Software Engineering II Project

Jorge Arregoitia, Cristofer Lombardo, Gabriela Haas, Jessica Lourenco

May 7th , 2019

INTRODUCTION

In this report we will address some of the requirements, challenges, and overall achievements of making a Pirate Fighting Crisis Management System web application. This document will serve as a collection of our ideas, struggles, and achievements throughout the development process.

REQUIREMENTS

We built a web application using React, NodeJS, MongoDB, HTML5, CSS3, JavaScript, version control was done through GitHub, and our editor of choice was Atom. Our decision was based on curiosity, desire to learn, and the interests of most team members. In today's ever changing world, web applications are getting infinitely more complex; leading the way to faster, more abstract forms of information delivery. We followed a conventional approach to web development with a modern twist. Our application has both a front end, and a back end, but, this architectural style is masked with a more dynamic style, which is part of one of the technologies used; React.

In a conventional web development environment, the client would handle presentation, the server would have the database, and the logic would be on one of them or on both. While this increases the scalability of the application and separates the display and the database, it still doesn't allow for true specialization of layers, so most applications will outgrow this model, hence the creation of React. With React, data can seamlessly weave from the front end to the back end, in reach of the database. This might seem a bit abstract but it is the foundation of most modern web application today.

Our application can be viewed/accessed through any of the following browsers: Mozilla, Chrome, or IE. I'm positive that thanks to the portability of the application, it could also be viewed on other browsers, but those are the major ones.

Performance wise, the application loads surprisingly fast, even if the user suffers from a slow internet connection, due to it not having a complex interface, or heavy graphics. It was unnecessary for us to time the application loading time since the application is small, and never struggled with volume.

Since our project was mostly developed in JavaScript, we took on a functional paradigm for the application to be easily testable. Most of the functionalities were developed through functions, and React components through classes.

CHALLENGES

We encountered several challenges when developing our application. One of the main ones was our inexperience using modern tools such as React and NodeJS. Throughout development we kept pushing ourselves to the limit, trying to get a better understanding of the technologies being used. We ran into a bump trying to get React to render HTML, until we learned about JSX. With JSX we were able to combine the power of HTML and JavaScript to create truly dynamic elements. We also ran into a design problem when it came to the views, more specifically the user view. We discussed the benefit of removing or adding certain elements, and decided on a few that would improve the user's experience overall.

Following the learning and design challenges came a series of development challenges, like the one mentioned previously, that hindered our ability to profoundly create a quality application. We had trouble with npm (Node Package Manager) and other necessary tools for development. Between other classes and projects it took us the better part of a week to fix those issues. Once that was done, one of our teammates ran into GitHub issues, but we were able to keep on learning and coding in the meantime.

