

Prototype photos:

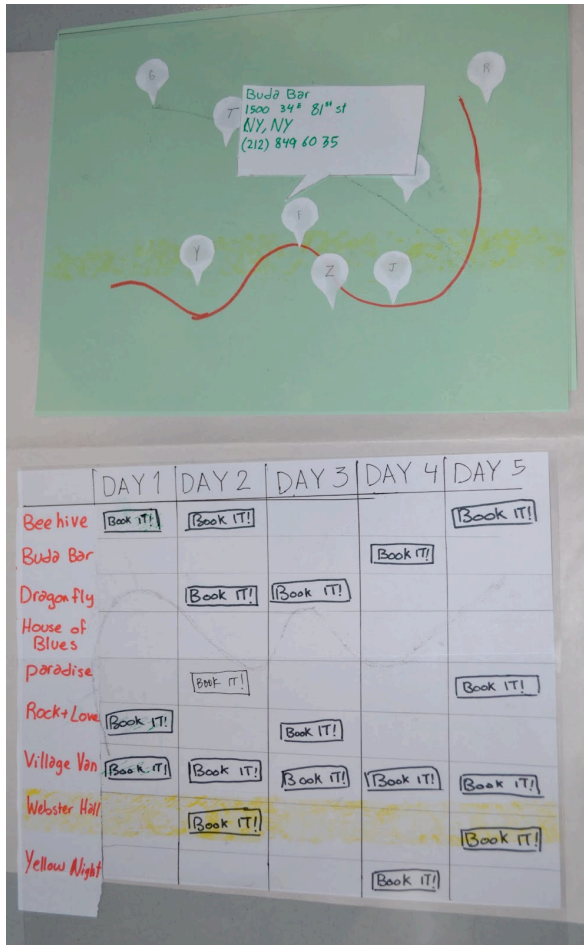


Figure 1: Iteration 1: Relative node view. When a venue is clicked in the node-view it is scrolled to (if necessary) and highlighted in the list-view.

*Venues in this test are sorted in alphabetical order.



Figure 2: Iteration 1: Available times for scheduling venue A on Day1, this time with venues sorted by distance (as opposed to alphabetically).

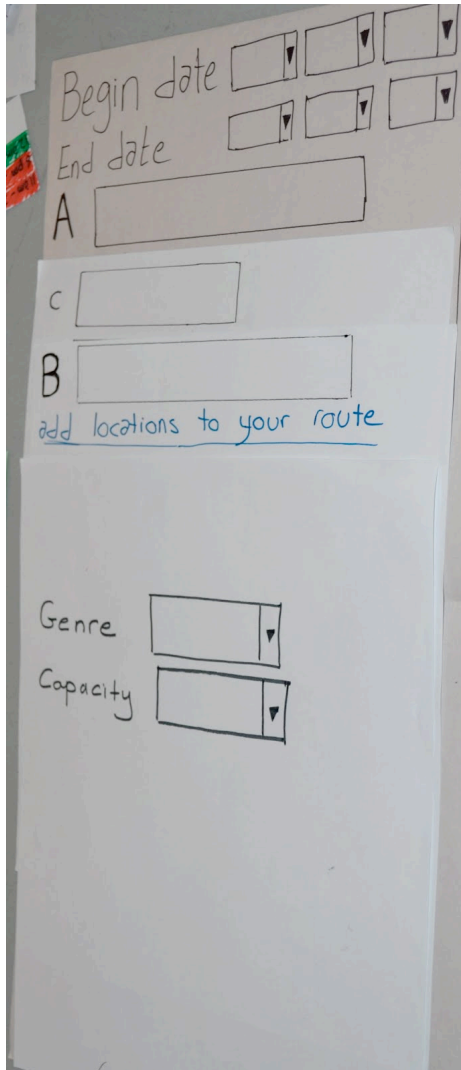


Figure 3:

Round 1 Task 1: Enter start and end cities (along with cities the band wants to “hit” along the way) and travel dates. Filter output based on venue genre and size.

