Project Main Field : Software

Project Thematic Field : Natural Disasters and Disaster Management

Project Name (Title) : ADYS - Acil Durum Yardımlaşma Sistemi (Emergency Situation Communication System)

Summary

When it comes to upsetting the social order and leaving massive wreckage in their wake, natural disasters have a remarkable capacity and a high success rate. After major disasters, teamwork and collaboration are essential to conducting efficient search and rescue operations. Because teamwork and collaboration allow faster aid to the people in need. Aiding and abetting saves lives by limiting the damage with its collective capability.

We can say that aiding and abetting, at its core, is a glue that sticks communities and organizations together. Instant and coordinated emergency responses play a life-saving measure when it comes to sending out help.

However, there are a few roadblocks that limit more effective support. One of these roadblocks is the logistics of aid reaching its target.

Broken infrastructure and impassable roadways are some of the factors that make it difficult for emergency personnel to be on the scene in time.

The lack of resources is another issue that makes it challenging to assist. Despite the good intentions of many volunteer communities and organizations providing aid, the effectiveness of aid can be severely limited by a lack of financial resources.

In these types of situations, most of the time, the groups that get affected by the disaster and can't receive help tend to reach out for it on social media platforms while fighting for their lives. But because these platforms weren't designed mainly for these types of situations, it makes it harder to communicate for both the ones in need and the ones that want to help.

My primary goal with this project is to develop a stand-alone system that will enable easy location detection, assistance dispatch, and general communication to make it easier between two groups.

With just a button click, those in need of assistance can send out an emergency signal with the use of the smartphone app. Those who wish to assist can easily see those emergency signals through the website.

My expected result with this project is to make solitary between individuals easier and faster, thus lowering the damages caused by the destruction.

Keywords: Natural disaster, teamwork, communication, aiding, damage, effect, risk

Purpose

My project's primary function is to send out aid signals. This method allows anyone who is in an emergency and is unable to contact the emergency personnels to use a mobile app to send

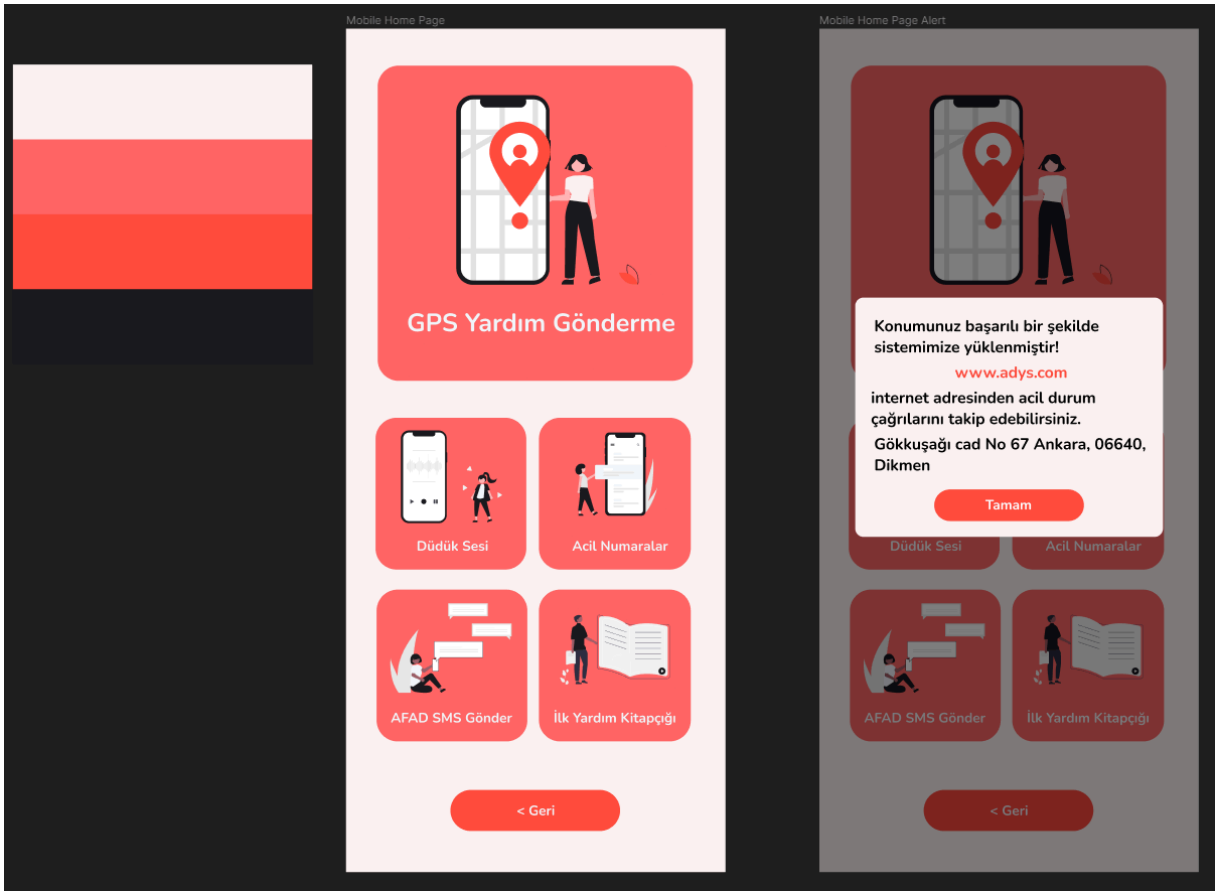
out a help signal that includes their location. Those who wish to lend a helping hand can use the website to filter and view the information of the nearest aid calls to their location.

