

# Rubato

## Communal Playlists in Real-Time

Parker O: Product Manager, Gio J: User Testing & Design, Peter W: Developer.

Studio: Creation

Assignment 5

### Introduction

*Mission Statement/Value Proposition*

Value Proposition: Communal playlists in real time.

Problem: Music is a dynamic force that brings communities together, but there is currently no way for all listeners to engage with each other in the creation process in real time--together, right here, right now.

Solution: Rubato allows everyone at an event to contribute in real time to the music. The result is a constantly evolving playlist, curated by the entire community.

### Sketches

*Overview of All Sketches*

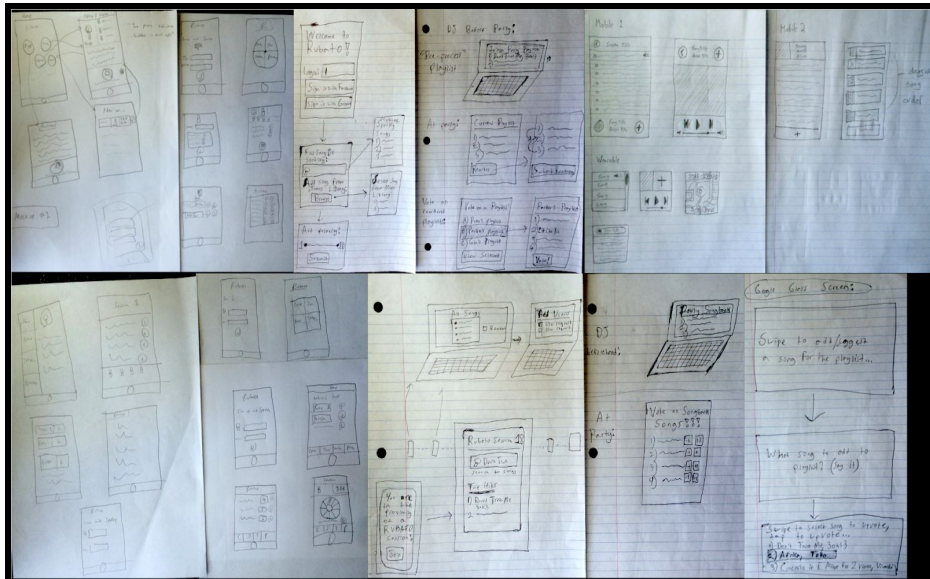


Figure 1: Overview of all of the sketches we came up with. To save paper, there are multiple sketches on some of the pages, especially on the left 4 sheets of paper shown.

### Storyboards for Top-2 Sketches

The idea for our first sketch is that partygoers can join a Rubato session and start adding songs right away. As seen in Figure 2, no songs are preselected - rather, the users play songs on-the-fly. There is no voting for this idea because we thought that users wouldn't want the burden of constantly voting during a party.

