2/7: Meeting 1 with RVR

Components needed

- DB backend
- Web server

Architecture considerations

- Need to discuss frameworks vs individual DB, backend, frontend connectivity
- Eventually host on cheap server- Bluehost is an option

Portion 1: Website (essentially a graph of pages)

- Each page should have a simple URL identifier
- Front page should have option for registration
- Someone should either be able to register as an organizer or musician in a band
- Create a login with username, email address, password
    - Store this information in a DB table for emails. See Portion 3 below

Organizers

- Able to create a new Porchfest instance
- Enter information including date of event, timeslots for performances, location of event
    - Simple way to specify location: GPS coordinates with radius
    - Best way: Actual google maps integration
- Organizers should be able to upload a logo and put text on their Porchfest instance page
- Set a deadline for musicians to register their bands

Musicians

- Register himself/herself and band
- Potentially could register a porch to play on as well, but this is not necessary
- Input time conflicts, location conflicts

Portion 2: Scheduling – NP hard constraint optimization

- There will be a list of bands, people in each band, which times the band can or cannot play
- Need to create a schedule with no conflicts
    - Some conflicts are explicitly listed
    - Some conflicts need to be inferred
        - i.e. A musician is part of multiple bands. Those bands obviously cannot be scheduled to play at the same time
- *Phase 1*: Software generator that places all bands on porches in different timeslots viewable to the organizer
    - Organizer should be able to move things around and fine tune by hand on our interface

- *Phase 2*: Organizer sets second deadline for musicians to fix mistakes/update conflicts
    - Generator should run again but not the full algorithm because of time efficiency issues

Portion 3: Email

- Mail listservs stored in DB tables to send mass emails
- RVR wants to be able to queue emails first before sending- need to clarify this with him
- Iteration 1: simple txt messages are fine
    - More advanced: macros to substitute portions with information from DB tables

Group Organization

- 3 main task groups
    - Emails and DB backend – large component
    - Scheduling – large component but comes into play later on
    - UI, ideally mobile compatibility so people can check schedule of events on the go
        - Smaller team should be fine
- Set up Gantt chart for timeline of subgroups
    - Map out dependencies

General Outline

- UI needs to be reviewed by client early on – this is the main first step
- Will have an initial demonstration of simple functionality followed by more sophisticated demos upon receiving client feedback
- Client should be involved in the whole process in order to hand off easily at the end and maintain after we leave