**HistViewer.js**

Data visualizations are extremely useful and interesting to see in the current age of technology. Not only are they useful to understand information, but they also allow for lots of information to be shown in an understandable way. Unfortunately, most visualizations are individual; meaning that there is only a single visualization that is used at a time in order to display the needed information. We are going to be creating an open source JavaScript library that will combat this. Our library will be a historical data visualization which will encompass anything that can fit within the 4 simple categories of who, what, when and where. Any developer that uses our library will expect our library to work if they provide us with data that has at least those 4 columns although we will have support for additional information if it is given. The use of these categories will allow us to link different historical events together in order to create multiple views to display this data.

In order to represent this information we are going to have 3 distinct views: timeline view, bubble view and map view. These views will have the ability to communicate between one another creating an intuitive flow of information. The timeline view will allow for multiple timelines organized in a list-like fashion. This will allow for the user to compare multiple timelines together and see where there are similarities. The bubble view will be the main intermediary view. This view is the main “web of data” in which this will allow for the user to navigate through to determine information about a particular subject. This view will consist of the “center of attention” bubble which is the item you are currently looking at and arranged around it will be the link bubbles which represent related items that are similar to the center of attention and will be clickable to load the links as new centers of attention. Lastly, the map view will be a view to display the events on a map generated by Google Maps API that will show similar events that were happening in the same time at the same place. This view will implement a time slider that will allow you to scroll through time to see how things changed.

In order to achieve the creation of this library we are going to use several languages and tools to aide us. In order to create the front end of our library we are going to use the AngularJS Javascript framework as well as Bootstrap. Specifically for our demo of our library we are going to use Heroku in order to host our site as well as our database (through the use of Postgres). We are going to use node.js in order to run the back end side of our demonstration product. Lastly we will be using Github for version control as well as allow the ability to open source our code for other developers.

Once again it is our aim to create an interesting, understandable and easy-to-use data visualization library for any historical event. We will be creating multiple interacting views that will display this information. The usefulness of the interactivity between each view will push us ahead of other similar visualizations that are already in the market.

**More Details**

Timeline (2).png

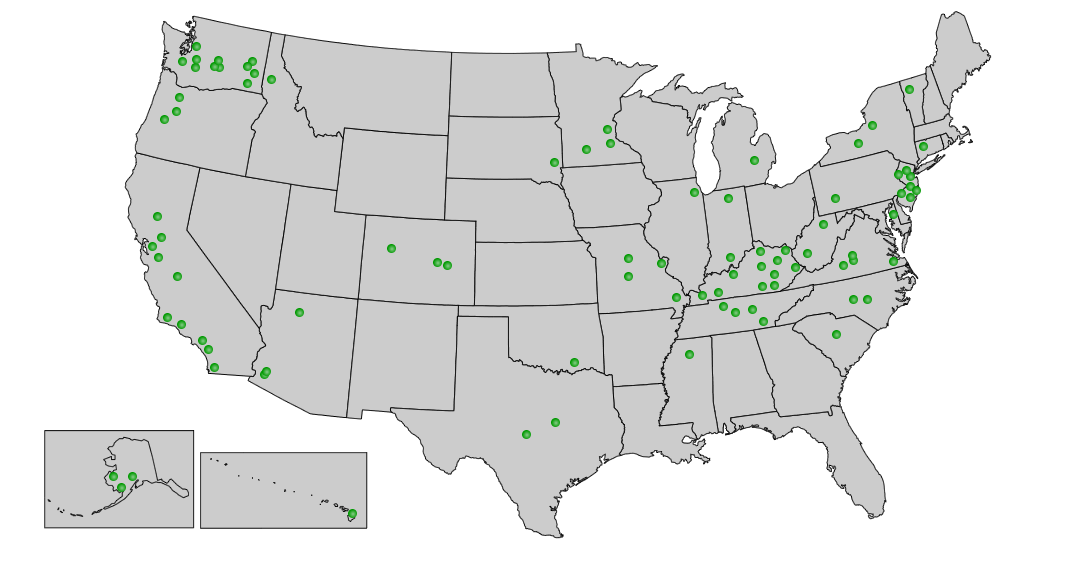
The timeline view is the base view for the library. The view will be unique from other timelines in a variety of ways. The first feature of the view that is original from other timelines is that it allows the user to compare the timelines of multiple individuals simultaneously using the same view. Another feature that the timeline view provides is the ability to transition to other views within the library seamlessly. This provides the user with an eye-pleasing and useful visualization of the data being displayed.

