Rubato

Gio Jacuzzi, Parker Odrich, Peter Washington Assignment 6: Medium-Fi Prototype Studio: Creation

Value Proposition and Solutions

Value Proposition: Communal playlists in real time.

<u>Problem</u>: 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.

<u>Solution:</u> 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.

Representative Tasks

Task 1: Create a communal playlist (easy).

Task 2: Update the playlist in real-time (medium).

Task 3: Play songs from any event attended in the past.

Problems Found in User Testing

- Users played songs with "malicious" intents.
- Users found the interface to be too confusing due to the overcomplexity of the design.
- The meanings of buttons were ambiguous (specifically for settings and adding songs).



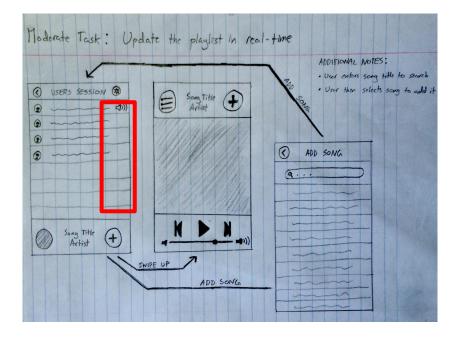
Design Change 1: Add Upvoting To Songs

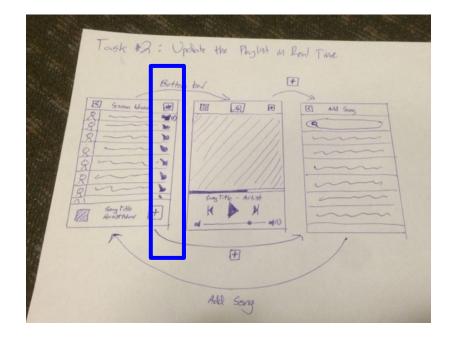
- Upvoting prevents a single user from hijacking the playlist.

- Adds more user interactivity to the party.

- Doesn't require excessive interaction during the party.

Design Change 1: Add Upvoting To Songs





Before

After

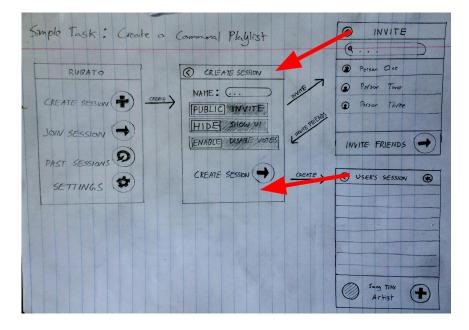
Design Change 2: Centralized Home Screen

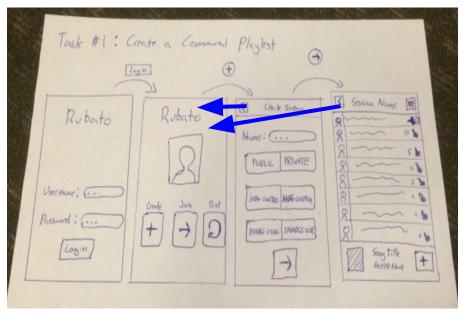
- A central screen makes it easier for user to navigate throughout the app.

- Prevents the user from being "stuck" in a certain screen.

- Prevents the user from clicking an unnecessary number of buttons and navigating through many screens.

Design Change 2: Centralized Home Screen





Before



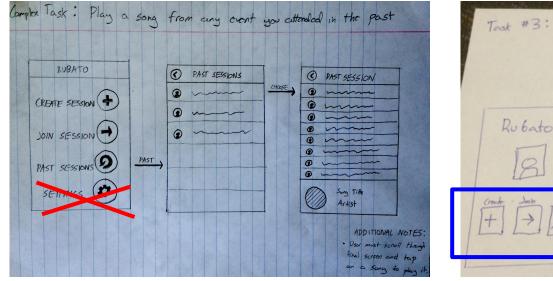
Design Change 3: Remove Global Settings

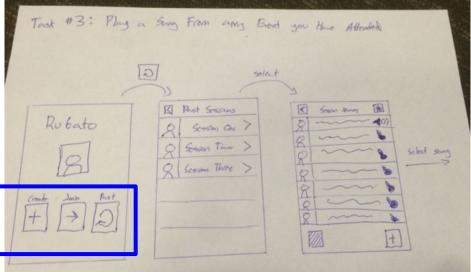
- Global app settings were largely irrelevant to the app.

