

# Developing a Web Mapping Application for Collaborative Database Management

Author: Travis Bock

## INTRODUCTION

The United States Environmental Protection Agency's (EPA) Emergency Response group is tasked with preparing response plans in preparation for major hazardous material spills; including, but not limited to, oil. These plans are grouped into distinct geographic areas and are thus called Geographic Response Plans (GRPs). At this time these plans are only maintained in text form. The plans are written up using Microsoft Word and filed as hard paper copies or saved as PDFs to be retrieved in the event of an actual emergency. While this has worked in the past, there has always been the issue of revising these plans as the situation changes in the real world. Additionally, these plans often require input for many different subject matter experts and stakeholders that work inside and outside of the EPA. Since EPA regions cover large geographic areas these experts can be in many distant locations from the Emergency Response group that is tasked with the development of these plans. GRPs are spatial by definition and therefore I believe the utilization of a Geographic Information System (GIS) can be of great service to the development and maintenance of these GRPs.

## GEOGRAPHIC RESPONSE PLANS

Geographic Response Plans cover a general geographic area and are comprised of several distinct components. The size of these geographic areas vary since they "may be defined by geographic features, jurisdictional boundaries, or both, at the discretion of the [Area Coordinator]" (US EPA, 2013). The area they cover is general and is often identified only by a common name such as "Feather River" or "Guam." Once an area has been identified additional experts and stakeholders can be identified based on the region(s) covered by the GRP. "Within the [GRP] boundaries, sub-areas may be defined where there are unique circumstances that require tailored response strategies." (US EPA, 2013).

The EPA covers all of the United States plus outlying islands, either directly or through grant funded local government agencies. Due to this large coverage area, the individuals that are involved in building these plans are dispersed over a very large area. This community of planners, first responders, and subject matter experts collaborate to accurately create the overarching incident action plan as well as location-specific sensitive site strategies. Without these plans the first responders to environmental incidents would be unprepared. Perry and Lindell (2003) define preparedness as "a state of readiness to respond to environmental threats". The EPA Emergency Response teams know the importance of planning and be ready for the worst.

The two main components of a GRP, within which the content is contained, are *Sensitive Sites*, represented as points, and *Incident Action Plans (IAP)*, represented as polygons. Sensitive Sites are locations inside of the geographic area that is covered by the response plan that are of significant interest and/or unique and sensitive environments. IAPs provide a starting point for first responders and environmental response experts to help contain, mitigate and clean up hazardous material discharges. IAPs at their most fundamental level are a list of goals and the steps necessary to meet these goals.

The sensitive sites components of these plans are represented by points in space with textual descriptions of the sensitive areas. Sensitive Sites can be identified because they are critical infrastructure, environmentally sensitive, culturally sensitive, high-risk locations or areas highly impacted from natural disaster (US EPA, 2013). For example, the sites could be previously identified as nature preserves, turtle nesting areas, historical archaeological sites, well heads and other areas needing protecting in the event of a spill. After identifying sensitive sites, strategies are developed to aid in protecting them. Strategies identify the equipment needed to protect a given sensitive site; this usually includes containment booms and the boats and anchors that may be needed to deploy and secure them. For any given site, multiple strategies for booming may be developed to handle several potential conditions on the ground (eg. flood vs ebb tide). The containment booms and equipment needed to deploy them may be pre-staged in the event of an emergency. While a strategy tells you what do you with this equipment it does not tell you who stores it and where to find it. This is where an IAP, whose geographic area encompasses the sensitive site, comes in.

An IAP contains a large amount of information needed to quickly respond to potential environmental disasters, including who has the equipment to respond to a spill and where it is stored. IAPs are created for areas that may be particularly at risk to spills. This may be because storage of oil or chemicals is nearby or transported through the area. The topography and types of potential spills materials are components that determine the area that a IAP may cover. Since a GRP can cover a varying amount of land it may contain one or many of these types of areas, so several related IAPs may be developed. The geographic area that an IAP covers, represented by a polygon, is not defined with a high level of precision. Instead, the area is loosely drawn based on the individual planner's knowledge of the area, and information contained in the plan may be applicable to environmental incidents nearby the area. While many components of an IAP may be the same or similar as another IAP, such as objectives that cover safety, public information/notification or documentation, other parts may not be able to be defined until the geographic area covered by the IAP has been defined.

There are many different components to an IAP but a major part is the identification of predefined groups or organizations (eg. local environmental protection, fire and police departments and local and state governments) and assigning them tasks in aid in the effort of achieving the objectives laid out in the IAP. Another important component of IAPs is a specific medical plan for the geographic area. This includes area medical facilities and services as well as on site medical staffing needs. While an IAP contains general and overarching objectives with accompanying steps to achieve them it contains a comprehensive contact list of people and entities relevant to any response in the defined area. The IAP also contains a starting point

Incident Command Structure (ICS), similar to an organizational chart, that defines who reports to whom during a response. Just like many parts of the GRP in general this command structure will likely be adjusted once a response has begun.

An IAP is created using approximately 10 different types of Microsoft Word forms. Information in some forms is the same or related to data found in a different form. While other forms contain nearly identical information in all IAPs. Several of these forms are specific to an organization and thus the plan may contain multiples of these forms to cover multiple organizations. Additionally, until now, the only mapping components of an IAP included a very simple overview map and references to NOAA nautical charts or USGS Topo maps that covered part or all of the geographic area the IAP covers. After reviewing and discussing these forms we broke them down into a entity relationship diagram (ERD) that helped us visualize and understand how the forms interacted with each other. Figure 1. shows the ERD that was developed after analyzing and existing IAP.

