

Team: 1

Team members: Joseph Duong, Hoang Nguyen, Hunter Glass, Grant Henry, Liam Connolly

Project name: SeeSaw

Project Synopsis

Social media application for the purpose of having a centralized location for users to find local events. Purposed to fill the lack of such an app.

Project Description

SeeSaw is a project born out of the need to condense the search for local events into a single application rather than be scoured through a variety of sources. These sources include university club websites, facebook group pages, local business instagrams, posters, and much more. Due to the unorganized nature of this type of search, events often find users by accident; SeeSaw will allow the user to have more efficiency in searching for events and thus build community for his or her friends and the local community at large. The end result of this project would be a multi-feature application centered around local events. These features include personal and business/organization accounts, tags to filter by event type, ability to follow friends, ability to make groups, private events, an interactive map, and others. We'd also like to implement an event promotion system where people can boost their event for a fixed monetary price.

Project Milestones

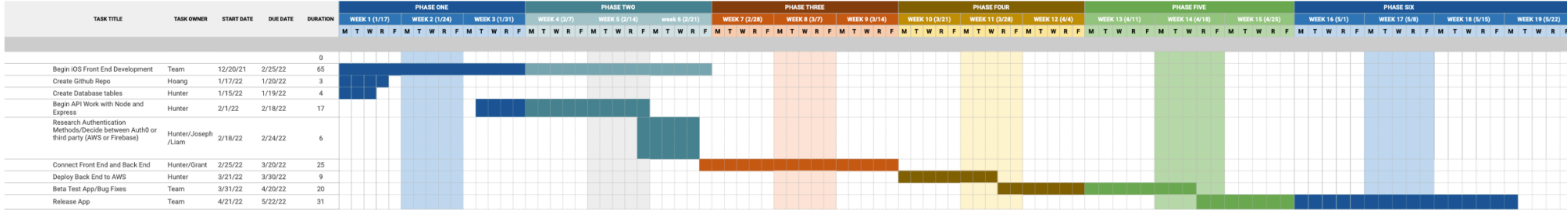
Item	Projected Date	Status	Date Completed
Team Assembled	9/20	Completed	9/20
Project Defined	9/20	Completed	9/20
Gantt chart	9/27	Completed	9/27
Rough Draft of App Design (mobile app)	10/15/2021	Changed/Completed	10/23/2021
Set up figma accounts	10/13/2021	Changed/Completed	10/20/2021
Set up xcode	10/22/2021	Completed	10/22/2021
Figure out Mac in Cloud	10/22/2021	Completed	10/22/2021
Figma app final rough design	10/29/2021	Changed/In progress	
Acquire funding for Mac in Cloud	11/20/2021	Changed/In Progress	
Research Docker Usage for Database	11/25/2021	Completed	12/1
Create Github Repo	1/17	Completed	1/20

FINAL GANTT CHART

GANTT CHART TEMPLATE

Smartsheet Tip → A Gantt chart's visual timeline allows you to see details about each task as well as project dependencies.

PROJECT TITLE: SeeSaw COMPANY NAME: SeeSaw
 PROJECT MANAGER: Team 1 DATE: 1/31/22



Project Budget

MacInCloud:

\$100

- 1 license @ \$25 a month from February - May

AWS Servers:

- Amazon RDS (host postgresql db): Free tier for 12 months- 750 hours per month
 - 20GB of storage
- Amazon EC2 instance (host node api): Free tier for 12 months- 750 hours per month
- Amazon API gateway: Free tier for 12 months- 750 hours per month

Apple Developer Account (to put on app store/most likely only need one)