The view will include each of the timelines that the user wants to display, events on the timeline, and view options for the user. Each individual timeline will display events color coded to correspond to a given category. Furthermore, if two timelines on the same timeline view share an event, this will be visibly linked in the view. Clicking on a given event will allow users to get extra information about the event as well as allow them to transition to another view.



The bubble view will be the “web of data” for our library. It will allow an ever expanding web that will give information that is in some way relevant to the item that they are currently viewing. The view will contain the main bubble, or key bubble, which will hold the information of the item in question as well as link bubbles that are surrounding it with information that is related in some way to the key bubble. This view will be generated from one of the other views in our library. This means that from the timeline view the user will be able to select an event and the bubble view will allow them to explore that event in further detail.

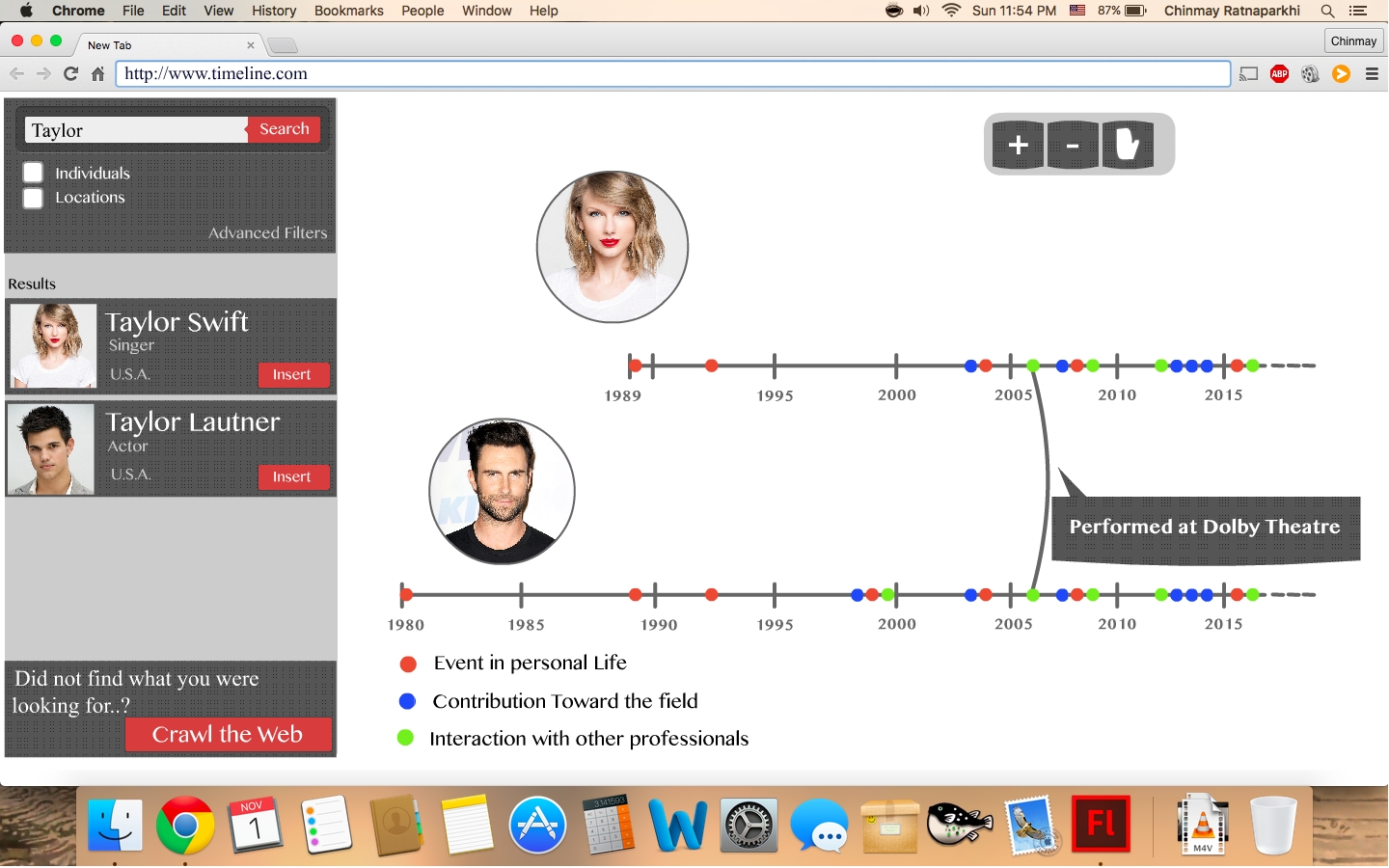
This view will allow the information to be separated into smaller pieces so that a large block of text is not the way that information is gathered. This will make it much more pleasing to explore information and will also allow those that are much more geared for visuals to not become overwhelmed when trying to research a certain idea.