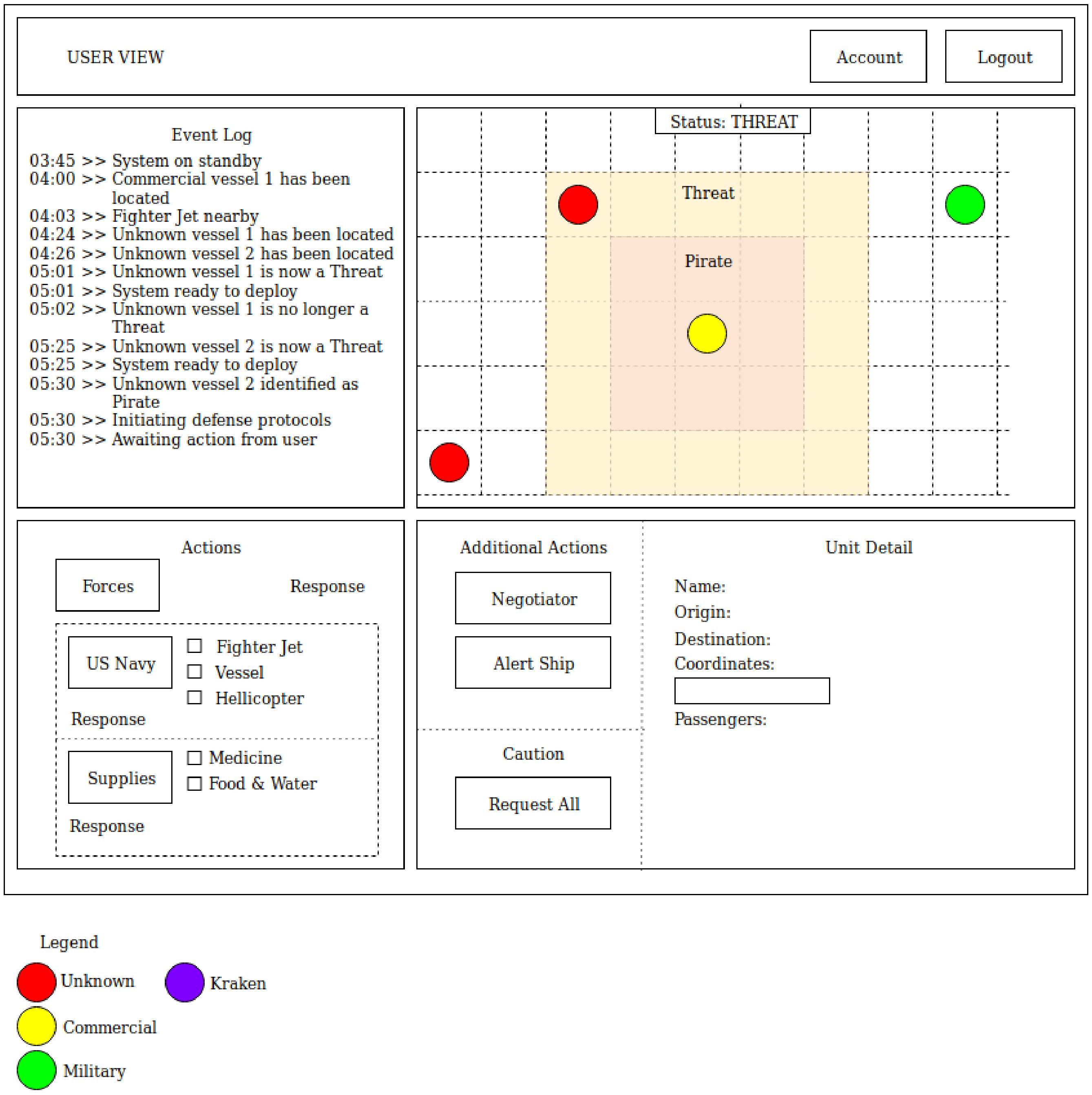
Our biggest challenge, one that we carried to the finish line, was our inability to figure out React's dynamic rendering. We would send data from the server but the front end would not render the data properly unless the user refreshed the page. This was a huge set back as our application was lacking comfort. We decided to focus our attention instead in generating the right data that we wanted and successfully sending it to the right React component, whether rendered dynamically or not.

The application itself was easy enough were only a couple dozen lines of code would have met qualifications, but we needed the application to be more complex. Implementing the AI component was another challenge that increased the difficulty of the project. The AI component was a single JS file that communicated with the main React component but at first we didn't know how to make it communicate. Thankfully there are many tutorials online on how to make JS files communicate across the development spectrum seamlessly, without bugging down performance with nonsensical requests.