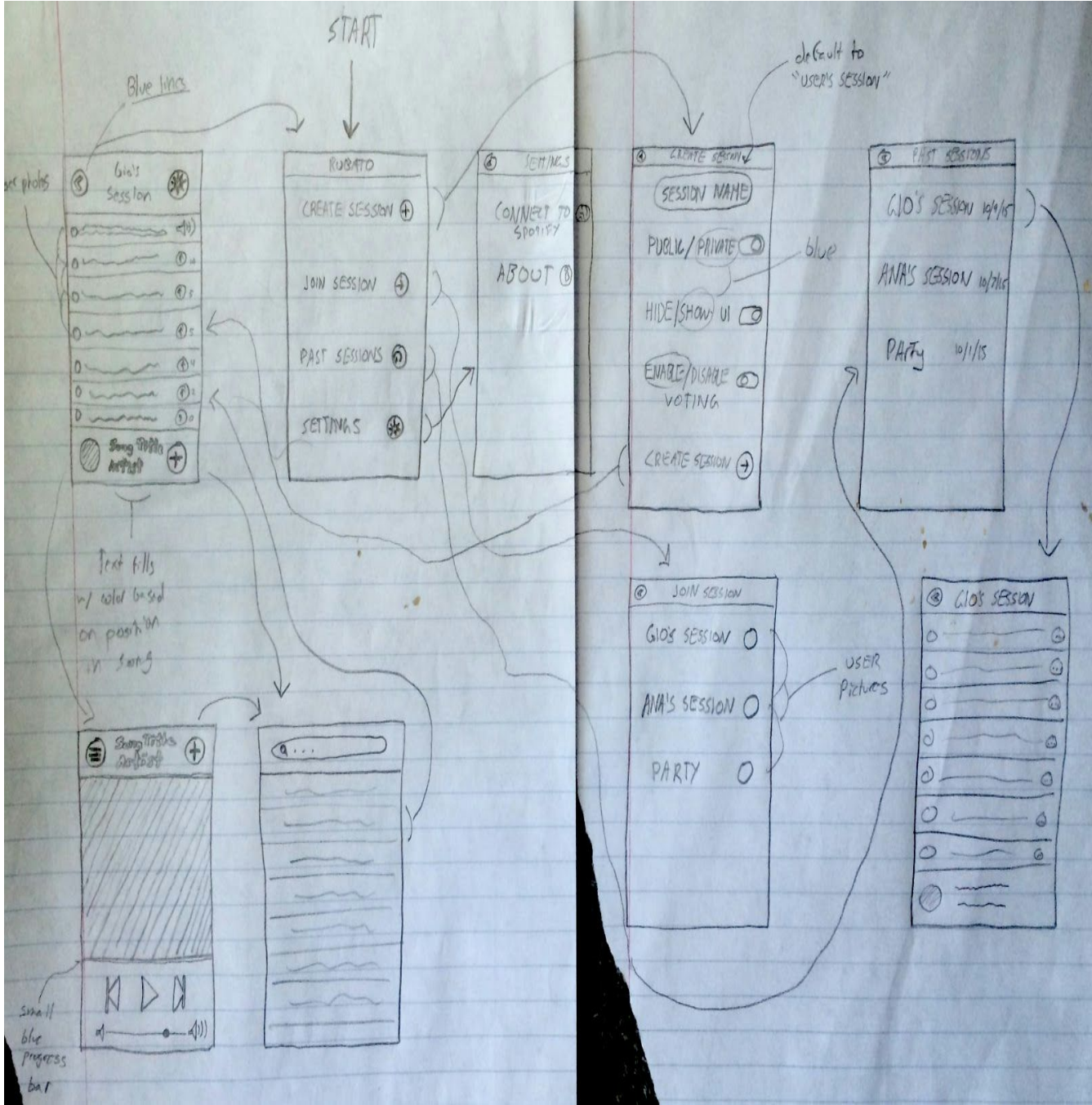


Figure 2: The first of our top-2 sketches.

The idea for our second sketch is that the party host pre-selects all of the songs beforehand, creating a song bank that users can select songs from. As seen in Figure 3, sessions to join are password-protected, with the party host setting a password beforehand. The user still performs the three tasks, but from pre-selected songs rather than any song in the Rubato library. We also didn't including voting for this version of the app.

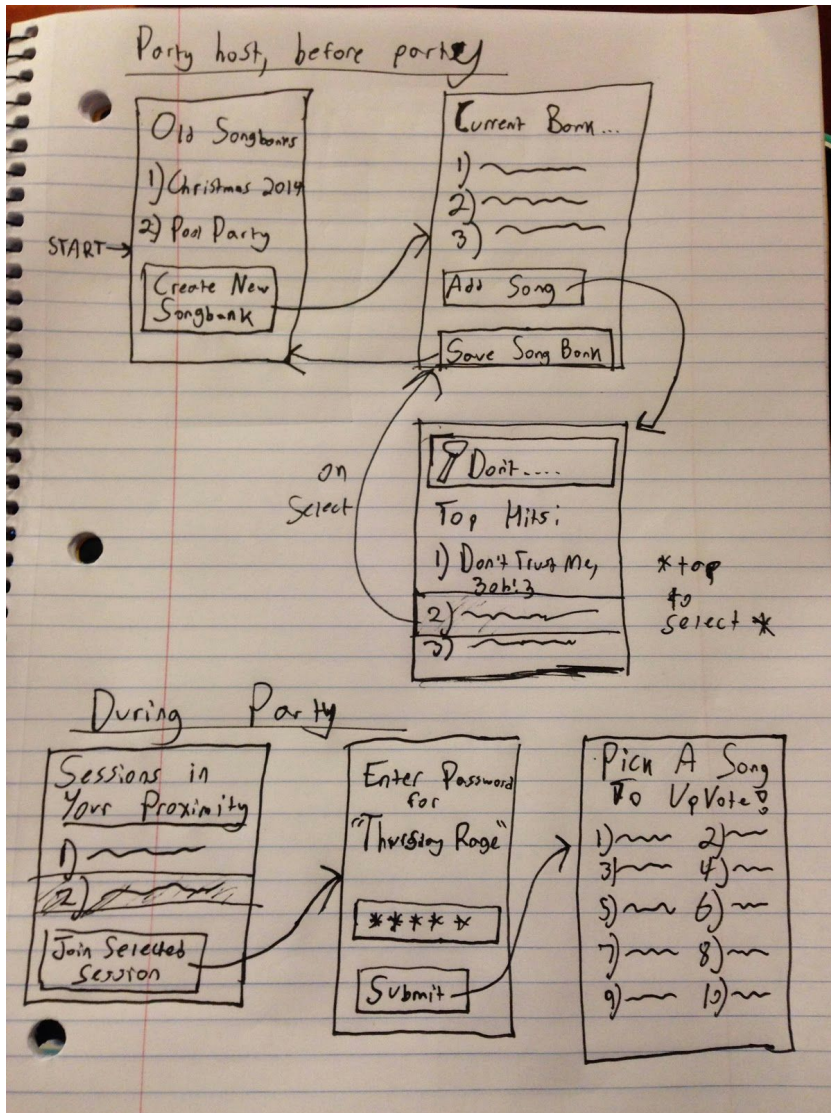


Figure 3: The second of our top-2 sketches.

