Mole-bile Voice Project Report

Wenqin Chen | Pi-Tan Hu

1. Executive summary

Mole-bile Voice started out as an explorative project for identifying and solving a blind users' need. Through user research, we discovered that many blind smartphone users experienced difficulties navigating their phones and a voice user interface (VUI) could make navigation easier. Based on this research, we created two VUI mobile applications that explored blind user friendly VUI designs and built a web-based development tool that help people create new VUI mobile applications with ease.

We hope the Mole-bile Voice project will stimulate more interest around designing for user groups with accessibility needs. We plan to continue the project beyond the end of the semester and make our demo apps and development tool open-source resources available to anyone interested in making VUI mobile apps. We hope there will be more user friendly VUI mobile apps available to blind users in the near future.

2. Initial Research: into the world of a blind user

In INFO 213 –User Interface Design and Development, Wenqin worked on an interesting group project that developed a Google Glass application to help the visually impaired navigate. Our team wanted to continue exploring designs for the blind user community.

We started the exploration by brainstorming different areas that technology might help make lives more enjoyable for blind users. At the end, we narrowed our focus to three areas – interactions with machines, social activities, and physical activities.

With these focus areas in mind, we conducted generative user research with six blind users with various degrees of familiarity with technology, various occupations, various levels of education and various period lengths of being blind.

User Demographics, Generative Research 1

User #	Age	Gender	Occupation	Tech Familiarity	Cellphone Type	Blind since
1	60s	Female	Professor	Medium	iPhone	High School
2	50s	Female	Tech Worker	High	Android	-
3	40s	Female	Teacher	High	iPhone	-
4	60s	Female	-	Low	Feature Phone	20s
5	50s	Female	-	Low	-	40s
6	60s	Female	Teacher	Medium	-	Birth

We asked about their experiences using computers, smartphones, wearables, and appliances, about their experiences sharing and communicating with friends, and their experiences doing physical exercises. (See interview guide in Appendix A)

We also conducted extensive literature review to understand the existing accessibility technology landscape.

Informed by our research, we went back to the whiteboard and brainstormed potential product ideas. We envisioned many products that would help address observed user needs, but due to implementation constraints, we had to let many of them go. Here are some notable early ideas:

- Dance instructor: A soft wearable suit that would gently guide users through learning nuanced body movements. Help solve the challenge blind dancers and yoga practitioners face when learning new physical movements.
- Text editor for the blind: A physical machine that generates tactile text blocks as users enter text. Then users can manipulate the text blocks in different ways to edit and format text. Help solve the challenge blind writers face when editing and formatting text on a computer.
- Smart braille board: A refreshable digital braille board capable of rendering text, graphs, and images using movable pins. Help solve the challenge blind students face when learning the visual components of math and science. (Finalist of UC Berkeley Big Ideas Competition. See proposal here.)

Unfortunately these ideas are not feasible technology-wise given the time and resources available to our project. We hope to revisit these ideas in the future.

3. Finding The Need: easier smartphone navigation

We eventually decided to address the challenge of smartphone navigation for the visually impaired. Below describes the research during which we discovered this need. For this research, we collaborated with a team from INFO 214 – Needs and Usability Assessment (Safei Gu, Jong-kai Yang, and Eric Pai), who did user research for Mole-bile Voice as part of their class project.

Research Methods, Recruitment, and User Demographics

We conducted user interviews and contextual inquires. We asked interviewees about how they accomplish daily tasks such as transportation, communication, file management, getting information etc. If an interviewee used smartphone, we asked them the types of mobile apps they used and how they navigated their smartphone. If an interviewee did not use a smartphone, we asked about reasons that had kept from using it. Lastly, we observed smartphone users using an application on their phones. (See interview guidelines in Appendix B)

We recruited from the UC Berkeley campus and East Bay Center for the Blind. In total, we interviewed 11 participants. Among these interviewees, 1 used a feature phone, and 9 used smartphones (1 Android and 9 iPhones).

User Demographics, Generative User Research 2

User #	Age	Gender	Occupation	VoiceOver Proficiency	Cellphone Type	Blind since
1	60s	Female	Professor	Medium	iPhone	High School
2	50s	Female	Tech Worker	High	Android	-
3	40s	Female	Tech instructor	High	iPhone	-
4	60s	Female	-	Non-user	Feature Phone	20s
5	42	Female	-	Low-medium	iPhone	-
6	32	Female	Usability Tester	High	iPhone	-
7	40s	female	-	High	iPhone	Birth
8	40s	female	-	Medium low	iPhone	Birth
9	60s -70s	female	Tech instructor	high	iPhone	-
10	60s -70s	female	-	low	iPhone	-
11	60s -70s	Male	-	high	iPhone	-

Observations

Smartphone Users and Learners

The current technology blind users use when navigating their smartphones is a screen reader that reads the descriptions of selected UI element (e.g. buttons and text fields). Users can select UI elements by either tapping on a particular element or by swiping left and right on the screen to go through elements in special order. The iPhone version of this screen reader is called

VoiceOver, and the Android version is called TalkBack. Both screen readers are very similar in functionality, so for the purpose of our research, we refer to both screen readers as VoiceOver.

From our interviews, we learned that blind smartphone users use the same rich variety of apps as sighted people. They used social apps such as Facebook, leisure apps such as YouTube and Pandora, scheduling apps such as Reminders, transportation apps such as Uber and Google Maps, shopping apps such as Amazon and Orbitz, and productivity apps such as Email. They also used image recognition apps.

We asked interviewees why they used smartphones as opposed to calling a human operated or automated phone line. We learned that smartphone apps get things done faster since there is no wait time, and cheaper since there is no service charges. In addition, there are many things that are simply not possible to do via phone services. For example, GPS service, user account login, payment and address preset, coordination with other apps, calendar, music player, sensors and camera.

