

Project Retrospective Report

Team [06]

[Citrus]

Rumsha Khan

Therisa Tran

Umang Dhawan

Zain Merchant

April 24, 2020

Introduction/Project Idea/How it came about etc:

Our team, team #06 also known as Team Citrus was formed within the first week of this Capstone class. The four of us knew each other from the beginning of Freshman year and throughout our four years worked with each other in various capacities in other Informatics classes. We decided to register as a team before we even knew what our product idea was for the project.

On the first or second day of class, a leading representative of the IU Fire Safety Department came to our lecture and pitched an idea for some of the teams to think about working on. They told us that they needed a team to create an automated scheduling system for them as they were currently manually inputting data for the fire drills that were taking place around campus. This was an idea that instantly clicked with the four of us which is why we went forward with getting more information on it before committing to the idea. When meeting with the representative we got more insight into what problems the IU Fire Safety Department was facing and what their needs were. After much discussion, the four of us agreed that this was a project idea that really clicked with us and allowed us to give back to IU in some way. We also liked the idea of working for a specific client as it would motivate and push us to do the best that we can do to satisfy the client and stand out from the other teams also working on this.

The IU Safety Department had a few specific needs and problems that they pointed out to us. Their current manual system included communicating with other personnel through email about what drills were to take place or had already taken place. With no proper scheduling system, the managers would schedule events through such

email methodologies or by creating Google event ideas. All data was manually inputted into an excel sheet. For feedback, they expressed that they wanted an effective way to get feedback from the other building managers. What they were currently doing was having the managers email a simple google form individually to each involved employee after a scheduled drill took place. They wanted this system to be simple to navigate for the user, assuming they are not very tech-savvy. They wanted the scheduling system to be straightforward and to the point. They were looking for a flexible and easy way to create and delete scheduled drills, communicate that with involved employees, receive feedback and manage all of that data in an easy way where there would no longer need to be any manual input of data of any sorts.

Taking all of this into account, our team "Citrus", believed that we could make the drill scheduling system that they were looking for. Taking each of our strengths into account, the four of us, in general, felt that we would prefer to work on a web-based application and would do well in it. This scheduling idea fell right in place with that preference of wanting to work with web-based. After a few meetings and brainstorming sessions with the AI and professors, we started to make a list of all the requirements we need for the product, what we imagine our product to look like, etc. We would meet every week as a team to sit together and better understand what each member's strengths were and how we would go about working on this product that we would present to the representatives of the IU Safety Department.

How the Team Functioned:

Overall, our team worked very well together. We all stayed professional and most importantly always communicated with each other. We communicated through GroupMe mainly, used software like 'Discord' and 'Microsoft Teams' for group call meetings where we could send each other links and files and work collaboratively, and when we were still on campus we would schedule certain meals together every week to sit together for 3-4 hours and work together on top of the individual hours we would put in. The first semester of Capstone, I-494 required us to submit individual tech proficiency assignments. This gave us a very good idea of who was strong in programming and who was not. We wanted our team functionality to be as structured as possible so we assigned each other specific roles. Zain and Therisa were the primary developers, Rumsha was Scrum master and with Umang, the two of them would take on the role of product managers. Of course, this did not mean that Umang and Rumsha would not be doing programming, it just meant that because Zain and Therisa had stronger programming skills, they would be required to solely work on programming related tasks and maybe more advanced programming parts of our product. Initially, we split Rumsha and Therisa to focus on the frontend part of the web application and for Zain and Umang to do the backend. Over time our roles overlapped but by constantly communicating we were able to keep things running smoothly.

One of our biggest successes in keeping our team on track with our assigned tasks and goals was the implementation of Scrum. Although this was mandatory, it was something our Scrum master put extra time in and continuously updated as our team made changes along the way in our vision and what requirements we needed. The

Scrum sheet kept us on track and also communication. Our communication was key and the consistency in it enforced certain team rules to the extent where nobody on the team slacked or missed a meeting/work session unless it was a valid excuse. These things kept us on track throughout the year, especially in the second semester. Our AI was also very helpful and very good at keeping us on our toes at all times. Every week those 15-minute meetings would be very effective and we made notes at every meeting, kept everything in a shared folder of google drive, and documented each step of the way to keep us as organized as possible.

High-Level Breakdowns:

Capstone provided an opportunity to work with the Indiana University Fire and Safety Department as a client. They had mainly two requests: an interface that can record the drills that need to take place, and the feedback reports associated with those drills. With some of the members' past experience and courses taken, our members had some experience working with PHP, Html, CSS, and a bit of Javascript. It was determined that a website would be the best platform for us to choose and satisfy the client's needs.

The client provided some data and information on how they currently store their data and schedule their drills. The information was then taken to establish our databases determining whether the information should be a field or its separate table connected through foreign keys. A total of five tables were created to build the database to keep a record of the buildings, drills, drill types, personnel, and feedback reports recorded after

each drill. As the final version of the database was being created, two of our members worked on creating Html frameworks and implementing basic CSS that would, later on, be linked to the database retrieving information and updating the tables.

