



TED UNIVERSITY

CMPE 491 SENIOR PROJECT I

SEARCH AND RESCUE OPERATION PORTAL (SAROP)

Project Analysis Report

Project Name: SAROP (Search and Rescue Operation Portal)

Project Url: sarop.tech

Team Members:

- Arda Gök
- Saliha Nursu Baltacı
- Ceren Özdoğan
- Mert Çıkla

Name of the supervisor: Emin Kuğu

Names of the juries: Tolga Kurtuluş Çapın - Venera Adanova

TABLE OF CONTENTS

1. Introduction	3
2. Current System	3
3. Proposed System	4
3.1 Overview.....	4
3.2 Functional Requirements.....	4
3.3 Non-functional Requirements.....	5
3.4 Pseudo Requirements	5
3.5 System Models.....	6
3.5.1 Scenarios.....	6
3.5.2 Use case model	12
3.5.3 Object and class model	13
3.5.4 Dynamic models.....	14
3.5.5 User interface - navigational paths and screen mock-ups .	16
4. Glossary	17
5. References	18

1. Introduction

In recent years, our country has witnessed a series of devastating disasters, and it has become evident that search and rescue teams face challenges in effectively managing their operations during these crises. In response, we are introducing the "Search and Rescue Operation Portal" (SAROP) project, which aims to provide a platform for search and rescue teams to enhance their operational efficiency.

The SAROP project, in its current state, will offer basic features that include an authentication system to limit access to registered search and rescue team members. Once inside the portal, users can enter new operations and list existing ones with basic details. In addition to that, it will be a mapping application which meets the needs of a search and rescue team.

The current planned features of SAROP include Operation Management, Operation Tracking, Route and Path Mapping, Team Management and Offline Capability.

It will be a general mapping application with a basic team management board to manage the operations.

The goal of SAROP is to minimize the damage caused by disasters and enhance the efficiency of search and rescue teams.

2. Current System

To apply this project, we met with a search and rescue team in our country and learned the current system they use. In the current system, they use different applications. For example, for mapping applications they generally use global mapper to convert their maps and different mapping apps such as Sartopo. To manage their project, they also use another third-party application. Also, for navigation they generally use devices to track the person who does the task. Our goal is to merge all these applications into one platform with this project.

3. Proposed System

3.1 Overview

In this project, we will implement a mapping application with a basic operation management feature for search and rescue teams. Basically, our application will show maps of the foundation and public maps for general usage. In the maps section, the user can search for a location, draw a path between two locations and share their location via the map. In the operation board, the admins can enter an operation, assign it to someone, add a description and add a deadline to it. At the end, our project will include two different pages which are mapping and operation. These pages will help the Search and Rescue teams to manage their operations easily.

3.2 Functional Requirements

REQ-1: People can mark their location and write notes on this location.

REQ-2: The program should show the path of people which they took.

REQ-3: There will be an authentication system for the members of the foundation.

REQ-4: The user can choose between different maps and coordinate systems.

REQ-5: The users can import and export maps in different formats.

REQ-6: There will be different roles and admins can add/delete users.

REQ-7: Admins can open an operation, edit the progress of it and close it. Users can only edit the progress of a task.

REQ-8: The information of closed operations can be exported as a word file.

REQ-9: The application can save the user's location when the user is offline and show it when user is online.

3.3 Non-functional Requirements

REQ-1: The application should work in the background.

REQ-2: The application should work when there is no internet. The user can use the downloaded maps in offline state.

REQ-3: Everyone can see the public maps without need of any role.

REQ-4: Users can share their location or operation information in third party applications such as WhatsApp.

REQ-5: The system will have the login with Google option for the users in the database.

3.4 Pseudo Requirements

REQ-1: The program should run on Linux as well as it can run in Windows.

REQ-2: The program should use an open-source map hosting server like Geoserver.

REQ-3: The program should let the upload of .ecw format for the maps since Search and Rescue teams use this format.

REQ-4: For map converting and storage, the program should use a map file converter or a database plugin for gis systems such as POSTGIS.

REQ-5: In order to display maps, the system should use the OpenLayers library.

REQ-6: In the front-end, React will be used.