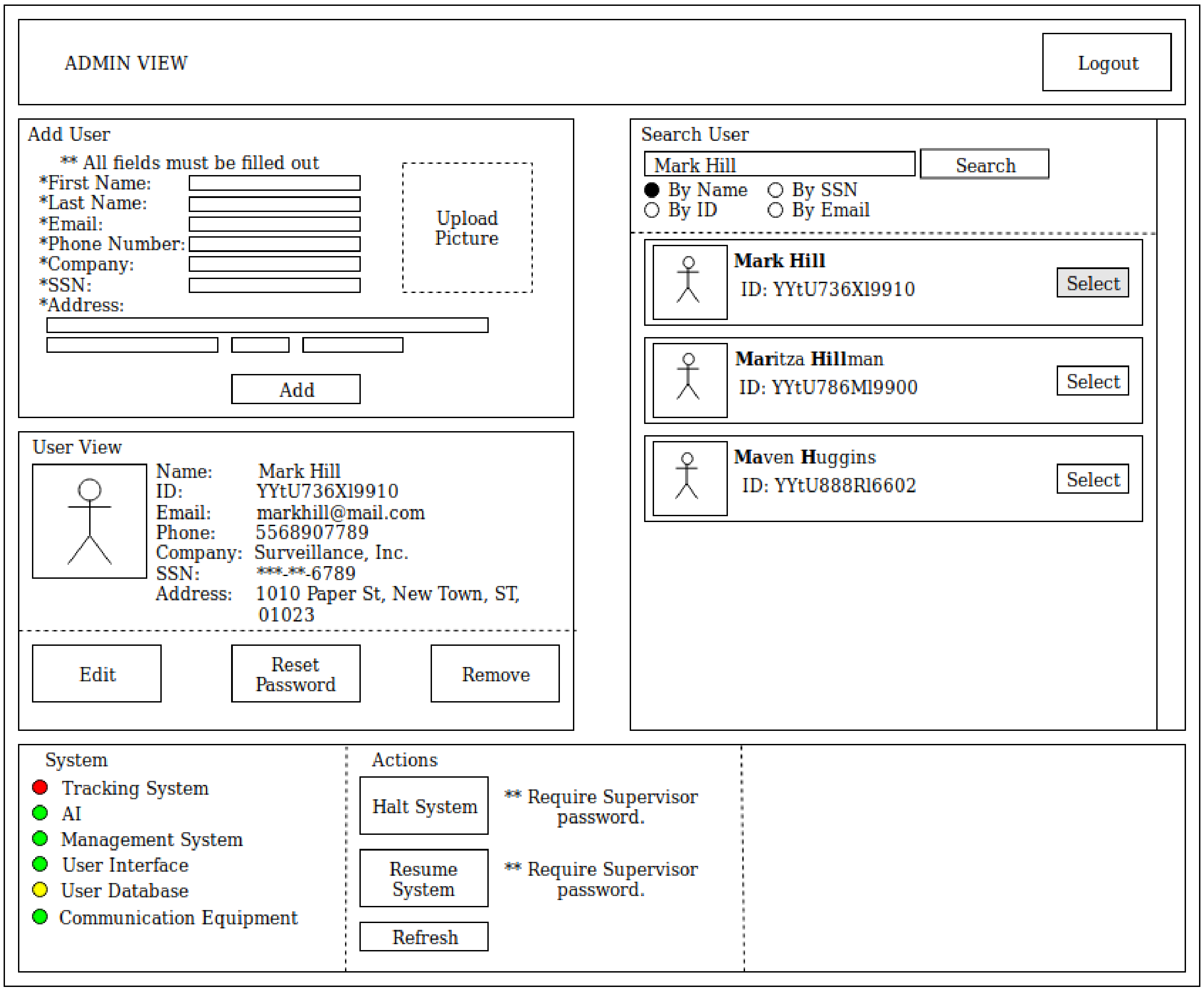
Lastly, after having worked on the application for nearly a month we started to work with the database. That did not prove to be difficult, however, we decided not to take security too seriously in this project, even though we know it's a very serious issue. Our passwords were not encrypted, nor salted, making the application quite volatile. We also discovered several critical bugs between the views that we managed to patch before the presentation.