All the smartphone users we interviewed used VoiceOver to navigate their phones. We encountered three types of blind smartphone users in our research process and they had different experiences with VoiceOver:

- Advanced users (6 users) who are very familiar with the VoiceOver technology. They know its shortcut gestures, remember button locations and think voiceover is quick and reliable. “I think voiceover is very useful. It helps me a lot,” says one advanced user.
- A regular user (1 user) who knows the basics about VoiceOver. She knows the basic gestures such as swiping left and right to scroll through buttons and text fields, and double tapping to select. She remembers buttons locations for apps that she uses every day, but for less frequently used apps and new apps, it takes her a long time to find the buttons she needs, and he sometimes gets stuck on a screen and cannot find ways to either go back or go forward. But given enough time, she can usually figure out how to complete a task using VoiceOver.
- Novice users (3 users) are in the process of learning VoiceOver. They need to learn the concept of voiceover, the UI layout, and the gestures. This learning often takes a long time, especially among older users. “I’m still learning. I’ll get there,” said one novice user.

<u>Non-smartphone Users</u>

We also talked to 1 non-smartphone user. She owned an iphone at home but chose not to use it. She instead preferred using a Samsung feature phone that came with a keypad. She used a combination of keypad touching and voice commands (the phone understood voice commands) to make phone calls. “I don’t use the iPhone because it’s all flat! If it had buttons I can feel then I would use it,” said the user.

Findings

Based on the above observations, we think that VoiceOver navigation poses usability challenges to three of the four user groups that we identified:

- Regular VoiceOver users
- Novice VoiceOver users
- Non-smartphone users

For these groups, VoiceOver poses four usability challenges:

- Users need to understand the conceptual model. Since VoiceOver corresponds to locations on a graphical user interface, users need to understand the visual concepts of buttons and text fields and how their gestures correspond to these visual elements.
- Users need to memorize a number of different gestures in order to navigate efficiently.
- It takes substantial time to learn how to use a new application. We observed that users had to spend much time searching up and down the screen to discover new button locations.
- Users need to memorize the locations of screen buttons in order to use an application quickly.

In short, for non-advanced smartphone users, VoiceOver is challenging to learn, and once a user has learned it, it is hard to operate it fast. We felt there was a need for an alternative navigation method that blind smartphone users can try when VoiceOver does not suit their needs.

4. Solution Part 1: usable VUI mobile apps

Now having discovered the need, we went back to the drawing board to brainstorm an alternative navigation method. We found voice user interface (VUI) as a potential alternative; it might be easier to learn and faster to operate for the three groups of blind users we identified during need finding. To further investigate this hypothesis, we went on to develop user personas, analyze existing voice technologies, prototype VUI mobile apps, and conduct usability testing.

Personas

Since our mobile VUI targets three user groups, we created three personas.

Devin (regular VoiceOver user)
Devin is a financial analyst in his 50s. He lives in San Dieago with his wife and has two children who now work in other states. He lost his vision due to an illness 15 years ago. Devin bought his first smartphone, an iPhone 5, in 2014. Devin uses his iPhone on a daily basis. He uses the phone to make calls, send text messages, check emails, get directions, request ubers and read news articles. Occasionally, he hears about a new useful app and downloads it from the App Store. Sometimes it takes Devin a long time to complete a task on his phone using VoiceOver. He wonders if there is a faster way.

Jasmine (novice VoiceOver user)
Jasmine is a customer service representative in her 50s. She lives in San Francisco with her dog Chipper. Jasmine was born blind. Jasmine used a Nokia flip phone for many years, but recently purchased an iPhone based on friends' recommendation. She hopes the iPhone can help her be more independent and connected with family and friends. Jasmine is taking a class on how to use VoiceOver at the San Francisco Lighthouse. It has been challenging to learn and remember gestures and buttons, but she thinks the eventual benefit is worth the work.

Teresa (Non-VoiceOver user)
Teresa is a retired teacher in her late 60s. She lives in Michigan with her husband. Teresa lost her vision two years ago due to illness. Teresa's daughter gave her an Android phone last Christmas hoping it could help stay more connected. Teresa felt the phone would be hard to learn and still hasn't tried it.

Competitive Analysis

We evaluated some current voice UI products and concluded that neither of them could serve as an alternative to VoiceOver.

Siri & Google Now

- Does not provide in-app navigation for most applications. Mostly performs system level navigation such as launching an app.
- Longer System response is shown as visual only.

Amazon Echo

- Works on a stationary device. Cannot be used mobile.
- Does not track state. Therefore cannot perform any tasks that require state.

Dragon Dictation

- Specific to text editing

Since there were no VUI smartphone apps out there, to find out whether blind users would want to use VUI mobile apps, we decided to design two example VUI applications by ourselves and collect user feedback.

Prototyping

We selected two mobile apps from the use cases we gathered during user research – Uber for requesting rides, and Orbitz for booking plane tickets. We chose these two apps partially because they were apps that blind users actual used, and partially because they presented some challenging interactions: requesting a Uber involves GPS locations, address presets, and multiple rounds of input and output; booking a plan ticket requires a large amount of information entry and output. We wanted to design VUI apps that could fulfill these interaction needs.

We referenced design patterns for Interactive Voice Response, a technology that allows a computer to interact with humans through the use of voice and DTMF tones input via keypad.[1]

[1] https://en.wikipedia.org/wiki/Interactive_voice_response

Uber Voice

Uber Voice is a VUI mobile app that helps users request Uber rides just like its GUI counterpart. You can view our final design here.

We iterated over five prototypes. We worked closely with two blind users, initially using the wizard-of-oz method, and later using a functional prototype. (See testing script in Appendix C; see functional prototype in Appendix G)

The users gave us valuable feedback for improvement:

- Added contextual help
  Users wanted to know what options they could say at any given step in the interaction. We added a feature where when a user says "help" while using the app, the app would give them information on what they can say at that specific step.
- Added verifications for address input
  A user wanted to make sure the address she entered was for the correct city. We added an interaction for checking the city.
- Added presets for home and work addresses
  Users wanted to be able to say "home" and the phone would populate a preset home address. We added this interaction.
- Enabled multiple user says to trigger a system response
  Users would say different commands in an effort to trigger the same system response. We added all these user commands as valid triggers.