3.5 System Models

3.5.1 Scenarios

Scenario Title: Admin authentication.

Objective: Admin logs in to the system.

Actors: Admin, System.

Precondition: Admin information must be included in the database.

Scenario:

- Admin writes username and password in the log in page.
- The information in the database is checked.
- System displays admin panel.

Scenario Title: Add new user.

Objective: Adding new users to system.

Actors: Admin, System.

Precondition: Admin must log in to admin panel.

Scenario:

- Admin logs in to the admin panel.
- Admin presses the “add new user” button.
- Admin enters information to create new users.
- Admin presses “save” button.
- System saves user information and displays “user created successfully” message.

Scenario Title: Edit user.

Objective: Editing users in the system.

Actors: Admin, System.

Precondition: Admin must log in to admin panel, user must be in the system.

Scenario:

- Admin logs in to the admin panel.
- Admin selects user to edit.

- Admin presses the “edit user” button.
- Admin enters information to edit users.
- Admin presses “update” button.
- System saves user information and displays “user updated successfully” message.

Scenario Title: Delete user.

Objective: Deleting users in the system.

Actors: Admin, System.

Precondition: Admin must log in to admin panel, user must be in the system.

Scenario:

- Admin logs in to the admin panel.
- Admin selects user to delete.
- Admin presses the “delete user” button.
- System displays “are you sure about delete this user?” message.
- Admin presses “yes” button.
- The system deletes user information and displays “user deleted successfully” message.

Scenario Title: Add operation.

Objective: Adding operation in the system.

Actors: Admin, System.

Precondition: Admin must log in to admin panel.

Scenario:

- Admin logs in to the admin panel.
- Admin presses “add operation” button.
- Admin enters necessary information.
- Admin presses “save” button.
- The system adds operation and displays “operation created successfully” message.

Scenario Title: Edit operation.

Objective: Editing operation in the system.

Actors: Admin, System.

Precondition: Admin must log in to admin panel, operation must be in the system.

Scenario:

- Admin logs in to the admin panel.
- Admin selects operation to edit.
- Admin presses the “edit operation” button.
- Admin changes operation information.
- Admin presses “save” button.
- The system updates operation information and displays “operation updated successfully” message.

Scenario Title: Delete operation.

Objective: Deleting operation in the system.

Actors: Admin, System.

Precondition: Admin must log in to admin panel, operation must be in the system.

Scenario:

- Admin logs in to the admin panel.
- Admin selects operation to delete.
- Admin presses the “delete operation” button.
- System displays “are you sure about delete this operation?” message.
- Admin presses “yes” button.
- The system deletes user information and displays “operation deleted successfully” message.

Scenario Title: Assign task.

Objective: Assigning task to user.

Actors: Admin, System, User.

Precondition: Admin must log in to admin panel, operation must be in the system, user must be in the system.

Scenario:

- Admin logs in to the admin panel.
- Admin selects operation to select task.

- Admin selects task to assign user.
- Admin selects users to assign.
- Admin presses the “assign task” button.
- System displays “are you sure about assign this task to user?” message.
- Admin presses “yes” button.
- The system assigns tasks to the user and displays “task assigned to user successfully” message.
- System displays to the user this task.

Scenario Title: User authentication.

Objective: User logs in to the system.

Actors: User, System.

Precondition: User information must be included in the database.

Scenario:

- User writes username and password in the log in page.
- The information in the database is checked.
- System displays user interface.

Scenario Title: User adds note.

Objective: User adds note to the task.

Actors: User, System.

Precondition: User must log in to admin panel, task must be in the system.

Scenario:

- User selects task to add note.
- User presses “add note” button.
- User enters note.
- User presses “save” button.
- The system updates operation information and displays “note saved to task successfully” message.

Scenario Title: User edit task.

Objective: User edit the task.

Actors: User, System.

Precondition: User must log in to admin panel, task must be in the system.

Scenario:

- User selects task to edit.
- User presses “edit task” button.
- User enters necessary information to edit.
- User presses “save” button.
- The system updates operation information and displays “task edited successfully” message.

Scenario Title: User chooses map.

Objective: User chooses map to display.

Actors: User, System.