- \$99

Final Project Design



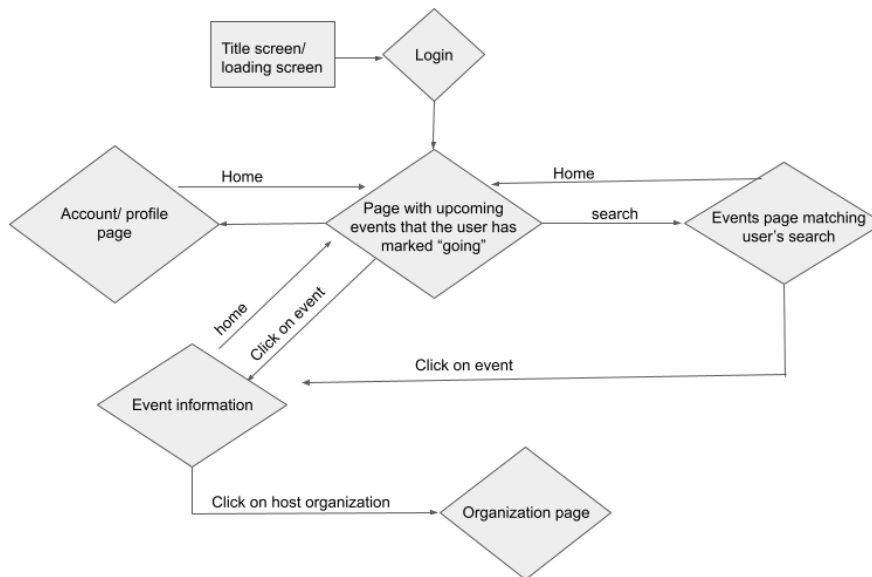
Final(rough) Homepage Design

When users enter the application, they will be able to log in or go straight through to the home page. When logging in, users will either provide their SeeSaw credentials or log in through Google. Logging in through Google will be handled by Firebase Authentication. When a user reaches the home page they can see a list of upcoming events that populate the home page based on people and particular tags that they follow. Users will also see a small banner reminding them of upcoming events that they have marked as “going”. Events within a radius of 5 to 10 miles will be given preference when populating the feed. Clicking on a particular event will expand the event out to a full page and show information about the time, location, user, and tags associated with the event as well as a set of buttons for the user to mark the event as “going”, “save for later”, or “not interested”.

associated with the event as well as a set of buttons for the user to mark the event as “going”, “save for later”, or “not interested”.

Tags are labels for events. They function similar to hashtags and allow users to find and create events of various types. These tags will be critical in allowing other users to search for events that align with their interests. At the bottom of the application will be a set of navigation buttons that will allow users to switch between the home feed, the search page, the event creation page, and the user account page. The search page will allow users to see for events, users, and tags. A filter will be used to organize these different types of information.

The event creation page will allow users to make an event. Users will fill out information about the event such as the event name, the location, the event date, and whether or not the event is private or public. A private event means that only the user’s followers can have access to the event. The last page is the user account page and will allow the user to manage their following and followers lists as well as their user settings which include changing their username and password. The user account page will also allow the user to update their location and manage their blocked contacts list.



Final (Rough) Data Flow Diagram

When considering the design constraints as the non-negotiable “must have” features of SeeSaw, we have a few in mind. Firstly, we want SeeSaw to be able to provide a consolidated location for event promotion in a community. For the purpose of defining what “local community” means, we can say for now that it means all events taking place in a city. It is critical to the mission of SeeSaw that the stream of events shown on the app are not from only one place, like a municipal parks & recreation website or university club website. Rather, our defining feature would be for anyone to be able to promote their event to everyone.

