*First Response Team Application*

*Capstone Project Phase B*

*by. Sharon Mor & Yuval Badihi*

*advisor. Dr. Naomi Unkelos Shpigel*

*Software Engineering Department*

*Project Code : 24-1-D-14*

*Project Domain:* [*https://www.frtproject.com*](https://www.frtproject.com/) *GitHub Phase A :* [*https://github.com/SharonMor/First-Response-Team*](https://github.com/SharonMor/First-Response-Team) *GitHub Phase B:* [*https://github.com/First-Response-Team*](https://github.com/First-Response-Team)

*Project Walkthrough:* [*FRT Project - Walkthrough*](https://www.youtube.com/watch?v=lYiZtuzUW_Y)



***Table Of Contents***

[*Abstract 2*](#_heading=h.30j0zll)

[*1. Introduction 3*](#_heading=h.1fob9te)

[*2. Literature Review 4*](#_heading=h.3znysh7)

[2.1 First Response Team (FRT) 4](#_heading=h.2et92p0)

[2.2 Community Emergency Response Teams (CERT) 4](#_heading=h.tyjcwt)

[2.3 Existing Application Introduction 4](#_heading=h.3dy6vkm)

[2.4.1 Key Differentiators 6](#_heading=h.1t3h5sf)

[2.5 Technological Solutions in Emergency Management 6](#_heading=h.4d34og8)

[2.6 User Experience (UX) in Emergency Applications 6](#_heading=h.2s8eyo1)

[*3. Engineering Process 6*](#_heading=h.17dp8vu)

[3.1 Development Process 6](#_heading=h.3rdcrjn)

[3.2 System Architecture Overview 7](#_heading=h.26in1rg)

[3.2.1 Technology Stack 7](#_heading=h.lnxbz9)

[*4. Work Artifacts 9*](#_heading=h.35nkun2)

[4.1 Use Case Diagram 9](#_heading=h.1ksv4uv)

[4.2 Working Flow Diagram 11](#_heading=h.44sinio)

[*5. User Guide 12*](#_heading=h.2jxsxqh)

[5.1 WEB 12](#_heading=h.z337ya)

[5.1.1 Homepage 12](#_heading=h.3j2qqm3)

[5.1.2 Navigation Bar (Logged in State) 12](#_heading=h.1y810tw)

[5.1.3 Login Page 13](#_heading=h.4i7ojhp)

[5.1.4 Teams Page 14](#_heading=h.2xcytpi)

[5.1.5 Export Team Data Modal 15](#_heading=h.1ci93xb)

[5.1.6 Create Team Modal 16](#_heading=h.3whwml4)

[5.1.7 Team Card 17](#_heading=h.2bn6wsx)

[5.1.8 Team's Control Panel 17](#_heading=h.qsh70q)

[5.1.9 Team Member Management Section 18](#_heading=h.3as4poj)

[5.1.10 User Details Dialog 19](#_heading=h.1pxezwc)

[5.1.11 Messenger Section 20](#_heading=h.49x2ik5)

[5.1.12 Map Interface 21](#_heading=h.2p2csry)

[5.1.13 Add Marker Modal 22](#_heading=h.147n2zr)

[5.1.14 Missions Section 24](#_heading=h.3o7alnk)

[5.1.15 Select Mission Type Dialog 25](#_heading=h.23ckvvd)

[5.1.16 Create Mission Dialog 26](#_heading=h.ihv636)

[5.1.17 Attendance Modal 27](#_heading=h.32hioqz)

[5.2 Mobile 29](#_heading=h.1hmsyys)

[5.2.1 Telegram Bot Interaction 29](#_heading=h.41mghml)

[5.2.2 Team Selection in Telegram Bot 29](#_heading=h.2grqrue)

[5.2.3 Default Location Request in Telegram Bot 30](#_heading=h.vx1227)

[*6. Maintenance Guide 30*](#_heading=h.3fwokq0)

[6.1 Installation 30](#_heading=h.1v1yuxt)

[6.1.1 Prerequisites 30](#_heading=h.4f1mdlm)

[6.1.2 Setup 30](#_heading=h.2u6wntf)

[6.2 Running the application locally 31](#_heading=h.19c6y18)

[6.3 Deploying the application 31](#_heading=h.3tbugp1)

[6.4 Database 32](#_heading=h.28h4qwu)

[6.4.1 Maps Collection: 32](#_heading=h.nmf14n)

[6.4.2 Missions Collection 33](#_heading=h.37m2jsg)

[6.4.3 Teams Collection 33](#_heading=h.1mrcu09)

[6.4.4 Telegram Users 33](#_heading=h.46r0co2)

[6.4.5 Users: 34](#_heading=h.2lwamvv)

[6.5 Kafka 34](#_heading=h.111kx3o)

[6.5.1 map\_topic 34](#_heading=h.3l18frh)

[6.5.2 mission\_topic 35](#_heading=h.206ipza)

[6.5.3 message\_stream\_topic 35](#_heading=h.4k668n3)

[*7. Evaluation 36*](#_heading=h.cj97w2em8s05)

[7.1 Reflection - Interview with Mr. Amir Dagan - Kibbutz Megiddo Health Cert Team Leader 36](#_heading=h.9ntm1jgtrdyg)

[7.1.1 Key Observations and Feedback 36](#_heading=h.1ffx4xdqt3wn)

[7.1.2 Real-World Use Case Validation 36](#_heading=h.w29i8n38p50q)

[7.1.3 Areas for Enhancement 36](#_heading=h.ib0holpopptt)

[7.2 SUS Survey 37](#_heading=h.3ndlnp17xxxv)

[7.3 Evaluation Summary 39](#_heading=h.jwxhuj3g2cnb)

[*8. Conclusion and Summary 40*](#_heading=h.2zbgiuw)

[8.1 Key Findings and Achievements 40](#_heading=h.1egqt2p)

[8.2 Challenges and Solutions 40](#_heading=h.3ygebqi)

[8.3 Implications and Future Research 41](#_heading=h.2dlolyb)

[8.4 Personal Learning Outcomes 41](#_heading=h.sqyw64)

[8.5 Final Thoughts 42](#_heading=h.3cqmetx)

[*9. References 42*](#_heading=h.1rvwp1q)

### Abstract

Currently, first response and emergency resilience teams face significant challenges in managing their operations efficiently due to the lack of a dedicated tool that integrates all aspects of their work. In many cases, coordination and communication are handled through basic, non-specialized platforms. This results in fragmented information flows and suboptimal decision-making, as these platforms do not offer features tailored to the unique needs of emergency management. Without a centralized system to manage the multitude of tasks, notifications, and real-time data, teams struggle to respond with the agility and precision required in crisis situations. This gap severely impacts their ability to effectively manage emergencies, from minor incidents to large-scale disasters.

This book describes an advanced web and mobile app solution that optimizes communication and operational efficiency for first response and emergency resilience teams, especially in high-risk and recently impacted areas. The tool supports real-time strategic decision-making and comprehensive emergency management. The book details the technical design, diagrams, and research underpinning this tool, including system architecture frameworks. It incorporates insights from interviews with first responders to shape features and functionality. The application includes real-time communication tools, mission management capabilities, interactive mapping systems, and cross-platform integration between web and mobile interfaces, providing teams with the comprehensive toolset needed for effective emergency response.

### 1. Introduction

Today's world is plagued by crises and emergencies that can strike at any moment, posing significant challenges for first responders and resilience teams. Effective emergency management systems are crucial, yet existing tools and processes often fall short, leading to communication breakdowns, delays, and inefficiencies that can have devastating consequences. Critical incidents like the events of 7.10.2023 in Israel, have exposed glaring shortcomings in emergency coordination efforts, highlighting the urgent need for improved systems that enable swift, precise, and reliable communication and coordination.

First response teams are often the first line of defense in emergency situations, tasked with managing crises efficiently to minimize damage and save lives. However, the effectiveness of these teams can be significantly hampered by outdated tools and fragmented systems. Studies have shown that integrating modern technology and improving inter-agency communication can greatly enhance the operational capabilities of first responders[3]. By leveraging cutting-edge solutions, these teams can ensure a more coordinated and effective approach, reducing response times and increasing the likelihood of positive outcomes.

Extensive dialogue with responders and teams in high-risk areas, such as the Golan settlements, has revealed that their current tools are inadequate for emergencies that require seamless information sharing, efficient task assignment, accurate location tracking and comprehensive incident documentation. Teams often struggle with a lack of real-time situational awareness, communication barriers between different agencies, and difficulties in allocating resources effectively.

To address these critical issues, we developed an innovative application that serves as a centralized platform for seamless coordination and decision-making. Drawing insights from leading practices on organizing effective emergency teams [8] and enhancing the capabilities of disaster responders [7], our application provides a comprehensive suite of features designed to streamline communication, improve situational awareness, and facilitate efficient task management.

The application offers features like task assignment, real-time location tracking and chat, hazard marking systems, and comprehensive task management tools. The system archives incident data and implements well-established emergency response principles.

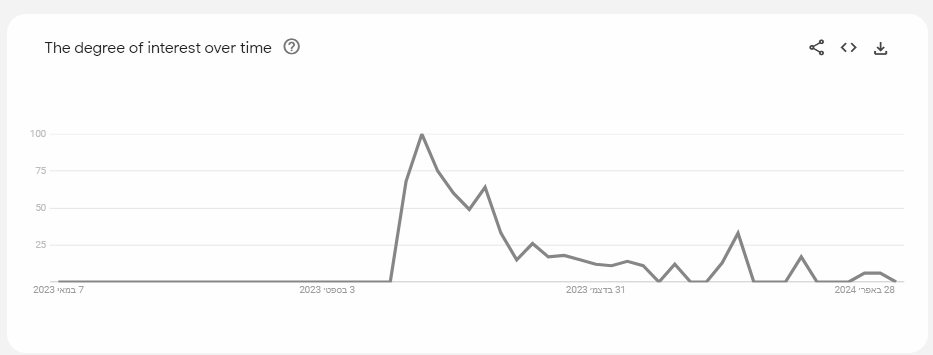
The application improves team preparedness, equips responders with vital capabilities, and overcomes communication barriers in chaotic situations. Through its accessible interface with advanced features, it simplifies complexities and boosts team confidence in crisis response operations.

