Sharon Mor | Yuval Badihi

Advisor – Dr. Naomi Unkelos Spiegel

First Response Team Application

Table of Contents

**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 proposes creating an advanced web and mobile app to optimize communication and operational efficiency for first response and emergency resilience teams, especially in high-risk and recently impacted areas. The proposed 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. Additionally, it examines the app's potential as a model for leveraging technology to elevate emergency management practices, aiming to significantly enhance responsiveness and effectiveness.

**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 ***[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.]***. 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 and enhance emergency response capabilities, our project aims to develop an innovative application that will serve as a centralized platform for seamless coordination and decision-making. Drawing insights from leading practices on organizing effective emergency teams ***[McWilliams, M. (2020).]*** *and enhancing the capabilities of disaster responders****[King, R. V., North, C. S., Larkin, G. L., Downs, D. L., Klein, K. R., Fowler, R. L., ... & Pepe, P. E. (2010)]****,* this application will provide a comprehensive suite of features designed to streamline communication, improve situational awareness, and facilitate efficient task management.

The application will offer features like alarm creation, task assignment, real-time location tracking, hazard marking systems, and comprehensive task management tools. Built using modern web and mobile technologies like React and React Native, it will intelligently differentiate between response and recovery phases ***[Brown, O., Power, N., & Conchie, S. M. (2021).]***, ensuring that appropriate protocols and actions are followed in each stage. Additionally, it will archive incident data for future analysis and draw upon well-established emergency response principles.

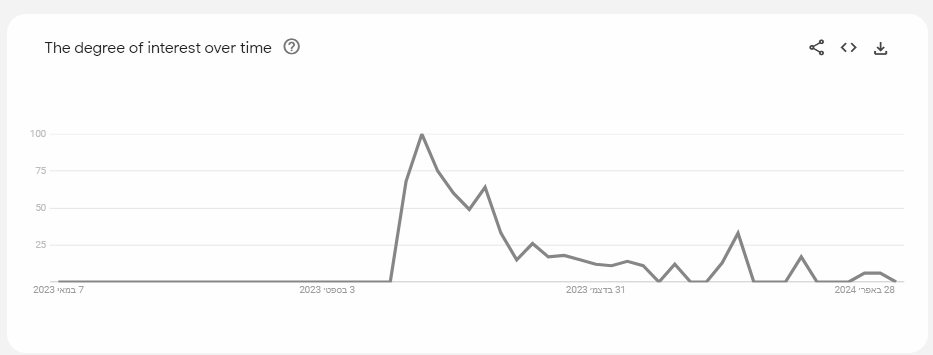
This tool promises to improve team preparedness, equipping responders with vital capabilities, and overcoming communication barriers in chaotic situations. By integrating advanced features into an accessible interface, this application aims to simplify complexities, boost team confidence, and transform how first responders operate during crises. The goal is to create an essential partner that substantially impacts response efficiency and ultimately saves lives.

The rest of the book examines the technological, organizational, and ethical aspects of advanced emergency management systems. It uses case studies and expert insights to explore challenges and solutions, highlighting the benefits of integrating advanced technology in this field.

**2. LITERATURE REVIEW**

**2.1 First Response Team**

**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* [Number Of Google Trends Ref]

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.**[6]**

**3. RESEARCH**

**3.1 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. **[7]** |
| --- | --- |
| 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.**[8]** |
| ‪Zello PTT Walkie Talkie – At the ECOM App Library | Get it here‬‏ | **Zello:** Zello is a modern and customizable walkie-talkie app (and more) that lives on any smart device.[9] |

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

**3.2 Existing Application Comparison Table**

| **Avuka Squad** | **Zello** | **AVIA** | **Feature** |
| --- | --- | --- | --- |
|  | ✔ | ✔ | IOS Support |
| ✔ | ✔ | ✔ | Android Support |
|  | ✔ | ✔ | Web/PC Support |
| ✔ |  | ✔ | Management App |
|  | ✔ | ✔ | Resident's App |
|  |  | ✔ | Wellness Check |
|  |  |  | Automatic Dialing Emergency Services |
| ✔ |  | ✔ | Google Maps |
| ✔ |  | ✔ | Dynamic Map |
| ✔ |  | ✔ | GPS Tracking |
| ✔ |  |  | Sectors In Map |
| ✔ |  |  | Freehand Marking |
| ✔ |  |  | Navigation Option |
|  |  |  | 3D Map |
| ✔ | ✔ |  | Push-To-Talk |
| ✔ | ✔ | ✔ | Chat |
| ✔ | ✔ | ✔ | SOS Alarm |
| ✔ | ✔ | ✔ | Alert For FRT Activation |
| ✔ |  | ✔ | Predefining Possible Emergency Events |
| ✔ |  |  | Execute Exercise |
| ✔ | ✔ | ✔ | Defining Groups |
| ✔ | ✔ | ✔ | Operation Logs |
| ✔ |  | ✔ | Message Board |
|  |  | ✔ | Forces Count |
|  |  |  | Performance Monitoring |
|  |  |  | Integration With Cameras |
|  |  |  | Smooth Registration |

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.