- Settings were per-session and could just be done during session create-time.

- Settings button clogged the home screen. Now, have a **much prettier home screen**.

Design Change 3: Remove Global Settings

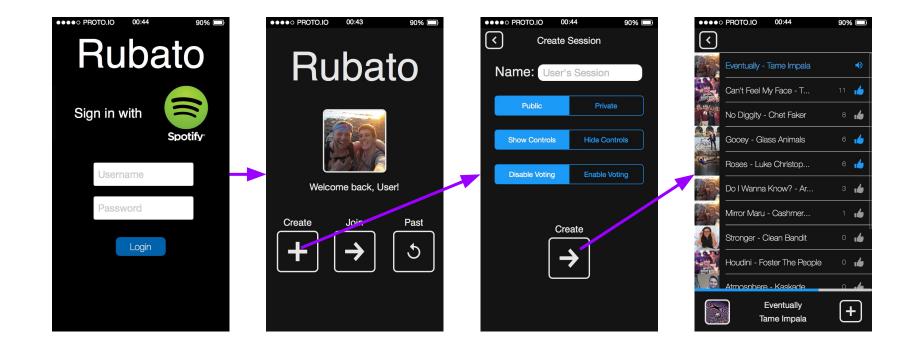








Task Flow #1: Create a Rubato Session



Task Flow #2: Update playlist in real time

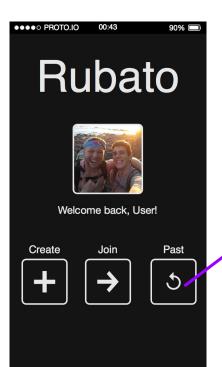
••••	PROTO.IO 00:44	90% 🗖
\checkmark	Join Session	
	Peter's Session	÷
B. Bisa	Parker's Party	
	Raquel's Roadtrip	÷

••••0	90%		
<	Peter's Session		
90	Eventually - Tame Impala		
	Can't Feel My Face - T	11	•
S. Bisian	No Diggity - Chet Faker	8	•
<u>3</u>	Gooey - Glass Animals	6	•
	Roses - Luke Christop		4
	Do I Wanna Know? - Ar		.6
	Mirror Maru - Cashmer		•
	Stronger - Clean Bandit		.4
	Houdini - Foster The People		•
	Atmosphere - Kaskade	0	
	Eventually Tame Impala	Η	

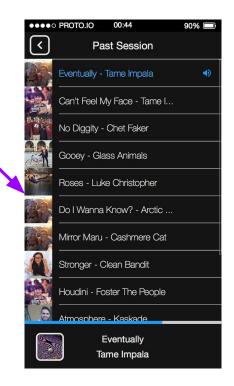
):44 90% 🔲	
E Add	Song	<
QS	Search	
Artist Result	>	
Album Result	>	S. Bisson
Playlist Result	>	
Song Result		

	PROTO.IO 00:44	90%	
〈	Peter's Session		
2	Eventually - Tame Impala		
	Can't Feel My Face - T	11	•
	No Diggity - Chet Faker		.4
X	Gooey - Glass Animals	6	
	Roses - Luke Christop	6	4
30	Do I Wanna Know? - Ar	3	.4
2	Mirror Maru - Cashmer	1	.4
	Stronger - Clean Bandit	0	.4
	Houdini - Foster The People	0	•
0	Atmosphere - Kaskade	0	.4
	Eventually Tame Impala	(Ð

Task Flow #3: Relive the playlist



••••	PROTO.IO	00:44	90%		
\langle	Pas	t Sessions			
	Peter's Se	ession		→	
S. Sing	Parker's F	Party		→	
	Raquel's	Roadtrip		>	



Prototyping Tool



- Used Proto.IO
- Makes iOS prototypes that seem real
- Lots of features
- Free trial

Limitations of Proto.IO

- No "conditional" logic prevented many simple prototyping features from being added
 - Selected songs couldn't actually be added to playlist
 - The results of upvoting a song can't be seen

- Long learning curve compared to Marvel and InVision

Wizard of Oz Techniques

 When a user wants to add a song, we will tell them that the song is added to the playlist.

 When a user wants to play a song, we will play the song ourselves. The prototype cannot play songs.



Hand-Coded Features

- Song names
- Session names
- User profile
- Number of likes per song



RUBATO MUSIC PLAYLIST CREATION IN REAL-TIME.



http://web.stanford.edu/class/cs147/projects/creation/Rubato/index.html