Precondition: User must log in to admin panel, map must be in the system.

Scenario:

- User selects map to display.
- User presses “open map” button.
- The system displays a map.

Scenario Title: User chooses coordinate system.

Objective: User chooses coordinate system to show map.

Actors: User, System.

Precondition: User must log in to admin panel, map must be in the system.

Scenario:

- User selects map.
- User presses “select coordinate type” button.
- System displays coordinate types.
- User selects coordinate type.
- User presses “open map” button.
- The system displays a map according to the selected coordinate system.

Scenario Title: User marks location.

Objective: User marks current location on the map.

Actors: User, System.

Precondition: User must log in to admin panel, map must be in the system.

Scenario:

- User selects map.
- User presses “mark location” button.
- System displays a map.
- User selects current location.
- User presses “save location” button.
- The system saves users location to the system.

Scenario Title: User import map.

Objective: User imports map to the system.

Actors: User, System.

Precondition: User must log in to admin panel.

Scenario:

- User presses “import map” button.
- User selects map to import the system.
- User presses “save map” button.
- The system saves a map to the system.

Scenario Title: User export map.

Objective: User exports map to the system.

Actors: User, System.

Precondition: User must log in to admin panel, map must be in the system.

Scenario:

- User presses “export map” button.
- User presses “select format type” button.
- User selects format type to export map.
- User presses “export map to device” button.
- The system saves a map to the device selected format type.