The rest of the book details the application's technological implementation, features, and evaluation results, using case studies and expert insights to demonstrate its effectiveness in emergency management scenarios.

### 2. Literature Review

#### 2.1 First Response Team (FRT)

Group of trained individuals, including professionals and volunteers, who provide immediate assistance during emergency situations. They act as the first line of defense, delivering urgent aid and coordination until additional support arrives [5].

****

*fig 1. ‘First Response Teams’ on google trends [19]*

The degree of interest in 'First Response Team' over time in Hebrew in Israel has risen significantly over the last year, following the events of October 7, according to Google Trends.

#### 2.2 Community Emergency Response Teams (CERT)

Trained civilian volunteers who assist professional responders during emergencies across sectors like health, education, and logistics. The CERT program teaches basic skills like fire safety, light rescue, first aid, and volunteer management. CERT members provide immediate support in these areas until responders arrive, enhancing a community's resilience and response capabilities in crises [5].

#### 2.3 Existing Application Introduction

We conducted research on tools designed to enhance the capabilities of first response and emergency resilience teams. Our focus was on evaluating existing platforms that aim to tackle the operational and communication challenges these teams encounter in emergency situations. Here are some notable examples:

| Home | Avuka-Squad System | **Avuka Squad:** System suite was designed and developed to supply a comprehensive location based C2 **(\*)** (Command and Control) solution for organizations with complex security needs [13]. |
| --- | --- |
| לנצח את האירוע עם טכנולוגיה מתקדמת - AVIA SECURITY | **AVIA:** System operates as an emergency population management system that enables control over complex security and civilian events [14]. |
| ‪Zello PTT Walkie Talkie – At the ECOM App Library | Get it here‬‏ | **Zello:** Modern and customizable walkie-talkie app (and more) that lives on any smart device [20]. |

***\* -*** [*https://intelligence.airbus.com/industries/defence/c2*](https://intelligence.airbus.com/industries/defence/c2)

**2.4 Existing Application Comparison**

| **FEATURE** | **Zello** | **AVIA** | **Avuka Squad** | **FRT APP (US)** |
| --- | --- | --- | --- | --- |
| IOS Support | ✓ | ✓ |  | ✓ |
| Android Support | ✓ | ✓ | ✓ | ✓ |
| Web/PC Support | ✓ | ✓ |  | ✓ |
| Management App | ✓ |  | ✓ | ✓ |
| Resident's App | ✓ | ✓ |  | ✓ |
| Wellness Check | ✓ |  |  | ✓ |
| Google Maps | ✓ |  | ✓ | ✓ |
| Dynamic Map | ✓ |  | ✓ | ✓ |
| GPS Tracking | ✓ |  | ✓ | ✓ |
| Chat | ✓ | ✓ | ✓ | ✓ |
| Alert For FRT Activation | ✓ | ✓ | ✓ | ✓ |
| Predefining Possible Emergency Events | ✓ |  | ✓ | ✓ |
| Defining Groups | ✓ | ✓ | ✓ | ✓ |
| Operation Logs | ✓ | ✓ | ✓ | ✓ |
| Message Board | ✓ |  | ✓ | ✓ |
| Forces Count | ✓ |  |  | ✓ |
| Performance Monitoring |  |  |  | ✓ |
| Smooth Registration |  |  |  | ✓ |

*Table 1. existing tools examination*

This table compares three emergency management applications—Avuka Squad, AVIA, and Zello—highlighting their key features to support first response teams. It outlines support for iOS and Android, web/PC compatibility, and specialized functionalities like GPS tracking and emergency services integration, helping to identify the best fit for specific emergency response needs.

The existing tools provide some features for first response teams, but lack a comprehensive, integrated solution that meets all their diverse needs.

To address this gap, the proposed First Response Team application aims to develop a unified platform that combines vital features like communication tools, mapping, task assignment, and real-time coordination in a user-friendly interface.

##### 2.4.1 Key Differentiators

* Simplicity for use in high-stress situations.
* Features tailored based on extensive interviews with first responders.
* Leveraging modern web and mobile technologies for scalability and accessibility.
* Advanced data handling, search capabilities, and secure authentication.

By offering a complete toolset streamlining communication, situational awareness, and decision-making, this application promises to revolutionize emergency management practices.

#### 2.5 Technological Solutions in Emergency Management

Studies on technological solutions for emergency management highlight the increasing role of advanced technologies in enhancing the capabilities of first response teams.

* **Real-Time Data Processing:** The capacity to process data in real-time is critical in emergency management, enabling responders to make informed decisions swiftly. The development of systems that manage data influx during crises helps mitigate cognitive overload on human operators by efficiently filtering and presenting only the most relevant information [1].
* **Location Systems:** Geographic Information Systems (GIS) play a pivotal role in disaster and emergency management by providing tools for mapping, resource tracking, and strategic planning. These technologies improve situational awareness and coordination during operations across various government levels [6].

#### 2.6 User Experience (UX) in Emergency Applications

**User-Centered Design in Emergency Systems:** Implementing user-centered design principles is crucial in the development of emergency management systems. Prototyping user interfaces based on these principles can significantly enhance the functionality of intelligent emergency management systems, making them more intuitive and effective during crisis situations [4].

### 3. Engineering Process

#### 3.1 Development Process

In developing the advanced web and mobile application for optimizing the operations of first response and emergency resilience teams, our engineering process was driven by the need for precision, reliability, and user-centric functionality.   
This section outlines the methodologies and practices we employed to construct a system that not only meets the technical requirements of robust software architecture but also addresses the practical demands of emergency response scenarios. We detail our approach to integrating modern technologies with structured development phases, from initial design through to testing and deployment, ensuring that the application is both scalable and adaptable. By focusing on a user-oriented development cycle, we aim to deliver a solution that enhances situational awareness and facilitates real-time decision-making for first responders operating in diverse and challenging environments.

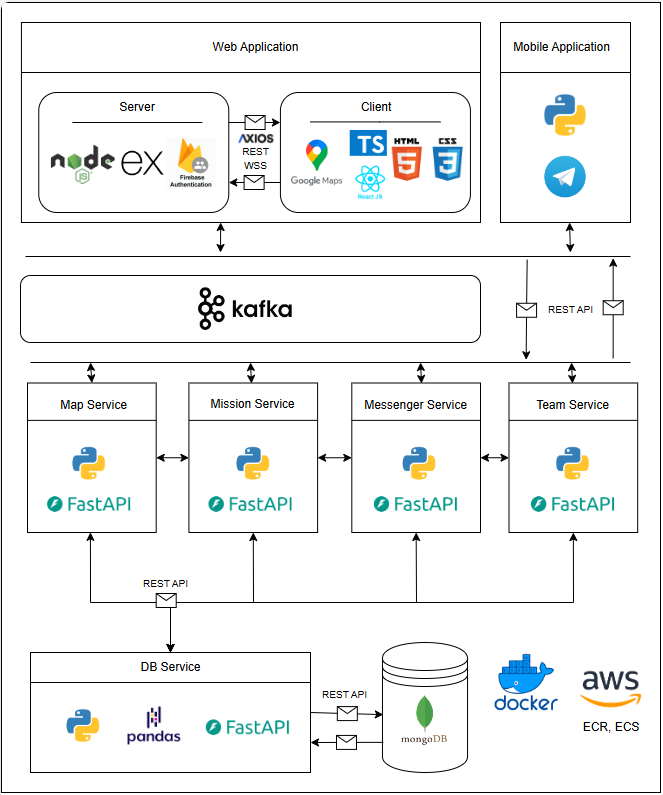
Our workflow begins with an extensive review of existing emergency response applications, case studies, and trends to deeply understand the unique challenges faced by first response teams. We conduct interviews with stakeholders from various settlements to gather insights on their specific needs and pain points.  
In the design phase, we architect a comprehensive system integrating web and mobile applications. We sketch wireframes for intuitive interfaces that cater to different user roles and permissions. Precise UML diagrams capture the application's features like communication tools, mapping, task management, and real-time coordination.  
Refining the project's scope is an iterative process, using feedback from interviews and data analysis on first responder operations. This helps set clear requirements for seamless team collaboration, data synchronization, offline functionality, and a user-friendly experience during high-stress situations.   
The final stages involve a robust testing strategy with unit tests, integration tests, usability checks and load simulations to validate functionality, performance, and security across web and mobile platforms. Tight coding practices ensure a reliable and effective application tailored for efficient emergency management by first response teams.

#### 3.2 System Architecture Overview

Our architecture is a hybrid model that merges microservices and distributed services to enhance the integration and functionality of web and mobile interactions. This design is aimed at delivering a cohesive experience across diverse platforms while maintaining robust scalability and flexibility.