## Selected Design Interface

### Storyboards for 3 Tasks

Our three storyboards for each of our three tasks are shown below.

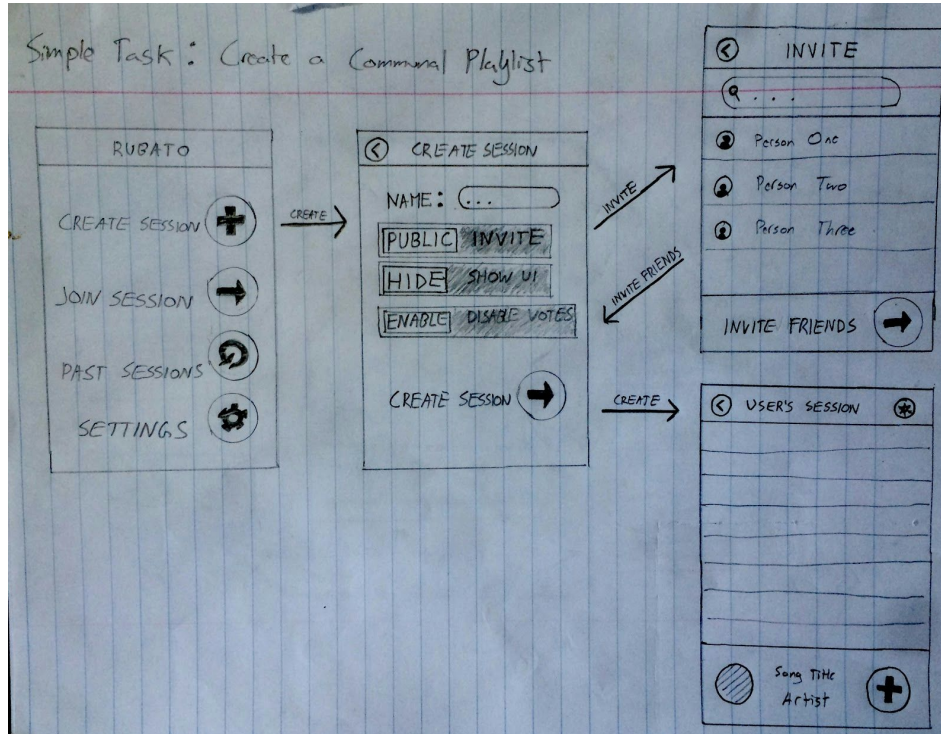


Figure 4: Storyboard for simple task.

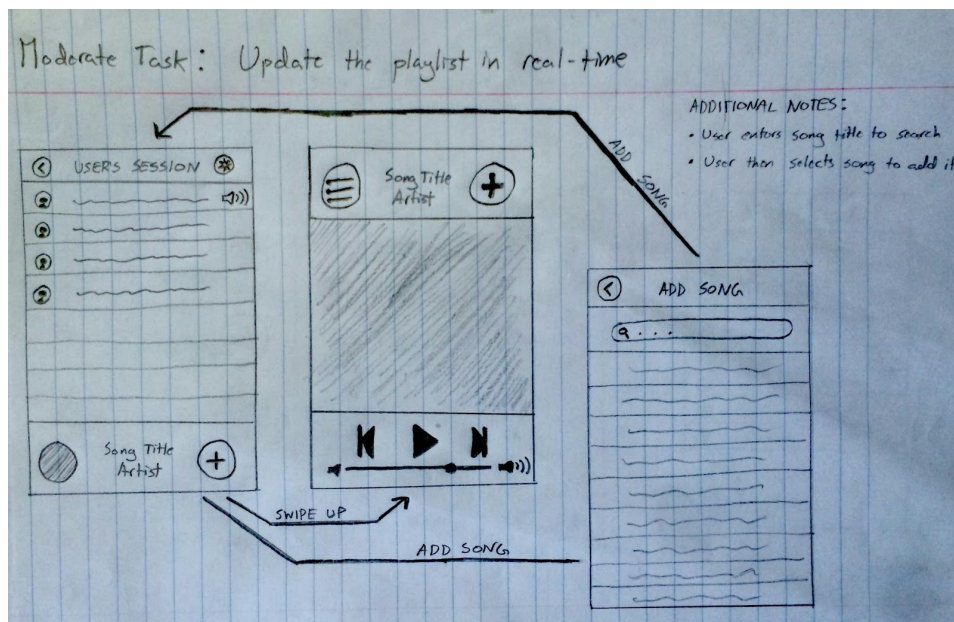


Figure 5: Storyboard for moderate task.