The INFO 214 Team performed a heuristics evaluation for the Uber Voice app (see details in Appendix D). Based on their findings, we made the following changes to the prototype:

Heuristics Evaluation Issues & Fixes

Heuristic Violation	**Fix**
Violations of "3. User Control and Freedom" Cannot exit and cannot go back.	Allow users to start over
Violations of "7. Flexibility and Ease of Use" The app doesn't implement the simplified sentence-based commands, such as "Set Home Address" or "Pick me up from here".	Updated design to accept multiple user commands
Violations of "10. Help and Documentation" No way to hear/repeat/discover available options.	Partially fixed, added contextual help
Violations of "10. Help and Documentation" Though it provides context-related help command, but it doesn't provide an initial tutorial or onboarding steps to inform the user that those shortcuts like "help" command exist.	Added tutorial

Orbitz Voice

Orbitz Voice is a VUI mobile app that helps users book plane tickets just like its GUI counterpart. You can view our final design here.

We iterated over two prototypes for Orbitz Voice. Our primary design goal for this app was to handle long user input and long system output. The INFO 214 team conducted a cognitive walkthrough on the Orbitz voice app, during which the team examined the cognitive process of all the steps involved in a blind user persona booking a round-trip flight from Oakland to Boston. Based on the evaluation results, we improved Orbitz Voice's following aspects:

- Refined wording
  Terms such as "two-way" and "departing" were potentially confusing to the user. Changed them to "round-trip" and "outbound".
- Switched question order
  Change the question prompts from Departing location > Returning location > Departing date > Returning date to Departing location > Departing date > Returning location > Returning date
- Added tutorial for how to navigate through the flight results.

Evaluation

Now that we have the Uber and Orbitz voice app designs finalized, we asked the Info 214 team to help us evaluate the apps' usability with blind users. The team conducted 5 usability testing sessions at the East Bay Center for the Blind. Due to recruiting limitations, while 3 of the 5 testing users fitted into our target users, and the other two were proxy users (advanced VoiceOver users). (See user details in Appendix E)

During each testing, the user was asked to perform 4 tasks using the Uber Voice app.
The subject was asked to perform 4 tasks:

1. Specify the pickup location with Uber Voice
2. Select a ride with Voice User Interface
3. Select a ride with Gesture Control
4. Check ride updates and ride details

(Tasks 2 & 3 used similar interactions as those used in the Orbitz Voice app for checking long output.)

Because the final Uber voice app design included interactions our prototyping technology could not realize at the moment, the evaluators chose to use wizard-of-oz testing. The evaluators would "become the app", using the mock-synthesized voice and only respond to predefined prompts (See testing script in Appendix F).

Testing Results

Task	Success Rate
Task 1	5/5
Task 2	4/5
Task 3	4/5
Task 4	4/4

General Feedback

- User 1, a regular VoiceOver user, said she “loved” the app.
- Advanced users liked VoiceOver more than Voice User Interface, while novice smartphone users preferred Voice UI.
- Advanced VoiceOver users mentioned some drawbacks of the Voice User Interface they noticed:
    - When the smartphone is speaking (outputting information to user), they generally don’t want to interfere the speech with their own input.
    - When the surrounding environment is noisy, the voice input would normally be less useful.
    - Dictations are sometimes error-prone, which can soon frustrate the user.
    - Dialogues-style Voice User Interface should implement the “repeat” function, for human would also mishear the synthesized voice, not only the machine.
    - One user suggested combining VUI with VoiceOver to give users options.

There are a few violations from the heuristics evaluation and one finding from the Cognitive Walkthrough that we have not be able to fix, we list them here in the evaluation as well:

Remaining Usability Evaluation Issues

Issue	**Source**
If the user wants to change a tiny error in her returning flight, right now she can only go all the way back to the departure flight and go through everything to edit one tiny error in returning flight.	Cognitive Walkthrough
Violations of “1. Visibility of System Status” In the long procedural interaction of getting a ride, there is no sound clue of where the user is.	Heuristic Evaluation
Violations of “3. User Control and Freedom” It doesn't allow user freedom to exit a loop, undo/redo, escape or go back.	Heuristic Evaluation
Violations of “10. Help and Documentation” No way to hear/repeat/discover available options.	Heuristic Evaluation

Conclusion

As a new technology, VUI mobile apps still have many usability issues that need to be addressed, but the high success rate from the usability testing and the positive feedback from our target non-advanced VoiceOver users and current non-users indicate that a more mature VUI design holds promise as a VoiceOver alternative.

5. Solution Part 2: usable mobile VUI development tool

Motivation

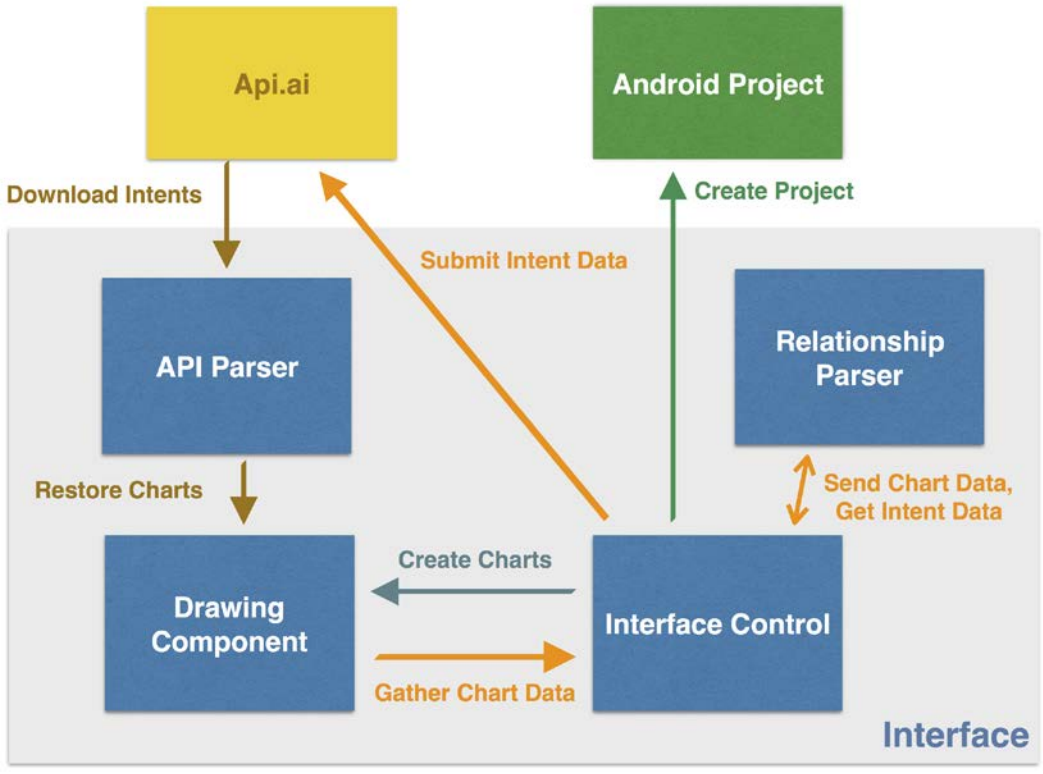
Since our research and design work shows VUI's potential as a VoiceOver alternative, we want to encourage more designers and developers to experiment with and create VUI mobile apps. Unfortunately, we were not able to find an easy to use VUI mobile app development tool. We believe that to encourage more designers and developers to make VUI mobile apps for the blind, they need to have easier tools for prototyping and development. The current shortage of usable development tools motivated us to build one ourselves.