3.5.2 Use case model

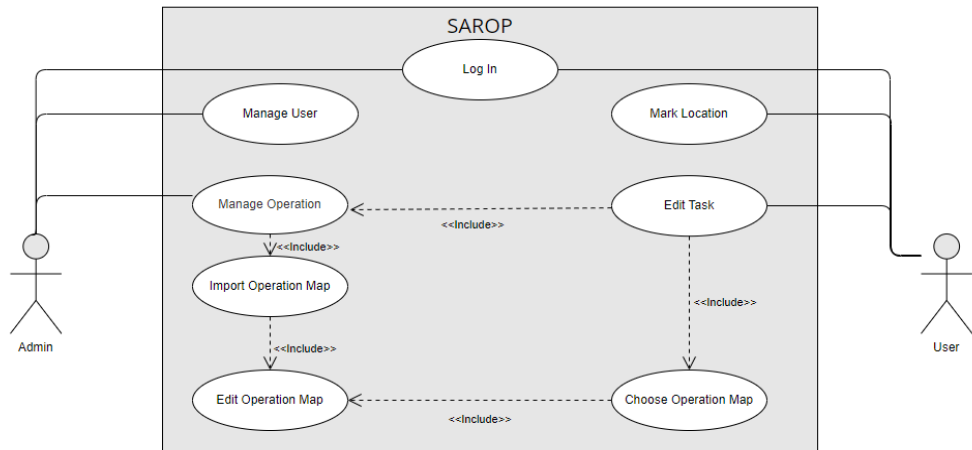


Figure 1: 1st Use Case Diagram

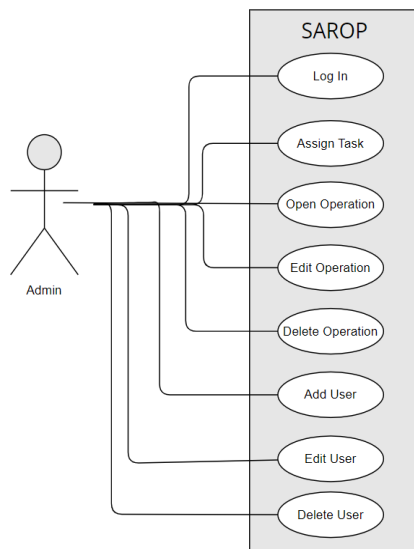


Figure 2: 2nd Use Case Diagram

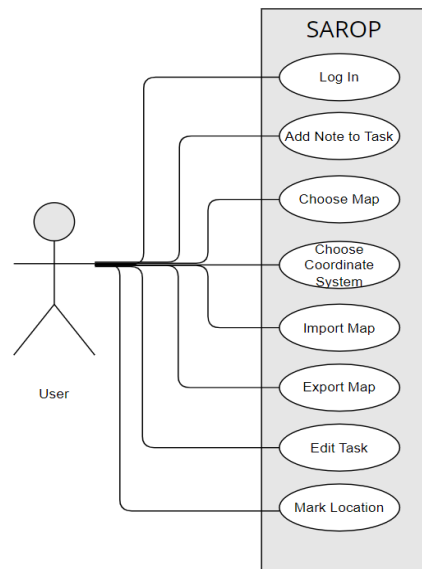


Figure 3: 3rd Use Case Diagram

3.5.3 Object and class model

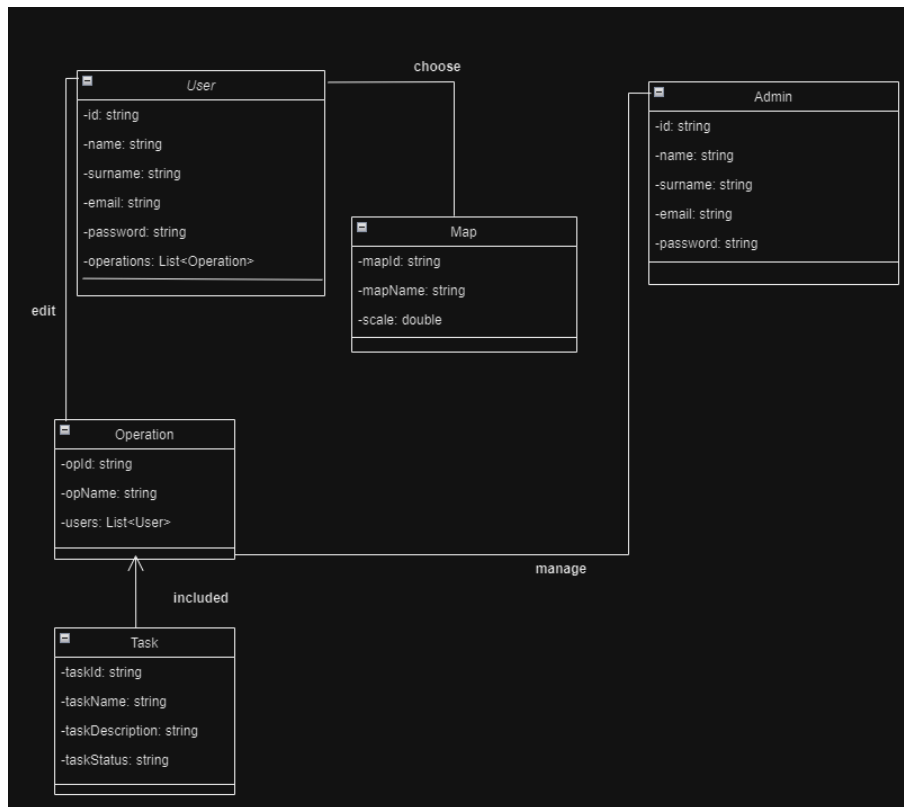


Figure 4: Object Diagram