The map view will be an implementation of the google API used specifically for our purposes. We will use the API so that we can place the events onto a map with accuracy as well as have the look and feel of it be pleasing. The events that are placed onto the map will all be of events that occurred during a specific year.

This view will also include a time slider which we will create ourselves. This time slider will allow the user to move through time of the location they are looking at. As the slider is moved along the map will reload with different events for the current year that is selected. This will give the user an idea of how things progressed through someone’s life or how one section of the world may have changed.

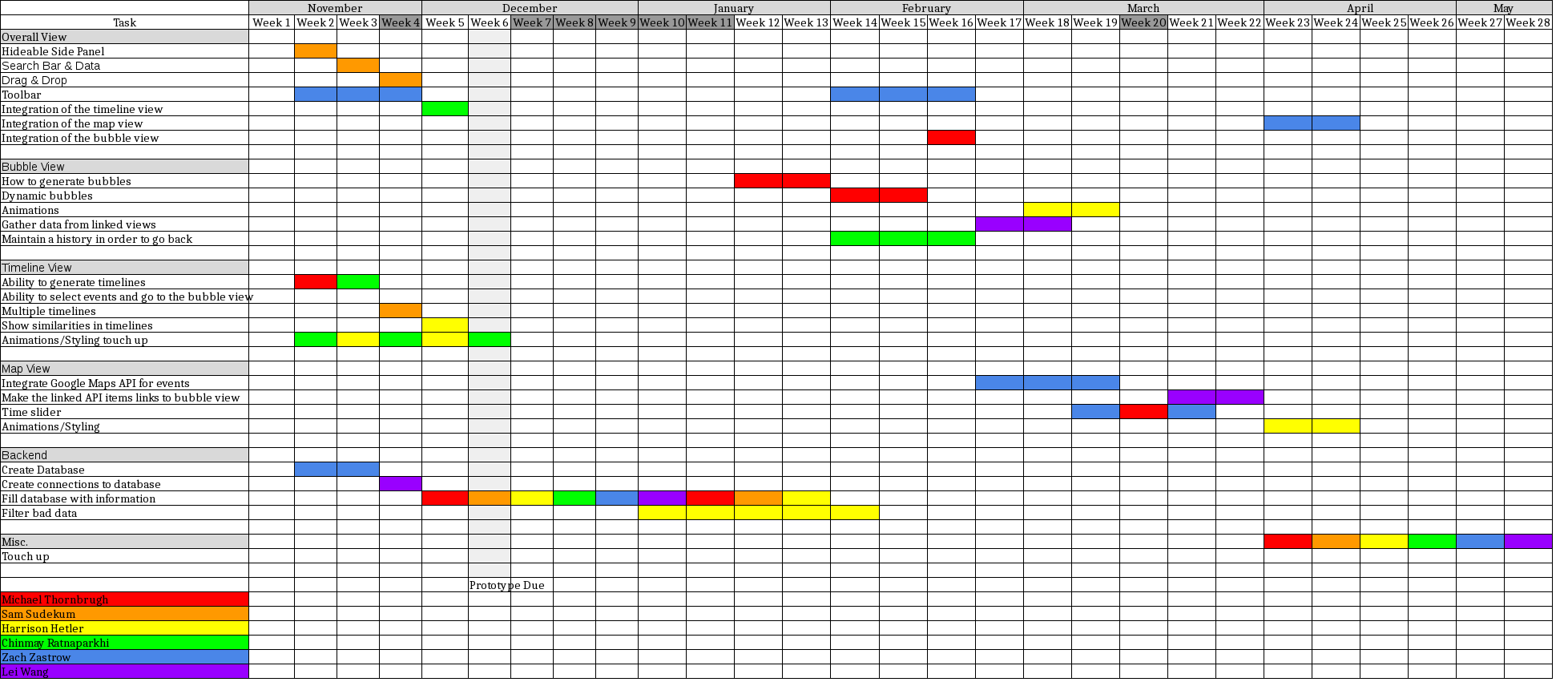


In order to allow for the views mentioned above to communicate together we are going to have an overall view that will incorporate navigation and tools that will help throughout the specialized views. The primary portion of this view will be blank space in which the primary views above will be stationed. Incorporated into this view will be a search bar where you can begin a search for information, helper items for use within the views themselves and the ability to drag and drop items from the searched items into the timeline view in order to add a timeline. The ultimate goal of this view is to help out the views while not taking up too much space.