Solution

We wanted to build a drag-and-drop graphic user interface where designers/developers can create versatile conversation flowcharts for their voice apps, and this flowchart would automatically be converted into code, and then easily deployed as a smartphone app.

Technical Overview

Our system is composed of three user flows which are creating charts, syncing with api.ai and producing a basic Android project. The followings are the detailed descriptions about what they can achieve.



Create Charts

The “Create Charts” flow is comprised of the arrows and the components withblue color on the chart. For this flow, when users click on node-creation buttons on the interface, a set of nodes which present conversational components, intents, user utterances or system responses will be created on the drawing canvas. Users can modify the information for each node, change the positions for those nodes on the canvas, and build the conversation concept by repeating this process.

Submit Intents to Api.ai

The representation color for this flow is orange. When users finish making modifications to the charts, they can save all the changes to Api.ai in real-time. Thehe “relationship parser,” described below, is responsible for identifying the intents and the relationships between them, and transforming the charting data format to the format that Api.ai accepts. In this way, all intents and the conversational connections will be preserved and users can access to Api.ai speech API built from our interface and use the speech API in other platform.

Download Intents from Api.ai

This flow is highlighted with brown color. If users close the browser and re-open our interface later, all the intents and conversation connections will be downloaded and restored to the previous state. We use another system component called “API parser” to create charting nodes and connections data from the intents data downloaded from Api.ai. As a result, users don’t need to worry about losing all the accomplished results or creating the same chart all over again.

Create Android Project

The last flow is colored with green. In order to let users quickly build a voice-navigation-enabled Android app without too much effort, we created a feature that automatically generates a basic Android application contains a simple button to interact with Api.ai’s speech API. Users are able to focus on extending and customizing the project to achieve their goal without spending time figuring out how to integrate with Api.ai.

System Component Detail

Interface

Our interface is built using technologies including React.js[2] from Facebook, Redux[3] and webpack[4]. We used React.js to manage and categorize our view components, along with single directional data flow library named Redux to handle all the data status change. Finally, all the view components and data status modules are be compiled into a single JavaScript file using webpack.

Drawing Component

To support chart-drawing function, we incorporated a charting library called Cytoscape.js[5], a graph theory library which allows users to create nodes and edges from plain JavaScript objects or JSON format files. It also supports basic event handling such as mouseover, touch-start and touch-end so users can create customized functions to support various interactions. We only used the basic functions such as drawing nodes and edges of this library.

API Parser

The data format downloaded from Api.ai, which is structured using the intent concept defined by Api.ai[6], and the format we use to draw nodes and edges in Cytoscape canvas are different. Therefore, we created an API parser to transform the data to plain Cytoscape format described in their website[7] and output the data to Cytoscape canvas.

Relationship Parser

The same thing happens when we need to extract data from Cytoscape canvas and submit them to Api.ai. The difference is that we need to identify the intent and maintain the connections between intents while transforming the data. Therefore, the purposes of this parser are to compose intents from nodes and edges, decide which edges are connection within nodes of intents or between intents and then produce the output data in the right format.

System Deployment

[2] React.js official website https://facebook.github.io/react/

[3] Redux official website https://github.com/reactjs/redux

[4] webpack official website https://webpack.github.io/

[5] Cytoscape.js official website http://js.cytoscape.org/

[6] Intent definition from Api.ai https://docs.api.ai/docs/intents#intent-object

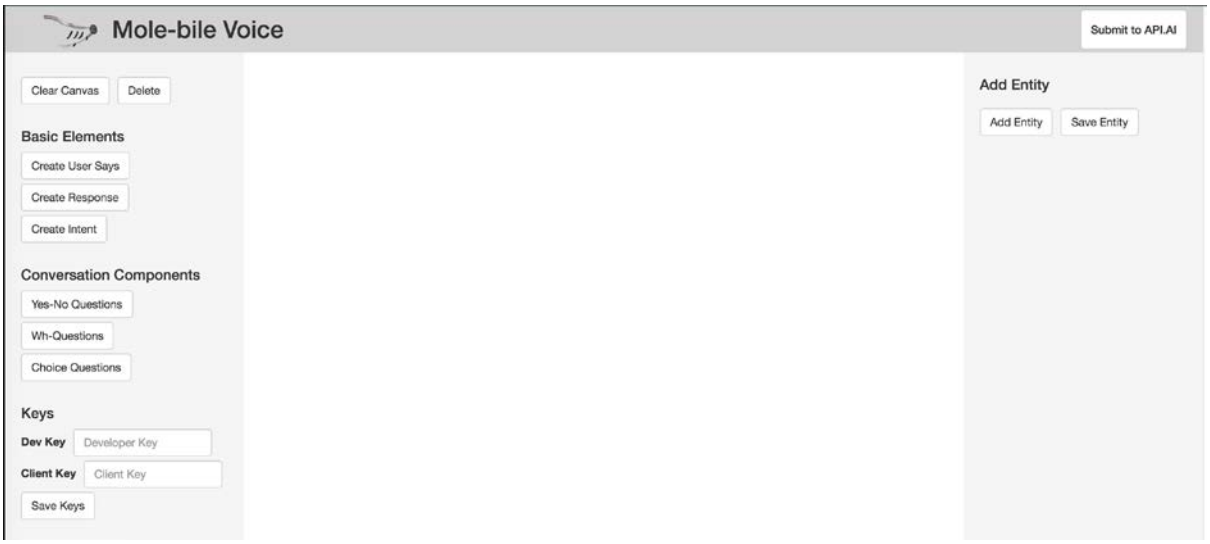
[7] Graph manipulation document of Cytoscape.js http://js.cytoscape.org/#core/graph-manipulation

We hosted our website on the Heroku[8] platform. We configured the server and enabled Heroku's automatic deployment feature so that our server is always running the latest version from our GitHub repository.