##### 3.2.1 Technology Stack

* **Clients:**
* **Web:**The web application leverages React.js along with TypeScript, HTML5, CSS3, and Google Maps API for building a responsive, performant, and interactive user interface.
* **Telegram Bot:**  
  Replaces the mobile application, providing a user interface through Telegram's platform for seamless communication and mission management.
* **Servers:**
  + **BFF:**powered by Node.js and Express.js, with Firebase Authentication integrated for secure user management. This server acts as the API gateway, handling communication between the client and backend services using REST API, AXIOS, and WebSocket (WSS).  
    This server produces/consumes messages from Kafka and delivers them right back to the client via secure web socket
  + **Python Microservices:**The system is composed of various microservices, each built using FastAPI and written in Python
* **Map Service:** Handles map-related operations such as adding or removing points of interest.
* **Mission Service:** Manages team missions.
* **Messenger Service:** Facilitates communication between team members.
* **Team Service:** Manages team-related data and operations.
* **Telegram Service**: Manages telegram data and serves the telegram client.
* **DB Service:** A service layer managing database operations for all other services. Pandas is used within this service to handle data analytics and manipulation tasks.
* **Messaging System:** Apache Kafka is utilized for message brokering to ensure the system's scalability and maintain loose coupling between various microservices.
* **Containerization & Deployment:** 
  + - **Docker** is used for containerizing all services.
    - **Amazon ECR** (Elastic Container Registry) for storing Docker images.
    - **Amazon ECS** (Elastic Container Service) for running containerized applications.
    - **Amazon ELB** (Elastic Load Balancing) for distributing incoming application traffic across multiple targets.
    - **Amazon Route 53** for [domain](https://www.frtproject.com/) management and DNS routing.

**

*fig 2. system architecture diagram*

### 4. Work Artifacts

#### 4.1 Use Case Diagram

* **Unregistered User:** An anonymous individual who is not recognized by our database. They have not registered or authenticated with our system.
* **Registered User**: This user is recognized and authenticated by our system, known through their verified login credentials.
* **Team Manager**: A user who created a team, can remove users and delete the team.

*A diagram of a team

Description automatically generated  
fig 3. Use Case diagram*

#### 4.2 Working Flow Diagram

****

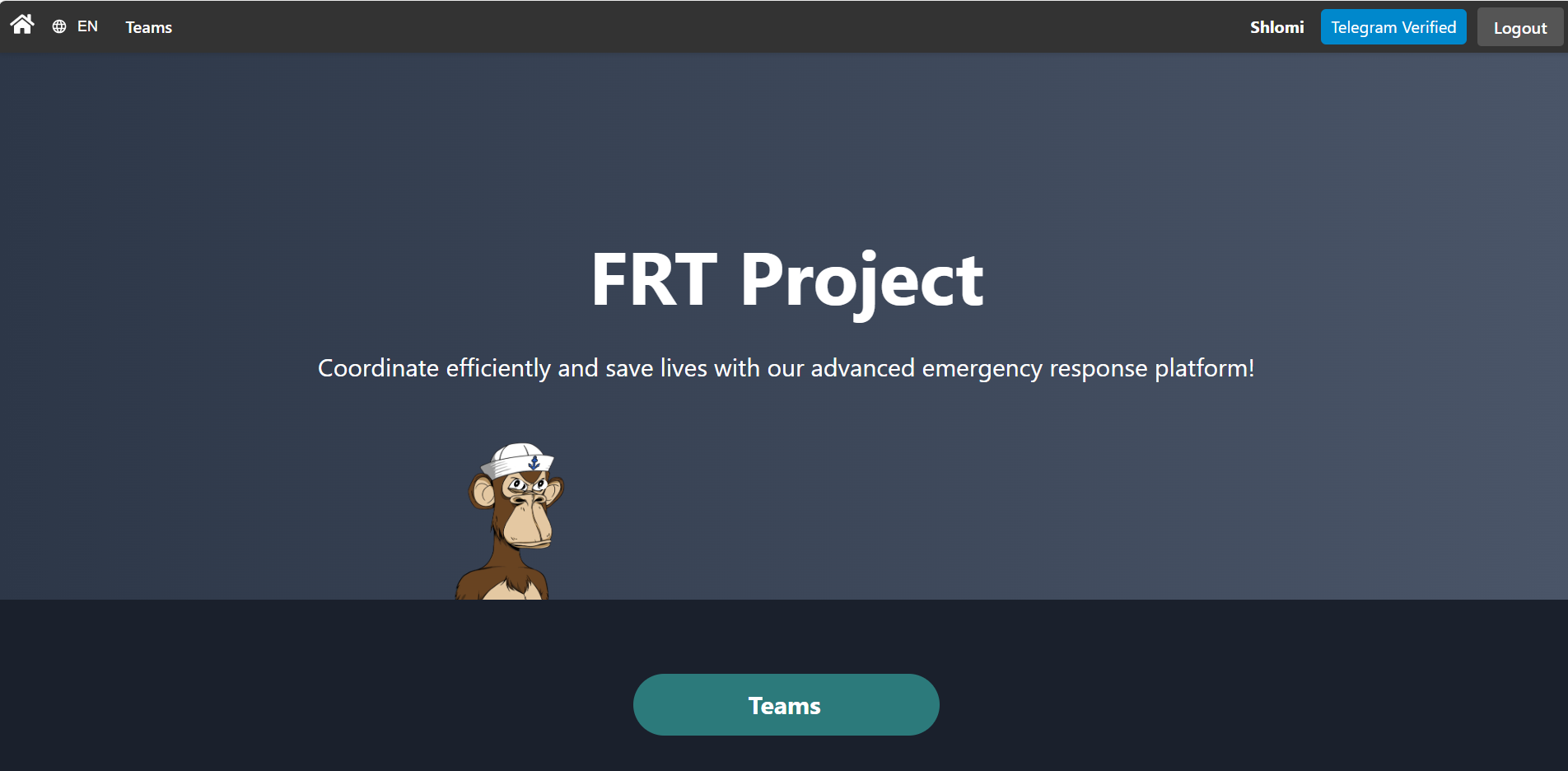
*fig 4. Working Flow diagram*

### 5. User Guide

#### 5.1 WEB

##### 5.1.1 Homepage

* **Purpose:** This is the main landing page of your website. It introduces the FRT Project with a clear message about its purpose, which is to coordinate emergency response teams efficiently and save lives. The page has a simple, clean design with a prominent call to action.
* **Key Elements**
  + **"Teams"** Button: A button that likely directs users to the "Teams" page to view or manage teams.
  + **Login Button:** Located at the top right, it prompts users to log in for full access to the platform.

****

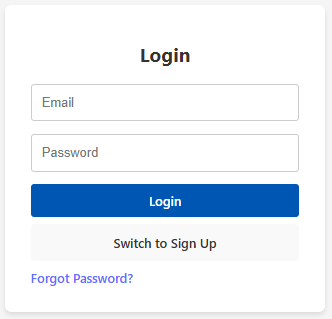
##### 5.1.2 Navigation Bar (Logged in State)

* **Purpose:** Once logged in, users can see their status and have access to additional features.
* **Key Elements**
  + **Home Icon:** Returns the user to the homepage.
  + **Language Selection (EN):** Allows the user to switch between languages.
  + **Teams:** A link to the team's section where users can view or manage teams.
  + **Username (Shlomi):** Displays the currently logged-in user’s name.
  + **Telegram Verified:** This button shows that the user has been connected to Telegram account.
  + **Logout Button:** Clicking this will log the user out of the system.

****

##### 5.1.3 Login Page

* **Purpose:** The login page is where users can access their accounts by providing their Email and Password.
* **Key Elements:**
  + **Email and Password Fields:** Users must input their registered email address and password to log in.
  + **Login Button:** Once the credentials are entered, pressing this button will log the user in.
  + **Switch to Sign Up**: This link allows users to switch to the sign-up page if they don't have an account.
  + **Forgot Password:** If a user forgets their password, they can click this link to initiate the password recovery process.

****

##### 5.1.4 Teams Page

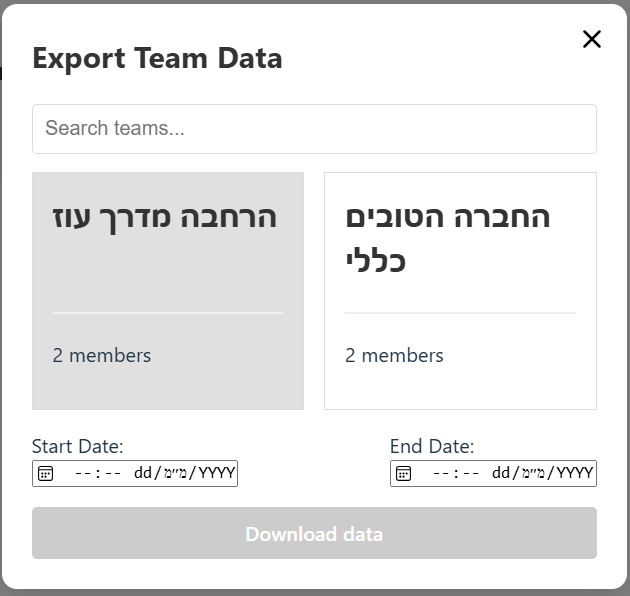
* **Purpose:** The Teams page allows users to view and manage the teams they are a part of or have created. This is a central location for team coordination and collaboration.
* **Key Elements:**
  + **Your Teams:** This section lists the user's teams. Each team has its own card for easy identification.
  + **Create New Team Button:** On the right-hand side of the page, there's a prominent button labeled "Create New Team", allowing users to create a new team quickly.
  + **Refresh Icon:** Located near the "Create New Team" button, this icon allows users to refresh the page, updating any new teams or changes made to their teams.
  + **Export Icon:** Located near the "Create New Team" button, this icon allows users to export team data.