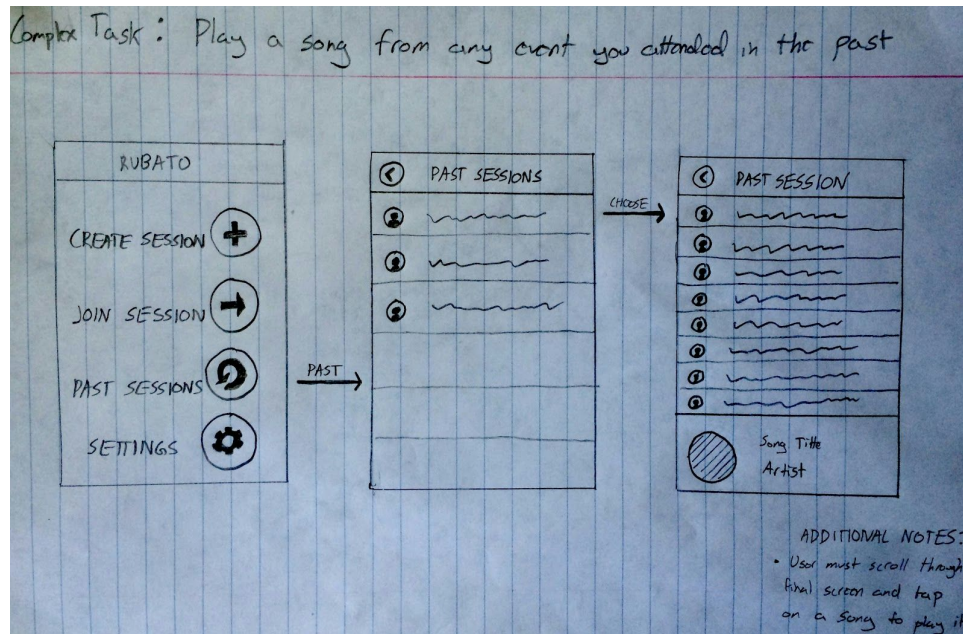


Figure 6: Storyboard for complex task.

### Reason for Selection

We ultimately chose this UI design over our other option because of its simplicity, both for users at the party and for the party host. The host has no preprocessing to do before the party, only being required to start the session and add an initial song to the playlist. The partygoers can add songs during the party, but then can enjoy the party without having to stress about choosing only pre-selected songs (as choosing from a list of songs you are not a fan of could be annoying and limit the potential for curation of the playlist). Furthermore, there is no password that needs to be entered for partygoers, further alleviating the complexity of using the app during the party.

### Prototype Description

As shown in Figure 7 (on the next page), our prototype consists of paper cut-outs used to represent different screens of the app. These cutouts were superimposed onto an actual iPhone 6 device in order to maximize the realness of the prototype. Pairings of these cut-outs represent our three primary task flows. The first flow starts with a home screen that allows the user to create a new “Rubato session”. Once the session is created, the second task flow involves the user adding songs or inviting friends to the current session. Finally, users can “re-live” previous playlists from the home page by accessing “previous sessions”.