ASPIRATIONS & ACHIEVEMENTS

Our first design for the Pirate Fighting Crisis Management System web application looked like this:



```
03:45 >> System on standby
04:00 >> Commercial vessel 1 has been
         located
04:03 >> Fighter Jet nearby
04:24 >> Unknown vessel 1 has been located
04:26 >> Unknown vessel 2 has been located
05:01 >> Unknown vessel 1 is now a Threat
05:01 >> System ready to deploy
05:02 >> Unknown vessel 1 is no longer a
         Threat
05:25 >> Unknown vessel 2 is now a Threat
05:25 >> System ready to deploy
05:30 >> Unknown vessel 2 identified as
         Pirate
05:30 >> Initiating defense protocols
05:30 >> Awaiting action from user
```


ADMIN VIEW

Logout

Add User

** All fields must be filled out

*First Name:
*Last Name:
*Email:
*Phone Number:
*Company:
*SSN:
*Address:

Upload Picture

Add

User View

Name:	Mark Hill
ID:	YYtU736X19910
Email:	markhill@mail.com
Phone:	5568907789
Company:	Surveillance, Inc.
SSN:	***-**-6789
Address:	1010 Paper St, New Town, ST, 01023

Edit · Reset Password · Remove

Search User

Mark Hill · Search

● By Name ○ By SSN
○ By ID ○ By Email

Mark Hill
ID: YYtU736X19910
Select

Maritza **Hill**man
ID: YYtU786M19900
Select

Maven **H**uggins
ID: YYtU888R16602
Select

System

🔴 Tracking System
🟢 AI
🟢 Management System
🟢 User Interface
🟡 User Database
🟢 Communication Equipment

Actions

Halt System — ** Require Supervisor password.

Resume System — ** Require Supervisor password.