תמונה שמכילה טקסט, צילום מסך, גופן, עיצוב

התיאור נוצר באופן אוטומטי

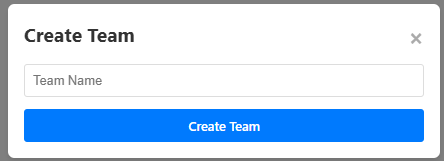
##### 5.1.5 Export Team Data Modal

* **Purpose:** This dialog allows users to export team data (such as chat, timeline, and actions) into an Excel file for record-keeping or analysis.
* **Key Elements:**
  + **Team Selection:** Users can search for and select the team they want to export data for. In this case, two teams are displayed: הרחבה מדרך עוז and החברה הטובים כללי.
  + **Start Date and End Date**: Allows users to define a specific date range for the data export, ensuring only relevant data within that timeframe is included.
  + **Download Data Button:** After selecting a team and defining the date range, users can click this button to download the team data in Excel format.



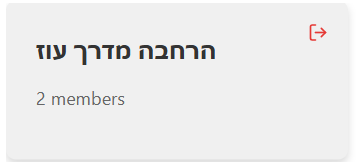
##### 5.1.6 Create Team Modal

* **Purpose:** This modal allows users to create a new team by entering a name for the team.
* **Key Elements:**
  + **Team Name Field:** A simple input box where users can type the desired name for their new team.
  + **Create Team Button:** Once the user has entered a name, they can click this button to create the team. The button is visually prominent in blue, making it clear where to click to complete the action.
  + **Close Icon (X):** Located in the top-right corner, this allows the user to exit the dialog if they decide not to create a team at that moment.



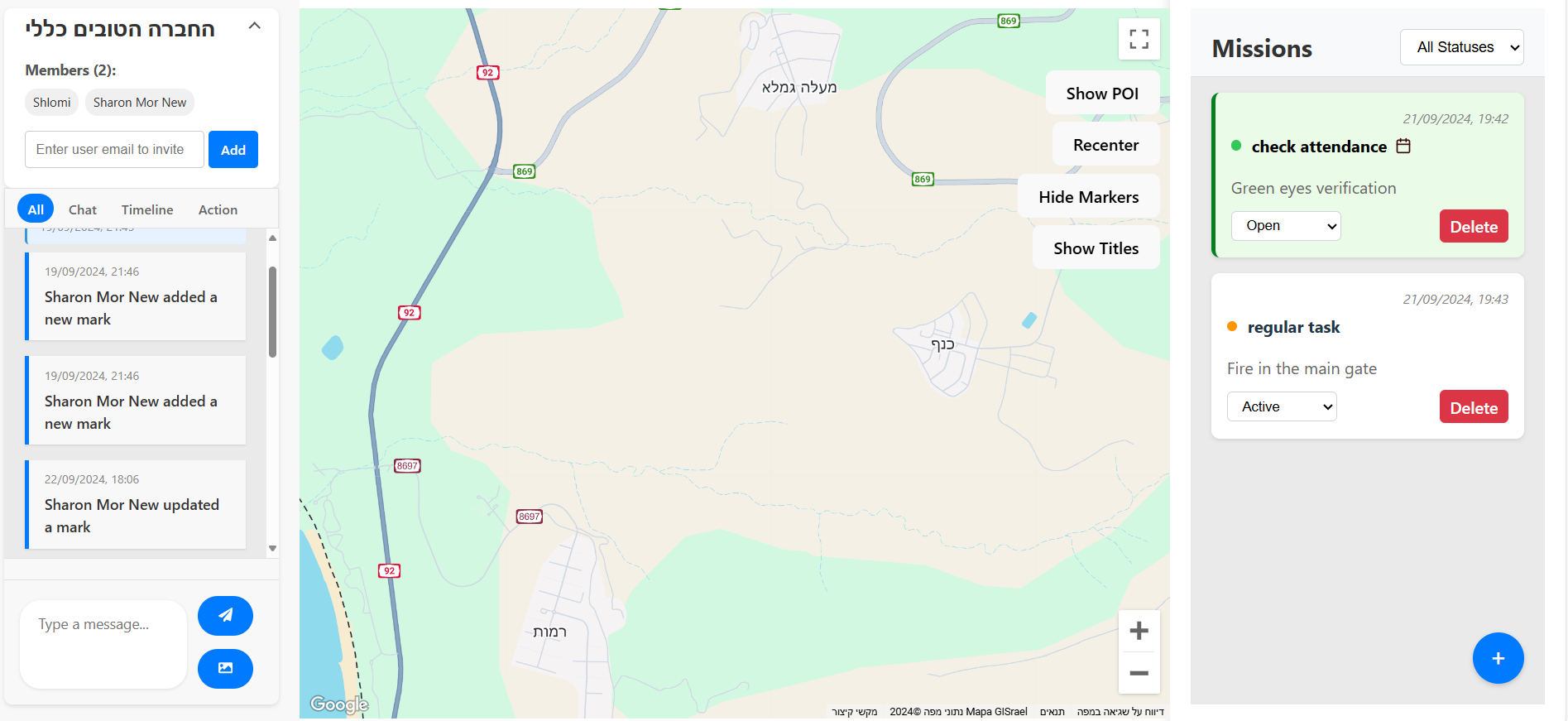
##### 5.1.7 Team Card

* **Purpose:** This card represents an individual team the user is part of, providing basic team information and options for interacting with the team.
* **Key Elements:**
  + **Team Name:** Displays the team's name, in this example written in Hebrew: "**הרחבה** **מדרך** **עוז**" (translated: "Extension Guide Oz").
  + **Members Count:** Shows the number of members in the team, indicating the current number of participants (2 members).
  + **Arrow Icon (Red):** This red arrow icon allows the user to leavetheteam. If the user is the teamowner, leaving the team will result in the deletion of the entire team.



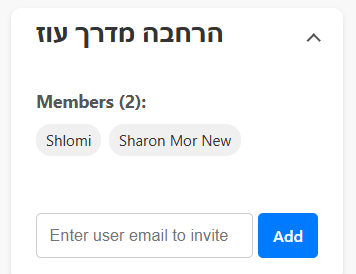
##### 5.1.8 Team's Control Panel

* **Purpose:** This page is designed for managing missions and interacting with team-related activities. It provides a map interface for location-based data, mission management, and communication among team members.
* **Key Elements:**
  + **Chat Box:** Allows users to communicate with team members by sending messages or images.
  + **Map Interface:** A central map showing the current geographical area and points of interest (POI) relevant to the team’s operations.
  + **Mission Management:** A list of current missions assigned to team members.



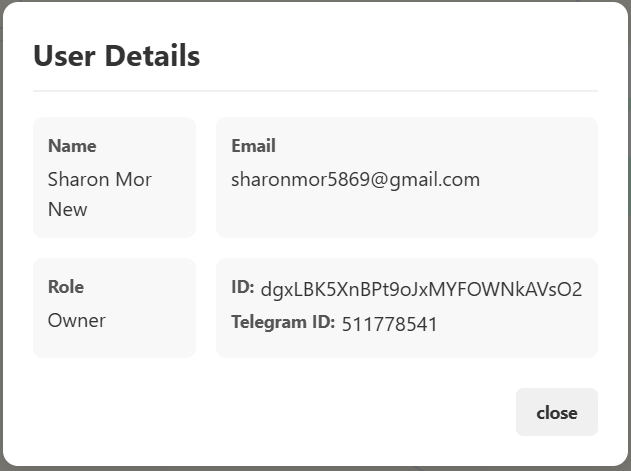
##### 5.1.9 Team Member Management Section

* **Purpose:** This section allows team owners or administrators to manage team members, displaying the current team members and providing functionality to invite new ones.
* **Key Elements:**
  + **Team Name:** Displays the current team’s name (in this case, "**הרחבה מדרך עוז**").
  + **Members List:** Shows the names of current team members, such as Shlomi and Sharon Mor New.
  + **Invite User Feature:** A field where users can enter an email address to invite new members to the team.
  + **Add Button:** After entering an email, clicking this button sends an invitation to the new user to join the team.



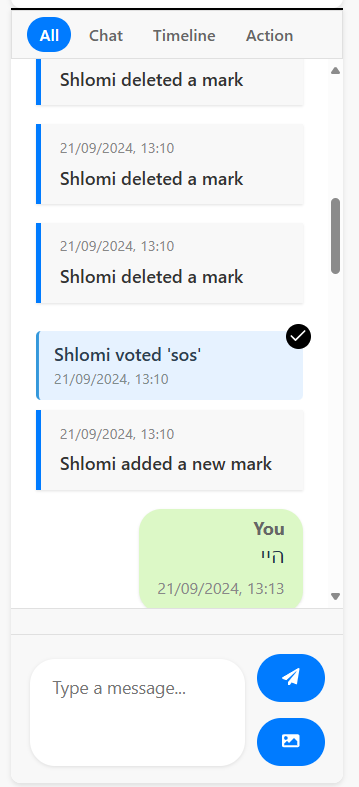
##### 5.1.10 User Details Dialog

* **Purpose:** This dialog provides detailed information about a team member when their name is clicked in the members section.
* **Key Elements:**
  + **Name:** Displays the full name of the user (e.g., Sharon Mor New).
  + **Email:** Shows the user’s email address (e.g., sharonmor5869@gmail.com).
  + **Role:** Indicates the user's role within the team (e.g., Owner).
  + **ID:** A unique identifier associated with the user in the system.
  + **Telegram ID:** Shows the user's Telegram ID for integration purposes (e.g., 511778541).
  + **Close Button:** Allows users to close the dialog and return to the previous screen.