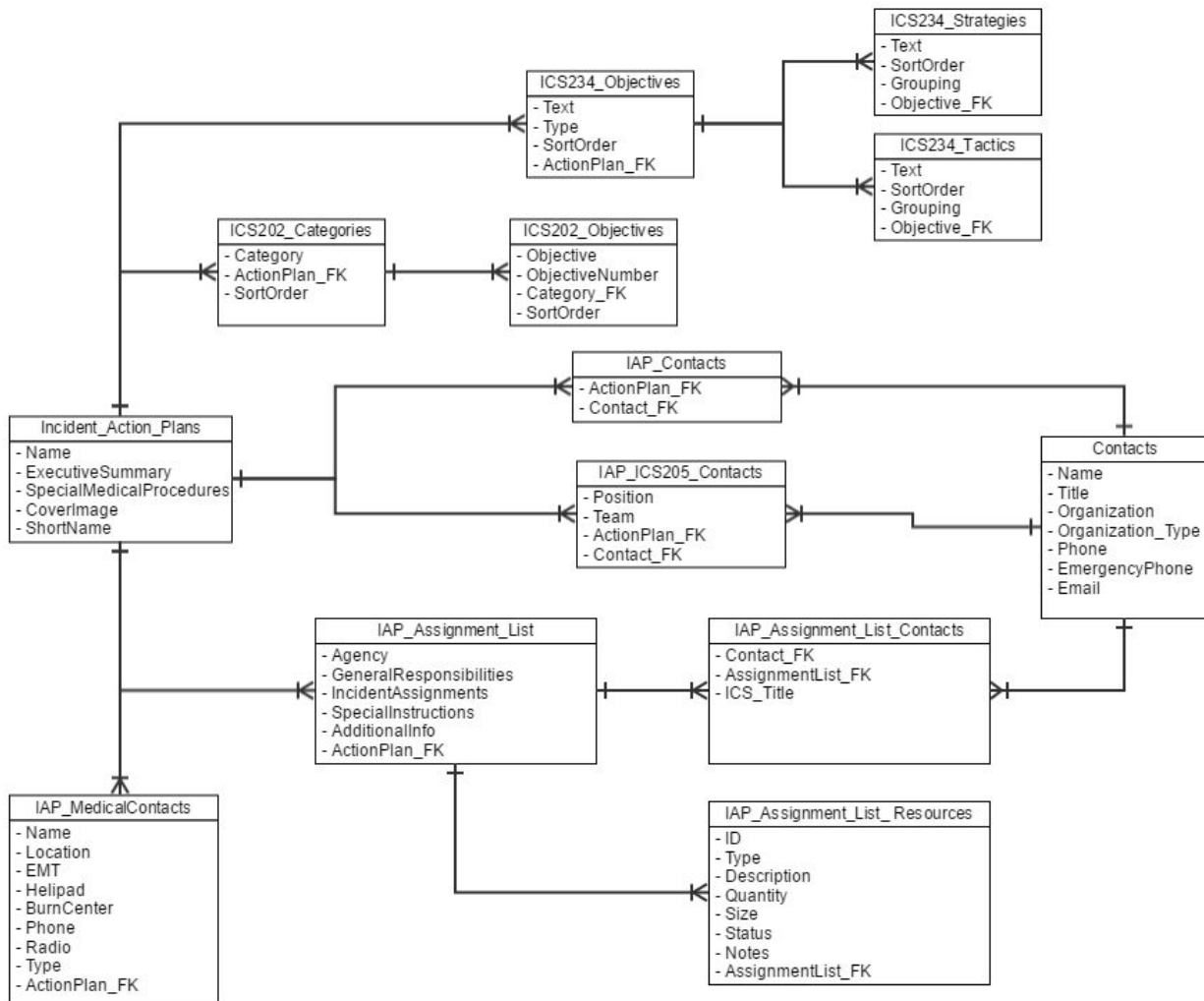


Figure 1. ERD of Incident Action Plans based on analysis of existing IAP

The ERD in figure 1 shows how different components of an IAP relate to each other. Near the middle left of the diagram we have the Incident\_Action\_Plans feature class. This is the single component that ties everything else to the polygon that represents the area covered by

the IAP. The forms contained in an IAP are covered by the tables that branch off of this feature class. Some forms were easily covered by a single table like IAP\_MedicalContacts, seen in the lower left, while others required multiple tables to store all the information. For example, there is a form in an IAP known as a Work Analysis Matrix that lists objectives along with strategies and tactics for each objective. The three tables in the top right corner prefixed with ICS234 were created to store all this information. Lastly, on the right we have a table for a list of Contacts. This contact list is referenced in many places in an IAP since many individual's information is repeated in multiple locations of a single IAP as well as other IAPs, allowing staff to only need to update a contacts phone number in a single location instead of across many forms in many different IAPs.