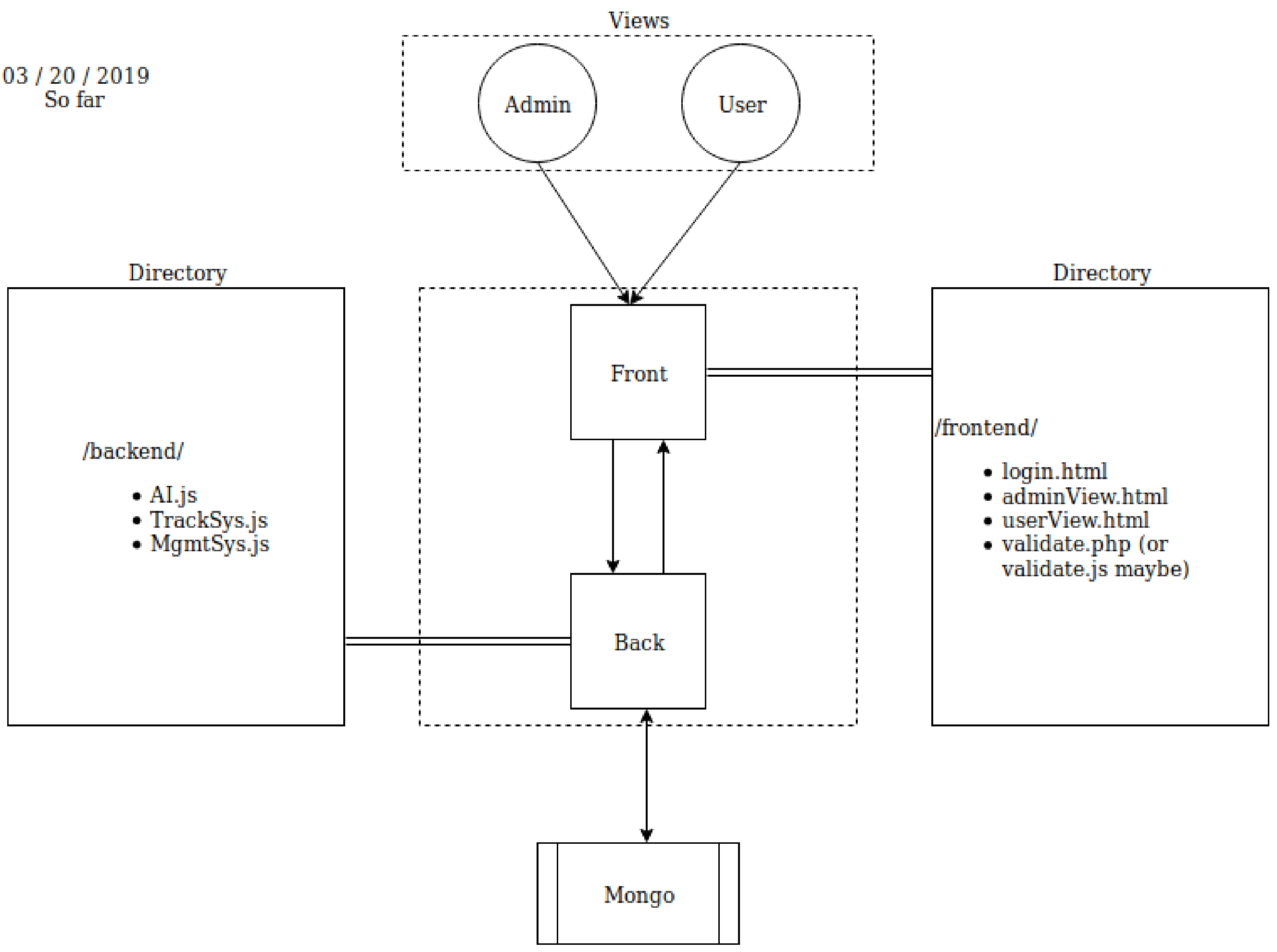
Refresh

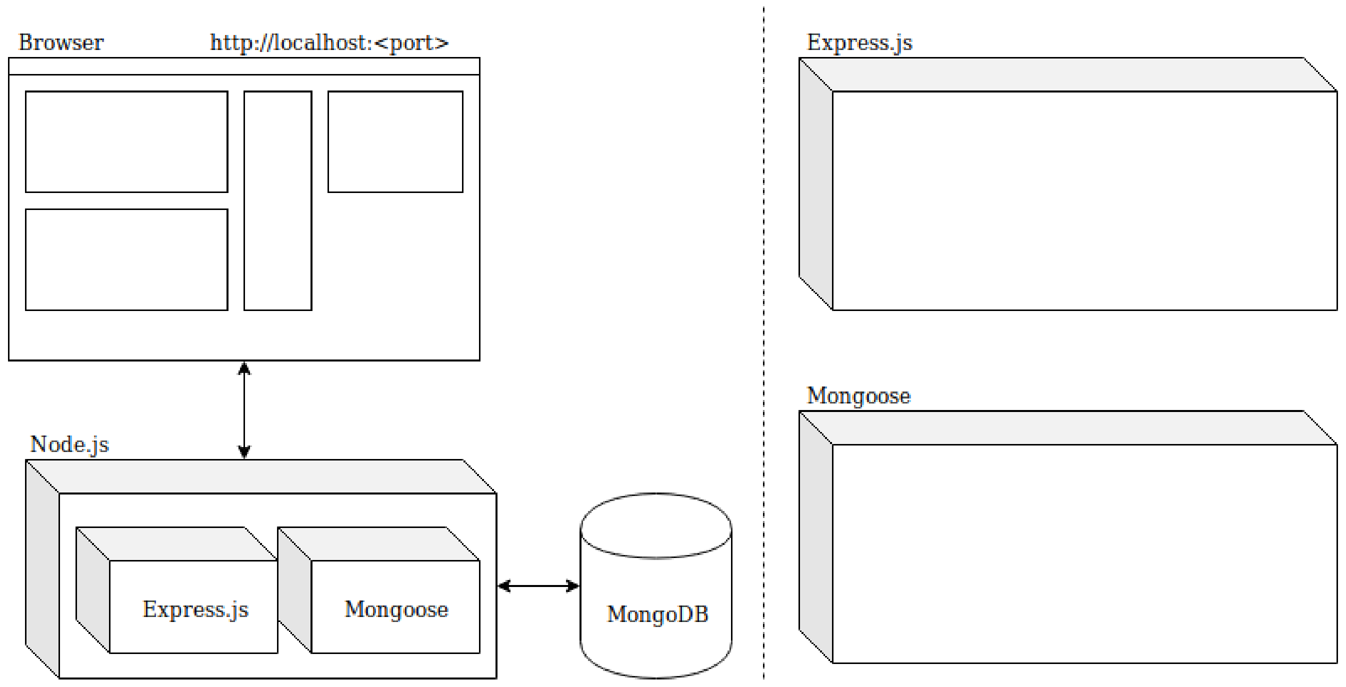
These were the original "templates" for the user and admin views respectively. We wanted to convey a sense of urgency and seriousness through the placement of certain elements and their purpose.

