**15-237 Term Project Weekly Deliverable**

Group: Kim Kardashboard;

Project: Chat Application from the Crevices of Justice (CAftCoJ);

**User Studies**

User Study 1: User Study-A-Thon

*Questions/Suggestions:*

1. How can I search for friends?
2. Need a refresh and back button, more user feedback/solid buttons
3. Have all pages on one HTML page, don’t need to refresh the top bar every time, and just change CSS display property  of different “page” elements/sections.

*Discussion/Answers:*

1. We haven’t yet implemented a search feature. We plan to have a default group document in the Groups collection (Mongo) where every user is added upon registration, and this will serve as your full contact list (temporarily). Ideally, we will integrate with Facebook, and a list of your Facebook friends that use the app will be available to you. There will be a separate page to add friends to a group.
2. In progress.
3. Essentially, this has been fully implemented. This is a huge change to the way our app is written.

User Study 2: Anonymous

[*Note*: this study was done Tuesday midnight, after we were already deep into making changes from the user study-a-thon so most of these changes will be made after this week’s deliverable, so we can focus on completing the tasks we are already on].

*Questions/Suggestions:*

1. When you click on marker, instead of popup above marker, have sidebar popup with event details, as well as chat window below it. This makes it much easier to access chat from map, so you can discuss the event with your group immediately.
2. For each user, make the marker a different color for distinguishing purposes. This way you don’t need to click on the marker each time to figure out which user it is. This can also correspond with the chat, so each message backgrounds match the same color. This will allow for memory over time so users learn to associate colors with users.
3. Change the person marker to something unique so it looks less like Google maps.

*Discussion/Answers:*

1. This is an excellent point and we will look to implement it by next week. As the marker info window is now, it’s pretty unattractive, so we’re inclined to modify it.
2. This is a relatively simple thing to implement, so we will look to implement this for next week’s deliverable. It makes a huge difference to the users, so it’s critical.
3. We see where this is coming from, but I guess the user didn’t realize that the User marker on the map was actually our own custom one, and not a Google Map inbuilt marker. We will revisit this idea later on.

In summary, this user study gave us a lot of good, and important feedback since it was after our first iteration following the study-a-thon, so we had two very different stable versions of our app tested. We now have multiple improvements that we can make before the final deliverable next week.

User Study 3: User Study-A-Thon

*Questions/Suggestions:*

1. Have an algorithm that calculates how far the user is from an event once the event has been created. This allows other group members to estimate how long that user will take to arrive to location of event.
2. Maps page should be able to calculate the midpoint of all friends in the group. This will let user know ideal location to drop pin and create event.

*Discussion/Answers:*

1. This is an excellent idea and will definitely make the user experience much more enjoyable, but it’s not very critical to the use of our app, so we’ll re-evaluate this later on to see if we have time. It’s not nearly important enough to implement by this week’s deliverable.
2. This is a fantastic idea. We had thought about this earlier, but it’s a secondary feature so we will think about it for next week’s deliverable. Not imperative for this week.

User Study 4: User Study-A-Thon

*Questions/Suggestions:*

1. Make scheduling better. Have a nice way to organize past events and future events.

*Discussions/Answers:*

1. Our idea for scheduling is fairly simple. We intend to have an event log that is sorted chronologically that includes all the group events ever created. We do not wish to make this app like an organizer, because the purpose of this app is quickly create events with friends, and too much structure and organization would take away from this purpose. With that said, we can implement a sorting feature on the event log that would allow the user to sort all events based on time, who created it, events that the user accepted, and starred events.

It is worth mentioning that based on these user studies, we gave the most priority to suggestion 3 under User Study 1. This user suggested that we have a single HTML page as opposed to multiple pages. At the last deliverable, we had a different HTML pages for different parts of the app. Our code redirected the user from one page to another based on the user’s inputs. This was making our app a little slow, because our app requires the client to have certain data from the server at hand, and this forced us to have to make multiple server calls once the new pages were reloaded. However, with one HTML page, there is no need to redirect the user from one HTML file to another, which means that the same JavaScript file runs through different parts of the app, allowing us to simply store this data in variables and objects.