## APPLICATION NEED

The EPA has been responsible for developing these plans since at least the passing of the Oil Pollution Act of 1990 (US EPA, 2013). Technology has changed drastically since the 1990's and continues to do so. In the earlier days, these plans were likely typed up, printed and stored as hard copies in filing cabinets. Today, the way the plans are stored is in slightly less static formats, such as Word Documents that have been converted into PDFs. GIS is being incorporated to varying degrees, but often in a disjointed and disconnected way. Separating GIS data from the text data in PDFs makes it difficult to keep everything consistent and up to date, and impedes the ability to visualize information spatially. Furthermore, this disjointed method of storing the plans increases friction within the continual revision process that is necessary to keep these plans accurate and up to date.

As part of the planning process, EPA prepares a collection of documents that make up the sensitive site strategies and IAPs in order to be able to respond quickly to an unexpected spill. In addition to the various goals and directives that comprise these documents they also contain a lot of information that is bound to change overtime. These pieces of changing information are mostly directly linked to available resources, including personnel, facilities and materials. Since these and other parts of the plan are always in flux "planning and training must be continual processes in order to establish and maintain emergency preparedness" (Perry and Lindell). A web based system that allows multiple individuals to share in the responsibility of maintaining plans as well as plans not being stored in a static flat file state all encourage this continual revisioning of the plan.

As previously mentioned, these plans are created by and for multiple individuals and organizations from all over the country. This is necessary because of the diverse nature of the areas for which the plans are being prepared. This diversity requires an equal or greater diversity of experts to help guide the development of these plans. Currently, emergency response planners at EPA have to reach out to the community of people involved in this planning and gather information. While the internet and email have made gathering data from many sources a lot easier the core planners still have to compile them into a single document. By sharing this tool with this community of people tasked with developing these plans we can reduce some duplication of data entry effort (Somers, 1998).

By creating a tool that is web based and allows secure multi-user access, these planners no longer have to spend time compiling so much information from many different sources. By giving stakeholders and subject matter experts direct access to quickly and easily edit sensitive sites and IAPs the GRP is more likely to be revised as the situation on the ground changes over time. With a web based tool participation will increase and “lead to strengthened organizational relationships and thus improve post-disaster action“ (Kapucu, 2008). Since the tool is fully integrated into a GIS many efficiencies can be achieved. For example, we can quickly and automatically generate various types of maps, collect water level data from the U.S. Geological Survey and overlay other valuable spatial data to aid with planning and response. Integrating the tabular data of a GRP along with their natural spatial components gives us many immediate as well as many potential unknown benefits in the future.

## APPLICATION DESIGN

Over the years many different technologies have been used to build web applications. Some of these technological options included Silverlight and Flex. Today the best available option is asynchronous JavaScript; often referred to as Ajax. However, “Ajax isn’t a technology. It’s really several technologies, each flourishing in its own right, coming together in powerful new ways”. (Garrett, 2016) Ajax was originally used to send and receive data in the extensible markup language (XML) format but the more common format now is JavaScript Object Notation (JSON). XMLHttpRequest, another technological part of Ajax, is what enables the ability to transfer data between server and client using the Hypertext Transfer Protocol (HTTP) as part of the representational state transfer (REST) style application program interface (API). RESTful services, as REST based APIs are often referred to, follow a loose standard for allowing remote devices to read and write data on a server. Using standard web technologies, like asynchronous JavaScript and REST, we can create an application that does not need to reload the entire page each time something changes, data needs to be read from, or written to the server.

Asynchronous JavaScript is a broad term that refers to JavaScript code that reads or writes data from a server without reloading the web page. The company at the forefront of much of this has been Google (Fawcett, McPeak, & Zakas, 2007). As such they have developed many libraries to utilize the Ajax methodology. Thankfully, Google has open sourced several of the their internally used JavaScript projects. I am referring to AngularJS and the accompanying Angular Material. AngularJS allow individuals and teams to quickly create usable and powerful web applications. The Angular Material library enhances the power of AngularJS and provides cutting edge user interface components. By using this framework we can rapidly develop beautiful applications at a fraction of the cost of developing from scratch.

This combination of technologies to send and receive data to a server without reloading an entire web page is what allows us to create web applications that are much more responsive and faster than traditional web sites. Without Ajax the current generation of click-and-drag web mapping tools be impossible. This is because, prior to Ajax, each time the end user interacts

with the map a call to a server is made and the entire web page reloads. For example, each time an end user wanted to see a different part of the map they could click an arrow button and the entire page would reload displaying the desired new map view. This process was very slow and cumbersome.

To fully develop the GRP application we decided to combine all of the technologies mentioned above by using AngularJS for most of the user interface (UI) components of the application and an ArcGIS Server for accessing the spatial data stored in a database on the server. Esri's ArcGIS Server provides web mapping services through a well documented RESTful API (Esri, n.d.). While it is possible to develop JavaScript code to communicate with the server from scratch, Esri has already provided most of the JavaScript needed to interact with their RESTful API in the Esri JavaScript API. We utilized the Esri JavaScript API to render the web map and as part of an AngularJS factory to interact with the map services and passed the data to the AngularJS framework to handle the UI and data validation.