**3.2.1 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.

**3.3 Interviews**

To gather accurate and actionable data for the development of our response and resilience team management application, engaging directly with experienced individuals in the field was essential. Interviews with key stakeholders from various settlements provided invaluable insights into the unique challenges and requirements faced by first response and residential resilience teams. Through these discussions, we were able to understand the complexities of their operations and identify specific features and functionalities that our application must include to effectively support their critical work. **3.3.1 Interviews Summary**

1. **Muli Shpigel** - Architect, Director of Engineering and Planning Department, the settlements in the Golan.  
   **Background:** Muli oversees logistics for his settlement's first response team, which is well-organized into specialized units.  
   **Key Insights:** The need for a system that effectively delineates and connects various operational teams, both geographically and functionally, while facilitating proper communication among them.
2. **Roee Tavor** - Chairman of the Residential Resilience Team in Kanaf settlement.  
   **Background:** Kanaf operates with a residential resilience team collaborating closely with the first response team.  
   **Key Insights:** *"Since the blackout on 7.10, we've realized the necessity for technology that works independently of GPS and internet connections,"* highlighting the essential need for a system that functions reliably in technologically limited environments.
3. **Hagit Geva & Itamar Cohen** - Chairman & Vice Chairman of the Residential Resilience Team in Givat Yoav settlement.  
   **Background:** Givat Yoav has several resilience teams, including logistics, health, education, spokesperson, welfare, and first response.  
   **Key Insights:** The importance of a strict hierarchy with appropriate permissions, a simplified mobile app for general team members, and a more complex web application for leaders. The need for a feature to mark significant locations on a map and assign tasks across teams. Hagit added, *"It's crucial our tools are intuitive—during a crisis, no one should struggle with complex interfaces."* This comment underscores the importance of user-friendly technology that can be swiftly adopted and utilized under stress.

**3.3.2 Interviews Insights**

* GPS connections cannot be relied upon during emergencies, so the application should function without GPS.
* Push-to-talk functionality is not required, as the teams already have walkie-talkies for communication.
* User experience (UX) and ease of use are crucial for the successful adoption of the application by the teams.
* The application should cater to different roles and permissions within the team hierarchy.
* Marking significant locations on a map and assigning tasks across teams are essential features.
* The application should work seamlessly during both routine operations and emergencies, with a simplified interface for emergencies.

**3.4 Research Summary**

Our research involved a thorough evaluation of existing applications aimed at bolstering the capabilities of first response and emergency resilience teams. The focus was on tools that address the specific operational and communication challenges encountered in emergency situations. Through this research, we analyzed a variety of platforms and conducted interviews with key stakeholders in high-risk areas. These interviews provided valuable insights into the unique needs and requirements of these teams, informing our understanding of the critical features that an effective emergency management system must possess. The outcomes of this research highlight the necessity for an integrated solution that offers comprehensive support through features like robust communication tools, efficient task management, and enhanced situational awareness.

**4. ENGINEERING PROCESS**

**4.1 Development Process**

In developing the advanced web and mobile application for optimizing the operations of first response and emergency resilience teams, our engineering process is 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.

תמונה שמכילה טקסט, צילום מסך, קבלה, תרשים

התיאור נוצר באופן אוטומטי

*fig 2. workflow diagram*

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. Stringent coding practices ensure a reliable and effective application tailored for efficient emergency management by first response teams.

**4.2 Web and Mobile Application Development**

Our integrated system utilizes both web and mobile applications developed with React and React Native to efficiently manage first response teams. The web application, built with React, offers a stable and scalable platform ideal for complex data management and operational oversight. The collaborative nature of React and the availability of numerous third-party tools facilitate quick updates essential in emergency scenarios. However, performance can vary based on browser capability, and internet reliance increases security risks.

The mobile application, developed using React Native, enhances operational effectiveness by ensuring first responders remain informed and connected as much as possible, even offline. It provides tailored user experiences optimized for handheld devices, crucial push notifications, and superior data security with encryption techniques like AES-256. Yet, development complexities arise from the need to support various devices and operating systems like iOS and Android, coupled with the necessity for frequent updates to address compatibility issues.

By combining these platforms leveraging React's code reusability, we harness their respective strengths to ensure comprehensive coverage, seamless communication, and robust emergency response management across web and mobile interfaces.