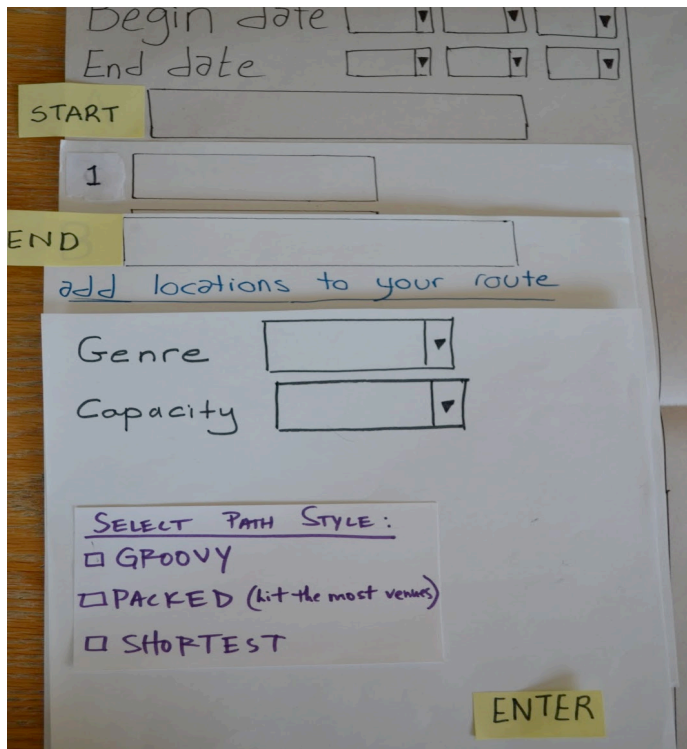


Figure 4:
Round 2 Task 1: Enter start and end cities (along with cities the band wants to “hit” along the way) and travel dates. Filter output based on venue genre and size.

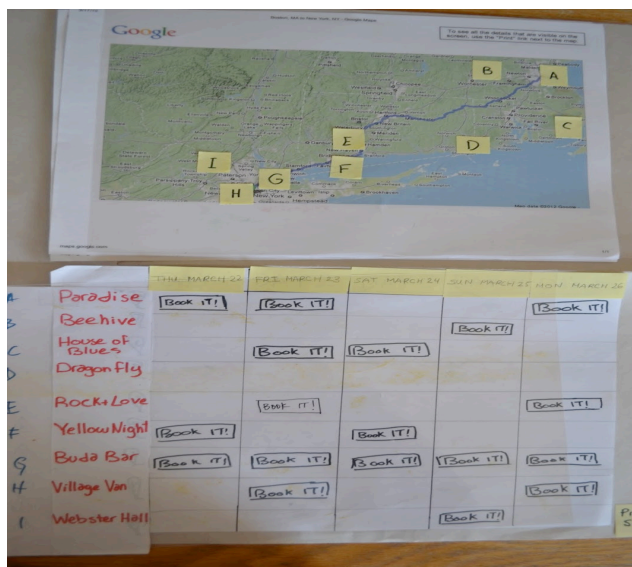


Figure 5: **Round 2**: map view and more specific labels for four dates (as opposed to Day1, Day2, Day3 in round 1).