The main goals of my system are to facilitate much simpler communication with aid organizations, swiftly transmit information to the impacted community, coordinate emergency responses, and supply resources. I aimed to support the search and rescue missions, supply distribution, and other related activities to minimize the damage caused by the disasters.

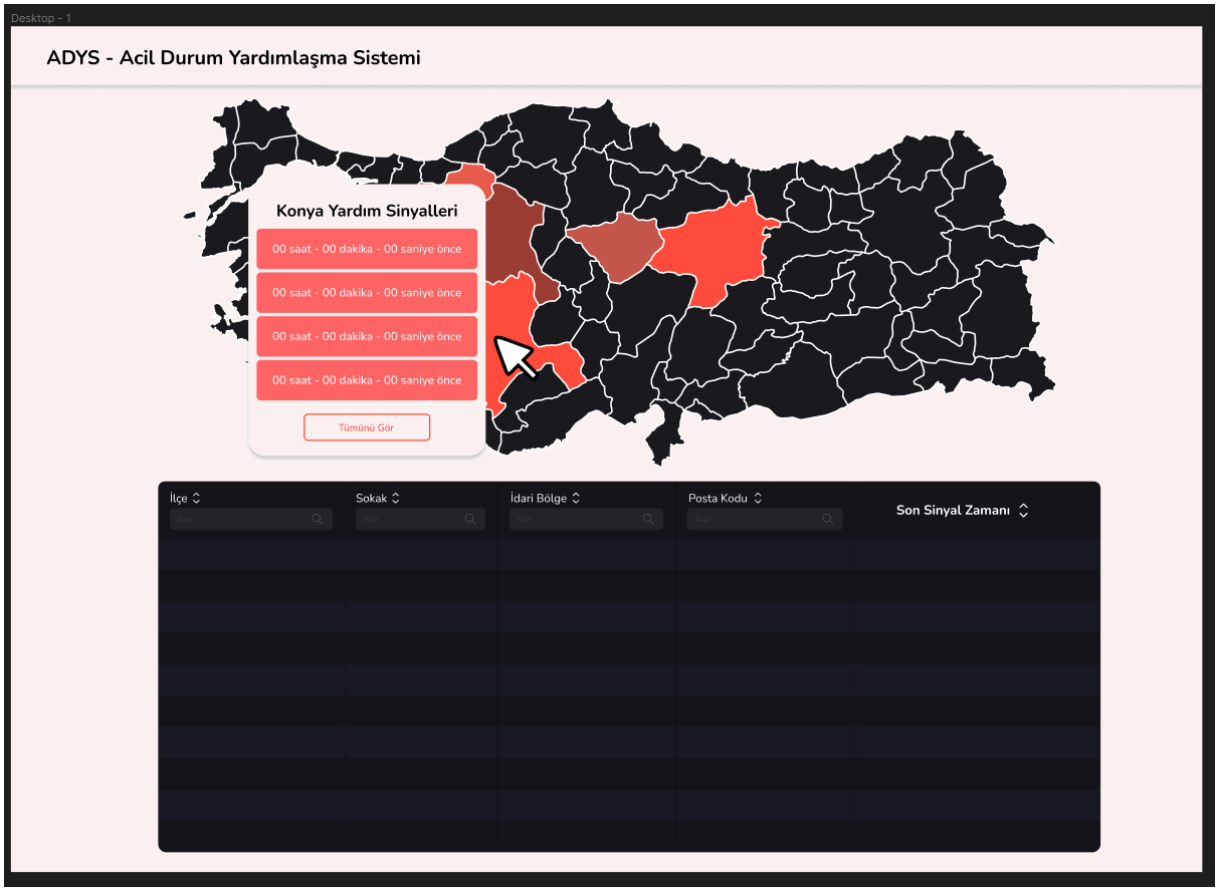
Even though there are already systems made for this purpose, through the poll results and feedback that I got, these existing systems tend to be slow and not very user-friendly with their interfaces. I made it my purpose to create something that puts upon the existing systems and fixes these problems.