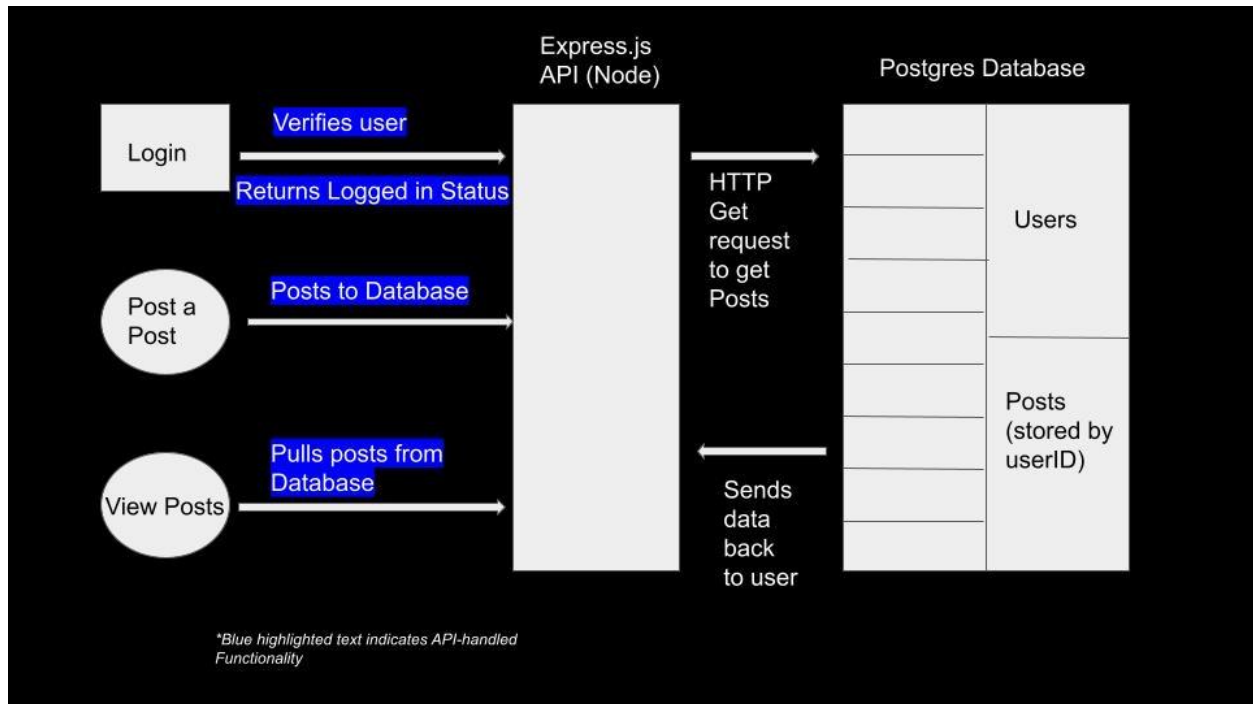
Another design constraint would be the two types of accounts: organizations and individuals. One of the biggest priorities in the development of this project is making sure events

are legitimate; this is why having accounts that can be verified is crucial to us. We'd also like to implement event tags so attendees can search for events fitting their interests such as sorting by types of events, price ranges, as well as filters for age such as 18+ or 21+. As for the attendee accounts, it is a design constraint for us that they are able to interact with events and other attendee accounts. The non-negotiable interactions are the ability to mark themselves as "going" to an event, the ability to be friends with other attendees and view their events, and search for events.

As for technical constraints, we also have some. The programming languages we will be using are Dart for the Flutter environment, JavaScript, and Postgresql. The platforms we will be using are Amazon web services, Node.js, and Flutter, Postgresql. Node-postgres is one of the node packages we will be using.

When it comes to business constraints, we have our budget to work with. As of yet, we are not sure if it's actually been approved yet, but everything else we are working with will be the free version. Our schedule constraint is the deadlines given to us by the class syllabus. Regarding team composition constraints, the only thing that limits us right now is each member's familiarity with the platforms and environments listed earlier, and time constraints as well. We predict that work will be divided up early on as we get a prototype working. After that, we expect that everyone on the team will become a "full stack" developer as we try to tie up the project.

The app will work by using the flutter framework written in Dart by google, for mobile devices. The reason flutter was chosen is because it is a portable framework that makes it easy to port over from iOS to Android, which is one of the goals in the long term. Data will be stored in a postgresql database, which is planned to be hosted on Amazon's aws RDS servers. The api to connect the database to the front end is going to be a node based api, that uses a package called node-postgres, which allows the node runtime environment to connect directly to a postgresql database in order to serve data via a rest api written in javascript. This will be hosted on an Amazon EC2 instance, which is basically a runtime environment that allows the choice to run any linux environment, and then the ability to do whatever is needed to host a server. Our plan is to run an ubuntu operating system on the EC2 instance in order to run the node app. Once this is hosted, using an Amazon API gateway instance to control api endpoints allows for us to set up specific endpoints in order to use basic http crud operations to transfer data to and from our postgresql database.



Final (Rough) Data flow Diagram

The front end will live on mobile devices, and will reach out to and from the api using the internet in order to transmit information as needed.