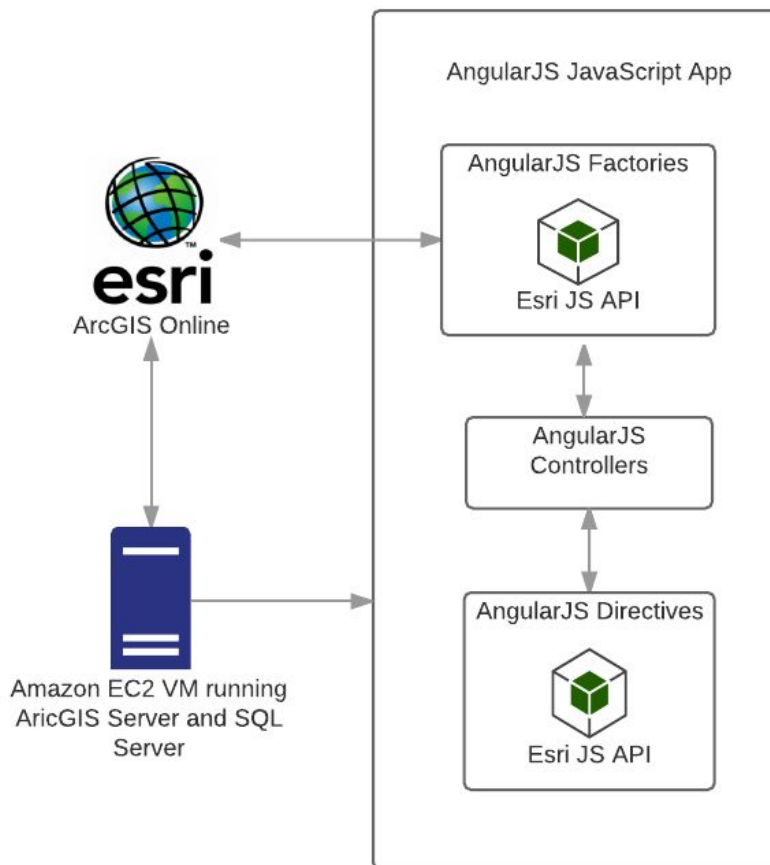


Figure 2. Diagram of interaction between server side and client side.

As can be seen in figure 2, data flows from the ArcGIS Server through ArcGIS Online (Esri, n.d.) to the JavaScript application. The EPA currently utilizes ArcGIS Online for a variety of mapping needs. For our application, the only reason the data flows through ArcGIS Online is to validate the user and to check if the user accessing the feature service has the permission to do so. Since EPA heavily utilizes the ArcGIS Online, they have implemented federated services integration so staff can use their existing Active Directory credentials which they use every day

to log into their work computers. Federated services is a technique to connect existing separate components, like a user login system, with a database. Additionally, ArcGIS Online can be used to securely share web mapping services to anyone outside the organization who has been given the proper permissions. Once the user is authenticated and verified to have the correct permissions the application can start requesting data from a feature or map service. Esri has exposed these services for use by developers.

Since Esri's JavaScript API is specifically designed to send and receive data from an Esri ArcGIS Server we will use their library. However, for our purposes we will wrap these portions of the Esri JS API in AngularJS factories. AngularJS factories are a core component of the AngularJS framework. As their name suggests these factories are used to generate custom JavaScript objects and help develop the application using an object-oriented approach. In this specific application we are creating objects that have methods to save any edits made by the end user. This functionality is achieved by calling the appropriate components of the Esri JS API to read and write data on the server when we ask the object to do so. This approach allows us to easily reuse code where possible.

AngularJS controllers and directives are where we start to interact with the user interface. Once a controller receives an object from a factory, it can send it through an HTML template that will be rendered in a web browser for the end user. In instances where we need to reuse the same combination of JavaScript and HTML repeatedly we can place that code into a directive. For our application we will need to develop a directive that wraps the actual map component of the Esri JS API. We are placing the map in a directive since it will be used many times and is a core visual component of the application. For most of the standard data entry portions of the application a controller will simply send the JavaScript object to a template that will allow the end user to edit data. This is where we will validate data, as it is being entered, when possible.

The data validation will have multiple components. First, the ArcGIS Server provides some limited metadata about each attribute. For example, it provides the maximum length of a string field. We used this information to notify the user when they've reached the limit before the data is even sent to the server. Another component of the data validation only takes place within the AngularJS portion of the application; we will identify special fields, such as email address and phone number, and apply regular expression (regex) validation to them. Again, when a user enters data into one of these fields they will be given immediate feedback as to the validity of their input prior to data being sent to the server. Submission to the server is now allowed until a valid format phone number or email address has been entered. Lastly, there is a limited amount of validation of the spatial data. This portion will exist entirely inside Esri JS API and is currently limited to validating that polygons do not self-intersect.

There are multiple options for deploying a system to an offsite location. One option is known as software as a service (SaaS). SaaS is software that is utilized through a web browser that is often paid for on a monthly basis (Dubey & Wagle, 2007). This type of licensing software provides many benefits besides the low cost of licensing startup. It also allows organizations to quickly and efficiently deploy large scale applications without having to maintain, in nearly any

way, the underlying technology needed to support the software. This may limit our ability to access certain features of the software or control where the servers are located. If limitations are too great we can still deploy our application into the cloud by using infrastructure as a service (IaaS). IaaS is very similar to SaaS but instead of just accessing the software you are responsible for maintaining the software and underlying operating system on a remote server just as you would in a traditional onsite configuration (Bhardwaj, Jain & Jain, 2010). The great benefit here is that you do not need to procure or maintain the underlying hardware of the server, but you still have full control over all other components. Additionally, the servers can be positioned in one or server multiple locations around the world. This provides resilience and a level of assurance that the application will always be available. In either case, there are tradeoffs and the best configuration will be slightly different for all projects and organizations.