How Did I Develop It?

"Yes, insufficient aid results in deaths in natural disasters, but how can I prevent this?" was the first question that needed to be answered. I did an extensive study on the issues people encounter, particularly when it comes to communication during emergencies. I started planning the system after I had an idea. Diagrams showing the interactions between the database, website, API, and app in the background were first made. Following that, a color scheme was chosen, and the website and mobile app designs were made.



Picture 1: Design of the mobile app.



Picture 2: Design of the website.

Made a concerted effort to produce a clean, quick, efficient, and user-friendly design. Ultimately, everything was coded, and a demo was ready.

What Goes On Behind The Scenes?

The four main components of the project are the website, mobile app, database, and API.

The simplest way to conceptualize this system is as a restaurant. We can think of the website and mobile app as the clients from the ground up. As we descend, we may say that the fridge is the database. You can use the database to store data for later use, in the same way that you use it to store ingredients for later use. The API is at the bottom layer. The server and endpoints are the two parts of the API that we may separate. Consider the waiters as endpoints and the server as the chef.

As an example, when a user clicks the send help button on the app, the user being the customer, it gives an order to the API endpoints with the user's location data as its context, location data being the user's latitude and longitude coordinates. Upon receiving your order, the server uses an API from a third party to prepare it by converting the coordinates into the whole

address. After that, it saves the completed data for eventual usage in the database. The user receives the finished data from the endpoint.

Picture 3: Help signal database model.

Technologies That I Used

These technologies were used to produce something as soon as possible.

1) React.js

React is a tech stack framework that belongs to the JavaScript UI Libraries group. It is a front-end JavaScript library for building user interfaces. It makes the website more interactive and faster. React is often used as the V in MVC architecture. It's simple to try out React on a small feature in an existing project because it makes no assumptions about the rest of the technology stack. (Ekanayake, 2021)

React's virtual DOM reduces the need to directly manipulate the DOM, which makes it possible to build websites with effective UI updates and ensuing performance benefits. Together with its declarative syntax, React became a perfect choice for me.

2) Flutter

Flutter is a mobile app development framework created by Google. It allows developers to build high-quality, native-like apps for iOS, Android, and other platforms using a single codebase. Flutter uses the Dart programming language and provides a rich set of pre-built widgets and tools to streamline the app development process. (Solutions, 2023)

Flutter lets me create apps without spending too much time on them because it provides developers with an extensive collection of customizable widgets right out of the box, allowing them to construct stunning and highly customized user interfaces. It would be incorrect to choose Flutter for this project given its rapid development and single codebase, which allow you to create code once and deliver it across different platforms.