After the user study, we immediately revamped the structure of our code and implemented only 2 separate HTML files, as opposed to 5 that we had before. The login/signup, groups, and add group ‘pages’ are now in the same HTML file. Also, the map page and add event page are in another HTML file. We feel that this has increased the efficiency of our app by reducing the number of server calls we have to make.

**Code writeup:**

Since our last submission we have improved certain aspects of our application, and have started adding a new feature.  NOTE: We still only have a single default group that holds all users.  Keeping that in mind, our improvements include:

Multiple Users:

We now can have multiple users register on our site and view the locations of the other users.  Currently each user’s location is updated each time they view the site’s map page.  We hope to find a way to automatically update each user’s location, but this is still in the works.

Events:

We now have it so each user can drop an event marker on the map.  The map then redirects to an event info page.  This allows the user to fill in information for this specific event.  After submitting this information the event marker appears on the map, and now has the information stored in the marker’s info-window. The event’s info-window can be accessed by simply clicking on the event marker.  Each user’s events are stored on the map, and will be redrawn the next time the user views the map.  Our problem is that the other group members can’t always see all other members’ events.

**Added feature:**

Chat:

One of the core features of our product is a group chat. In our last deliverable, we had no chat. Now we have a near-fully functional group chat. It can be accessed from multiple computers, and you have a chat with the other users. Initially, we had all the messages displaying on the left side, so you couldn’t distinguish your messages from the other users’ messages. After user testing, this came up as the biggest issue, and we have since fixed it. Now your messages display on the right, in a light blue background, while all others are on the left in white. This feature has two limitations: (1) it does not yet work on mobile devices, and (2) it isn’t integrated into the rest of our product.

**Significant changes through iterative design:**

**Events:**

* Initially we inputted event information through a pop-up like window.
  + The first problem we found was that users could place a marker and then by placing another one, the previous marker’s information is left blank.
  + On top of this we were encouraged by users to change how this was done because the way the pop-up appeared interrupted the flow of the application.
    - Our Solution:  We instead redirect the user to a separate page that allows the user to fill in the event information.  After submitting the information the user is redirected back to the map.  This stops users from placing empty markers, and it also gets rid of the disruption that the pop-up window was causing to the users on the map.
* Another small issue we came across with events was the information we were storing.
  + We were storing an event name, date, and start time
    - Our Solution:We discovered that we needed an end time because allowed us to eliminate markers from the map after it’s end time has passed.  We also decided that other information about the markers may be useful, and added more to each marker.  Currently each marker now holds: event name, start time, end time, date of event, date event was created.

**Server Calls:**

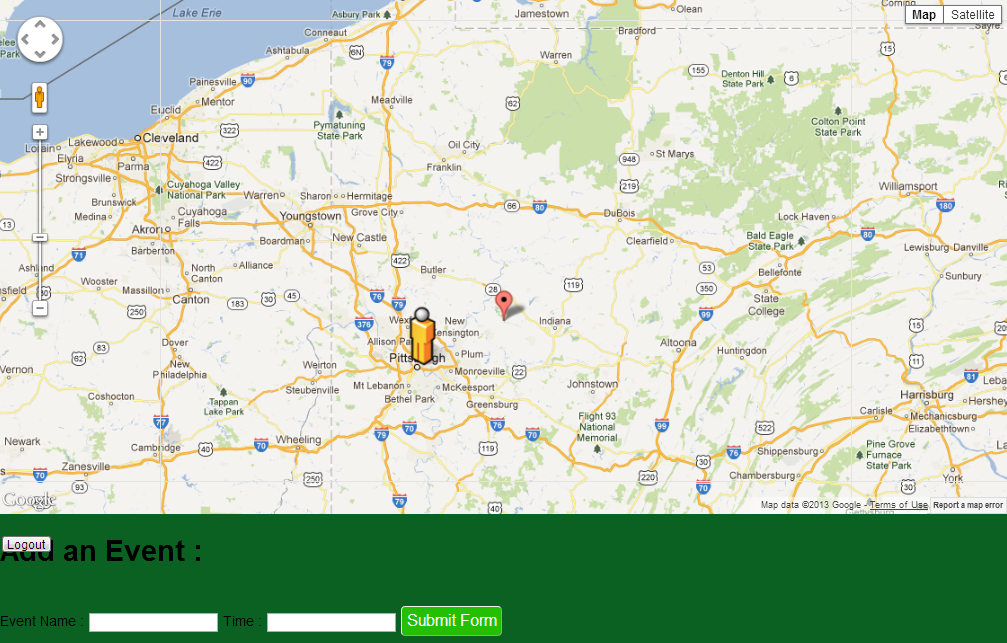
* We initially had separate HTML pages for each of our pages.
  + We kept running into problems with our client, because information is lost we the client changes to a new page
    - Our First Solution:We realized that the best way to avoid this problem was through the use of local and session storage.  This allowed us to store the information we needed, and switch between pages without losing this information.
    - Our Second Solution:Following a user study, we were given the advice that we didn’t need separate HTML pages, and we could take advantage of the CSS property display.  This was another way we were able to eliminate other server calls.  We took what each page had inside it’s body, and encased it in a div.  All the divs were then placed in a single HTML document, and by using the CSS display property, we were able to choose with div the user can see.