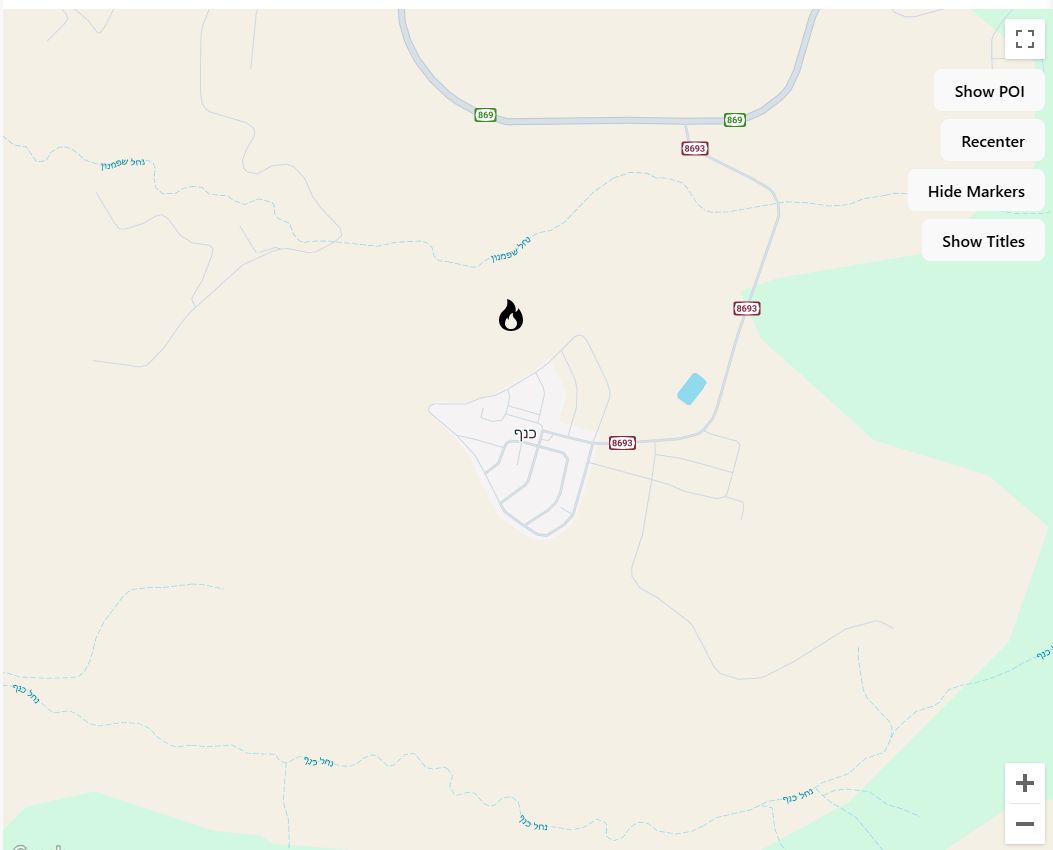
##### 5.1.11 Messenger Section

* **Purpose:** This section provides a detailed feed of team activities, messages, and actions, allowing users to track the team's events, communicate, and respond to interactive tasks.
* **Key Elements:**
  + **Filters (Timeline / Chat / Action):**
    - **Timeline:** Displays events that have occurred within the team (e.g., adding or deleting marks, updating tasks).
    - **Chat:** Displays user messages sent in the team's chat.
    - **Action:** Shows user responses to interactive messages with buttons (e.g., completing tasks, interacting with buttons).
  + **Activity Feed:** Displays actions, such as members adding or deleting marks or updating missions.
  + **Message Input Box:** Allows users to type messages to communicate with other team members.
  + **Send Message Button:** Sends typed messages.
  + **Send Media Button:** Allows users to send media, such as images, in the chat.



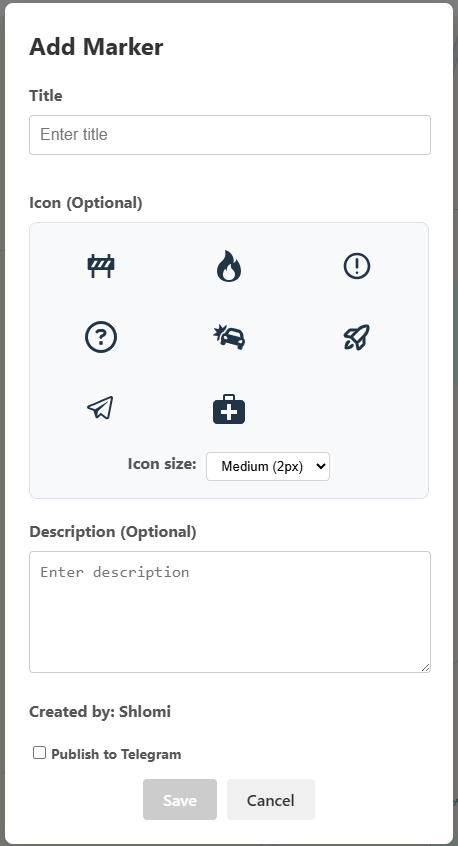
##### 5.1.12 Map Interface

* **Purpose:** This section allows users to interact with geographical data, view key points of interest (POIs), and manage map markers related to the team's operations.
* **Key Elements:**
  + **Map Display:** Shows the geographical area relevant to the team’s operations. In this case, a specific region is marked with an icon (e.g., fire icon for an event or POI).
  + **Show POI Button:** Toggles the display of points of interest on the map.
  + **Recenter Button:** Re-centers the map on the default location or main area of interest.
  + **Hide Markers Button:** Allows users to hide the markers displayed on the map.
  + **Show Titles Button:** Displays additional titles or labels related to the markers and locations on the map.
  + **Zoom Controls:** Located at the bottom right, these allow users to zoom in and out on the map for better detail or a broader view.



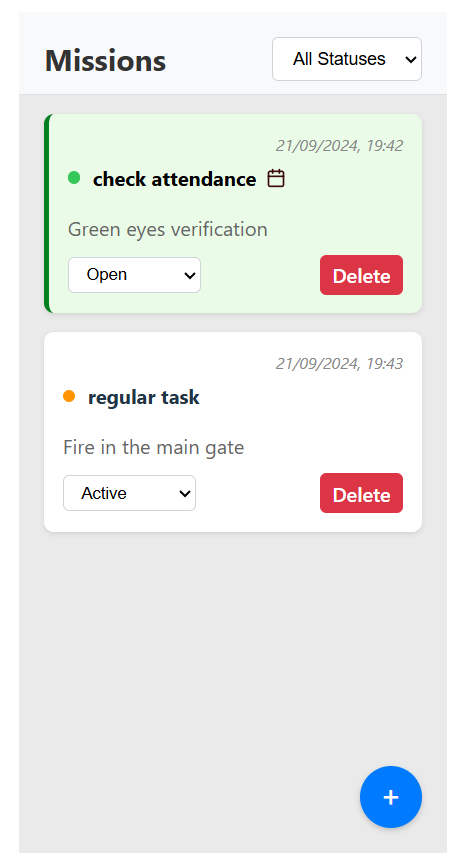
##### 5.1.13 Add Marker Modal

* **Purpose:** This dialog allows users to add a marker on the map for indicating points of interest, events, or important locations. Left click on the map to open the modal.
* **Key Elements:**
  + **Title Field:** A text input where users can enter the title or name of the marker.
  + **Icon Selection (Optional):** A set of icons users can choose from to visually represent the marker (e.g., fire, question mark, medical cross).
  + **Icon Size:** Allows users to select the size of the icon from options such as small, medium, or large.
  + **Description (Optional):** A field where users can add a description to provide more context for the marker.
  + **Created by:** Displays the username of the person creating the marker (e.g., "Shlomi").
  + **Publish to Telegram:** A checkbox to optionally publish the marker details to a connected Telegram bot.
  + **Save and Cancel Buttons:** Save the marker to the map or cancel the action.



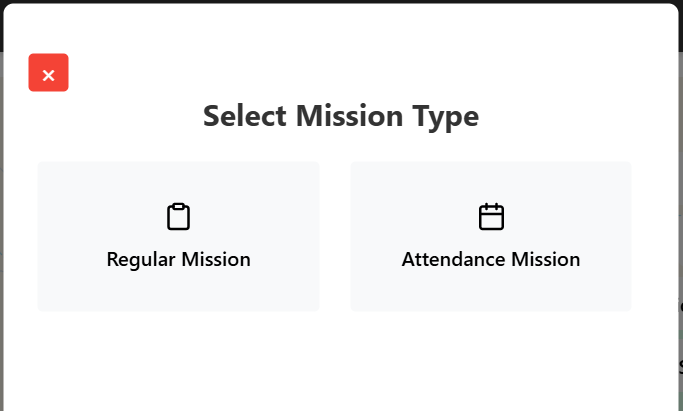
##### 5.1.14 Missions Section

* **Purpose:** This section is designed for managing, tracking, and updating the status of missions assigned to team members. Each mission is categorized by type and displays relevant details for quick reference.
* **Key Elements:**
  + **Mission Cards:** Each card displays a task with key information:
    - **Task Title:** For example, "check attendance" or "regular task" with visual indicators such as a calendar icon for attendance tasks.
    - **Task Description:** Provides additional context, such as "Green eyes verification" or "Fire in the main gate."
    - **Status Dropdown:** Allows users to update the status of the mission (e.g., Open, Active).
    - **Delete Button:** Enables users to remove the mission from the list.
  + **Add Mission Button (+):** A blue button for adding new tasks to the mission list.
  + **Status Filter:** A dropdown that allows users to filter missions based on their status (e.g., All Statuses).