For our application we decided to take advantage some Amazon's suite of IaaS products known as Amazon Web Services (Amazon, n.d.). In figure 2 you can see we are running ArcGIS Server, SQL Server and a web server on an Amazon Elastic Cloud Compute (EC2) instance. EC2 is an example of a IaaS service, in which hardware, software, and data can be hosted on a remote architecture, freeing the local IT staff from managing certain resources (Amazon, n.d.). While utilizing a local installation of the ArcGIS Server is a valid option, the EPA Emergency Response group required an off-site redundant system. Since the EPA regional office exists inside of the geographic area they protect, it is possible that the EPA office may be damaged during a disaster to which they themselves would need to respond. Having an off-site system allows them some level of certainty that these application and services will be available even during such an unlikely event.

Amazon Web Services is one provider of an IaaS option, but other IaaS providers like Microsoft Azure would likely suit our needs as well. IaaS provides the ability to create virtual machines in many different places around the world to host our web server, ArcGIS Server and database services. This does allow us to have off-site and redundant services; however, this option requires more hands-on maintenance than utilizing Esri's ArcGIS Online SaaS. Esri's ArcGIS Online encompasses nearly all of these needs and in fact uses Amazon EC2 to host what appears to us as a large instance of ArcGIS Server. With ArcGIS Online you are simply reduced to managing your services instead of the underlying software. In this scenario we still have a need for a local installation of ArcGIS Server because we plan on using custom geoprocessing services to produce PDFs of the plans for printing. Currently, ArcGIS Online does not support publishing of hosted geoprocessing services.

For the specific needs of this project we have put together a combination of technology we believe best suits the end user's needs. The user interface is a JavaScript application built using a combination of AngularJS and Esri JavaScript API. This application along with ArcGIS Server and Microsoft SQL Server will be hosted on an Amazon EC2 instance. The ArcGIS Server installation is required because we will use custom geoprocessing services to produce PDF print outs of the sensitive site strategies and IAPs. ArcGIS Online also being incorporated. The feature services we create are consumed by ArcGIS Online using embedded credentials and then shared via ArcGIS Online. This allows us to utilize ArcGIS Online's user authentication without having to maintain our own user store. This specific configuration allows



us the greatest amount of flexibility and dependability while utilizing as much off the shelf technology as possible.

## APPLICATION IMPLEMENTATION

After early discussions about the needs of the end users for this application we moved into the requirements gathering phase of development. First, an existing data model, which was nothing more than a single feature class, was evaluated and normalized. The feature class contained information in it that should have been contained in a related table. This data model essentially had a relationship inside a single table that allow one sensitive site to have up to six strategies. This was quickly determined to be unable to handle all the information needed to develop a Geographic Response Plan. After analyzing all of the available information we determined a new data model needed to consist of two entities. The ERD seen in figure 1 is the IAP side, and the ERD seen in figure 3 represents the sensitive site strategy. These two parts are not connected by any hard relationship in a database; instead, the sensitive sites relate to IAPs spatially. Since we are representing this data on a map, we can easily obtain and reproduce all the information available in the GRP for a given area without having to maintain traditional database relationships. Instead individuals just need to click on the features, either sensitive sites or IAPs, in the affected area.

