

1. Build Backend models to handle Users, Events, interface with YELP and Facebook APIs. (Trivial mongodb initially)

1.1 Make RESTful API to correctly relay data between each client and each server.

1. Start implementing basic responsive web view

2.1 different users need different interface: E.G. club owner and bouncers need different views to club goers.

## Rating system works with either a thumbs up or thumbs down system.

- A goer will be asked how the event was, using thumbs up or down,
- A owner will ask how each goer was (can leave blank which is nutral) or a great guest gets a bonus
- Good guests will occasionally get rewards from clubs (or from GoParty itself) where bad goers will be Blacklisted.

## Authentication

Facebook will be required for authentication, for real name and basic details such as age.

Will ask user if details he enters on facebook are correct. If not, deal with this later. **LOOK AT LEGALITY OF FORCING AGE CHECKS?**

## Events

- Events can be either public, private, or invitees can also invite.
- Events have a maximum number of goers.

## Cost

- Initially, this will **NOT BE TIERED**. ONE PRICE FOR ALL.
- A event can have a price.
- This price will be handled online using paypal or equivalent.

- A description can be presented for what the money will be used for.
  - This allows for a system where the host can prebuy alcohol or even food which helps alleviate the costs of an event.
- Initially, **NO STAKE IS TAKEN BY ANY PARTY OTHER THAN OWNER OF EVENT**

## Clubs

- Clubs can use GoParty to help alleviate queues due to time taken paying, and also pre checks age. (ID's still checked at door)
- Helps reduce fake IDs due to facebook name not lining up. (MUCH HARDER TO FAKE)
- Clubs get special version of the app which has a barcode scanner to scan each users 'ticket'.

## Funding

- Initially, the application will be funded purely through advertising. An example of this is clubs would pay and then get a higher preference in search results when compared to other clubs.
- Targetted advertising should also be in place e.g. beer companies.

This funding is required for Server Costs and maintainace costs.

## Components

### Models

#### Events

- if event is private, venue will not show on map for that event, unless you have access

#### Users

#### Venues

- if venue is private, then It will not be able to be seen on map, and will make all events private.

### Views

- Bouncer View Has a barcode scanner to scan a ticket of a user who wants to enter. Can line up facebook name used with actual name.
- Goer View Can freely browse events
- Owner View (Also has inbuilt bouncer view) Full Admin over event, can see who is attending etc and how much money has been accumulated.
- Main View Any user has access to this. They can see other Events in the area and create a new event if they so choose.

## Controllers

### Venue Database creation

- Handled using YELP API.
- Rerun automatically *every MONTH??*
- Users can add to Database at any point **NEEDS SANTISED**

### Event Creation

- Venue needs to already exist - if not create a venue.
- User can then create event easily.

### User accepting/rejecting event

- if user accepts the event, then he will have to pay the event charge. This is a flat charge, and will be done using card.
- A Barcode is then generated from your User private key, and a receipt of the transaction will be emailed to you. <http://davidshimjs.github.io/qrcodejs/>

### Bouncer barcode scanning

- Done using this library <http://blog.cdeutsch.com/2013/09/how-to-add-barcode-scanning-to-your-web.html>
- <https://github.com/LazarSoft/jsqrcode>