Ethical Issues

Enabling of Unsafe Activity

An ethical issue of SeeSaw is that a user could host an event in which unsafe or illegal activity was taking place. For instance, a fraternity's official account could be advertising a party in which illicit substances would be consumed. Although we as the app developers are not the ones promoting such activities, a level of responsibility falls upon us to prevent illegal activity. Another facet of this same issue would be hate groups using our platform. Unlike the previous scenario, there is the possibility that a completely legal group would be hosting unethical activity. An example would be the Ku Klux Klan, which has a legal right to assemble, but could be promoting an event at which hate speech would be espoused. In the worst case, hateful actions would be planned here, and people would be harmed.

There are some ways we have decided to mitigate these. One way is for there to be a user agreement that let's users know of the risk of using our app. Another is to give the users the ability to rate host users. This way, a dangerous person promoting harmful and unpleasant events could be easier avoided. Tantamount in preserving the safety of users would be verifying

organization accounts so that false events are not advertised, and that the host organization can be held liable for any incidents.

The prime issues for us as the creators would be figuring out how culpable we would be for any unethical activity and determining the level to which we are reasonably responsible for preventing illicit behavior.

Protection of Personally Identifying Information

According to the ACM Code of Ethics section 1.6, the privacy of users must be protected. With regard to the features mentioned in the project description section, user location would probably be the most sensitive of private information. This consideration would guide how we implement features related to user location.

An interesting complication is that we'd like to give the users the option to share their location, and so writing code that makes a user's location accessible but only to certain people would pose a challenge. A proposed solution to this issue would be to not actually live-track people's location but to make location be user reported.

Intellectual Property Issues

3rd Party Software applications and libraries

Through the creation of a social media platform, we will be focusing specifically on copyright issues regarding the creation of our own user experience design for our application. Our group plans to use an application called "Figma" to generate various ideas for our application's user interface, which should be credited and acknowledged in accordance with the ACM Code of Ethics. According to Section 1.5 of the ACM Code of Ethics, we must credit the owners of applications we plan to use for our own capstone's development. We will utilize other applications like Xcode, Macincloud, Amazon web services, Flutter, Postgresql, and Node.js, which will require a similar procedure for proper acknowledgement in our final version of our capstone application.

Issues with other event platforms

The University of Kansas currently supports an application called "Rock Chalk Central," which serves as a platform for various KU clubs and organizations to advertise events and meetings. While our capstone's scope includes student organizations from local universities, our application will differ significantly through its focus on the City of Lawrence itself rather than just on the University. We are targeting individuals and organizations throughout the entire community, spanning well outside the scope of a college campus. Our team is differentiating our capstone from Rock Chalk Central for not only intellectual property reasons, but also because we envision a platform being implemented on a much wider scale that could reach multiple cities and communities in the future.

Open Source Code Template

We are using an open source social media template found on GitHub to build this project. The social media app template allows our team to build off of the existing chat, page layout, and user account framework that is already available on the template. We will be adding new features and restructuring the page layout and account framework to suit the needs of the new application. The template's code will be cloned and renamed to a new GitHub repository that will only be accessible to our group. The template being used is licensed under the "Do What The [expletive] You Want To" Public License, meaning that the code can be freely used as long as the project is renamed. This means that the code can be reused in any capacity, even commercially. Usage of the template was mentioned without objection in our 1-on-1 with Dr. Johnson, but we are awaiting explicit permission.

Change Log

As of today, we have very much stuck to the wide majority of the original project proposal. The only noticeable deviation from the original plan has been our timeliness when it comes to starting on certain parts of the project. For example, we are a bit behind on the original start date of 10/15 for having complete mockups and templates. This can be attributed to us still trying to figure out the use of Figma.

We decided to use docker containers to run the databases as opposed to installing all the postgres/other required software as it simplifies deployment and allows for every team member to have the same images. Using docker allows us to deploy the database to AWS by using a docker-compose file, without the need for configuring all the tables again inside of AWS servers - essentially just uploading the tables we create in the development phase.

Another deadline that was changed was the iOS front end development - this was changed as team members had many errors trying to get our repo to run on a simulator. This was mainly caused by flutter not being the right version, so upgrading flutter solved that.