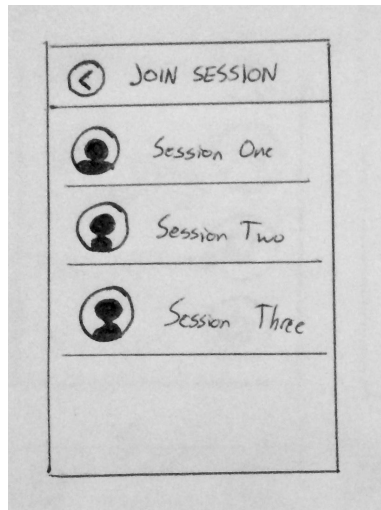
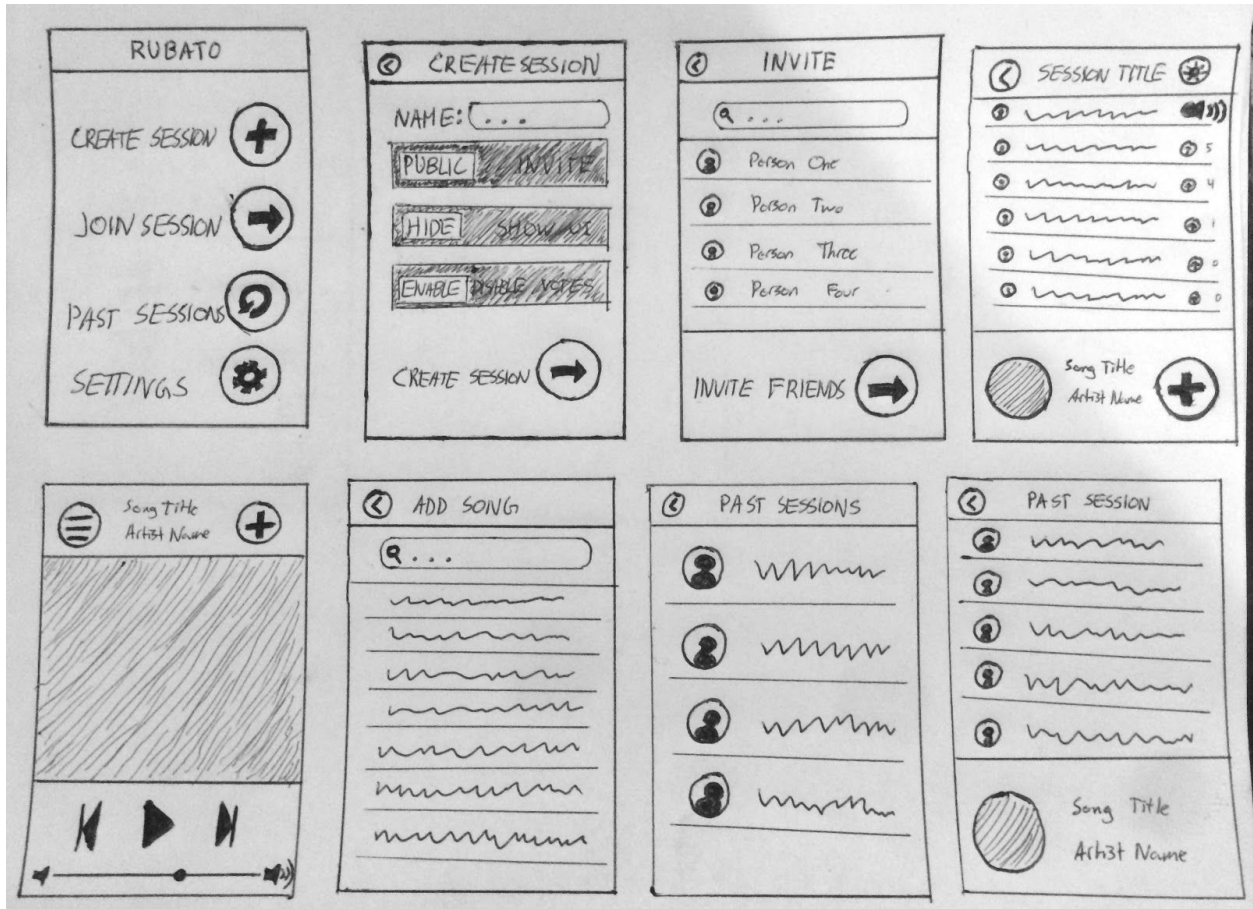


Figure 7: An overview image of our finalized low-fi prototype.

## **Method**

### *Participants*

We asked residents within the Roble Lounge if they liked going to parties and listening to music. If they responded affirmatively, we asked them if they would mind participating in a team experiment for our CS147 Class. We also offered them a couple of donuts from Happy Donuts as an incentive for participation.

### *Environment*

The team conducted our experiment in the Roble Lounge around 10 PM on a weekday night. There were plenty of residents socializing, playing ping-pong and working together. We figured this would be an ideal environment to test the various interactions with our low fidelity prototype because it was an active social scene.

### *Tasks*

Simple task: Create a communal playlist.

Medium task: Update the playlist in real-time.

Hard task: Play songs from any event you attended in the past.

### *Procedure*

Parker served as the main facilitator for the experiment with Peter and Gio operating the music selection as well as being note-takers. After reading and signing the consent form (see appendices), Parker explained that we would ask him to complete three tasks within the mock UI. For each task, Parker would instruct the participant to complete it to the best of his/her ability while Gio & Peter would take notes on any specific behaviors noteworthy during the simulation. When the participant chose a song initially to add to the playlist, Peter would play that song. However, when the participant added a second song to the playlist, Peter would play either "Back to Back" by Drake followed by "Don't Trust Me" by 3oh!3 to simulate the presence of other participants contributing to the active playlist. We played these songs regardless of the song choice of the user because in a real party, the songs that the user is choosing would have little to no bearing on the songs that other partygoers would choose. We then told the user that the party was over and that it was the next day. We asked them to use the app screens to replay the playlist that was created during the event. Once the user successfully navigated the screens and selected the appropriate session, we played the entire playlist for the user. Upon completion of the three tasks, the team would thank the participant for his/her time and would offer him/her a selection of donuts as compensation.

### *Test Measures*

When observing participant behavior during the testing we particularly kept an eye out for ease of use, quickness moving between tasks, and number of errors.

### **Results**

We kept track of the events that occurred during the testing via a log of critical incidents. We use a scale where 0 indicates no problem, 1 indicates a cosmetic problem only, 2 indicates a minor usability problem, 3 indicates a major usability problem, and 4 indicates a usability catastrophe.

#### Participant #1

*Task 1: Create a communal playlist.*

0: The user successfully created the Rubato session.

*Task 2: Update the communal playlist in real-time.*

0: Was able to create the playlist.

2: Wasn't sure what to do with the "Hide" vs "Show" switch.

2: Wasn't sure whether "Settings" button indicated session settings or playlist settings.

3: Added a song to the playlist just to "troll".

4: Added the song to the playlist again in order to keep on spamming the playlist.

*Task 3: "Re-live" the playlist.*

0: The user successfully accessed a previous session and listened to a song from it.