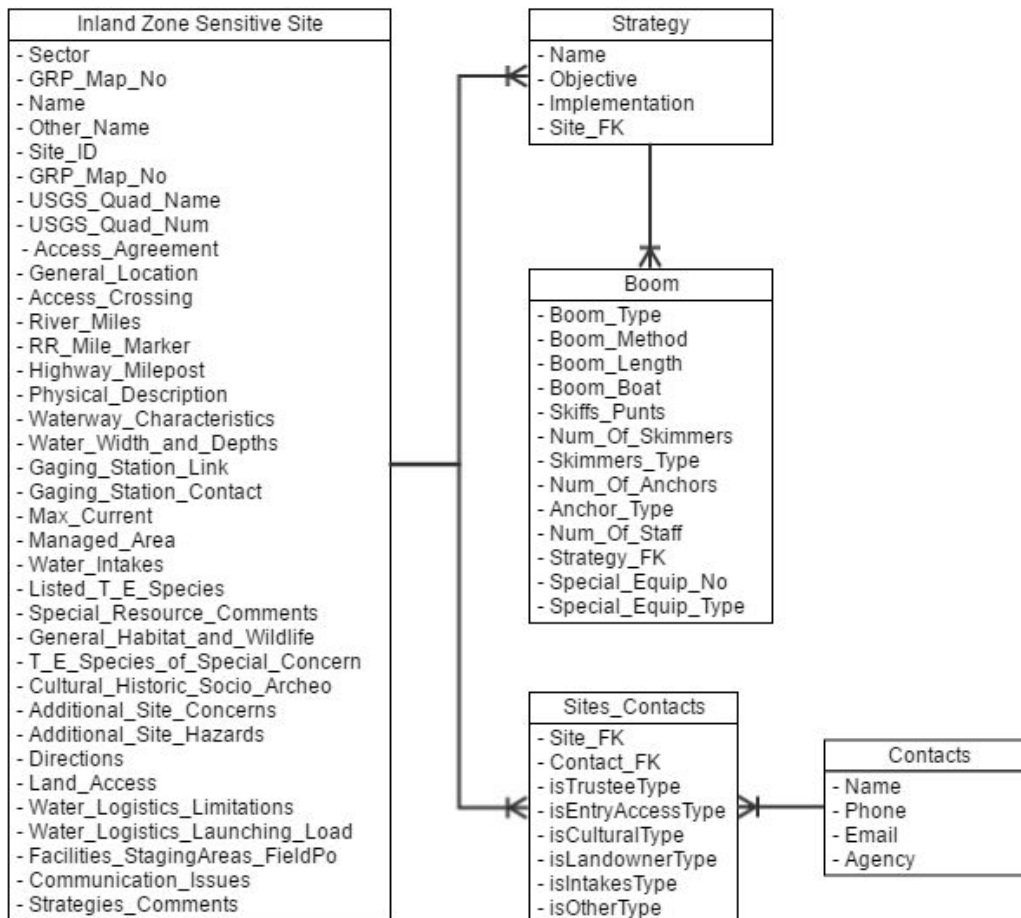


Figure 3. ERD of inland zone sensitive site strategy

Once the initial data models were agreed upon by stakeholders we moved into the design phase of the application. Together we worked with a graphic designer to convey the data model and the needs of the end users as discovered in the requirements phase. The mockups were presented to the stakeholders. After minimal revision, the graphic as seen in figure 4 was agreed upon as a user interface design for the application. This graphic provided a great starting point for development and helped ensure the creation of a high-quality user interface. As can be seen in figures 5 and 6 the actual application that has been deployed for end users did not diverge very much from the original mockup.

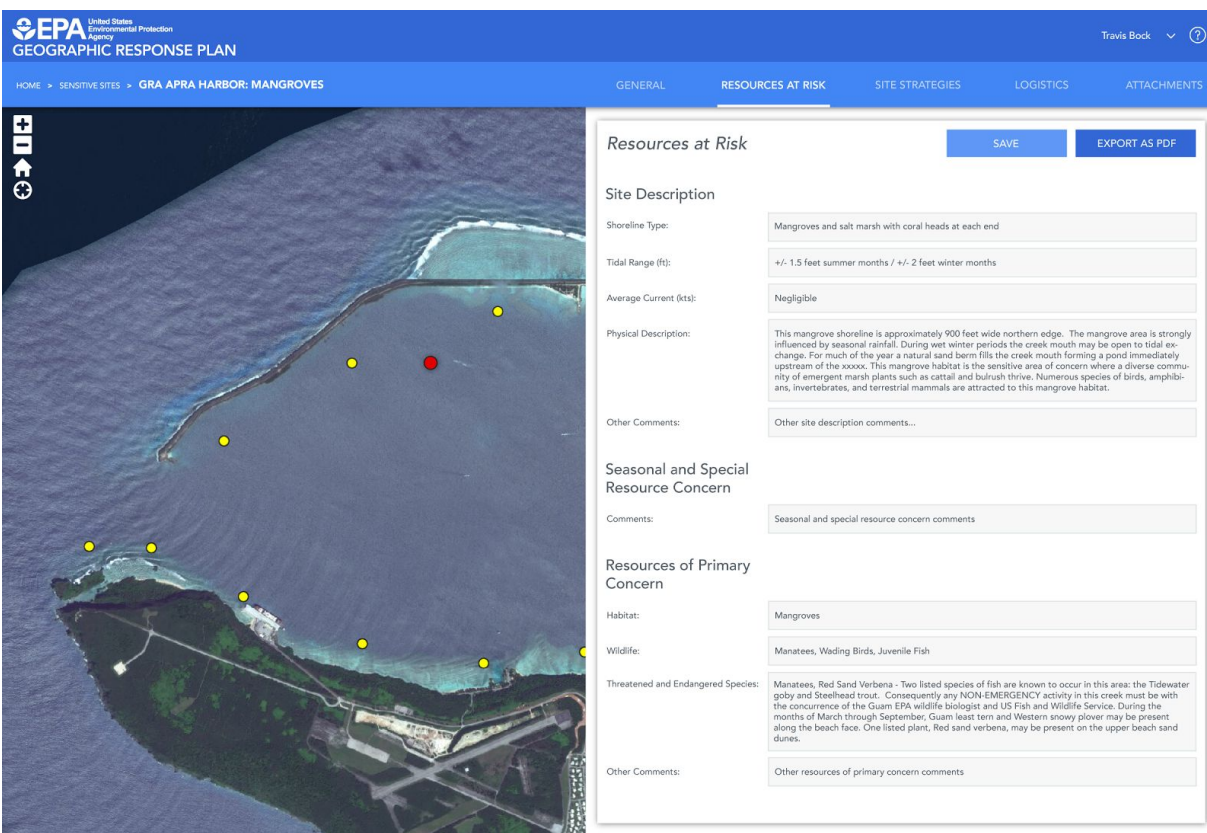


Figure 4. Design Mockup

Figure 5 shows the initial view of the application after a user has logged in. From here the end user can click on a point or polygon to begin editing all of the related data. Additionally, if they are only knowledgeable of the name of a site or plan they can search for either both sensitive sites and IAPs in a single search input. When the user does start editing they are presented with the view that can be seen in figure 6.