Figure 5: Class Diagram

3.5.4 Dynamic models

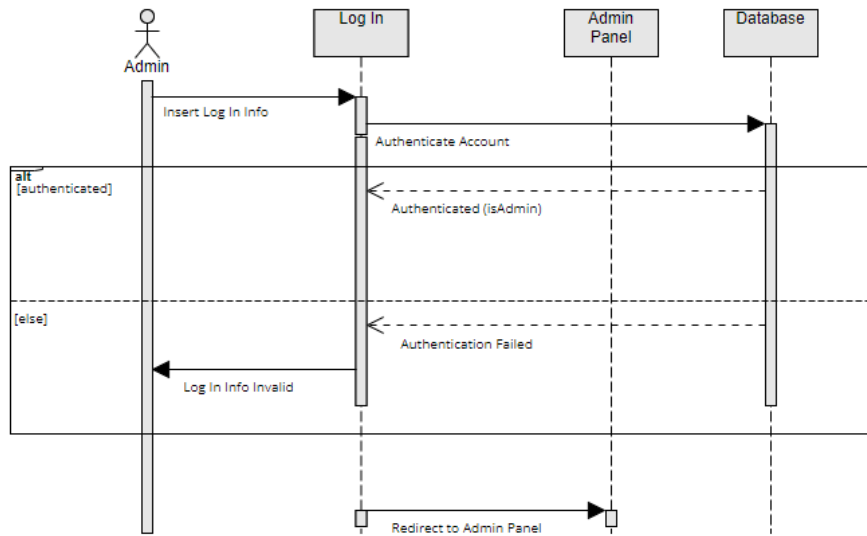


Figure 6: Admin Log In Sequence Diagram

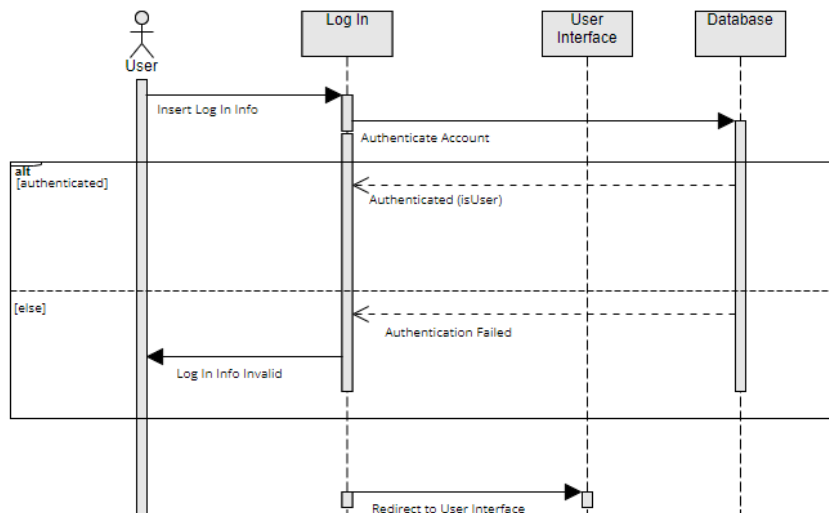


Figure 7: User Log In Sequence Diagram

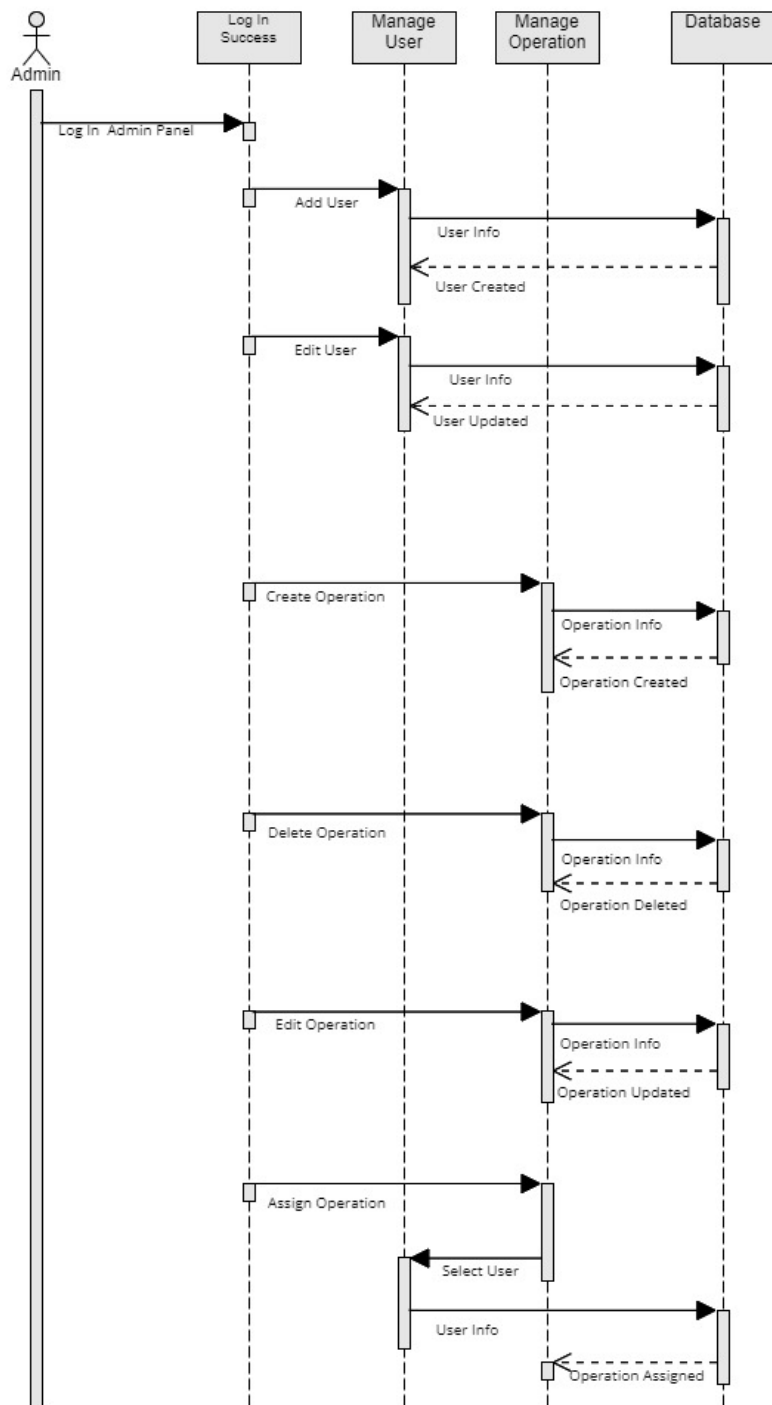


Figure 8: Admin Functions Sequence Diagram

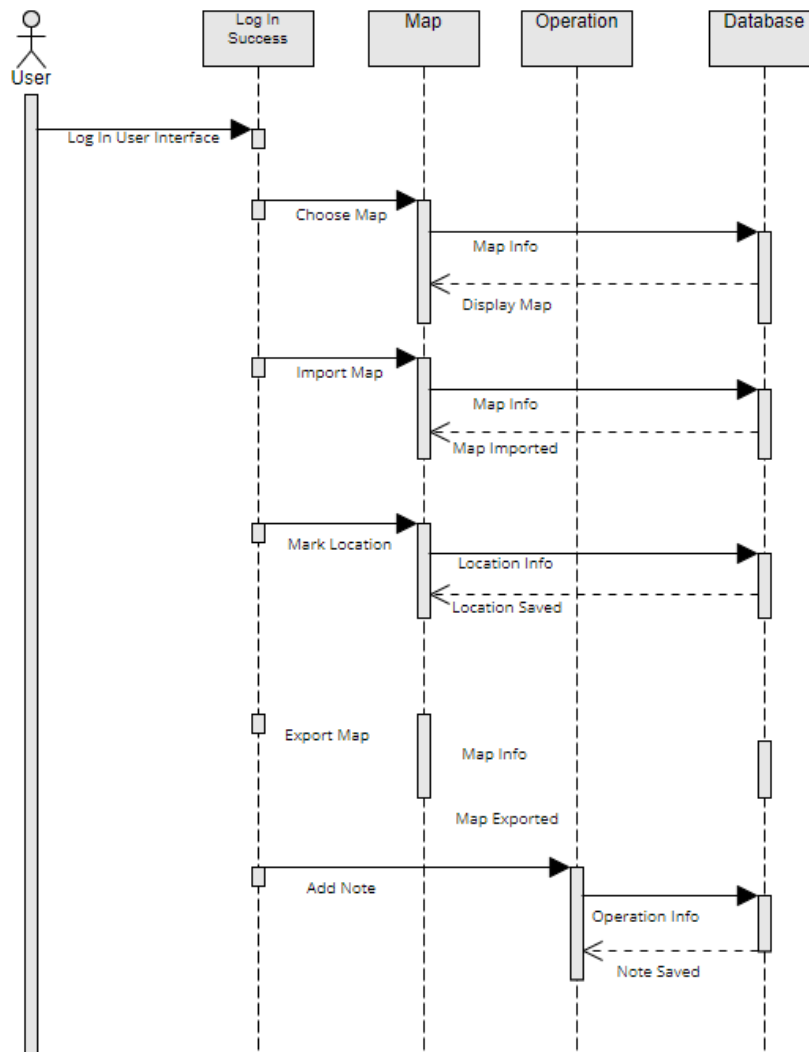