A couple of weeks into the spring semester, Rivet was recommended to be used to build our frontend of the website and transitioned to use Rivet's frameworks. Although this scrapped our old Html and CSS, Rivet quickened our design process in other ways, such as its prebuilt drop-down designs, forms, buttons, and modals.

However, Rivet did not include any calendar or map API. The calendar was intended for user interface so the user can visually see their drills and mapping of their locations. Google Calendar API and Bing Map API were the first API's we decided to work with before being recommended Fullcalendar.js and preferred Google Map API. We then used Fullcalendar.js and Google Map API. The transition from the Bing Map API to Google Map API was mainly due to user interface preferences and the popularity of Google products. As for the change from Google Calendar API, a few difficulties occurred working with the API because it required the involvement of other plugins and programs which the team wanted to avoid for risk of point deductions. The project was emphasized not to use other plugins or frameworks to help build our project and should be of original code. Fullcalendar.js was an API recommended to us from the Professor. Logan Paul, which made us comfortable in using it.

Fullcalendar.js has many callbacks with different properties that could be used in building the custom calendar. The calendar itself retrieves data from the database using PHP and automatically populates the calendar when it is first rendered. If an event is

dragged to a different date the event drop callback activates and updates the database, using ajax and PHP, with the new date for the event making it easier for the user to change event dates without having to delete and create a new event. When an event is clicked up upon, event click activates and provides the user with the information regarding the event. If the user wishes to delete an event, there is a "Delete Drill" checkbox where the user can activate and click on the event they wish to remove. This meets a second condition set on event click which will run asking for the user to confirm their action before sending the command to delete the event from the database. Whenever the user wishes to add an event/drill to the calendar, a "Create a New Drill" button is placed at the homepage of the website where the user can fill out a form. Once the appropriate data has been submitted, PHP will update the database with the new event and the full calendar will render the event on its display. Similar to creating "Create a New Drill Type," it was made to allow the user to create more descriptive drills that they could use to fill out the new event drill form.

Besides having the calendar be the main feature for the user to view all their drill or events for a certain month, week, day or as a list, the user can view all their drills and drill types by clicking their respective buttons on the homepage which will redirect them to the tables with all the information provided from the database. Also, the user can view other relevant information with the process of making drills on the different tabs: Drill History, Feedback Report, and Personnel Report. The commonality the three pages share is that the user can view all the information and modify the data, such as drills, feedback reports, and personnel. Also, the three pages have an area to search through

said data. What differs between the pages is that the feedback report and personnel report also have a button-form for the user to add in new entries, such as a feedback report of a drill that was just finished or a new personnel manager.

Lastly the map, we had the intention of using map API to have area markers for the different regions our client has in dividing the drill. As well within those regions, the appropriate marker would appear on the map with all the information regarding the drill on that location. However, with working with API, you cannot see the creators code and sort of blind-boxed and limited to the parameters the documentation of the API has provided because there wasn't an easy way to modify the API's code to match our needs it was resulted as being a simple map with all the locations the client currently works on labeled. This still allows the user to have an easier visual to map routes and plan drills back to back to create the most efficient paths in finishing each drill. In addition, the user can view and edit all of their buildings' information in a list, create new buildings, and review the building guide.

Due to Rivet being heavily recommended to our team, a lot of the user experience and user interface was finished for us since it was a framework that had pre-built designs, such as buttons, drop-downs, and modals. The only difficult task was learning and implementing the framework. This did change how we first approach the user interface designs to utilize more of Rivet's features, such as color scheme and structure. Nevertheless, our goal was to be following Indiana University's approved colors and fonts since we will be for Indiana University and to create something intuitive for the user to use.

Why and How We Handled Problems:

Referring to the high-break down some of our problems were due things we were not expecting to occur. We did not know that we would be implementing Rivet until a few weeks into the second semester. If we had known earlier, this would change our approach to creating features for the website and utilize our time for other areas than having to build an Html framework that would be later scrapped out. In addition, to being unaware of or unsure if we were allowed to use Fullcalendar.js after a few meetings later with the Professor. Due to implementing Fullcalendar.js after finishing our database, the fields from the database didn't align the best with Fullcalendar's properties.

How we handle these problems is diving into understanding these new elements and learning to implement them. We didn't let them stop production, but we made the most out of it by assigning each teammate which parts they were going to be responsible for. With Rivet, we had Zain and Rumsha read into the documentation for Rivet and implement Rivet since we knew this would affect all the pages. Fullcalendar.js only affected one page and Therisa took the responsibility to implement the calendar and bypass the difference in fields and parameters with trying different ways to format and deconstruct everything. Whenever one of us would get stuck or need assistance, we would ask each other for help, such as Umang helping with Rivet.

However, one problem we could have better prepared for was organization. In the early stages of the second semester, we had not yet established a strong SCRUM sheet and we hadn't taken the approach of assigning people certain features and approached building everything as a team. Upon our first meeting with Professor Richart, we soon learned that it wasn't effective, so we decided to change immediately. Rumsha took the task of organizing us, especially organizing an extremely detailed SCRUM sheet. From there, we gradually started claiming tasks and features each member is responsible for.