3: Had trouble navigating between songs during playback, as the interface (as it was) only allowed the user to listen to the past session rather than skipping between songs.

#### Participant #2

*Task 1: Create a communal playlist.*

0: The user successfully created the Rubato session.

*Task 2: Update the communal playlist in real-time.*

0: Was able to create the playlist.

3: Added a song to the playlist just to "troll".

*Task 3: "Re-live" the playlist.*

0: The user successfully accessed a previous session and listened to a song from it.

3: Had trouble navigating between songs during playback, as the interface (as it was) only allowed the user to listen to the past session rather than skipping between songs.



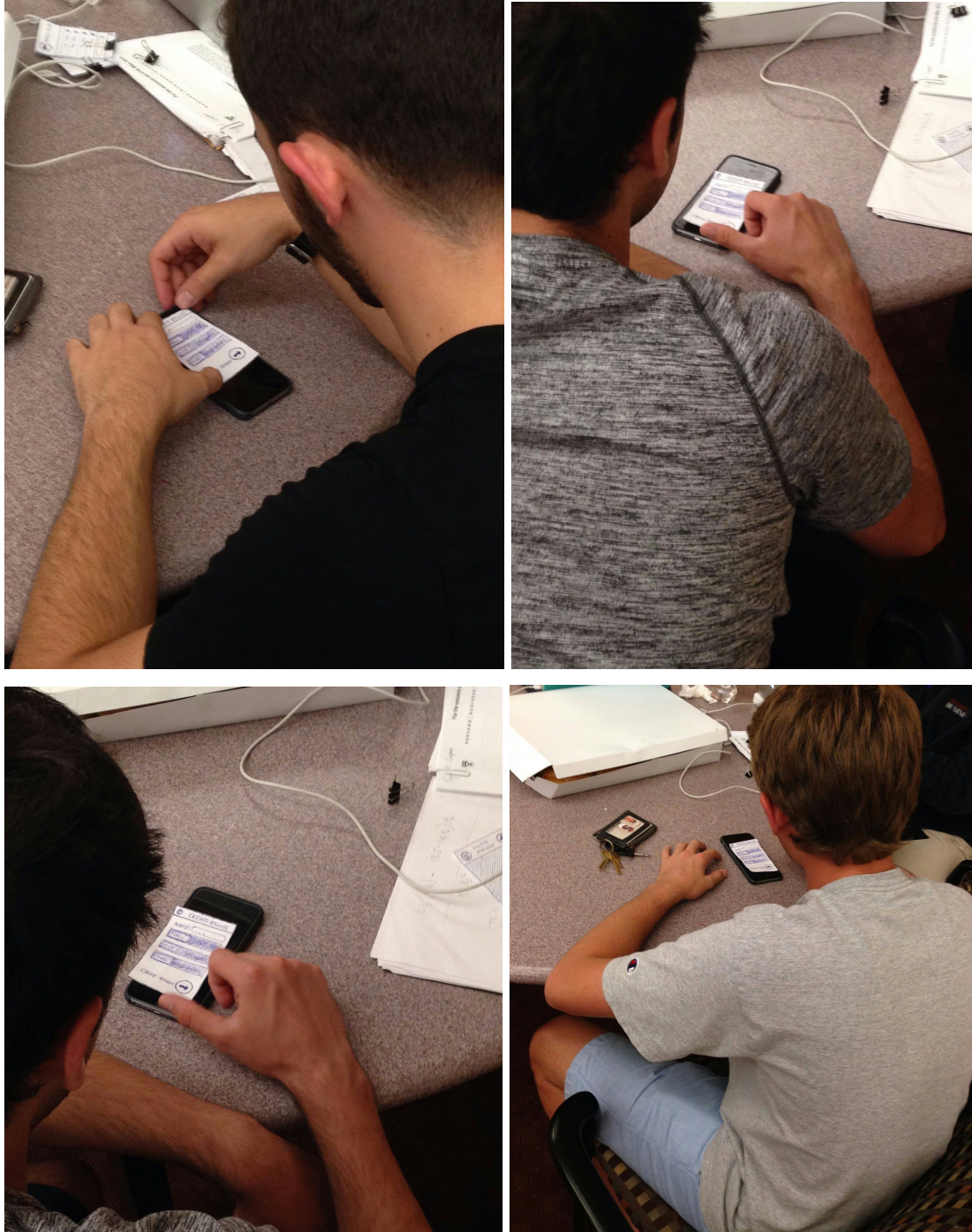


Figure 8: Participants testing the low-fi prototype in Roble Hall.