We agreed to not focus too much on the appearance of the application but instead to learn the technologies we were going to use, and focus on the back end first.

These were the first project templates developed as a guide and visual representation of what needed to be done. We ended up moving away from what's expressed in the pictures. By maintaining good communication we were able to not make more of these diagrams as they were time consuming and unnecessary passed the early stages of development.

03 / 20 / 2019
So far

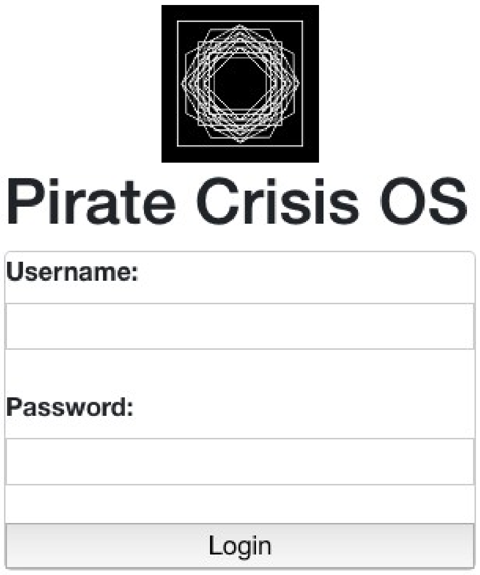




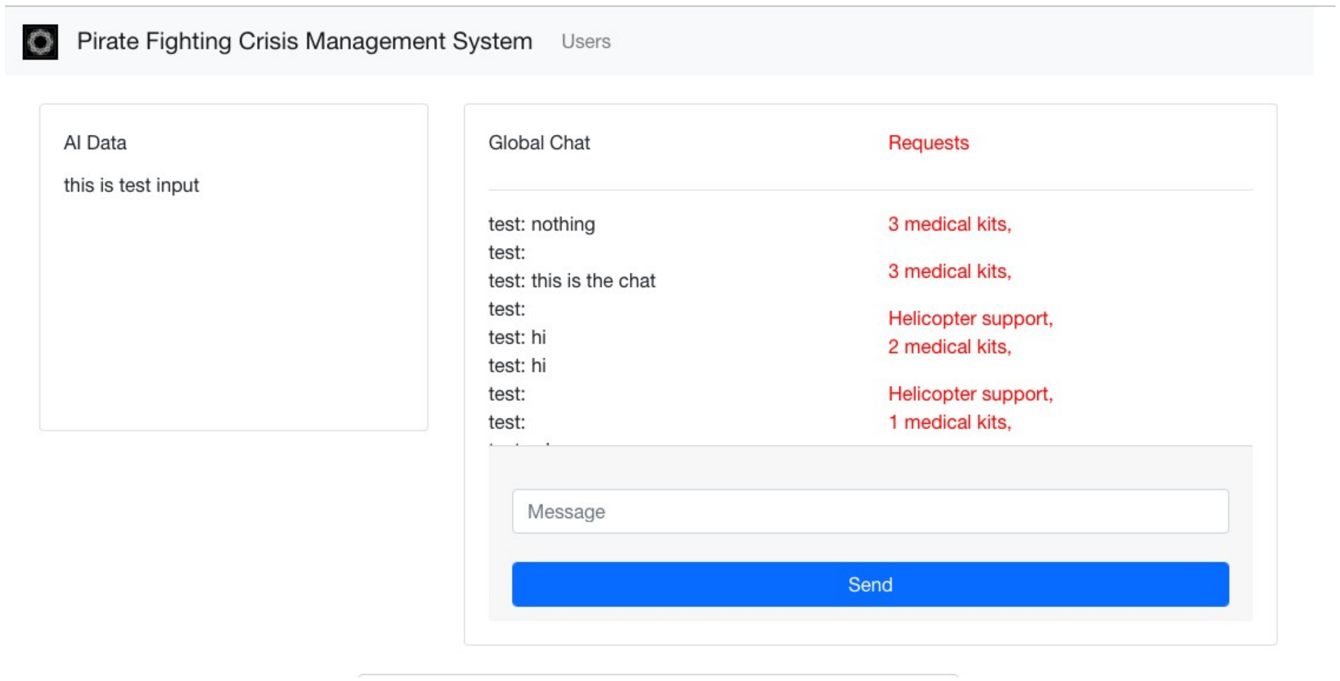
We ended up using React, not Express, which was a good decision since React is more flexible and robust.