3) Express.js

NodeJS is a platform for building scalable network applications. It runs on both the server and client sides, making it ideal for building real-time applications. (AlishaS, 2022)

Express is a web application framework built on the Node.js platform. It is one of the most commonly used frameworks to build web apps and APIs with Node.js. (AlishaS, 2022)

While developing the API, I wanted to use Javascript since it is the most comfortable language I am. To develop something fast, quick, and lightweight, I used the Express.js module. To maintain the API's organization, I've also implemented an MVC (Model View Controller) architectural structure.

4) Geoapify API

I needed to select a third-party API to convert user coordinates to address data. Geoapify became my preferred API since it was quick and simple to use, plus it included a free tier.

5) Database

MySQL is the most popular Open Source Relational SQL database management system that helps users store, organize, and retrieve data. (Prakash, 2019)

My sole motivation for using MySQL was that, in comparison to alternative databases such as MongoDB, PostgreSQL, etc., I felt more at ease using it.

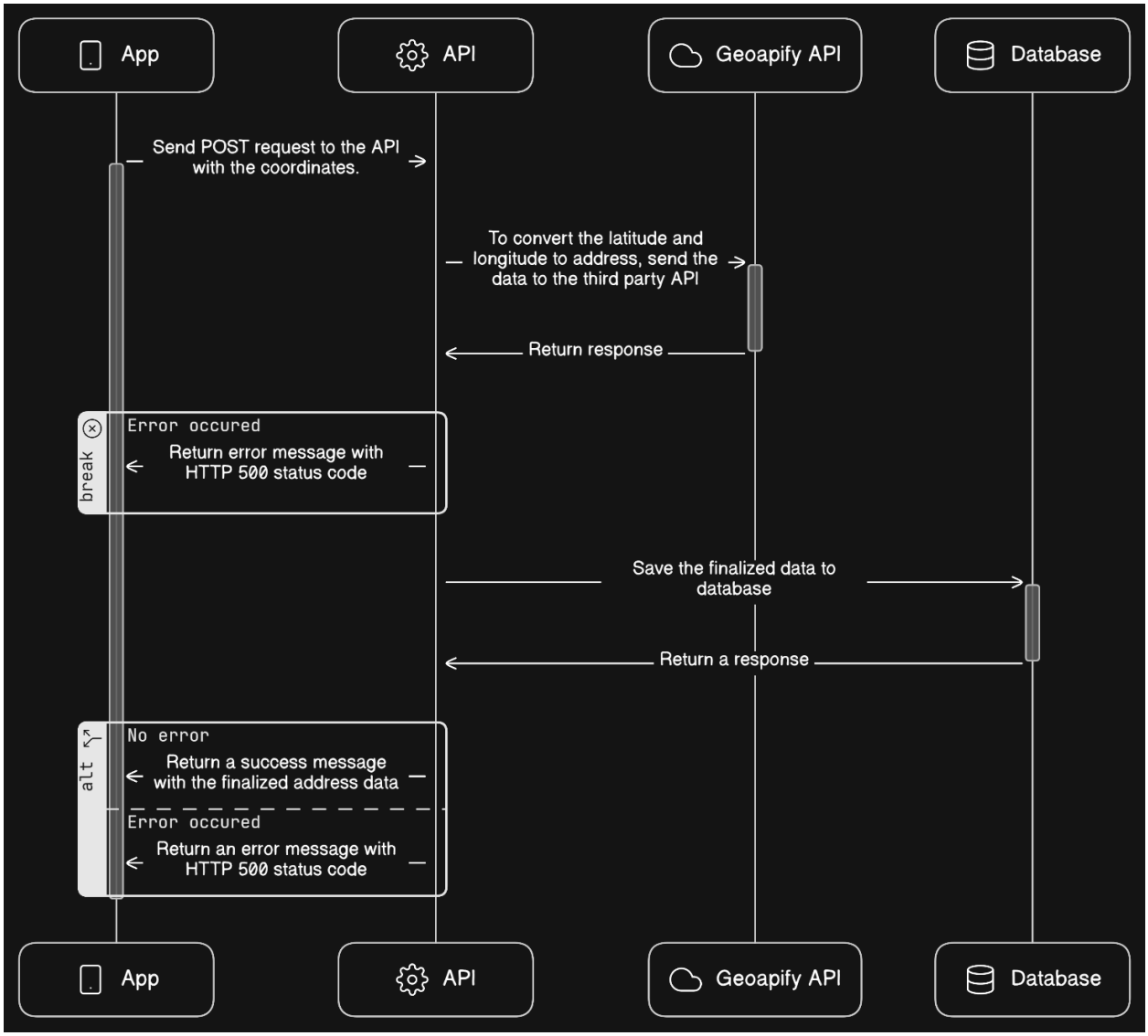
Main Function of The Project