Code Repository and Website

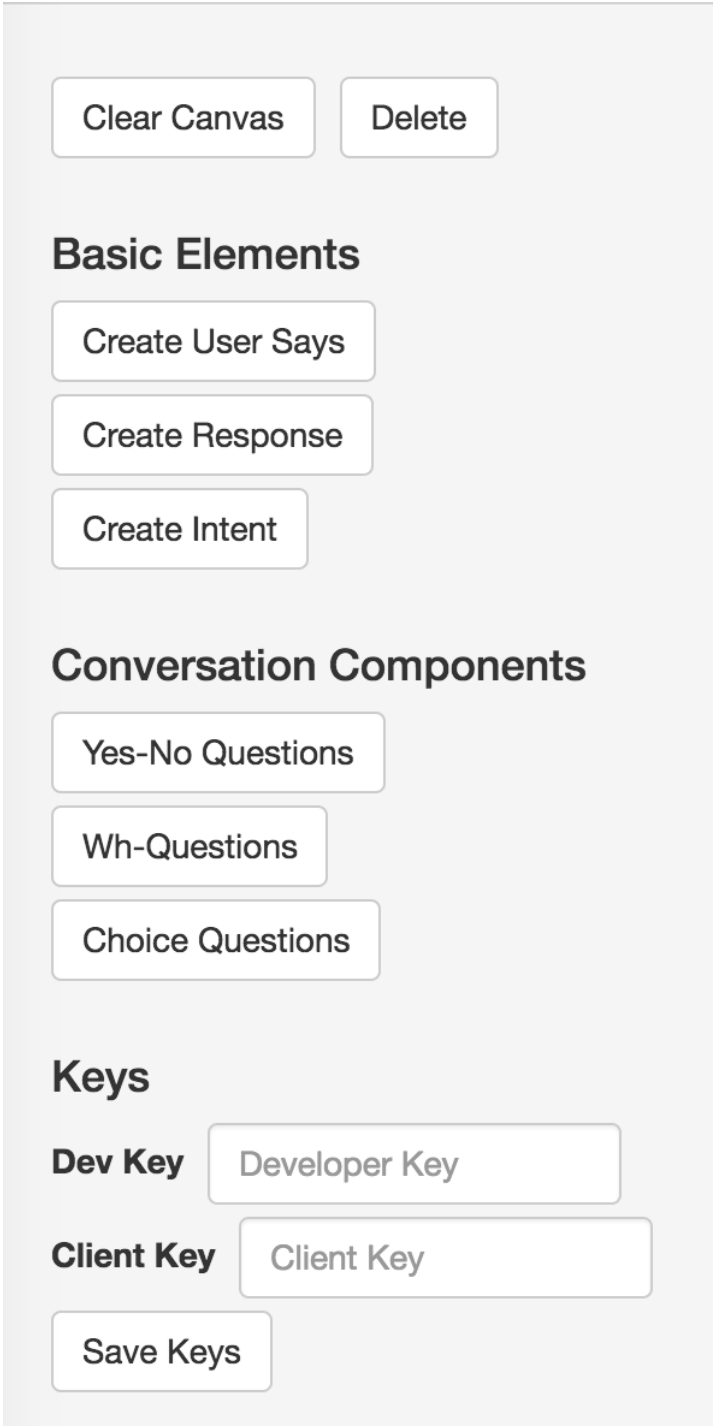
We used Github to cooperate between teammates. The link to our Github repository is https://github.com/LaContra/mole-bile-voice. The prerequisite frameworks and libraries that need to be installed are npm and webpack. After the repository is cloned from Github, run "npm install" in the project folder to install all dependencies. The final step is to start a local server with the root folder specified as the project folder and then open the browser to start playing with the interface. If there is an issue to run the project locally, we also have a website hosted online and the link is http://mole-bile-voice.herokuapp.com/.

Screenshots

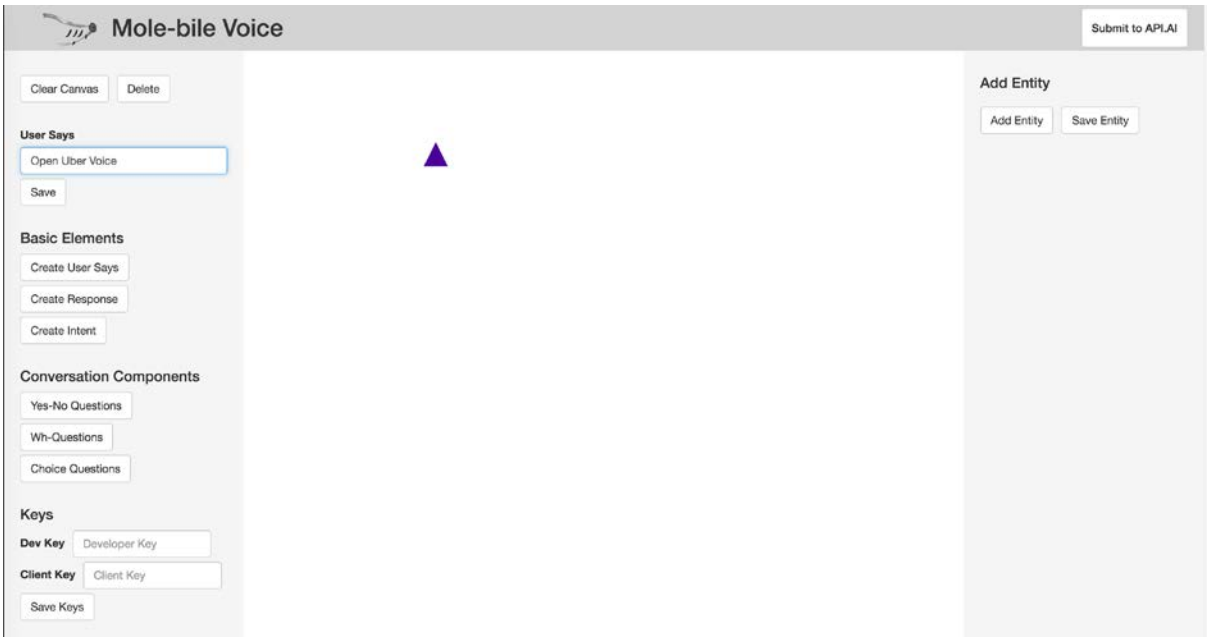


Website overview. The panel on the left contains all the chart-creating interaction buttons and the functions to set Api.ai development and client keys. The entity function panel on the right allows users to design their own phrase collections that can be recognized by Api.ai speech recognition.