Regardless of what we wanted to do, we are happy with the results. We learned a lot and though it was difficult, we finished. It took work and determination and that by itself is a great achievement. Here's our end product:

Login



Management System



AI Data

this is test input

Global Chat

test: nothing
test:
test: this is the chat
test:
test: hi
test: hi
test:
test:

Requests

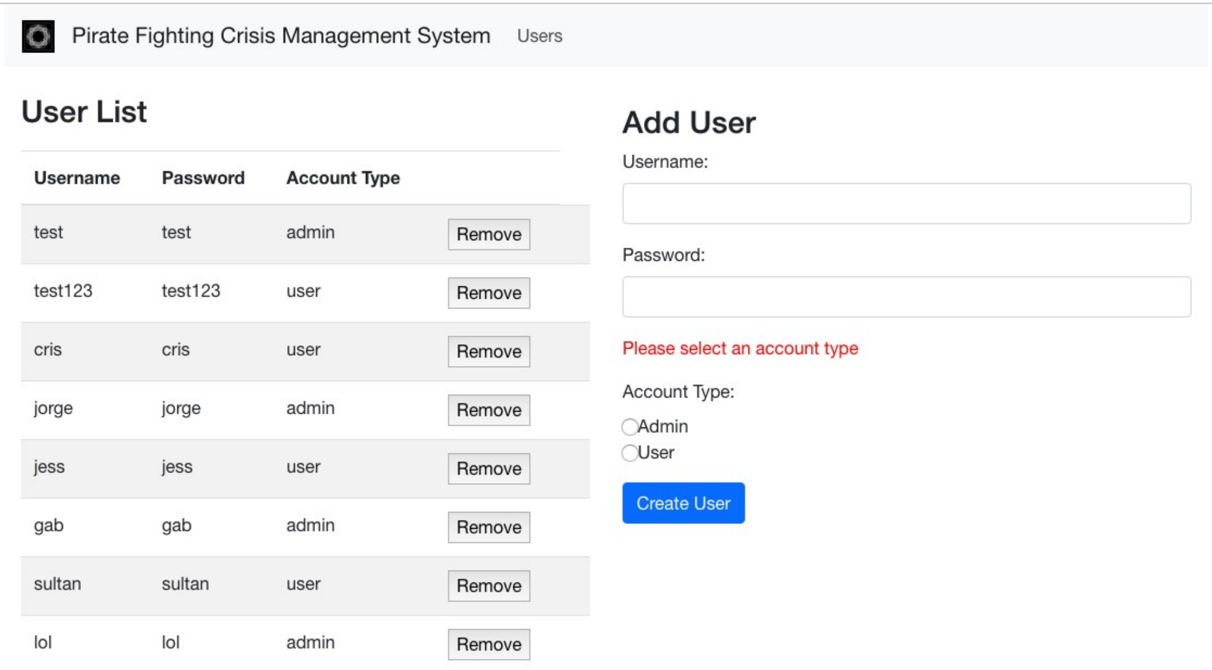
3 medical kits,

3 medical kits,

Helicopter support,
2 medical kits,

Helicopter support,
1 medical kits,

Admin tab



Pirate Fighting Crisis Management System Users

User List

Username	Password	Account Type	
test	test	admin	Remove
test123	test123	user	Remove
cris	cris	user	Remove
jorge	jorge	admin	Remove
jess	jess	user	Remove
gab	gab	admin	Remove
sultan	sultan	user	Remove
lol	lol	admin	Remove

