CS 316 Open Project
Sam Ginsburg, Ethan Gottlieb, Joe Jacob, Greg McKeon
Milestone #1: Project Proposal
10/15/15

Spawn

Project Description:

Spawn is a social networking web application that allows you to join or “spawn” events to enjoy spontaneous meetups. Whether you’re in the mood to get a quick bite, play pick-up soccer, see that new movie, go on a hike, or head to a party, Spawn brings the right people together at the right time.

Glance through upcoming events (all within 2 hours) created by your Facebook friends or by nearby users and view the event type, distance, time, attendees, etc. If an event strikes you, join and connect with other attendees! If you can’t find what you’re looking for, “spawn” an event of your own! In the process, you’ll catch up with friends you don’t see often and meet fun people.

By the end of this project, we plan to demonstrate a functioning web application that allows users to login and create or join events. The main page of the web application will consist of a list of events, and users will be able to query and sort events by certain attributes (type, time, location, etc.) to help them find the perfect event. Stretch goals include incorporating maps, messaging, advanced social features, etc. and enhancing the UI.

This application is useful and interesting because there currently are not other websites/apps that encourage spontaneous meetups. Most other event apps are about helping to plan future events, while Spawn stresses the here and now.

Technical Description:

We plan to use the Meteor JavaScript framework to build the web app, in addition to HTML and CSS. To build the UI of our web app, we’ll be using bootstrap. Meteor uses MongoDB, which we will use to store event and user data, and has features that help with processes like logging in. Although the possibilities of events are endless, they all have most of the same attributes including title, description, type (sport, party, meal, etc.), time, and location. Each event will also have a user that is the creator (with additional event options), and a list of users that are attendees. Users will also have similar attributes that we will populate mostly using Facebook. We hope to be able to create relationships between users based on connections like location and “friends.” All the data in Spawn will be user generated, although we may initially create hard coded events if traffic is low.

As mentioned previously, other apps exist that focus on future events instead of events going on now. We will focus on making the user experience simple and intuitive, by only including information that is important to users. Much of our app relies on creating a reliable

social network, which has been done in many different ways, so we can study other social networking apps to help build and design our initial system.
Database Description:

We plan on using MongoDB for our database. For MVP1, we plan on only having two tables: Events, with a primary key EventID, and User, with a primary key FB_ID. Their schema looks like:

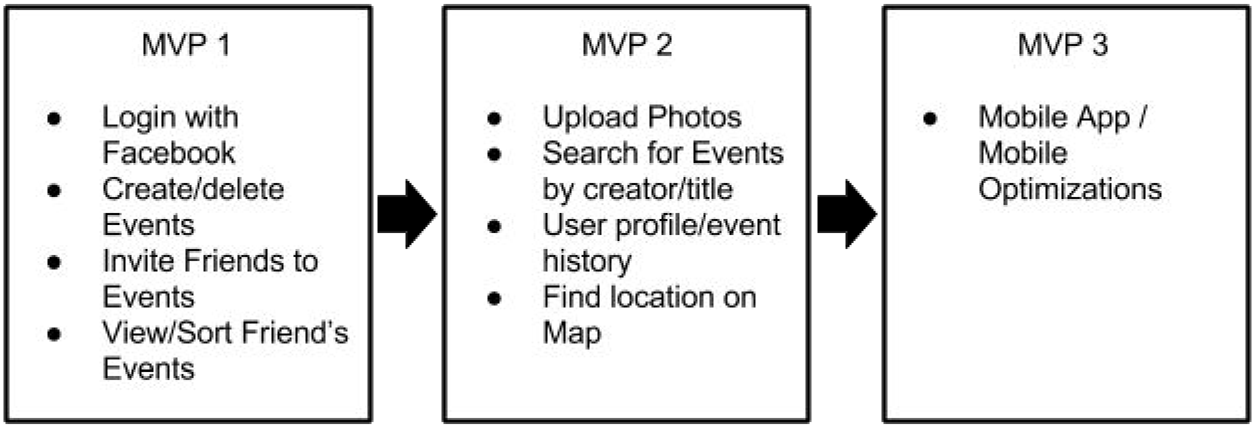
Event(id, Title, Description, Owner, attendees, location, date, duration, genre, maxPeople, private)

ID is a unique number representing the actual event. This is the primary key for the database, since it is the only property guaranteed to be unique. Title is a string which represents the name of the event, while Description is a String that contains the details of the event, as set by the owner. Attendees is a list of id's of Users who will be attending the event. Location is a pair of floats that represent the coordinates of the location (or, for MVP1, a String with the address). Date is a date object representing the start time of the event. Duration is an int representing the event length in minutes. Genre is a String containing the event category name. MaxPeople is an int containing the max number of people who can attend, while private is a boolean indicating whether or not the event is publically visible.

User(FB_ID, name)

For now, we do not store any settings information in the User table. As we develop the project, particularly the profile piece, we will use this table to store User preferences and settings, as well as profile information we cannot pull from the Facebook login.

Minimum Viable Products/Goals:



MVP 1

- Login with Facebook
- Create/delete Events
- Invite Friends to Events
- View/Sort Friend's Events

MVP 2

- Upload Photos
- Search for Events by creator/title
- User profile/event history
- Find location on Map

MVP 3

- Mobile App / Mobile Optimizations

We broke the project down into its core functionality, and features we think would make the product more successful. Our goal is to add additional functionality after completing MVP1, possibly changing later MVPs to include functions we find to be necessary from work in previous sprints.

A brief summary of your discussion with the instructor or TA (which is required before submitting the proposal).

Professor Yang liked the idea and didn't have any objections yet. He also supported the technical decisions of using Meteor and MongoDB as it would allow us to more quickly deliver a functional project. He said he'll give more feedback once he gets the full proposal.