**Storing Username:**

* We initially had issues keeping track of which user was logged in on the client after redirecting the user from one page to another. This is because the javascript file gets reloaded as the HTML page changes.
  + Our first solution: We solved this by encoding the user’s username using encodeURI(userString) and appending it to the URL of every page after a ‘#’. This worked, as once ever page loaded all we had to do was parse the URL of the page, find the index of the ‘#’, and store the user’s username. However, this was not very elegant, as the user’s username was visible so openly on the URL.
  + Our second solution (In the process of implementation): We are now going to use local storage to store information about the client on the client side. This includes the username, and some information about location that assists us with implementing the map. This not only is an elegant solution to the problem above, but it also saves us additional server calls to get information from the database.

**Screenshots Showing Iterative Design:**

* 1. Add event pop-up:

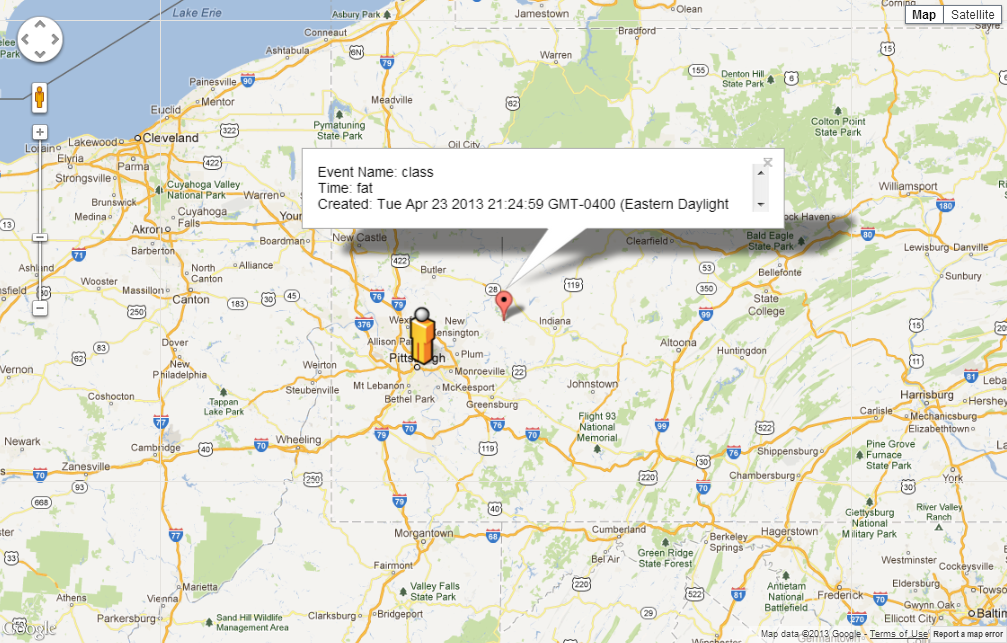


Before: Once the user dropped a pin on the map, a bar would open up at the bottom of the page allowing the user to create an event. However, the user could continue dropping pins on the map, which resulted in the first event not being created.