Figure 9: User Functions Sequence Diagram

3.5.5 User interface- navigational paths and screen mock-ups

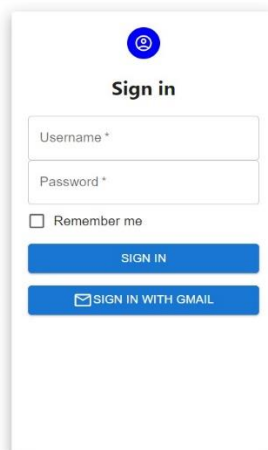


Figure 10: Log In Page

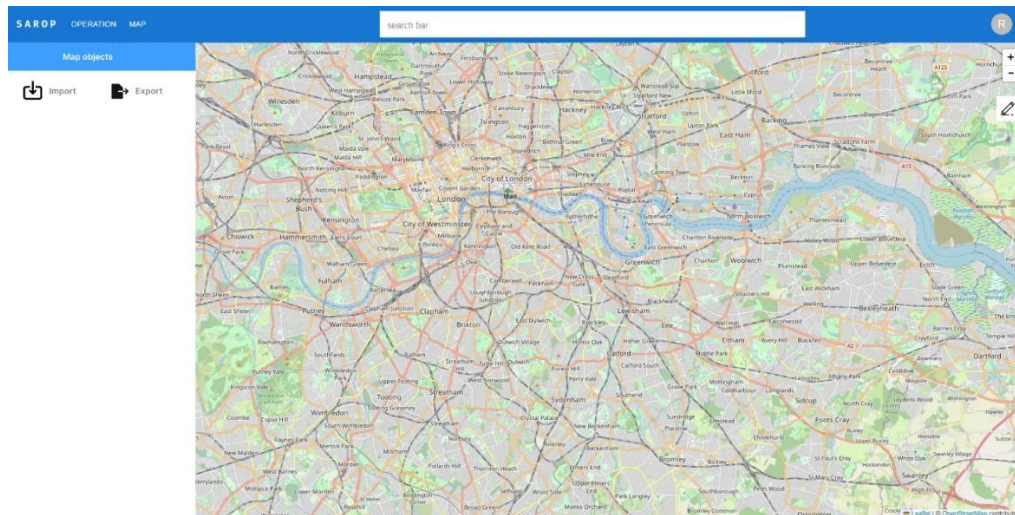


Figure 11: Main Page

Add a new operation

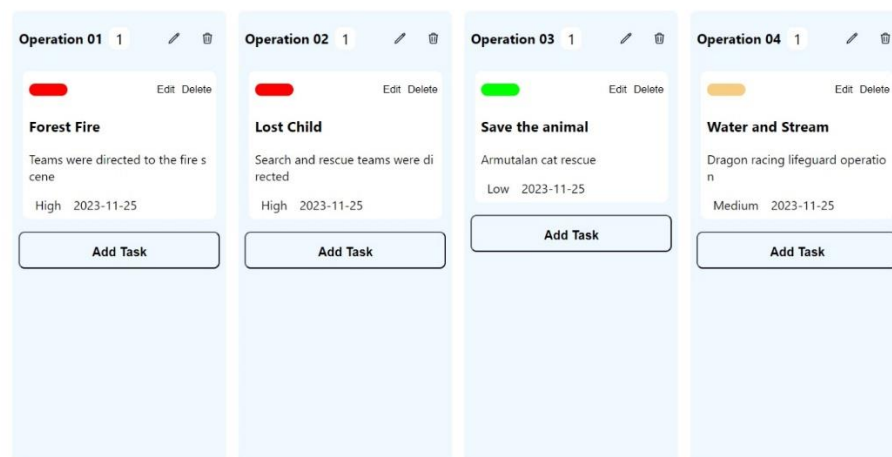


Figure 12: Operation Add Page

4. 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 as Enhanced Compression Wavelet. It is a photograph format which is used for the aerial mapping photos and satellite images.
- **Geoserver:** Geoserver is a 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.

5. 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>