&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

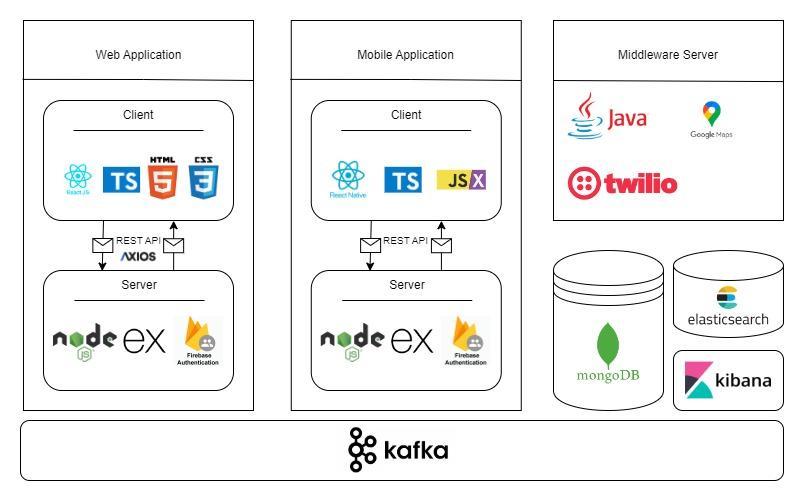
**4.3 System Architecture Overview**

Our system architecture is designed as a hybrid model that combines the principles of microservices and distributed services to ensure seamless interaction between web and mobile platforms. This structure features separate web and mobile application servers, both powered by:

Node.js, with Kafka ensuring loose coupling and scalability through its role as a messaging system.

MongoDB and Elasticsearch support decentralized data management as independent data services.

Additionally, our architecture includes a middleware server, which may function as an API gateway or service orchestrator, typical in microservices environments to streamline complex service interactions. Although the system incorporates several key elements of microservices, such as asynchronous communication and service-specific databases, it does not fully adopt all characteristics, such as complete service autonomy and domain-driven design. By leveraging modern frameworks and tools across both front-end and back-end implementations, supported by advanced data handling and authentication systems, our architecture enhances user experience while maintaining system integrity and scalability to meet growing demands.

fig 3. system architecture diagram

**Application Architecture**

The application is designed as a comprehensive solution comprising both web and mobile components, ensuring a seamless experience across various platforms. The architecture leverages modern technologies and frameworks to deliver a scalable, efficient, and responsive system.

**Web Application**

The web application is built using React, a powerful JavaScript library for creating responsive and efficient user interfaces. React's component-based architecture and virtual DOM implementation enable smooth rendering and optimal performance, ensuring a seamless browsing experience for users.

The server-side logic and API handling are managed by Node.js, a high-performance JavaScript runtime environment. Express.js, a minimalistic and flexible web application framework, is utilized to simplify the routing and middleware architecture, enhancing API development and server management.

For authentication and secure access, Firebase Auth is integrated into the web application, ensuring a consistent and reliable authentication process.

**Mobile Application**

On the mobile front, React Native is utilized to deliver a native-like experience across various mobile platforms. React Native's ability to leverage native components and its cross-platform compatibility make it an ideal choice for building high-performance and visually appealing mobile applications.

Similar to the web application, the mobile application's server-side logic is handled by Node.js and Express.js, providing a consistent and efficient backend infrastructure. Firebase Auth is also integrated into the mobile application, ensuring secure authentication and access control.

**Web and mobile application in high level**

Both the web and mobile applications leverage MongoDB as the primary data storage solution, enabling effective data management and retrieval across all components.

By leveraging a consistent technology stack, including React, React Native, Node.js, Express.js, Firebase Auth, and MongoDB, the web and mobile applications can seamlessly communicate and share data, enabling a cohesive and integrated user experience. This architecture not only promotes code reusability and maintainability but also facilitates efficient development and deployment processes.

**Data Processing and Analytics**

The application incorporates Kafka for real-time data processing and event streaming, Elasticsearch for efficient data storage and analysis, and Kibana for data visualization and insights.

**Java Server**

A dedicated Java server is implemented to handle computationally intensive tasks and complex calculations.