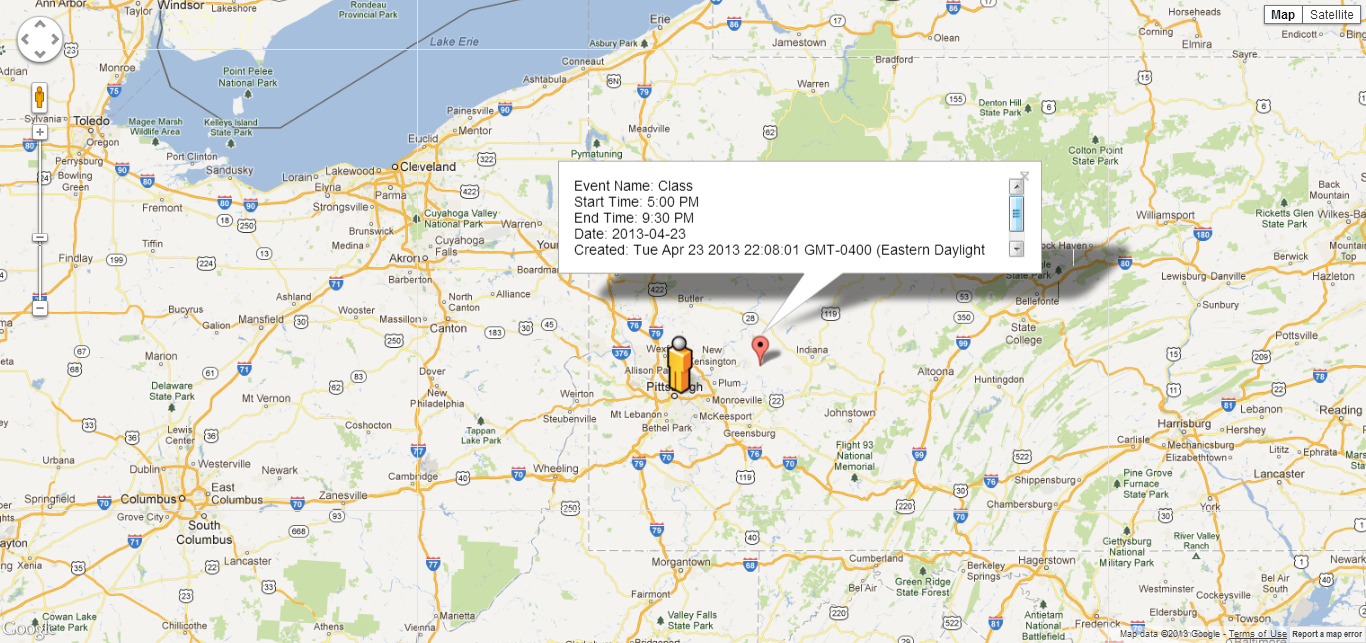
**

After: New add event screen altogether. This is a less disruptive and cleaner solution.

* 1. Event display on map:



Before: The events did not store accurate time information. It was just a string input, and the user could enter anything.



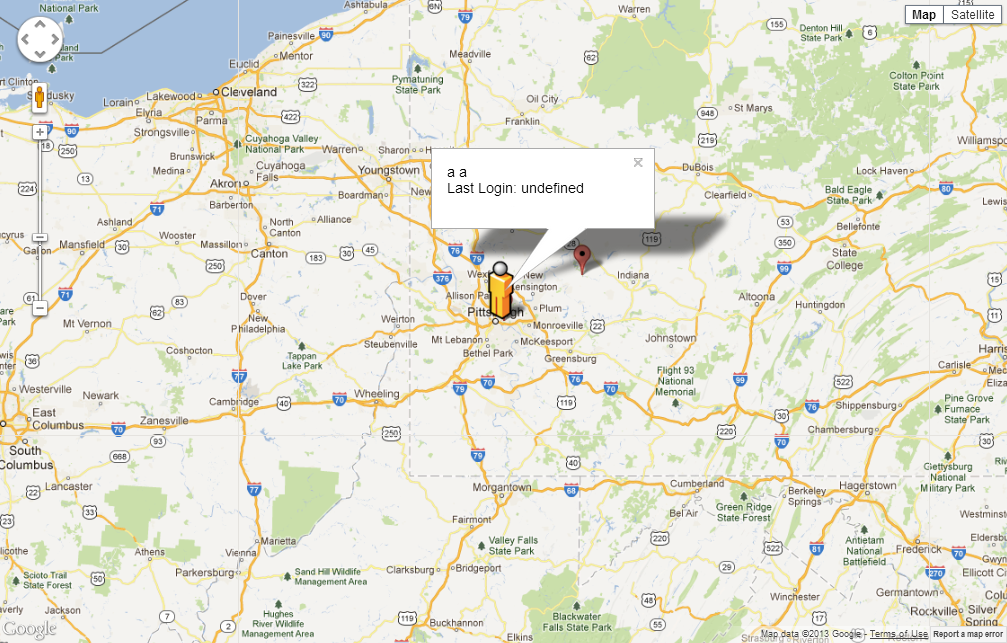
After: The events take a start time, end time, and date.