This project's primary goal is to transmit emergency signals. This technique makes it simple for people who were unable to call emergency services to signal for assistance with the use of the app so that others may notice it and get in touch. As well as individuals who want to help those in need can see the emergency signals closest to them with the use of the website.

1) Sending Out Emergency Signal With the App

After you press the send help button, the app sends out your coordinates to the API. If you've sent one before, your last help signal time gets updated. After that, to convert the coordinate data to address, the API sends a request to the third-party Geoapify API. This part happening in the server helps the user send fewer packets. If there are any problems after getting a response, the API returns an HTTP 500 (Internal Server Error). If not, it returns an HTTP 201 (Created) and stores the address information in the database. Lastly, after receiving the finalized data, the app sends an SMS to AFAD with the individual's address.

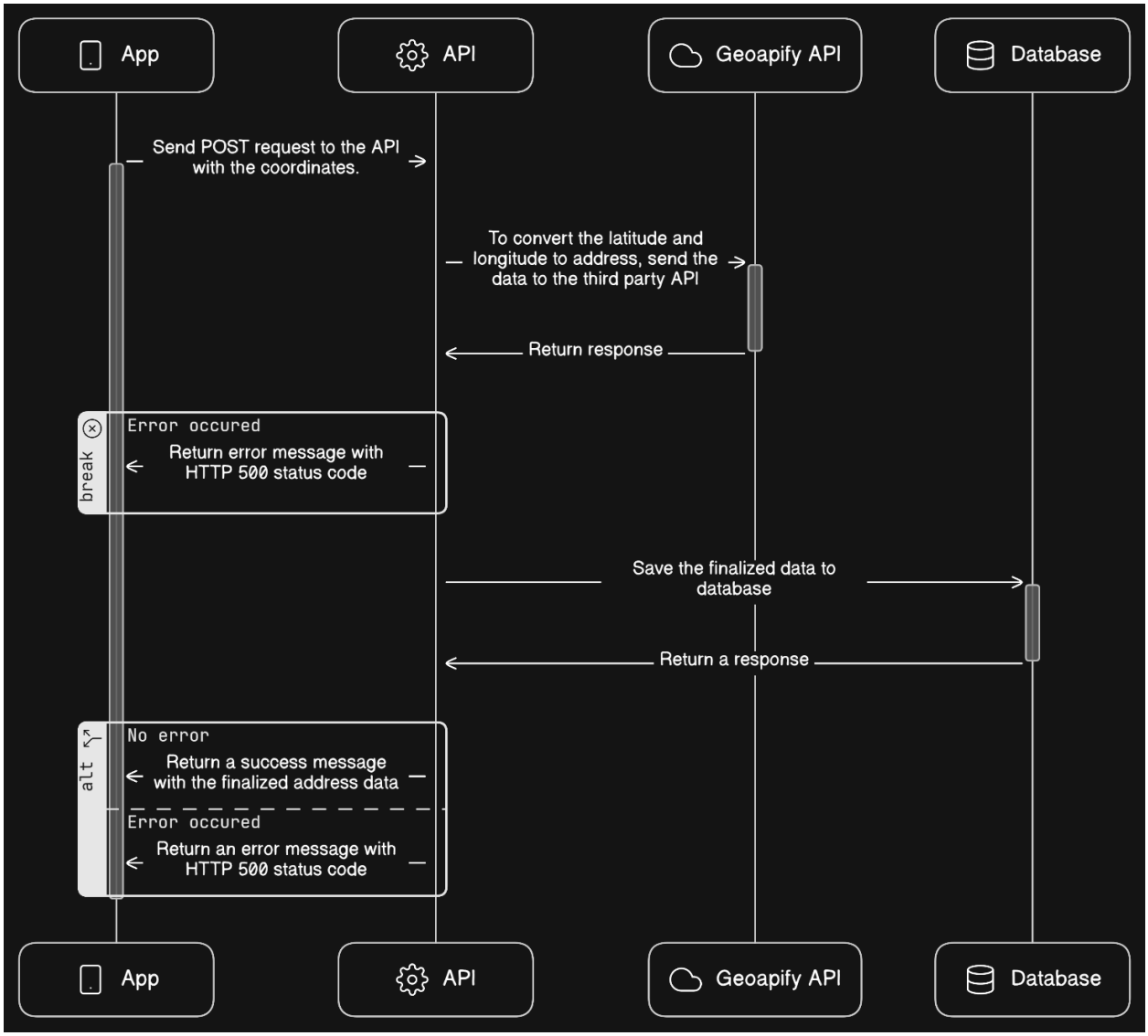
In the mobile app besides the ability to send out emergency signals, there is also a loud whistle sound, a list of emergency phone numbers, and a first aid booklet.