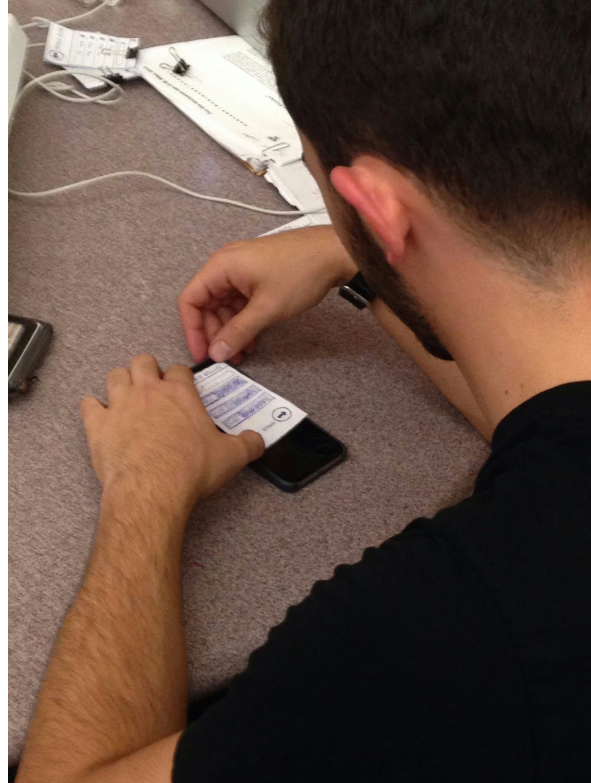


Figure 8.1: Additional pictures of participants testing the low-fi prototype in Roble Hall.

### Participant #3

#### *Task 1: Create a communal playlist.*

0: The user successfully created the Rubato session.

#### *Task 2: Update the communal playlist in real-time.*

0: Was able to create the playlist.

3: When trying to add another song to the playlist, he navigated back to the homepage of the app, ultimately doing a lot more work than was necessary.

#### *Task 3: "Relive" the playlist.*

0: The user successfully accessed a previous session and listened to a song from it.

3: Had trouble navigating between songs during playback, as the interface (as it was) only allowed the user to listen to the past session rather than skipping between songs.

### **Discussion**

Our testing revealed some expected and unexpected results. We were pleased that every participant was able to complete the simple and complex tasks with relative ease. However, within the medium task we noticed some confusion when adding a song to a playlist: participant 3 rejoined his own session when adding a new song. In addition, there was some confusion about settings and whether that led to playlist or session settings. We will look to clarify accessing the appropriate settings by making the labels for settings within specific sessions and within the application itself clearer and more distinct. To eliminate the confusion when adding a song, we noticed that certain buttons, such as Create Session and Add Song, cannot be ambiguous and must be made clearer. Furthermore, we noticed that the participants' freedom of interaction with past sessions was somewhat unclear, and needs to be redesigned moving forward.

Also, we were surprised by a particular behavior, "song trolling," of a participant when adding songs to the playlist. Seeing this behavior made us realize trolling could potentially disrupt the growth and the internal validity of the session, supporting a case for implementing up and down voting for songs. However, when choosing our top-2 design ideas, we chose not to implement this particular functionality in the prototype because we thought that users at a social event would want to focus on the event itself rather than spending a lot of time voting on songs currently in the playlist. After our testing, however, we decided that the pros of including real-time upvoting/downvoting greatly outweigh the cons, especially to combat "song trollers". We will definitely include upvoting and downvoting as features in our medium-fi prototype. Overall, our low-fi prototype testing revealed insightful and interesting behaviors that we will reflect on within the next stage of our design and prototyping iteration!

## Appendices

### *A.1: Raw Data:*

Person 1: Found that the UI was very easy to use. Navigation: everything was fine. Wasn't clear about all the buttons ("Hide" vs "show" switch). He wants a partially disabled vote mode where only a few people can add songs, but everyone can vote on songs. Was unclear about the "settings" button - did it mean session settings or playlist settings? Would be nice to have a way to deal with repeats - he added in songs just to troll ("Baby" by Justin Bieber) and added songs multiple times. Repeated a song he just played because he wanted to listen to it again.

Person 2: Found the UI very easy to use. He had no problem going through all 3 of the tasks. During the medium task, he trolled by playing "Party in the USA" by Miley Cyrus and upvoting it himself, admitting that he was spamming the playlist and found it to be funny.

Person 3: Also found the UI very intuitive to use. Found that the distinction between the plus sign and the progress bar on the voting playlist was very easy. He joined his own session again, so although he said the prototype was very easy to use, he went in a roundabout fashion through the app, basically starting from the beginning.

A.2: Consent Form:

## **Rubato Prototype Experiment Consent Form**

The Rubato application is being produced as part of the coursework for Computer Science course CS 147 at Stanford University. Participants in experimental evaluation of the application provide data that is used to evaluate and modify the interface of Rubato. Data will be collected by interview and observation.

Participation in this experiment is voluntary. Participants may withdraw themselves and their data at any time without fear of consequences. Concerns about the experiment may be discussed with the researchers (Parker Odrich, Peter Washington, Gio Jacuzzi) or with Professor James Landay, the instructor of CS 147:

James A. Landay  
CS Department  
Stanford University  
650-498-8215  
landay at [cs.stanford.edu](mailto:landay@cs.stanford.edu)

Participant anonymity will be provided by the separate storage of names from data. Data will only be identified by participant number. No identifying information about the participants will be available to anyone except the student researchers and their supervisors/teaching staff.