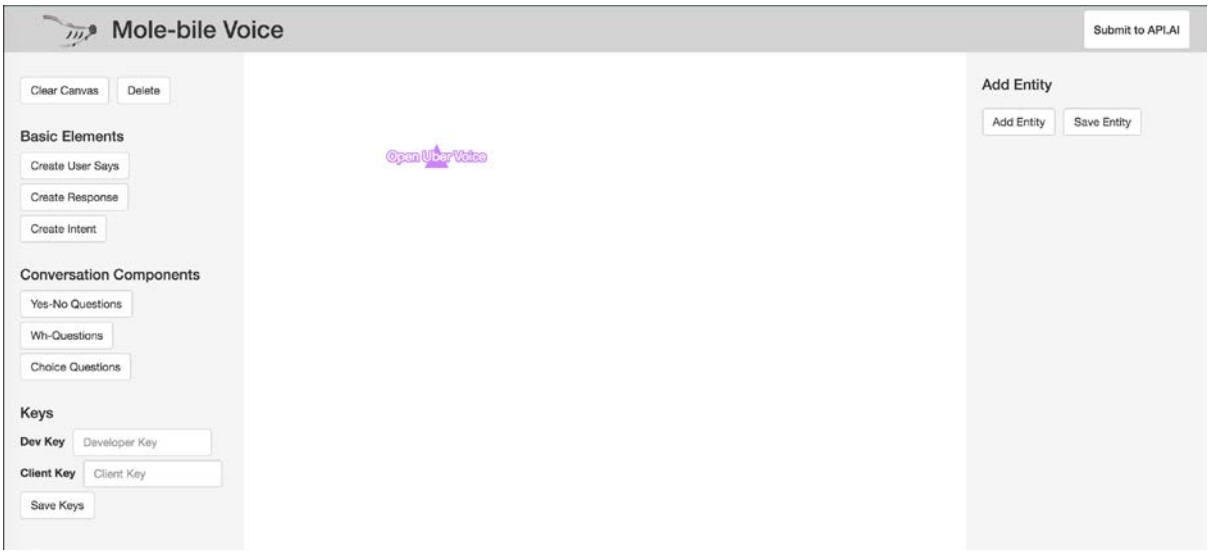
[8] Heroku official website https://www.heroku.com/



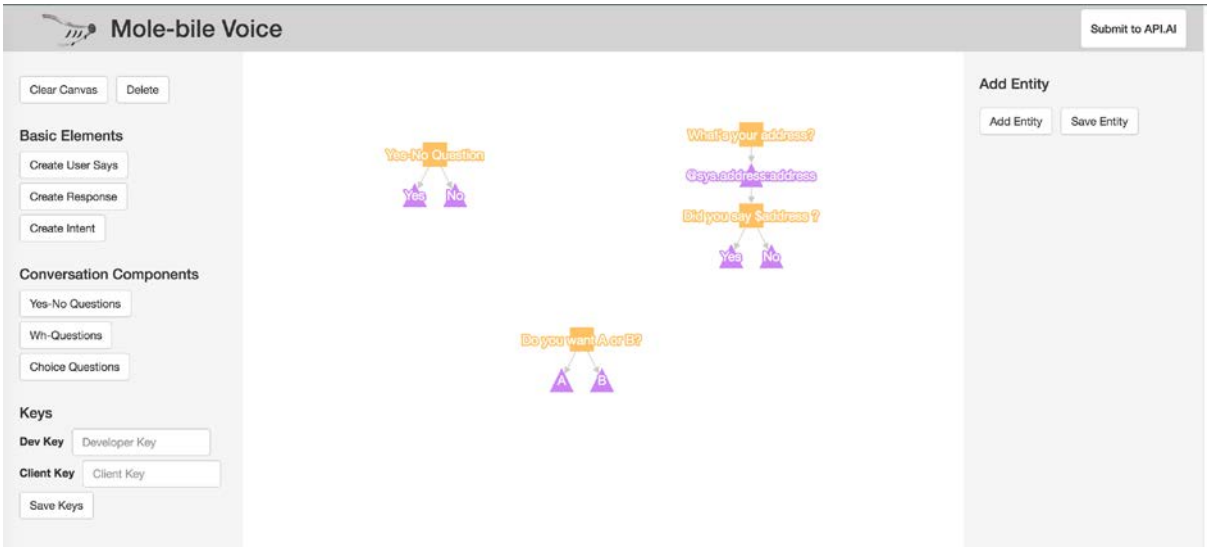
Clear version of the panel on the left.