Picture 4: Backend diagram of sending out the help signal.

2) Seeing Who Needs Help With the Website

Upon accessing the website, users will be greeted with a map of Turkey along with filtering options. As the number of emergency signals rises, the provinces on the map become increasingly redder. There's a table with the emergency signals under it. If there aren't any filters used, the nationwide help signals will be displayed. It will display a pop-up window with additional information about the signal when clicked on the rows. There's also a button that opens Google Maps and sets the path to the location of the signal.



Screenshot 5: Backend diagram of the website getting help signals data.

Findings

I presented the system to teachers and students following our school's natural disaster presentation, and I got a ton of great feedback. It has been observed that this method is far more useful and user-friendly than the prior ones. After the problems were identified, attempts were made to enhance the system's functionality.

Conclusion and Discussion

We can use lightweight, quick, contemporary systems made expressly for use in disaster situations instead of depending on social media platforms that weren't intended for aid and

abetting in such circumstances. With these solutions, we can drastically lower the damages that happen within the first 72 hours.

Suggestions

Security is one of the main issues with this system. We are unable to determine if someone sending a help signal needs assistance or is acting maliciously. It is possible to add a video recording feature for this. A mobile application for this feature should be available on the website as well. When a user selects one of the help calls, a video recording will play on the screen. The video may be uploaded to the server in segments after the user hits the record button. After watching these videos, one can determine whether or not there is a security issue.

The results of the third-party API used to translate coordinates to addresses are not precise. APIs such as Apple's location API can help solve this by increasing the accuracy several times.

Sources

Ekanayake, Piumi. (2023, April 17). Introduction to React. Medium. https://medium.com/geekculture/introduction-to-react-75744511253d

Solutions, Grow. (2023, September 15). What is Flutter? A Complete Guide to Google’s Framework. Medium. https://medium.com/@growsolutions/what-is-flutter-a-complete-guide-to-googles-framework-b51ae5e2fbd5

Alisha S. (2022, November 17). What is Express in Node.js. Medium. https://medium.com/geekculture/what-is-express-in-node-js-f51d178056b9

Prakash, Arun. (2019, September 20). A Basic MySQL Tutorial. Medium. https://medium.com/@reach2arunprakash/a-basic-mysql-tutorial-c22b5ad23181