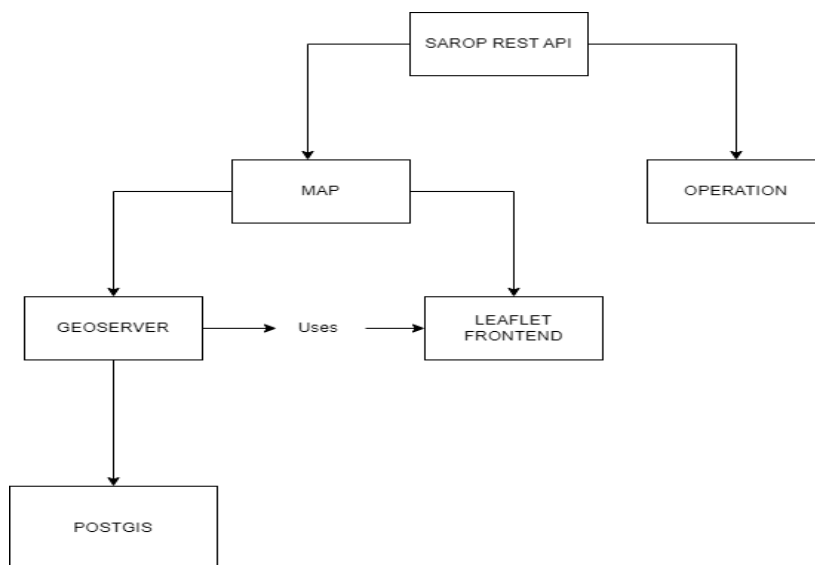


Figure 3: Subsystems General Overview

4.1 Map subsystem

In this subsystem, Geoserver will be used Geoserver as a map server to store and process the spatial data. Also, Leaflet will be used to add features and display the data. At the backend of the project Java will be used since it has some libraries such as GeoTools or GeoServer integration which will help us to implement GIS. Also, for the storage of the data, PostGIS will be used as an extension of PostgreSQL database.

4.2 Operation subsystem

In this subsystem, we will implement a basic calendar to-do list interface. Users can enter the operation information and manage it easily. We will use Java at the backend with the PostgreSQL database and we will use React at the frontend.

5. Glossary

- **Sartopo:** Sartopo is a map service for search and rescue teams all around the world. It uses a map server called “caltopo” to display high quality maps.
- **.ecw file extension:** It is called Enhanced Compression Wavelet. It is a photograph format which is used for the aerial mapping photos and satellite images.
- **Geoserver:** Geoserver is an open-source map server which is developed with Java. It allows users to display and edit maps.
- **GIS:** GIS is geographical information system which manages the spatial data.
- **POSTGIS:** POSTGIS is an extension of POSTGRESQL which makes database to store spatial data.
- **Openlayers:** Openlayers is an open-source Javascript library which allows us to display and edit maps.

6. References

- <https://sartopo.com/about/>
- [https://en.wikipedia.org/wiki/ECW_\(file_format\)](https://en.wikipedia.org/wiki/ECW_(file_format))
- <https://geoserver.org/about/>
- <https://www.esri.com/en-us/what-is-gis/overview>
- <https://postgis.net/>
- <https://github.com/openlayers/openlayers>