**Technologies Review**

**Client-side**

**React**

React uses JavaScript and JSX to create efficient, dynamic web interfaces. Its virtual DOM improves application performance by only updating necessary parts of the DOM, making it ideal for high-performance applications that require frequent updates. React is compatible with various testing frameworks such as Jest, facilitating robust component testing. It also supports ARIA standards, enhancing accessibility. [https://react.dev/]

**React Native**

React Native extends React’s capabilities to mobile app development, translating React components into native platform elements. This provides a smooth user experience and leverages React's development practices for mobile environments. React Native allows mobile apps to integrate with device capabilities like the camera and GPS, ensuring they perform well on both iOS and Android platforms. [https://reactnative.dev/]

**React VS React Native**

React and React Native are two powerful frameworks developed by Facebook for building user interfaces. While React is primarily used for web development, React Native extends its capabilities to mobile app development. Both frameworks share many similarities, such as their language and syntax, but they also have distinct features tailored to their respective platforms. In this comparison, we'll explore the key differences between React for web and React Native for mobile, focusing on various aspects such as performance,

testing, user interface, scalability, and community support.

During our research about the client-side possible tools we found out that for our use case of building an application both for web and mobile, react and react native a good solution [4].

**React vs. React Native - features comparison**

| **Feature** | **React (Web)** | **React Native (Mobile)** |
| --- | --- | --- |
| **Language and Syntax** | JavaScript and JSX for creating UI components with the ease of HTML-like syntax embedded in JavaScript. | JavaScript and JSX, translating to native platform components for a seamless mobile experience. |
| **Performance** | Uses Virtual DOM for efficient updates. Calculates differences and re-renders only the changed parts on the real DOM. | Employs Virtual DOM and native views for a performance-optimized mobile experience. |
| **Testing** | Jest for testing framework. Enzyme and React Testing Library for testing components in isolation. | Jest for testing framework. React Native Testing Library for mobile-specific testing, including native behaviors. |
| **User Interface** | Component-based, declarative views making the code predictable and debuggable. | Component-based, declarative views with native platform UI elements integration. |
| **Web Support** | Cross-browser compatibility and SEO-friendly features. | Cross-platform compatibility with access to native device features like camera and storage. |
| **Scalability** | Designed for large-scale web applications, it can handle complex architectures and traffic volumes. | Suitable for large-scale mobile applications with the ability to integrate with native modules for enhanced performance. |
| **Community** | Large, active developer community with a rich ecosystem of libraries and tools. | Rapidly growing community with increasing contributions and an evolving ecosystem. |
| **Accessibility** | ARIA (Accessible Rich Internet Applications) standards support for web accessibility. | ARIA support plus native mobile accessibility features like VoiceOver and Talkback for an inclusive user experience. |

**Server-side:**

Our application architecture is designed to effectively serve both web and mobile clients through a robust backend that scales and adapts to the needs of both platforms.

**Node.js:**Node.js is a versatile platform for building a variety of server-side and networking applications. Renowned for its non-blocking, event-driven architecture, it’s particularly adept at managing multiple simultaneous connections with high throughput, which makes it a suitable choice for high-load applications. Node.js brings the familiarity of JavaScript to the server-side, allowing developers to use a single programming language across both front-end and back-end.[10]

**Express.js:**Built on top of Node.js, Express.js is a minimalist and flexible web application framework that provides a robust set of features to develop web and mobile applications efficiently. It simplifies the server creation process with its middleware framework, making it easy to perform tasks like parsing request bodies, managing cookies, and implementing sessions. Express.js is designed to be a lightweight framework that gives you the tools to create a server quickly while also being robust enough to manage web application routes, handle requests, and serve responses. Its ability to create a REST API makes it a powerful tool for backend development, facilitating seamless front-end to back-end integration.[11]

**MongoDB:**MongoDB is an open-source NoSQL database that leverages a flexible document model for storing data. It’s well-suited for applications that need quick iterations and can manage diverse sets of data types. MongoDB’s schema-less nature allows it to handle large volumes of disparate data, which can be particularly useful for applications that require agility in data management and design.[12]

**Elasticsearch:**  
Elasticsearch is a powerful open-source search and analytics engine that allows applications to perform complex searches, real-time data analysis, and visualizations. It enhances our application by enabling fast and scalable search capabilities, crucial for handling large datasets and providing real-time responses to user queries.[13]

**Kibana**

Kibana is employed alongside Elasticsearch to provide powerful and user-friendly visualizations of the data indexed by Elasticsearch. It aids in monitoring, troubleshooting, and securing the application by offering dashboards that can display key performance metrics and logs in real-time. [13]

**Kafka**

Kafka is used as a high-throughput distributed messaging system that efficiently processes streams of records. It plays a critical role in our architecture by facilitating real-time data feeds and enabling decoupled data processing, which is vital for ensuring scalability and reliability across our distributed systems. [14]

**Firebase Authentication:**Firebase Authentication provides a complete identity solution, supporting a variety of authentication methods, including social media accounts, phone numbers, and email/password. It integrates with Firebase's real-time database, which synchronizes data across all clients in real-time. This combination allows for the implementation of secure, authenticated user access to your application's data. Firebase's real-time database uses JSON documents and offers offline data synchronization and retrieval, enhancing the user experience even when connectivity is intermittent.[15]4. **Work Artifacts**