I hereby acknowledge that I have been given an opportunity to ask questions about the nature of the experiment and my participation in it. I give my consent to have data collected on my behavior and opinions in relation to the Rubato experiment. I also give permission for images/video of me using the application to be used in presentations or publications as long as I am not personally identifiable in the images/video. I understand I may withdraw my permission at any time

Name \_\_\_\_\_

Participant Number \_\_\_\_\_

Date 10/21/2015

Signature \_\_\_\_\_

Witness name \_\_\_\_\_

Witness signature \_\_\_\_\_

## A.2.1: Participant #1 Signed Consent Form

### **Rubato Prototype Experiment Consent Form**

The Rubato application is being produced as part of the coursework for Computer Science course CS 147 at Stanford University. Participants in experimental evaluation of the application provide data that is used to evaluate and modify the interface of Rubato. Data will be collected by interview and observation.

Participation in this experiment is voluntary. Participants may withdraw themselves and their data at any time without fear of consequences. Concerns about the experiment may be discussed with the researchers (Parker Odrich, Peter Washington, Gio Jacuzzi) or with Professor James Landay, the instructor of CS 147.

James A. Landay  
CS Department  
Stanford University  
650-498-8215  
landay at cs.stanford.edu

Participant anonymity will be provided by the separate storage of names from data. Data will only be identified by participant number. No identifying information about the participants will be available to anyone except the student researchers and their supervisors/teaching staff.

I hereby acknowledge that I have been given an opportunity to ask questions about the nature of the experiment and my participation in it. I give my consent to have data collected on my behavior and opinions in relation to the Rubato experiment. I also give permission for images/video of me using the application to be used in presentations or publications as long as I am not personally identifiable in the images/video. I understand I may withdraw my permission at any time

Name Juan Plansa

Participant Number 1

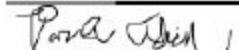
Date 10/21/2015



Signature \_\_\_\_\_

Witness name Parker Odrich

Witness signature \_\_\_\_\_



## A.2.2: Participant #2 Signed Consent Form

### **Rubato Prototype Experiment Consent Form**

The Rubato application is being produced as part of the coursework for Computer Science course CS 147 at Stanford University. Participants in experimental evaluation of the application provide data that is used to evaluate and modify the interface of Rubato. Data will be collected by interview and observation.

Participation in this experiment is voluntary. Participants may withdraw themselves and their data at any time without fear of consequences. Concerns about the experiment may be discussed with the researchers (Parker Odrich, Peter Washington, Gio Jacuzzi) or with Professor James Landay, the instructor of CS 147:

James A. Landay  
CS Department  
Stanford University  
650-498-8215  
landay at [cs.stanford.edu](mailto:cs.stanford.edu)

Participant anonymity will be provided by the separate storage of names from data. Data will only be identified by participant number. No identifying information about the participants will be available to anyone except the student researchers and their supervisors/teaching staff.

I hereby acknowledge that I have been given an opportunity to ask questions about the nature of the experiment and my participation in it. I give my consent to have data collected on my behavior and opinions in relation to the Rubato experiment. I also give permission for images/video of me using the application to be used in presentations or publications as long as I am not personally identifiable in the images/video. I understand I may withdraw my permission at any time

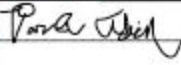
Name Nathaniel Okun

Participant Number 2

Date 10/21/2015

Signature  \_\_\_\_\_

Witness name Parker Odrich

Witness signature  \_\_\_\_\_

### A.2.3: Participant #3 Signed Consent Form

#### **Rubato Prototype Experiment Consent Form**

The Rubato application is being produced as part of the coursework for Computer Science course CS 147 at Stanford University. Participants in experimental evaluation of the application provide data that is used to evaluate and modify the interface of Rubato. Data will be collected by interview and observation.

Participation in this experiment is voluntary. Participants may withdraw themselves and their data at any time without fear of consequences. Concerns about the experiment may be discussed with the researchers (Parker Odrich, Peter Washington, Gio Jacuzzi) or with Professor James Landay, the instructor of CS 147:

James A. Landay  
CS Department  
Stanford University  
650-498-8215  
landay at cs.stanford.edu

Participant anonymity will be provided by the separate storage of names from data. Data will only be identified by participant number. No identifying information about the participants will be available to anyone except the student researchers and their supervisors/teaching staff.

I hereby acknowledge that I have been given an opportunity to ask questions about the nature of the experiment and my participation in it. I give my consent to have data collected on my behavior and opinions in relation to the Rubato experiment. I also give permission for images/video of me using the application to be used in presentations or publications as long as I am not personally identifiable in the images/video. I understand I may withdraw my permission at any time

Name Tommy Kimmelman

Participant Number 3

Date 10/21/2015

Signature 

Witness name Parker Odrich

Witness signature 

---

Word Count: **1927 words**