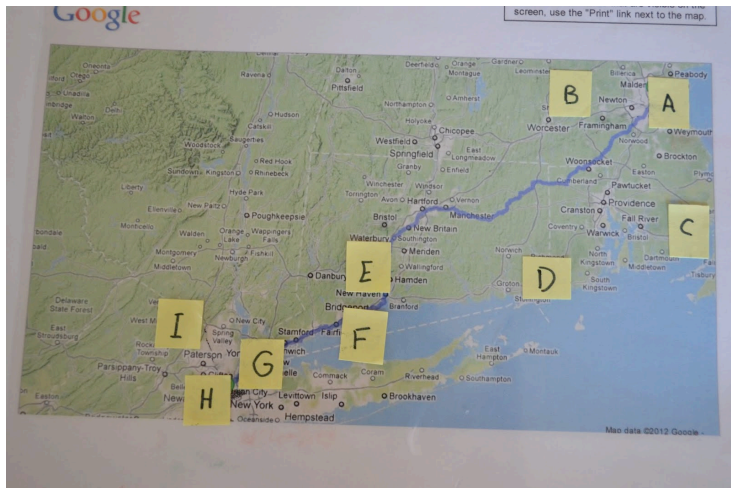


Figure 6: Round 2: close up of map view

	THU MARCH 24	FRI MARCH 25	SAT MARCH 26	SUN MARCH 27	MON MARCH 28
A Paradise	Book IT!	Book IT!			Book IT!
B Beehive				Book IT!	
C House of Blues		Book IT!	Book IT!		
D Dragon Fly					
E Rock + Love		<input type="checkbox"/> 10 am - 1 pm <input checked="" type="checkbox"/> 5 pm - 9 pm <input type="checkbox"/> 10 pm - 1 am			Book IT!
F Yellow Night	Book IT!		Book IT!		
G Buda Bar	Book IT!	Book IT!	Book IT!	Book IT!	Book IT!
H Village Van		Book IT!			Book IT!
I Webster Hall				Book IT!	

Figure 7: Round 2: close up of list with “Rock + Love” highlighted, and “book it!” button on day 2 selected, a click menu appears for user to choose time to book.

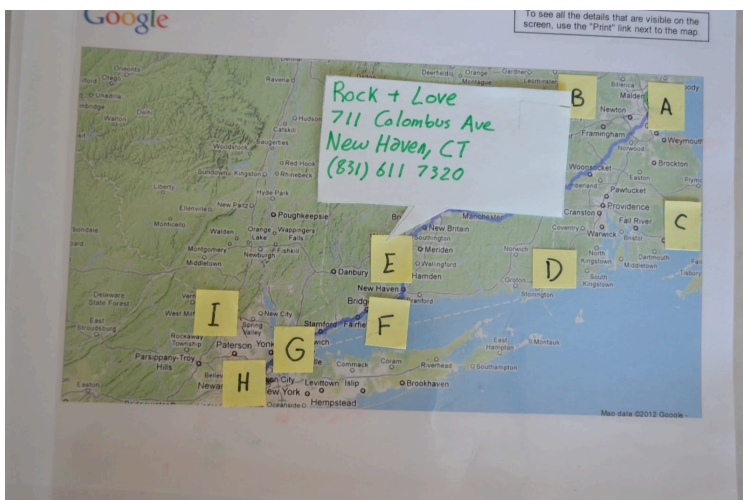


Figure 8: Round 2: User hovers over venue “E” to reveal venue information.

Briefing:

VenView is an application that helps musicians create and optimize a tour. VenView is designed to display multiple paths weighted on different metrics (shortest-travel time, venue density, and an average of the previous two metrics). Using this tool, musicians can make more informed decisions as to how to make the most out of their travel time.

You are a Jazz band from California. You are flying to Boston on March 21st, to start a tour from Boston to New York City on the 22nd. The tour is five days long and you want to use *VenView* to help you find venues to book along the way between your concerts on the 22st (in Boston) and on the 27th (New York). You've heard that New Haven has an emerging underground jazz scene and so you decide that you definitely want to pass through.

*Note: This was the explicit briefing we used for Round 2 of testing. For Round 1, the briefing was slightly more generalized and we modified it for Round 2 testing.

Scenario Tasks:

Task 1: See all smaller Jazz venues on your path (max capacity < 300). And add New Haven as a city you'd like to visit.

Task 2: Book venues for at least 3 days of your tour.

Task 3: Process your final itinerary.

Observations: (1st round)

Testers were confused by the relative node view:

One of the testers did not actually refer to the node-view at all. He felt it was unnecessary and booked venues with only the list-view. Another tester was a bit confused with the node-view, and compulsively cross-referenced the list with its matching letters in the node-view to check where the venue was located.

Overall, we observed that the node-view obfuscated users' understanding of venue's physical locations relative to their tour path.

Updated path:

While in the process of 'booking' a particular venue, one tester wondered how his booking-action would affect the original plotted path between cities.

Venue List Ordering:

Alphabetical ordering of venues in the list view compromised some users' efficiency.

Even though clicking on a venue in node-view would scroll and highlight to the relevant venue in the alphabetized list, many users did not discover this feature and manually tried to match venues in the node-view with venues in the list.