**4.3 FR & NFR Requirements:**

**Functional Requirements Document:**

| **No.** | **Requirement** |
| --- | --- |
| 1 | The System shall provide communication Tools. |
| 1.1 | The System shall support alerts |
| 1.2 | The System shall support alarms |
| 1.3 | The System shall support chat. |
| 2 | The System shall support voice commands. |
| 3 | The System shall include a map. |
| 3.1 | The System shall support marking on a map. |
| 3.2 | The System shall support defining map sectors. |
| 3.3 | The System shall support offline maps. |
| 4 | The System shall Monitor Performance. |
| 4.1 | The System shall record action logs. |
| 5. | The System shall Analyze Performance. |
| 5.1 | The System shall identify trends and bottlenecks. |
| 6. | The System shall ensure Web Compatibility. |
| 7 | The System shall ensure Mobile Compatibility. |
| 8 | The System shall include Management Interface. (BO) |
| 9 | The System shall support different user roles. |
| 9.1 | The System shall support adding new roles. |
| 9.2 | The System shall support setting roles. |
| 10 | The System shall include a Wellness check. |
| 11 | The System shall support Events. |
| 11.1 | The System shall support adding events. |
| 11.2 | The System shall support removing events. |
| 11.3 | The System shall support joining events. |
| 12 | The System shall support authentication. |
| 13 | The System shall support task management. |
| 13.1 | The System shall allow task creation. |
| 13.2 | The System shall allow task assignment. |
| 13.3 | The System shall allow task tracking. |
| 14 | The System shall provide data synchronization. |
| 14.1 | The System shall synchronize data across devices in real-time. |
| 15 | The System shall support customizable dashboards. |
| 15.1 | The System shall allow users to customize views. |
| 16 | The System shall provide secure data storage. |
| 16.1 | The System shall encrypt sensitive data. |

**Non-functional Requirements:**

| **No.** | **Requirement** | **Type** |
| --- | --- | --- |
| 1 | The Application should be quick and easy to use. | Usability |
| 1.1 | The interface shall require no more than 2 clicks or taps from the home screen to initiate any form of communication. | Usability |
| 1.2 | The application shall load its main content and be ready for user interaction within 3 seconds of launch. | Performance |
| 1.3 | The system must support up to 10,000 concurrent users without degradation of performance. | Scalability |
| 2 | The System shall ensure all chats, recordings and locations are encrypted. | Security (cyber and physical) |
| 3 | The System shall use mapping API from one of the providers:  Google Maps, Leaflet, Mapbox, OpenLayers. | Compatibility |
| 3.1 | Marking on a map should be by selecting the option after right clicking (pc) or clicking (on mobile). | Usability |
| 3.2 | The System shall provide a 'Free Draw' feature enabling users to define map sectors by manually drawing the sector boundaries directly on the map interface. | Usability |
| 4 | The System shall monitor the performance of the follows:  Time took between important actions, Number of actions until important action. | Latency |
| 5 | The System should analyze application by recording data and showing it on a graph on the manager side | Usability |
| 6 | The System should record and store the bi events on Google Analytics 4. | Auditability |
| 7 | The Application will allow the leaders of each circle to manage the different roles by names and permissions. | Flexibility |
| 8 | Alerts should be one of the following: Real Alarm, Test Alarm, Recorded Voice, Wellness Alarm | Usability |
| 9 | The Wellness Check will be by alarm and confirmed by each user and will calculate the time it took for everyone to confirm. | Response time |
| 12 | The Application will allow managing events and tasks by setting their name, description, participants and date. | Modifiability |
| 13 | The Web Application will be consisted of the next pages:  1. Login Page 2. Sign Up Page 3. Contact Us 4. Request to join Page 5.Users Page 6. Map Page 7. Chat Page 8. Alert Page 9. Logs Page 10. BO Page 11. Files Page 12. Events Page 13. Tasks Page | Usability |
| 14 | The Mobile Application will be consisted of the next pages: 1. Login Page 2. Sign Up Page 3. Contact Us 5. Users Page 6. Map Page 7. Chat Page 8. Alert Page 9. Tasks status page | Usability |
| 15 | The Application is going to be developed both for web and for mobile. | Compatibility |
| 15.1 | Web Tech Stack: 1. ReactJS (Front Library) 2. CSS / TailwindCSS 3. TS (Superset of JS) 4. NodeJS (JS Runtime) 5. ExpressJS (Server) 6. MongoDB / FirebaseDB (DB) 7. Firebase Authenticator (Auth) 8. Axios (HTTP Calls) 9. Jest (testing) 10. Render or AWS (hosting) | Compatibility |
| 15.2 | Mobile Tech Stack: 1. ReactJS (Front Library) 2. CSS / TailwindCSS 3. TS (Superset of JS) 4. NodeJS (JS Runtime) 5. ExpressJS (Server) 6. MongoDB / FirebaseDB (DB) 7. Firebase Authenticator (Auth) 8. Axios (HTTP Calls) 9. Jest (testing) 10. android / app store (hosting) | Compatibility |

**3.4 Use Case Diagram**

* **Unregistered User:** This is 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.
* **Super User**: A user with administrative privileges who manages at least one registered user, overseeing their activities and permissions within the system.