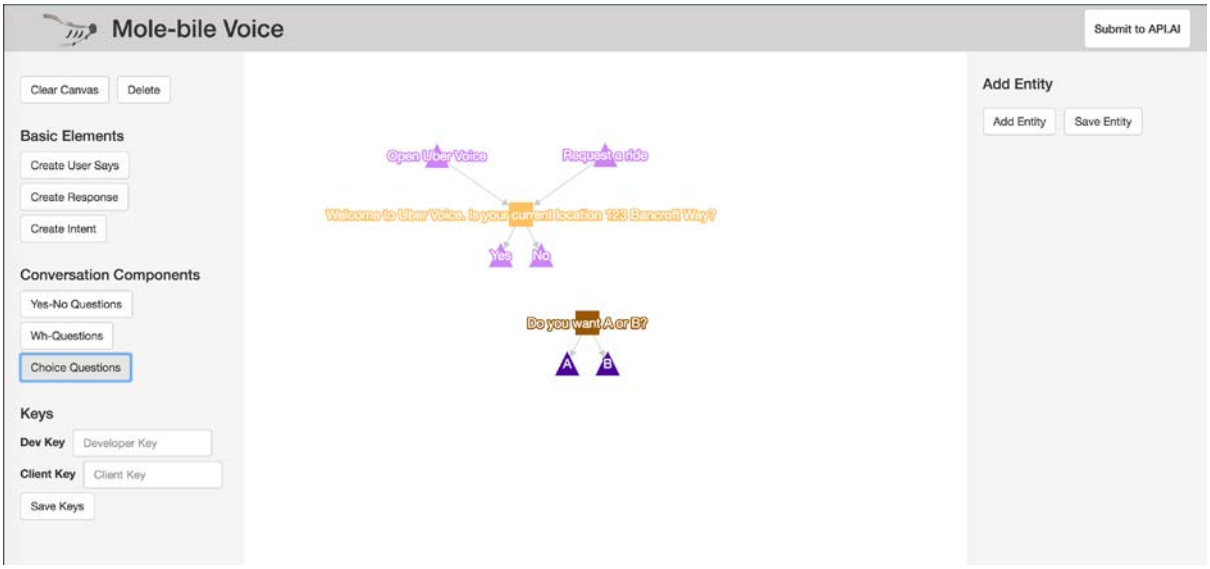
When clicking on the node, an input section will show up and users can input speech sentences for “user says” or “response” nodes.



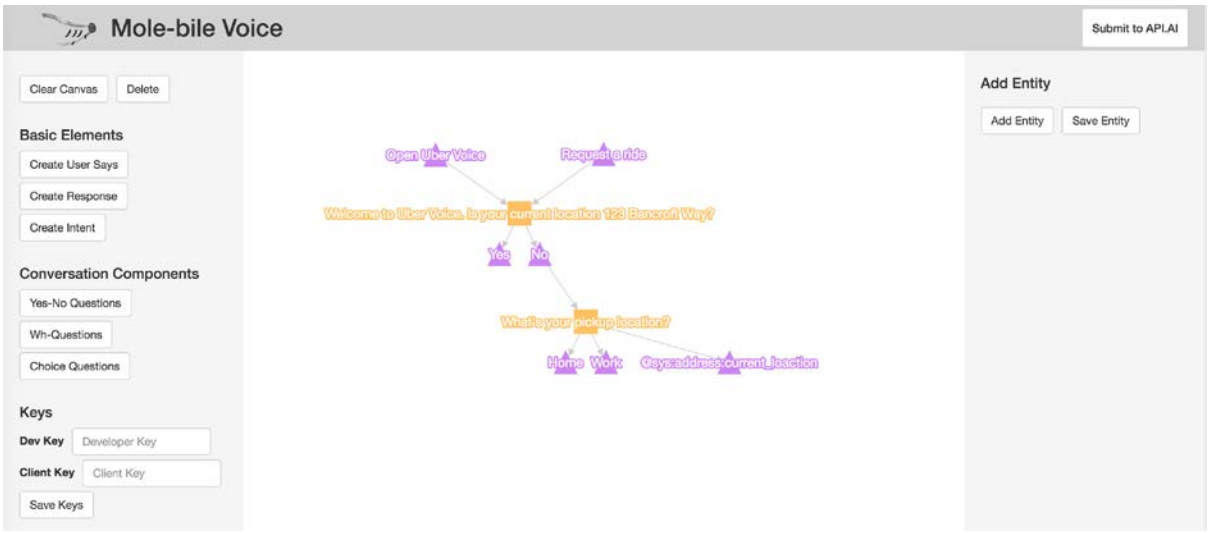
When users finish entering the speech for those node, clicking on the Save button or pressing the Enter key will save the sentences to the node.



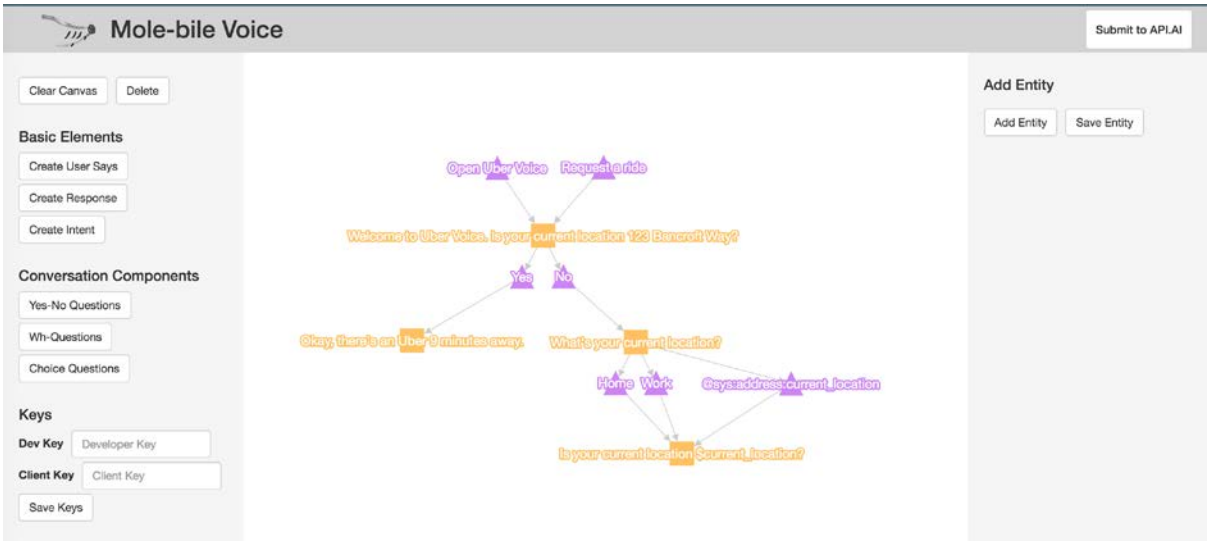
We provide three type of conversation components which are Yes-No questions (on the left), WH questions (on the right) and multi-choices questions (in the middle).