Accomplished & Unaccomplished:

For a table of all the features we achieved or did not achieve, it is listed below under appendices. Following the client's request, we successfully built a system to record the drills and feedback reports (for the drills). Specifically, we incorporated the CRUD framework (Create, Read, Update, Delete) in PHP to have these forms. Beyond these, we made a calendar interface for better usability purposes. Also, we added references to IU's building data and maps so that the client could have an all-in-one system for tracking drills and information related to them. The client will also be able to type to search for related information and data and have the needed result returned, saving time from having to look through large tables of information. Lastly, we implemented a method for the client to open access to their e-mail via the website to make updating other personnel easier for the client.

Overall, we delivered the requirements according to the client but decided not to attempt the "Dark Theme" goal that we had initially set. This was decided when we applied the IU Rivet framework to our website. Also, we weren't able to provide all the alternate methods to edit or create an event to fully cover the whole user experience.

Retrospect:

Looking back, this capstone project has not only taught us how to apply the concepts and skills we learned in the classroom but has also shown us the importance of the various roles played in the software development lifecycle, by challenging us to deliver a working product under the conditions that the industry requires. Some of the most important learning outcomes of this project include: effectively communicating with team members and stakeholders, efficiently preparing documents prior to implementing a working solution, and being able to find solutions to problems more independently. If there was one thing that we could have done differently, it would have been establishing more solid requirements in the beginning. As we worked on building the MVP, a lot of time was spent reorganizing our product description to be more realistic. This is something that could have been done earlier. That being said, we all were comfortable enough with each other to highlight an area of weakness and trust that we can work together to get better results. With this in mind, we managed to collaborate to meet the deadlines and deliver the product that we promised to.

In conclusion, we can confidently say that along with sharpening our code-debugging skills, we are more confident with our technical communication, time

management, understanding of the need to collaborate with professionals from diverse backgrounds, and be stronger problem solvers.

Appendices:

Client Details

Client: Indiana University Emergency Department (Person of contact: Alison Sinadinos)

Contact: asinadin@iu.edu

Business/role and expectations: The IU Emergency Department manages the fire/emergency drills across the IU Bloomington campus. It is looking to implement a web-based scheduling system to plan and carry out their fire drills in an organized and efficient way. The client is expecting us to create a web interface that will allow them to a) schedule fire drills across the campus, and b) record feedback from fire drills that have taken place.

Potential Issues: An issue that may arise is that of scheduling during the user-testing phase of our product development. This is because our product will be catered to their preferences, therefore determining the potential changes/improvements to make will mostly depend on their usability. To counter this issue, we are trying to establish all of their requirements in detail so that we have fewer iterations during the user-testing phase.

Feature Name	Priority	Notes	Status
--------------	----------	-------	--------

Calendar view for Scheduled drills	High	A new drill can be created using "Create a New Drill." Once a new drill has been added, it will instantly populate into the calendar for the user to keep a track of.	Achieved
Event removal via calendar view (creating drills, repeating drills, add/change information about a specific drill)	High	Including what was mentioned above, some additional things that can be done with events are the events will also update when moved to a different date on the calendar, and can be removed from scheduling via the calendar.	Achieved
Feedback create and search by event	High	The user is able to generate feedback reports which link to a specific building event allowing users to evaluate through comments. Moreover, the user has the ability to search feedback by location with its drill ID.	Achieved
Feedback Management (view all, update, delete)	High	The user has the ability to view all feedback report, edit or delete them accordingly	Achieved
Personnel Management (Building Managers, Fire Department Managers, Users, other important personnel)	High	The personnel report page allows the user to create responsible personnel based on the building location while also allowing them to write their specific descriptions. Moreover, It requires their IU username to help generate emails later.	Achieved

Generating Reports to view all events by history	High	The user has the ability to generate reports over how events were done. They generate reports based on drill ID, drill type, location.	Achieved
Building/Location Management (create, read, update, delete)	High	The user has access to generating a new building or looking up previously made buildings. Moreover, additional building information is also present in the form of a guide.	Achieved
Credential Management for admin (Permission Management, Credential Management)	High	The user has to be linked with the university duo system in order for them to utilize this application. It uses Duo-dual authentication to login as the user.	Achieved
Email Management (for communication with building managers and responsible personnel)	Medium	We created a convenient "send mail" button at the "view all personnels" page to open up the email client on the user's computer for emailing any personnel.	Achieved
List view scheduling	Medium	The user has the ability to see all the created events in a list form through the calendar on the home page.	Achieved
Ability to manage different Event	Medium	The user has the ability to create personalized event types which are	Achieved

Types (To-do event, fire drill, other drills)		stored and can be modified according to the needs of the user.	
Map view	Medium	The digital map section allows users to view different buildings across campus. It offers different viewing options alongside street view to make it convenient while creating a drill plan	Achieved
Theme Management (switch to dark mode)	Low	We preferred the theme to stay on the original Rivet Framework settings, and didn't see the importance of dark mode to implement.	Not Achieved