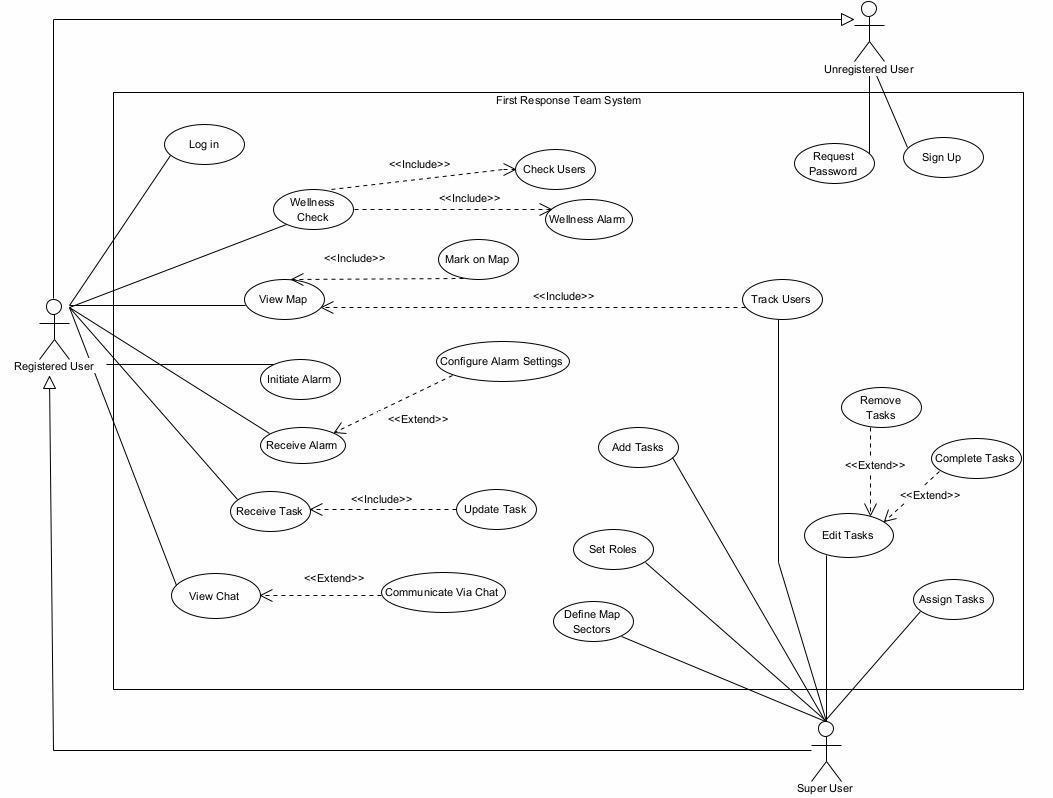


fig 4. Use Case diagram

**3.5 Activity Diagram**

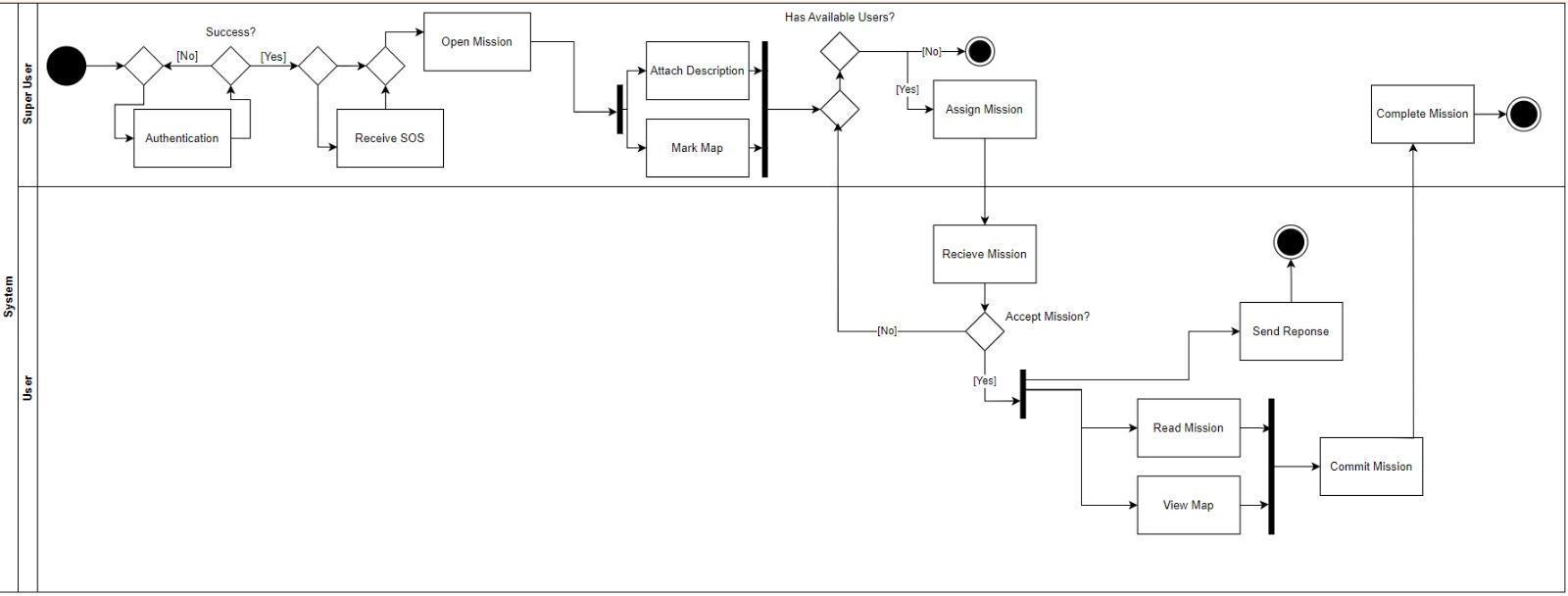


fig 5. Activity diagram

3.6 **Application Screen Sketches**

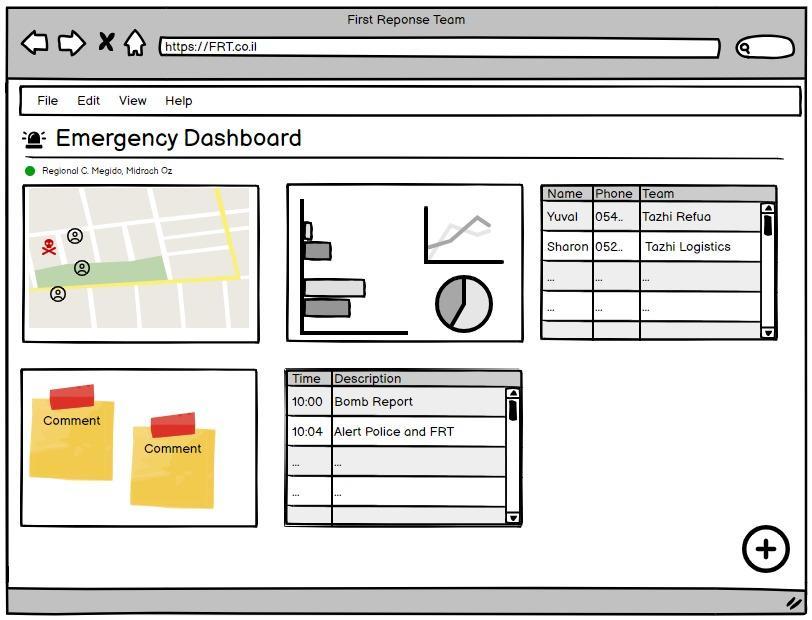


fig 6. application’s dashboard sketch

תמונה שמכילה טקסט, מפה, צילום מסך, קו

התיאור נוצר באופן אוטומטי

fig 7. application’s live map sketch

תמונה שמכילה טקסט, צילום מסך, מפה, טלפון נייד

התיאור נוצר באופן אוטומטיתמונה שמכילה טקסט, מפה, צילום מסך, טלפון נייד

התיאור נוצר באופן אוטומטי

fig 8. application’s emergency alarm interface fig 9. application’s event map

4. **Expected Achievements**

* **Enhanced Coordination:** The application will facilitate real-time coordination among different teams, improving response efficiency during emergencies.
* **Reliable Functionality Offline:** Key features will be accessible without constant internet connectivity, ensuring functionality under varied conditions.
* **Dynamic Role-Based Access:** Customizable interfaces and functionalities based on user roles and permissions, ensuring that each team member has the tools and information necessary for their responsibilities.
* **Mapping and Task Management:** Features for marking significant locations and assigning tasks, enhancing operational awareness and efficiency.
* **User-Friendly Design**: A focus on intuitive navigation and accessibility to encourage adoption and minimize training requirements.