3: Groups page



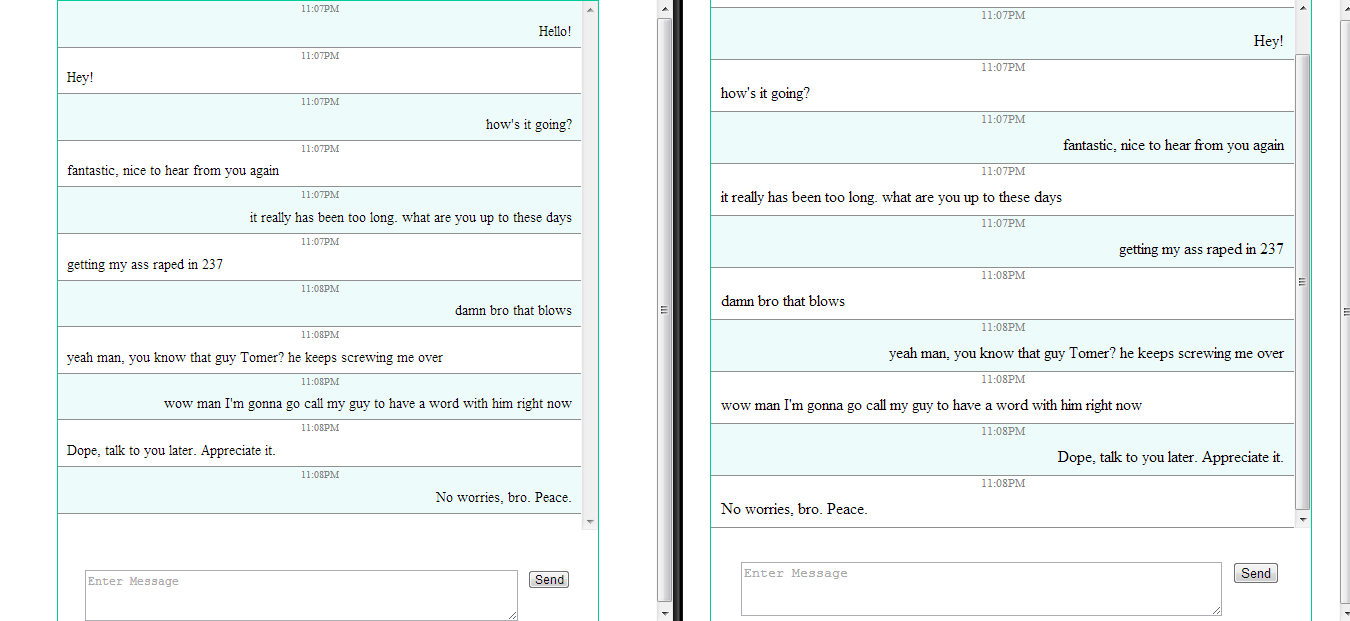
We created a new groups page that would allow the user to view all groups that he/she is in. (not fully implemented).

4) viewing other users on map

**

Each user is able to view other users on the map. In this example, we are able to view the location of “a a”, the last time he was logged in. This is, of course, the same location as we currently are in, so the icons overlap.

5) Chat



This is a screenshot of a chat conversation between 2 clients on the same server. The client that sends a message has the text aligned to the right, and the block shaded blue.