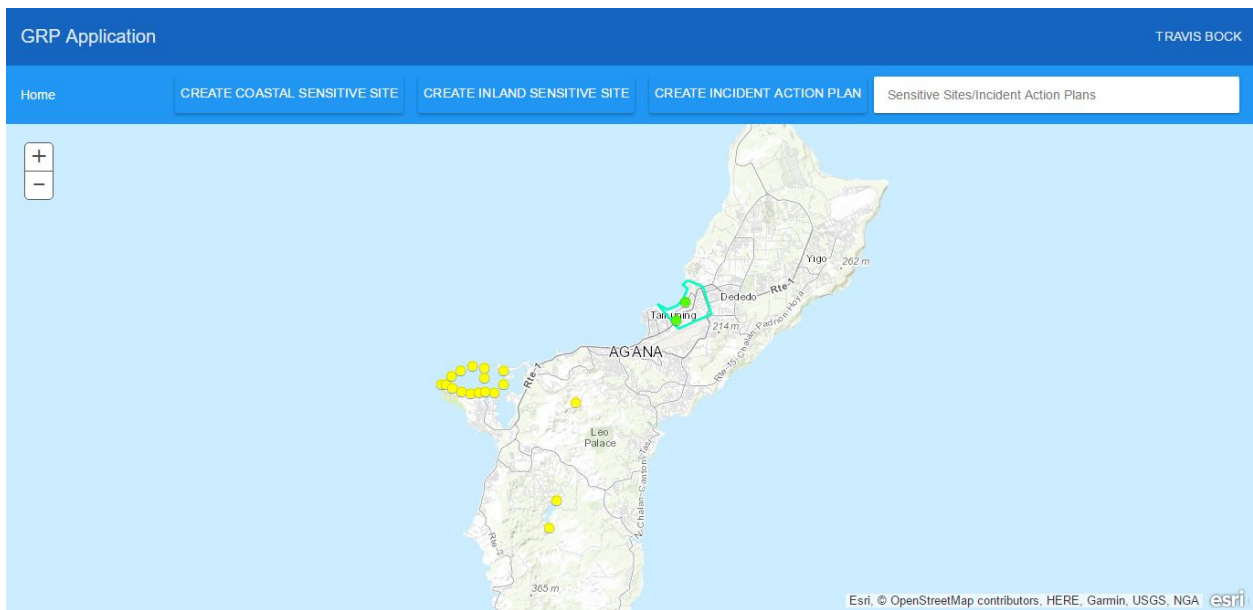


Figure 5. Starting view of deployed application

Figure 6 shows the editing view for an IAP. The editing view for a sensitive site only changes in its content. From here the user can edit all of the attributes of the feature or select another tab at the top right to start editing related tables. Both of these views allow the uploading of attachments. This includes pictures as well as additional documents that may provide additional insight to the plan for the area.

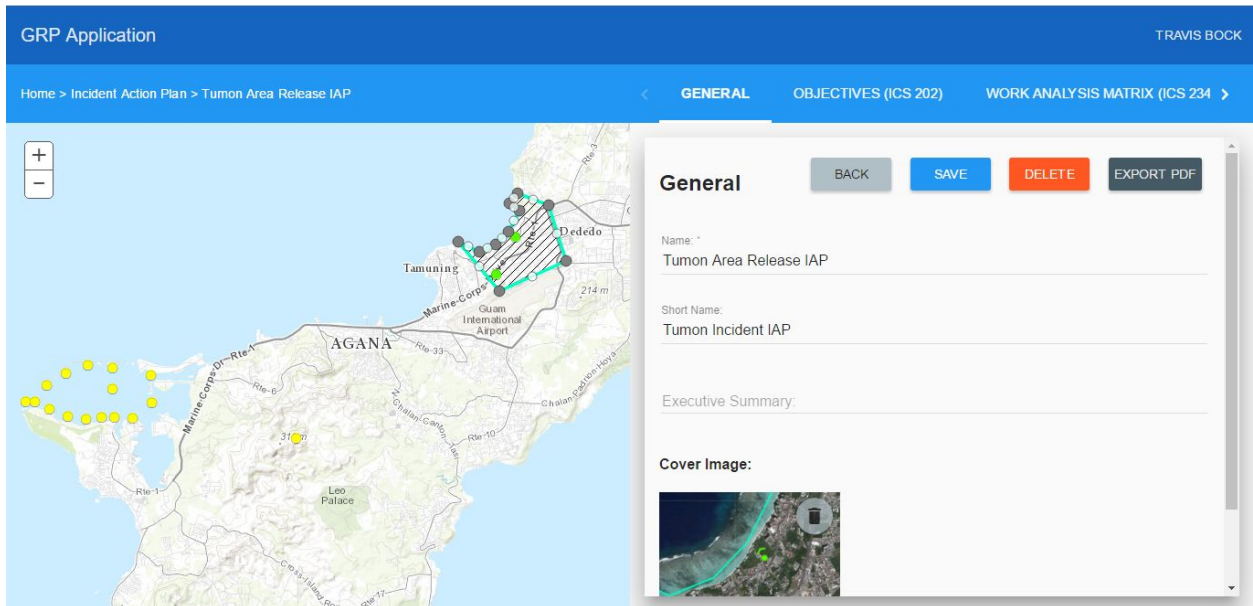


Figure 6. Edit view for an IAP

As you can see the original design see in figure 4 did not change much during development. Having worked with a specialist in user interface design we were left with a great starting point to developing the actual user interface.