4.1 **Challenges**

* **Data Synchronization:** Ensuring seamless sync across devices in fluctuating network conditions.
* **Interface Simplicity vs. Feature Richness:** Balancing a simple user interface with the complex functionalities needed by different teams.
* **Scalability:** Preparing the system to handle increasing amounts of data and users without performance loss.

4.2 **Success Criteria**

* **User Engagement and Retention:** Targeting an engagement rate of 85% among all users, with feedback loops to continuously refine the application.
* **Operational Impact:** Aiming for a 30% reduction in response times, demonstrating improvement in coordination during emergencies.
* **Stakeholder Satisfaction:** Seeking a 90% satisfaction rate from team leaders and members on the utility and usability of the application.

5. **Testing Plan:**

.

**Scope**

Comprehensive testing of all web and mobile interfaces, backend functionalities, offline capabilities, and user interactions under simulated real-world conditions.

**Objectives**

* Ensure all features function according to specification.
* Validate the user interface for ease of use and accessibility.
* Confirm application performance under normal and peak loads.
* Safeguard sensitive data through rigorous security testing.

**Testing Approach**

We will employ Jest, a popular testing framework, to conduct unit and integration tests across our application. The focus will be on achieving a high level of code coverage, targeting specific metrics to ensure that a significant portion of our codebase is tested. This approach will help us identify and rectify any potential issues early in the development process, enhancing the overall quality and reliability of the application. Additionally, automated and manual testing techniques will be utilized to simulate user interactions and load conditions, providing a robust evaluation of the application's performance and stability in diverse scenarios.

**Test Cases**

| **Test Area** | **Test Name** | **Description** | **Procedure** | **Expected Result** |
| --- | --- | --- | --- | --- |
| **Login Functionality** | Successful User Login | Verify that a user can log in with correct credentials. | Enter username: admin and password: 12345. Press "Login". | User is successfully logged in. |
|  | Incorrect Password Login | Check the system's response to an incorrect password. | Enter username: admin and password: wrongpassword. Press "Login". | An error message "Incorrect username or password" is displayed. |
|  | Re-login Attempt | Ensure the system correctly handles attempts to log in by a user who is already logged in. | Login as admin with password 12345. attempt to log in again in another browser tab. | An error message "User is already logged in" appears. |
| **User Registration** | Complete User Registration | Test successful registration process. | Enter all required details in the signup form and submit. | A success message "Sign up successfully" is displayed. |
|  | Incomplete User Registration | Verify the system's handling of incomplete registration attempts. | Omit the last name in the signup form and submit. | An error message "Need to fill the last name field" is displayed. |
| **Task Assignment and Mapping** | Task Creation and Assignment | Test the functionality for creating and assigning tasks. | Access the task creation form, fill in details, and assign it to a team member. | A message "Task assigned successfully" is displayed. |
| **Emergency Mode Operation** | Offline Functionality Check | Confirm that critical features are accessible offline. | Simulate a network outage and attempt to use critical features. | Core functionalities remain operational. |
| **User Permissions and Roles** | Permissions and Roles Check | Confirm that users can only perform actions appropriate to their permissions level | Simulate a user without permissions and attempt to use a feature requires permission | The user is unable to reach the wanted interface. |
| **Scalability Tests** | Scalability Check | Assess the system’s capacity to handle an increased load, such as many users logging in simultaneously | Simulates high volume of requests | The system is working flawlessly |
| **Usability Testing** | Usability Check | Conduct sessions with real users to identify potential usability issues before the product launch. | Interviewing users to get their opinion of our application | Positive results with action items to improve our application |

**REFERENCES**

**סידור לפי שם משפחה של כותב ראשון**[1] McWilliams, M. (2020). Creating an effective emergency response team. Professional Safety, 65(6), 66-70.‏

[2] 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.

[3] 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.

[4] Boduch, A. (2017). React and react native. Packt Publishing Ltd.

[5] FEMA (Federal Emergency Management Agency). (2022). First Responders.

[6] FEMA (Federal Emergency Management Agency). (2022). Community Emergency Response Team.

[7] <https://www.avuka-app.com/squad/about>

[8] <https://www.avia-app.org>

[9] <https://zello.com/product/frontline-communications/>

[10] <https://nodejs.org/en/about>

[11] <https://expressjs.com/>

[12] <https://www.mongodb.com/docs/atlas/>

[13] https://aws.amazon.com/what-is/elk-stack/

[14] <https://kafka.apache.org/documentation/#gettingStarted>

[15] <https://firebase.google.com/docs/auth?hl=en>

[16][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)