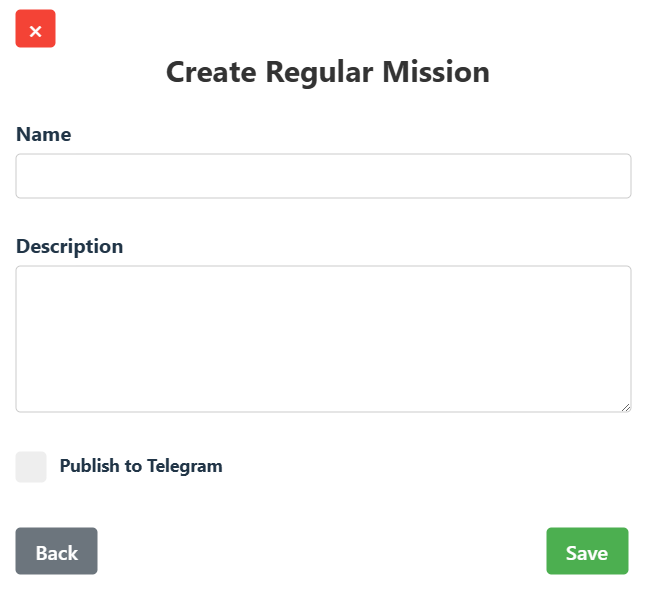
##### 5.1.15 Select Mission Type Dialog

* **Purpose:** This dialog allows users to choose the type of mission they want to create, ensuring that tasks are categorized correctly based on their nature (e.g., regular tasks vs. attendance-related tasks).
* **Key Elements:**
  + **Regular Mission Button**: Creates a standard task for the team, such as general activities or operational tasks.
  + **Attendance Mission Button:** Specifically for attendance-related tasks, which might involve checking in or verifying the presence of team members.
  + **Close Button (Red X):** Allows users to close the dialog without selecting a mission type.

****

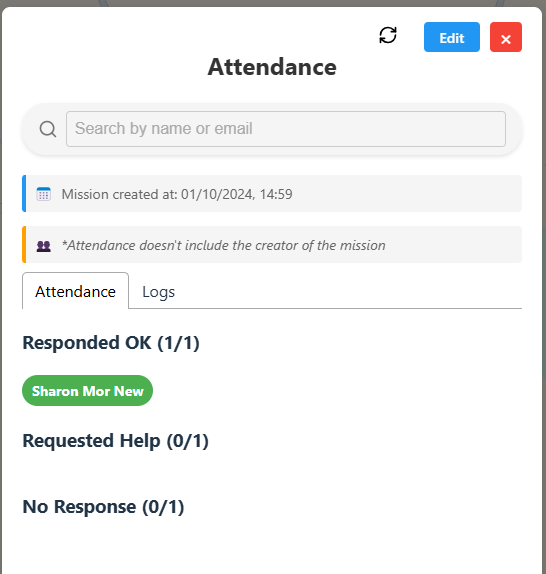
##### 5.1.16 Create Mission Dialog

* **Purpose:** This dialog allows users to create a new Mission, providing a space to define the task’s name, description, and optional publishing settings.
* **Key Elements:**
  + **Name Field:** A text input field for entering the mission's name or title.
  + **Description Field:** A larger text box for providing details or instructions related to the mission.
  + **Publish to Telegram Checkbox**: An option to publish the mission details to a connected Telegram bot.
  + **Back Button:** Allows the user to return to the previous screen without saving the mission.
  + **Save Button:** Confirms and saves the mission to the team’s list of tasks.

****

##### 5.1.17 Attendance Modal

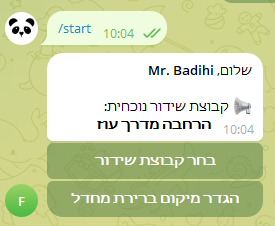
* **Purpose:** This dialog displays the status of team members’ responses to an Attendance Mission, allowing the team leader to track who has responded and whether anyone needs help.
* **Key Elements:**
  + **Search Bar:** Allows users to search for a specific team member by name or email.
  + **Mission Details:** Shows the creation date of the attendance mission and includes a note that the mission creator is not included in the attendance count.
  + **Attendance Tabs:**
    - **Attendance Tab:** Lists team members based on their response status, such as:
      * **Responded OK:** Displays team members who have confirmed their attendance (e.g., Sharon Mor New).
      * **Requested Help:** Shows team members who have requested assistance.
      * **No Response:** Lists team members who haven't responded yet.
    - **Logs Tab:** Displays activity related to the attendance mission.
  + **Refresh Button:** Allows the user to refresh the attendance status to ensure it’s up to date.
  + **Edit Button:** Lets the user modify the attendance mission details.
  + **Close Button (Red X):** Exits the dialog.

****

#### 5.2 Mobile

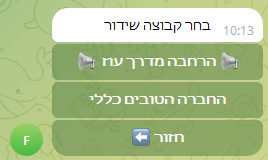
##### 5.2.1 Telegram Bot Interaction

* **Purpose:** This interface in the Telegram bot allows users to select which team they want to send a message to and define a default location for cases when GPS is unavailable.
* **Key Elements:**
  + **Message Greeting:** The bot greets the user by name and displays the current broadcast group (in this case**, "הרחבה מדרך עוז"**).
  + **First Button ("בחר קבוצת שידור"):** Allows the user to select from a list of teams to choose which team the message should be sent to.
  + **Second Button ("הגדר מיקום ברירת מחדל"):** This button is for setting a default location to be used when GPS data is not available.

****

##### 5.2.2 Team Selection in Telegram Bot

* **Purpose:** This screen in the Telegram bot allows the user to select which team to send a broadcast message to.
* **Key Elements:**
  + **Team Options:** A list of available teams is shown, such as "**הרחבה מדרך עוז"** and "**החברה הטובים כללי**."Each team has an associated broadcast icon next to it, indicating that the selected team will be the target for the broadcast.
  + **Back Button ("חזור"):** Allows the user to return to the previous menu if they do not want to select a team at this moment.

****

##### 5.2.3 Default Location Request in Telegram Bot

* **Purpose:** This message from the Telegram bot prompts the user to share a default location, which will be used when GPS data is unavailable.
* **Key Elements:**
  + **Bot Message:** The bot asks the user to share the location they would like to set as the default: ***"אנא שתף את המיקום שתרצה להגדיר כברירת מחדל"*** (translated: "Please share the location you would like to set as the default").
  + **User Reply:** The user can reply by sharing their location through Telegram’s location-sharing feature

**תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי**

### 6. Maintenance Guide

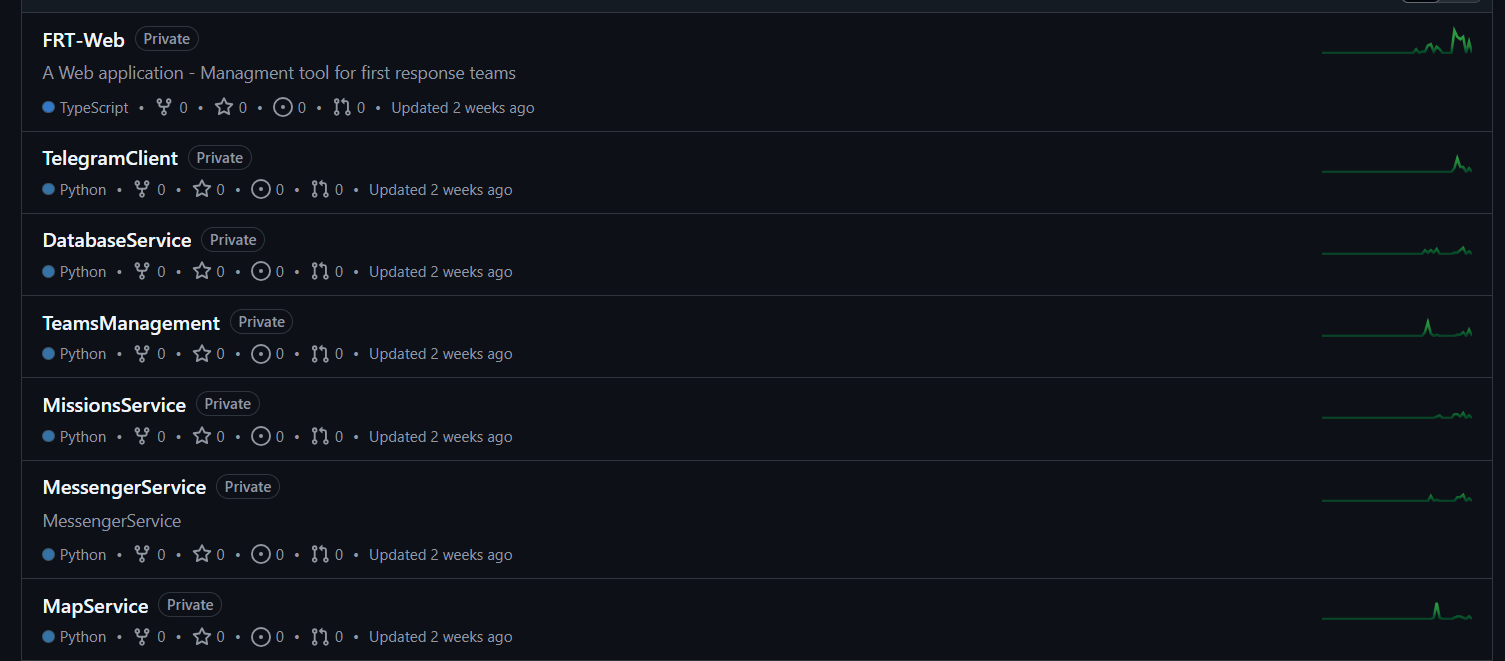
#### 6.1 Installation

##### 6.1.1 Prerequisites

* Node.js and npm (for server-side and client-side JavaScript)
* Docker (for containerization)
* AWS account (for deployment)
* MongoDB (for database)
* Apache Kafka (for message brokering)
* Git
* Telegram App