In order to test our product we are going to be creating a backend for our product that will be used solely for our purposes; this will not be a part of the library itself. We are going to be hosting our back end services using Heroku and Heroku Postgres. This will allow us to upkeep and manage a database that we will be able to use to gather information for our library. This will also allow us to simulate how our library may gather information from other developers once it becomes used within the community. We will be able to test giving our library as little as the 4 necessary columns (who, what, when and where) as well as giving it additional information such as pictures.

<http://people.eecs.ku.edu/~mthornbr/webpage/gantt_chart.pdf>



**How our project will benefit the world**

The need for a product such as this is quite large. People that are visual learners do not enjoy scavenging through wikipedia pages in order to determine what happened to someone when they were in their 20’s. They then do not want to have to go through an entirely different page in order to find out what other events happened in the same neighborhood or the same time period. With this in mind we came up with the idea to create a library that would serve as a “visual wikipedia”. This will have information in much the same way that wikipedia is, but instead of having to read through hundreds of lines of text you can easily navigate between small amounts of text that go along with images and search for what you want to learn.

The idea of a “visual wikipedia” is meant to create a story. When the user uses our timeline they will select a story that they want to learn about and the data visualization will allow them to travel through the story in the direction that they desire. They may start out with the question “When was Beethoven born?” and once they find that out they will continue on to wanting to know. “When did he compose Symphony No. 5” and then after that want to learn what other pieces were composed around this time. These are the kinds of stories that would need multiple page searches or the scanning of a large amount of text if using existing information utilities such as wikipedia, but with the use of our library the user will be able to see the answers to all of these questions very easily and quickly.

All of these abilities will be available to developers that would be able to generate a similar idea, but will now be able to create what would take several hundred lines to create in just a few simple calls. This will allow for developers that are developing products or experiments that use large amounts of historical data will have a simple way to organize their information in a way that is pleasing.

**Previous related works**

Currently, there are some similar libraries already available to the public. However, each of these libraries are inadequate for what this project hopes to accomplish.

The most common timeline views at the moment include TimelineJS[1] and vis.js[2] . The timeline view for this project will differ from these libraries in a variety of ways. Primarily, the timeline view will allow for the comparison of multiple timelines simultaneously as well as providing comparisons between these timelines. Currently, no other known javascript library provides this functionality.

For the bubble view, BubbleTree[3] is already quite similar. However, while BubbleTree is structured as a tree with a definite starting point and endpoints, the bubble view for this project will be structured as a web where the starting point is up to the user, and there are no definite endpoints.

The map view will use the Google Map API[4]. The functionality will admittedly not be particularly unique from other libraries that already exist, but it will be compatible with the data from the other view in this project.

**Team Breakdown**

Michael Thornbrugh

Background: Javascript, HTML, CSS, SQL, Google Maps API

Interests: Javascript libraries, web service back end and the implementation process

Zach Zastrow

Background: HTML, CSS, Databases, Google Maps API, javascript (limited)

Interests: Front end angularjs and bootstrap

Lei Wang

Background: C++, Java, HTML, SQL

Interests: Learning Javascript, Bootstrap and many other APIs, also the front end.

Chinmay Ratnaparkhi

Background: JavaScript, HTML, CSS, Bootstrap, AngularJS, Flash (2D Animation)

Interests: Datastructures, Front end and the implementation process

Harrison Hetler

Background: Javascript, HTML, C

Interests: Front end angularjs, CSS animation, and back end services

Sam Sudekum

Background: Javascript, PHP, SQL, CSS

Interests: Bootstrap, Learning different JS libraries

**Budget**

Since we are planning on having a website for people to access our content, we will need $64 for a 7 month web hosting subscription from HostGator[5] and $6 for a domain name from them as well.

$64.00 - Web Hosting

* $6.00 - Domain Name

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

$70 - Total

**Acknowledgements**

[1] - TimeLineJS - <https://timeline.knightlab.com/>

[2] - vis.js - <http://visjs.org/>

[3] - BubbleTree - <https://github.com/okfn/bubbletree/>

[4] - Google Maps - <https://developers.google.com/maps/?hl=en>

[5] - HostGator - <http://www.hostgator.com/>