## OUTCOMES

Prior to this application, geographic response plans were cumbersome to create and keep up to date. It was difficult to gather all the people necessary to develop these plans and even more difficult to make sure they stay update with everyone's input. Even using email and newer document sharing tools, an Area Coordinator needed to perform a large amount of organizing, proofing and editing. A web based tool provides a central place to develop and maintain plans as well as to access the most recent plan in an emergency. The web application also provides a level of automatic data validation that is unavailable through manual compilation or through newer share document editing. Our integration, through this application's UI, of a large amount of tabular data with related spatial data to allow visualization and collaboration increases the consistency and accuracy of GRPs.

The current version of the application has started to be adopted by US EPA Region 9 and Guam EPA. They are working together to develop, via this application, and fully assemble all the necessary information to develop a Guam GRP. As the first adopters of this application, they have been tremendously helpful in providing feedback about the UI and underlying data model structure. Their feedback has been immediately or nearly immediately incorporated into the application as they continue to develop their plan. While their overall feedback has been very positive; we fully expect to continue to receive and incorporate feedback as the application continues to be adopted.

One point of improvement we have seen as potentially being necessary is how the application is structured and deployed from a very high level. Currently, a single deployment of the application represents a single GRP. This is generally acceptable since most of the participants in developing a plan vary from one GRP to the next. However, since the Area Coordinator is involved in all of these plans we must keep a list of all the deployed applications. While this is not yet an arduous task we can see it becoming one. Ideally, the application would be restructured slightly to handle multiple GRPs and provide access to each GRP based on a user's credentials and the permissions they've been granted in ArcGIS Online. Additionally, we may develop a landing page for the application where the user is presented with a list of plans they have access to and must select one to launch the application. Eventually, Area Coordinates are likely going to need an alternative to a simple list of links.

From the technology maintenance side of the application we have also discovered one potential issue as adoption increases. Currently, simply deploying a single version of this application can take over an hour. Most of this time is spent creating a new database from an empty "template" database and publishing the necessary map service that is connected to this new database. Again, while this is not immediately prohibitive, we have noted it and have begun considering solutions. One potential solution is to automate the process of creating the database and publishing the required map services automatically via Python scripts.

As we continue to support and refine the application lifecycle we are faced with a very common issue in software development and technology in general. Since we began the project

newer versions of the core components used to build the application have been released. While software updates are not always an issue, in this case it is a significant hurdle. As previously mentioned, this application was developed using Angular 1.x and ArcGIS JS API 3.x. Newer versions of both of these have been released, and in both cases moving to the new versions require a complete rebuild of the application. It is certainly unclear if this will be necessary and will largely be determined by the success of the current release of our GRP application.

Overall, everyone, from developers to the client, have been very happy with the outcome of the project so far. We all believe that this tool will increase collaboration and in turn produce more accurate geographic response plans while reducing the time it takes to develop one. Additionally, with the application just a click away, the plans are significantly easier to update than in the past. Early feedback suggests the application, with minimal explanation, is easy to use and relatively self-guiding. We all look forward to supporting the adoption of this application to help make sure that everyone is prepared in the event of an environmental emergency.

## REFERENCES

Amazon. (n.d.). Amazon EC2. Retrieved May 04, 2017 from [https://aws.amazon.com/ec2/?nc2=h\\_m1](https://aws.amazon.com/ec2/?nc2=h_m1)

Amazon. (n.d.). What is AWS?. Retrieved May 04, 2017 from <https://aws.amazon.com/what-is-aws/>

Bhardwaj, S., Jain, L., & Jain, S. (2010). Cloud computing: A study of infrastructure as a service (IAAS). *International Journal of engineering and information Technology*, 2(1), 60-63.

Dubey, A., & Wagle, D. (2007). Delivering software as a service. *The McKinsey Quarterly*, 6(2007), 2007.

Esri. (n.d.). ArcGIS Rest API. Retrieved May 04, 2017, from [http://resources.arcgis.com/en/help/arcgis-rest-api/#/The\\_ArcGIS\\_REST\\_API/02r30000005400000/](http://resources.arcgis.com/en/help/arcgis-rest-api/#/The_ArcGIS_REST_API/02r30000005400000/)

Esri, (n.d.). What is ArcGIS Online?. Retrieved May 04, 2017, from <http://doc.arcgis.com/en/arcgis-online/reference/what-is-ago.htm>

Fawcett, J., McPeak, J., & Zakas, N. C. (2007). *Professional Ajax; 2nd Edition*. Wiley.

Garrett, J. J. (n.d.). Ajax: A New Approach to Web Applications. Retrieved June 21, 2016, from <http://adaptivepath.org/ideas/ajax-new-approach-web-applications>

Kapucu, N. (2008). Collaborative Emergency Management: Better Community Organising, Better Public Preparedness and Response. *Disasters*, 32(2), 239-262. doi: 10.1111/j.1467-7717.2008.01037.x

Perry, R. W., & Lindell, M. K. (2003). Preparedness for Emergency Response: Guidelines for the Emergency Planning Process. *Disasters*, 27(4), 336-350.  
doi:10.1111/j.0361-3666.2003.00237.x

Somers, R. (1998). Developing GIS Management Strategies for an Organization. *Journal of Housing Research*, 9(1), 157-178. Retrieved April 12, 2017, from <http://www.jstor.org/stable/24833663>

United States Environmental Protection Agency (2013). *Area Contingency Planning Handbook Version 1.0*