##### 6.1.2 Setup

1. **Clone GitHub repositories:**

****

1. **Install required modules:**
   1. **FRT-Web:** run npm install for client and server
   2. **Python Services:** run pip install requirements.txt
2. **Private credentials:**
   1. **.env files**: Add environment files with private credentials

#### 6.2 Running the application locally

* **Web Client:** run npm run dev
* **Web Server (BFF):** run npm run dev
* **Python services:** run app.py, starting with Database service

#### 6.3 Deploying the application

* **Containerization:**
* Create a dockerfile for each service
* Create a docker-compose.yml file
* Run docker-compose build (or alternatively build each service using docker build)
* **Deploying:**
* Push docker images to ECR (Amazon’s Image registry) command: ‘docker-compose push’
* Create ECS cluster and 7 different Fargate services – each service for another image
* Create task definition for each service, filled with correct params and environment values
* Create a EC2 Load balancing application with relevant paths and values
* Buy a domain from Amazon 53 and attach it to the client’s elastic public ip
* Open the domain in the browser

#### 6.4 Database

MongoDB Cluster with the following collections

**A screenshot of a phone

Description automatically generated**

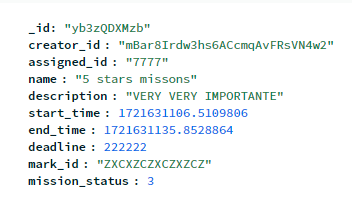
##### 6.4.1 Maps Collection:

**A screenshot of a computer code

Description automatically generated**

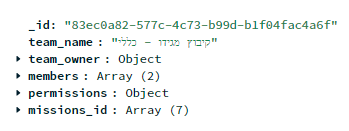
Each map collection item contains a unique id, active marks array, scale (zoom) and what is the initial location.

##### 6.4.2 Missions Collection



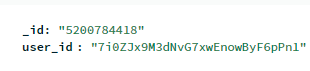
Each mission collection item contains a unique id, who created the mission, assignee id, name, description, start and end time, deadline, id of the mission’s mark and the status.

##### 6.4.3 Teams Collection



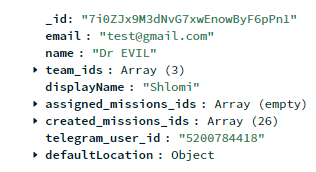
Each team collection item contains a unique id, name of the team, who is the owner, members, permissions and related mission ids.

##### 6.4.4 Telegram Users



Each telegram user’s collection item contains a pair – id in the system and the relevant telegram bot id.

##### 6.4.5 Users:



Each user’s collection item contains unique id, email, name, team ids that the user is part of, a display name, assigned mission ids, mission ids that the user created, telegram user id if exists and the user’s default location

#### 6.5 Kafka

In the context of our project, we use several Kafka topics to manage different aspects of communication and data processing. These topics facilitate the real-time streaming of information between components of the system. Below is a brief overview of each topic and the structure of the messages they carry.

##### 6.5.1 map\_topic

The *map\_topic* is responsible for handling data related to map markers. Each message contains details about a marker that will be displayed on the map, including its location, description, and status.

**Message format:**

**תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי**

##### 6.5.2 mission\_topic

The *mission\_topic* manages data related to missions, including their creation, assignment, and status updates. Each mission has details like start and end times, the assigned user, and a status indicator.

**Message format:**



##### 6.5.3 message\_stream\_topic

The *message\_stream\_topic* is used for handling messages within the system, such as chat communications. The message format includes metadata like the user ID, chat ID, and message content, as well as interactive elements via inline keyboards.

**Message format:**

****

### 7. Evaluation

Our evaluation process focused on assessing the FRT Application's performance and usability through two methodologies: expert feedback and user testing.

To evaluate our application's effectiveness and real-world applicability, we conducted two main assessment approaches:

* An in-depth interview with Mr. Amir Dagan, Health Cert Team Leader at Kibbutz Megiddo, who tested the system's core functionalities
* A System Usability Scale (SUS) survey administered to 20 potential users following a demonstration of the system's functions

#### 7.1 Reflection - Interview with Mr. Amir Dagan - Head of health CERT, Kibbutz Megiddo

A significant validation of the FRT Application's practical utility came through an in-depth interview and system demonstration with Mr. Amir Dagan, the Health Team Leader at Kibbutz Megiddo. The interview provided valuable insights into the application's real-world applicability and potential impact on emergency response operations.

#### 7.1.1 Key Observations and Feedback

Mr. Dagan's hands-on evaluation of the system included testing core functionalities such as user authentication, map marking, and cross-platform integration between web and mobile interfaces. His assessment was notably positive, particularly regarding the application's potential for improving coordination among various kibbutz teams, with specific emphasis on health team operations.

#### 7.1.2 Real-World Use Case Validation

A particularly illustrative example emerged from Mr. Dagan's recent experience during a training exercise. He described a situation where response teams were directed to "Moze's house," but encountered delays and confusion due to team members' unfamiliarity with the location. This real-world scenario highlighted one of the application's key strengths - the ability to instantly mark and share locations via the integrated map system and Telegram functionality. Mr. Dagan emphasized that this feature could significantly reduce response times and potentially save lives in actual emergencies.

#### 7.1.3 Areas for Enhancement

While the overall feedback was positive, Mr. Dagan provided valuable suggestions for future improvements:

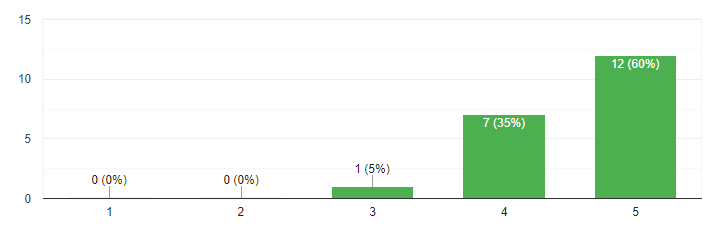
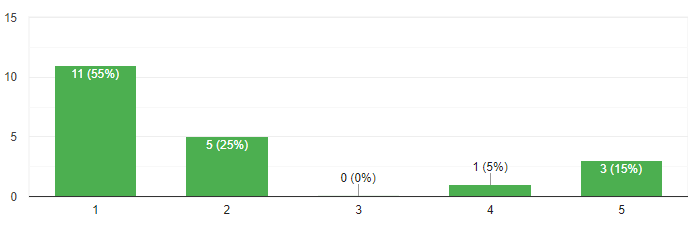
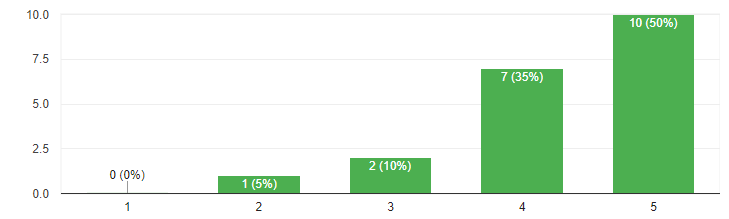
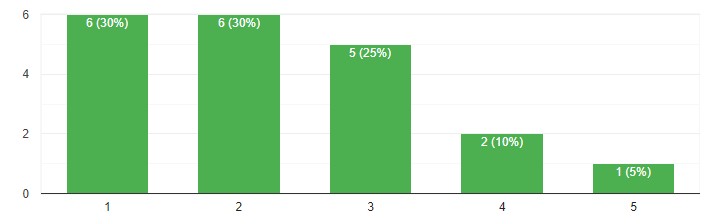
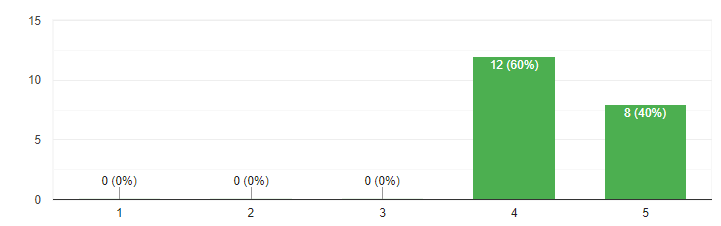
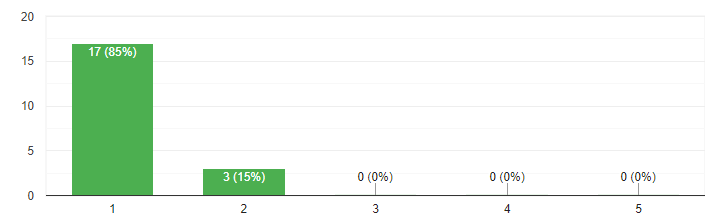
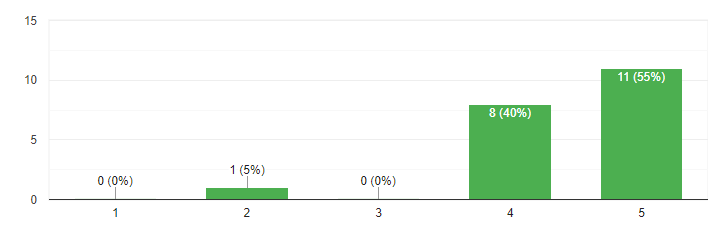
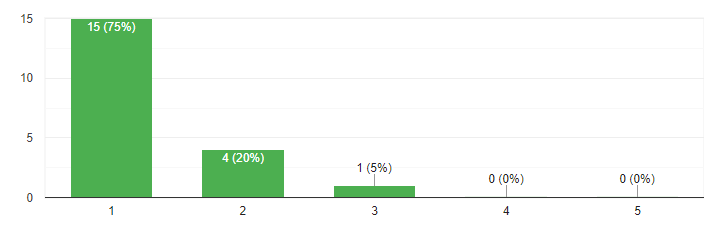
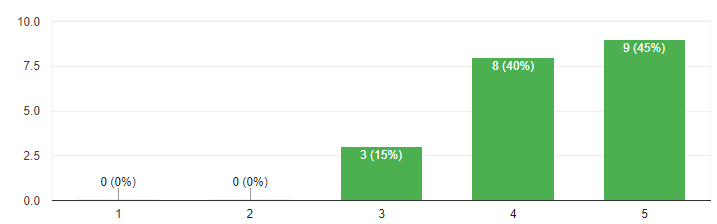
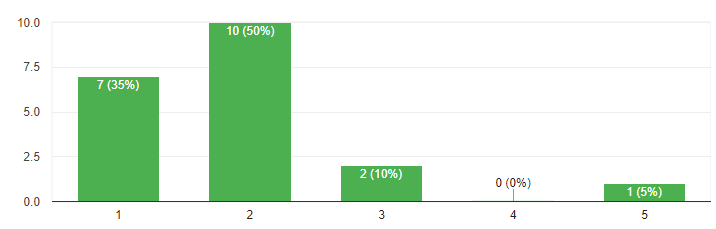
**Voice Command Integration:** A key recommendation was the implementation of voice command functionality. Mr. Dagan noted that during emergencies, the current interface, while effective, might be too time-consuming when rapid response is crucial. He suggested incorporating:

* One-click voice recording capability
* AI-powered voice processing
* Automatic conversion of voice commands into system actions

Example use case: Converting a voice command like "I need help near Moze's house" into both a chat message and an automated map marker

#### 7.2 SUS Survey

From October to December 2024, we surveyed 20 potential users from various First Response Teams and CERT organizations to evaluate the system's usability. The survey used the standard System Usability Scale, a reliable tool for measuring system usability. Each question's results are presented below with a corresponding bar chart showing the distribution of responses on a scale from 1 (strongly disagree) to 5 (strongly agree):

1. **I would like to use this system frequently.**
2. **I found the system unnecessarily complex.**
3. **I thought the system was easy to use.**  
   
4. **I would need technical support to be able to use this system.  
   **
5. **I found that the various functions of the system were well integrated.**
6. **I thought there was too much inconsistency in this system.**
7. **I think most people would be able to learn to use this system easily.**
8. **I found the system very tough to use.**
9. **I felt very confident when using the system.**
10. **I have to learn a lot of things before I can use this system  
    **

#### 7.3 Evaluation Summary

The application received positive feedback through both evaluation methods. The interview with Mr. Dagan highlighted the system's potential impact on emergency response operations, particularly praising the location-sharing capabilities that could prevent confusion during emergencies. He provided a practical example where the app could have prevented delays during a training exercise when teams struggled to locate "Moze's house."

The SUS survey yielded a score of **83.375**, indicating strong user acceptance. Users particularly appreciated the well-integrated functions and found the system easy to use. However, some concerns were noted that several users indicated they would need technical support to use the system effectively, and several participants, including Mr. Dagan, requested voice command functionality to replace manual typing and location sharing during emergencies.

### 8. Conclusion and Summary

This paper has presented a comprehensive examination of the First Response Team (FRT) Application, a project designed to enhance the operational efficiency and communication capabilities of emergency response teams. The project successfully evolved from its initial conceptualization in Phase A to a functional prototype in Phase B, demonstrating significant progress and adaptability in its development process.

#### 8.1 Key Findings and Achievements

* User-Centered Development: By employing a user-centered design approach and integrating feedback from first responders and emergency management professionals, we ensured the application met real-world needs.
* Feature Implementation: The majority of features proposed in Phase A were successfully implemented, including real-time communication tools, mission management capabilities, and an interactive mapping system.
* Technological Integration: The project successfully integrated a range of modern technologies, including React for web development, Node.js for server-side operations, and various AWS services for deployment and scaling.
* Adaptation and Innovation: A notable change from our initial plan was the shift from a native mobile application to a Telegram bot interface, demonstrating our ability to adapt to user needs and leverage existing platforms.

#### 8.2 Challenges and Solutions

Throughout the development process, we encountered several significant challenges:

**Mobile Development:** Initial plans involved native mobile app development, which would have required significant resources and expertise in mobile technologies.  
**Solution:** We pivoted to developing a Telegram bot interface, which provided immediate cross-platform mobile access while leveraging existing messaging infrastructure.

**Deploying Multiple Services:** Implementing and managing a microservices architecture proved complex, particularly in orchestrating multiple services and ensuring seamless communication between them.  
**Solution:** Implemented Docker Compose to orchestrate container deployment and used AWS ECS for service management. Created separate task definitions for each service with proper environment configurations.

**Parallel Development:** Working simultaneously on interconnected domains presented synchronization challenges, requiring careful coordination and version control management.  
**Solution:** Used Git for version control while we tried to focus on separate domains.

**HTTPS Implementation:** Transitioning from HTTP to HTTPS was necessary to enable WebSocket functionality beyond local environments, which involved additional security considerations and configuration.  
**Solution:** Configured AWS Certificate Manager for SSL certificates and set up Route 53 for domain management. Implemented proper security headers and CORS policies.

**Real-time Data Synchronization:** Ensuring consistent and timely data updates across all user interfaces presented technical hurdles.  
**Solution:** Utilized kafka and websockets.

**Cross-platform Compatibility:** Maintaining a uniform user experience across web and Telegram interfaces required careful design and implementation strategies.  
**Solution:** Used same service structure for the different services and reused several components

**User Research:** Conducting and synthesizing interviews with first responders presented logistical challenges but provided invaluable insights for feature prioritization and user interface design.  
**Solution:** Created a SUS survey in order to receive feedback and focused on a single long and in depth system walkthrough reflection. In addition, we pre-recorded a video that is showcasing our app’s different functions ([FRT Project - Walkthrough](https://www.youtube.com/watch?v=lYiZtuzUW_Y)) .

#### 8.3 Implications and Future Research

The FRT Application has significant potential for further development and expansion. Future research could explore:

* Implementing a more granular roles and permissions system
* Developing super-admin accounts for overseeing multiple teams
* Enhancing analytics capabilities for post-incident review and training
* Expanding integration with other emergency services and platforms

#### 8.4 Personal Learning Outcomes

This project provided invaluable learning experiences beyond technical skills. We gained significant insights into:

* Effective teamwork under strict deadlines
* Real-time application development challenges
* Cloud deployment strategies, particularly with AWS
* Microservices architecture implementation and management

#### 8.5 Final Thoughts

We believe the FRT Application represents a significant advancement in emergency management technology. By combining user-centric design with cutting-edge technology, we've created a tool that has the potential to significantly improve the effectiveness of first response teams. The current state of the FRT Application is more than just a concept; it's a working prototype that can be immediately utilized by emergency response teams seeking to improve their coordination and efficiency.

As we move forward, we remain committed to refining and expanding this platform to meet the evolving needs of emergency responders worldwide. This project not only contributes to the field of emergency management but also serves as a foundation for future research and development in crisis response technologies.

### 9. References

1. Blum, J. R., Eichhorn, A., Smith, S., Sterle-Contala, M., & Cooperstock, J. R. (2014). Real-time emergency response: Improved management of real-time information during crisis situations. Journal on Multimodal User Interfaces, 8(2), 161-173.
2. Brown, O., Power, N., & Conchie, S. M. (2021). Communication and coordination across event phases: A multi‐team system emergency response. Journal of Occupational and Organizational Psychology, 94(3), 591-615.
3. Damaševičius, R., Bacanin, N., & Misra, S. (2023). From sensors to safety: Internet of Emergency Services (IoES) for emergency response and disaster management. Journal of Sensor and Actuator Networks, 12(3), 41.‏
4. Figueiredo, D. (2019). Applying the User-Centered Design approach for Prototyping the Interfaces of an Intelligent Emergency Management System (Doctoral dissertation, Universidade Nova de Lisboa).
5. Flint, C., & Brennan, M. (2006). Community emergency response teams: From disaster responders to community builders. Rural realities, 1(3), 1-9.‏
6. Johnson, R. (2000). GIS technology for disasters and emergency management. Journal of Geographic Information and Decision Analysis.
7. King, R. V., North, C. S., Larkin, G. L., Downs, D. L., Klein, K. R., Fowler, R. L., ... & Pepe, P. E. (2010). Attributes of effective disaster responders: focus group discussions with key emergency response leaders. Disaster medicine and public health preparedness, 4(4), 332-338.
8. McWilliams, M. (2020). Creating an effective emergency response team. Professional Safety, 65(6), 66-70.‏
9. <https://aws.amazon.com>
10. <https://expressjs.com/>
11. <https://firebase.google.com/docs/auth?hl=en>
12. <https://kafka.apache.org/documentation/#gettingStarted>
13. <https://nodejs.org/en/about>
14. <https://www.avuka-app.com/squad/about>
15. [https://www.avia-app.org](https://www.avia-app.org/)
16. <https://www.geeksforgeeks.org/advanced-encryption-standard-aes/>
17. <https://www.mongodb.com/docs/atlas/>
18. <https://react.dev/>
19. [https://trends.google.com/trends/explore?geo=IL&q=כיתת%20כוננות&hl=iw](https://trends.google.com/trends/explore?geo=IL&q=%D7%9B%D7%99%D7%AA%D7%AA%20%D7%9B%D7%95%D7%A0%D7%A0%D7%95%D7%AA&hl=iw)
20. <https://zello.com/product/frontline-communications/>
21. [https://claude.ai/](https://claude.ai/%20) - Professional Plan
22. <https://chatgpt.com/> - Chat GPT