Add User

Username:

Password:

Please select an account type

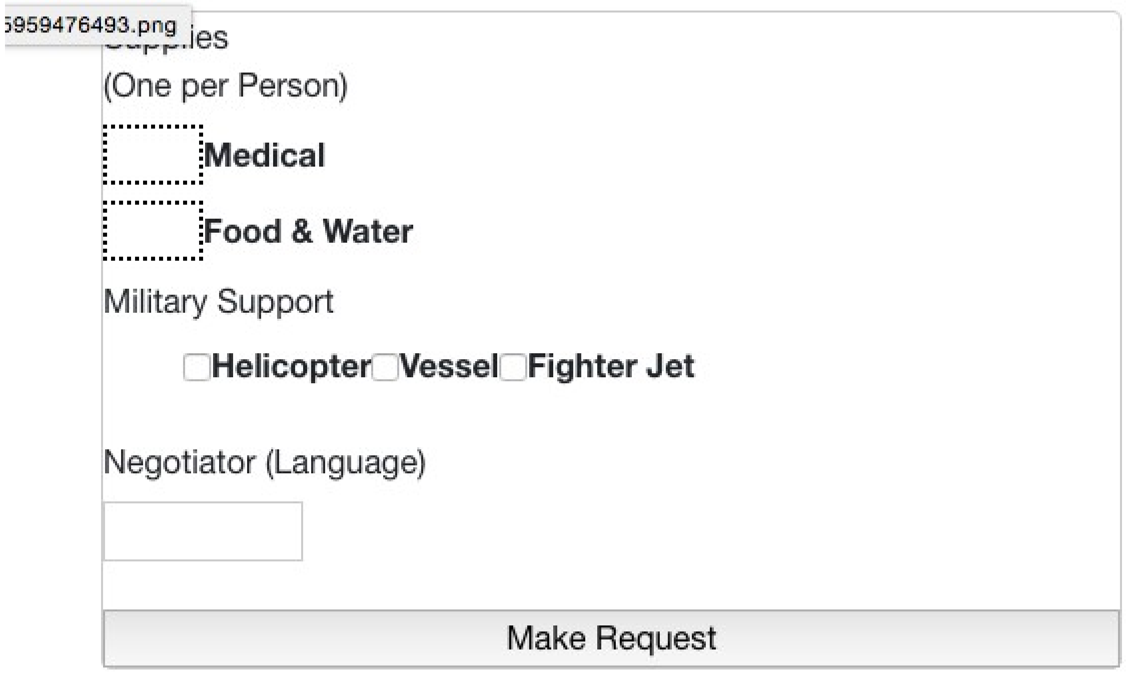
Account Type:

Admin

User

Create User

Request options



5959476493.png

Supplies

(One per Person)

Medical

Food & Water

Military Support

Helicopter Vessel Fighter Jet

Negotiator (Language)

Make Request

This was a huge learning experience for all of us. Some of us are interested in web development, some of us are not, but we all learned something new. These technologies are the future of applications, especially React. Most modern cross platform mobile applications are written in React Native. This entire project was an achievement for us and we wouldn't have asked for a better end product. Could it be improved? You betcha, but it was the learning experience that's worth a lot to us as developers.

RESPONSIBILITIES

Cristofer	-	Management System, Front & Back end logic, Database, React components
Jorge	-	AI stub, React components
Gabriela	-	Front-end User view, React components
Jessica	-	Front-end Admin view, React components