Booking:

Testers asked about making multiple bookings in a single day. Previously, we had assumed that users would just book one venue per tour-day (demonstrated by grey-ing venues open on the same day that another venue has been booked by the user).

Choosing Venue:

Testers weighed distance and time in their decision making process as to which venue to book.

While we did not have time to implement a test for a color-based system of “suggesting” certain venue times over others, such a scheme had come up in our design process previously and was suggested by our TA.

We outlined how such schemes would look to testers as follow-up questions after they had performed the tasks outlined above. All testers favored such an approach and were split in terms of whether a spectrum of colors would be too confusing versus simply “highlighting” suggested venue-times.

Performing Task 1 (Filling in Left-Hand Column):

Most users skipped the year dialog and seemed to assume it would automatically display the current year.

Selecting Generated-Path “Style” (Groovy, Shortest, Packed):

Selection of route type:

Users discovered what different path styles were by trying them out and inferring meaning. Often they were a little off and no other explanation in the app was provided.

List View Display: Tour “Days”:

As shown in Figure 1, our first prototype labeled tour dates as “Day 1, Day 2, etc.”.

One user explained that day of the week would impact his decision making process as to which venues to book (Sunday Brunch versus Monday Brunch, etc.)

Preparing Second Paper-Prototype:

Originally we had decided to use the node-view as outlined in GR2 for simplicity. Since users weren't actually accessing driving directions, it seemed unnecessary to explain lots of additional information. Through testing however, it seemed that users were skeptical of this distorted view. So often a relative-style view such as our node-view is used to distort a user's impression of space via some other metric eg. The map below is warped based on HIV distribution of all people aged 15-49.



We discovered (by talking to our TA), that we can use canvassing to simplify the map view which avoids our having to get rid of it entirely (as with the nodal-view). Switching our design to the map view was the principal change we made with our second prototype.

Additionally, we decided to add a dotted line that would be updated incrementally to mark the user's path based on bookings selected. This dotted-path will be displayed alongside the original path in the map-view.

We abandoned the alphabetized list approach in favor of ranking venues based on distance from the start location (radially outward).

We modified our greying-out scheme to be more lenient. So venues were only greyed out for a particular day if the exact time slot was taken compared with a venue that was already booked for the same day.

We also preset the year field with the current year (but still included an arrow-affordance for changing the year). Lastly, we updated "Day 1, Day 2, Day 3..." to read the day of the week and the date in the List-View.

Observations: (2nd round)

Cancellations:

Testers asked about how to cancel a booking once it has been selected.

Accessing the schedule of bookings:

Testers wanted to access the schedule card widget while in the process of booking another venue. Our current layout didn't enable this use case well.

Efficiency between List and Map views:

Testers still struggled with knowing the location of a venue when using the map-view and spent some time rectifying the two representations.

Adding cities between the start/end destinations:

The link labeled "add locations to your route" was ambiguous to one user; he was unsure as to if this would add locations to the end or middle of the tour.

Visibility Issues:

One user discovered a venue he definitely wanted to play at midway through his booking process. All of the times for that venue were greyed out since he had booked conflicting shows

and it was unclear to him how he could reverse these actions to rectify these conflicts and instead play at this new venue.

Round 2 Design Adjustments and Reflections

While users were able to better understand the non-list view, they still had difficulty with rectifying the different information displayed on both views.

Additionally, we need to think about our design more in terms of how users can incrementally revise their route even after they have booked most of their tour dates. We also need to make the user's own schedule (the schedule card widget) more visible/accessible in different steps of the tasks.

In addition we need to consider how the design will suggest relevant venues. We did not have time to implement a test for a color-based system of "suggesting" certain venue times over others. Although we did outline such a scheme as a follow-up discussion, and testers favored such an approach theoretically, we would need to confirm such a scheme by testing with a higher fidelity paper-prototype and researching other interfaces that implement a color-ranking approach.

Overall, it seems there was no extraneous information or overlooked features in our design. All of our test users did refer to both the map and list view. While we have visibility kinks to work out, it seems that the model that we exposed to the user was sufficient for him to complete all required tasks.