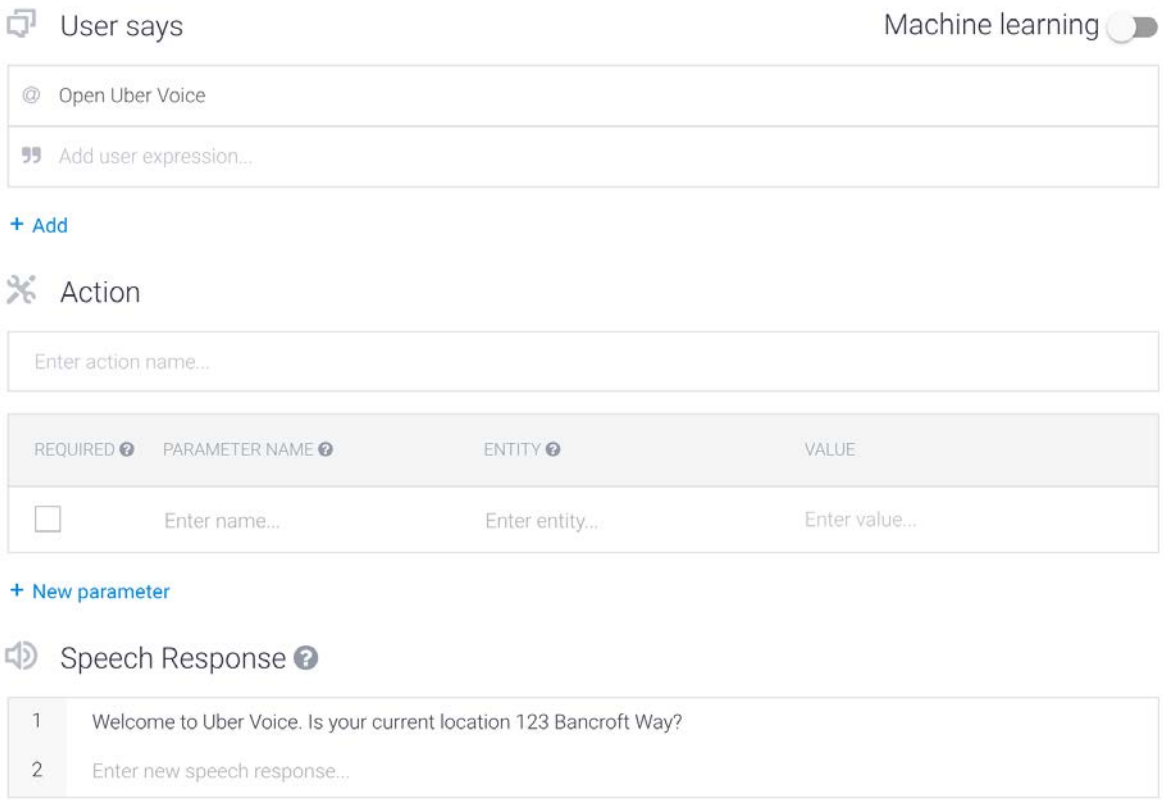
We were building our demo app, Uber Voice in this screenshot. We use multi-choices question component to build the conversation that will ask users their current locations.



After entering user says sentence and responses and saving them to the nodes, we created the first two intents with used in our demo App, Uber Voice. Note: those two intents are not complete since every intent has at least one “user says” and “response.”



We can repeat previous steps to create more intents. In this image, we created several complete intents that can be submitted to Api.ai without any error.



We can submit our design to Api.ai by pressing the submit button on the top-right corner of the website. The intent of this image shows that we successfully synced the first intent we created in our design to Api.ai.

6. Future Work

We plan to continue this project beyond the end of the school year. On the design side, we will

- Explore more voice interaction designs to address the remaining usability issues that surfaced during the design process.
- Design better error handling and tutorials.
- Conduct usability evaluations using functional prototypes.
- Explore potentials of combining VoiceOver and VUI, to allow flexible user choice, efficiency, and privacy.
- Explore other potential users of mobile VUI. E.g. Users with carpal tunnel or dexterity challenges.

On the SDK side, we plan to

1. Improve the UX design.
2. Implement more advanced features.
3. Implement better error handling and tutorials.

4. Develop context AI.
5. Make the SDK open-source.

7. Conclusion

Mole-bile Voice started out as an explorative project for identifying and solving a blind users' need. Through user research, we discovered that many blind smartphone users experienced difficulties navigating their phones and a voice user interface (VUI) could make navigation easier. Based on this research, we created two VUI mobile applications that explored blind user friendly VUI designs and built a web-based development tool that help people create new VUI mobile applications with ease.

We hope the Mole-bile Voice project will stimulate more interest around designing for user groups with accessibility needs. We plan to continue the project beyond the end of the semester and make our demo apps and development tool open-source resources available to anyone interested in making VUI mobile apps. We hope there will be more user friendly VUI mobile apps available to blind users in the near future.

8. Special Thanks

We want to thank our project advisor John Chuang for his guidance and support throughout the project, Safei Gu, Jong-Kai Yang, and Eric Pai from the INFO 214 team for their hard work on user research and testing, Georgina Kleege and Lucy Greco for their invaluable insights, East Bay Center for the Blind for their generous support, Jing-Wan Hsu for her great coding contribution, Steve Fadden for his guidance on user research, and Toshiro Nishimura for just being awesome in general. This project would not have been possible without all of you!

Appendix

Appendix A

Question guide:

a. Physical activities:
    i. Do you exercise?

       If the answer is no, ask them why and maybe jump to iv and v depending on the reasons; maybe they want to exercise but have some difficulties

    ii. What do you do for exercise?
    iii. How do you learn the movements?
    iv. Any difficulties with your current way of exercise?

v. Are there physical activities you would like to do? (whether it's a sport or entertainment, by themselves or with others)

b. Screen device interaction
    i. Do you use screen devices? What types do you usually use?
    ii. Can you show us how you usually interact with a screen device?

c. Social media
    i. Do you share your experience with family and friends?
    ii. What platform do you use?
    iii. What types of content do you like sharing with them?
    iv. What types of content do you like them to share with you?
    v. Anything else you would like to share or see other people sharing?

2. Entertainment:
    a. what activities do you do for fun? (whether it's by themselves or with others

Backup question:

3. Reading
    a. Do you read magazines or newspapers?
    b. What are your main sources of information?

Appendix B

Mobile Voice: Interview Guidelines for East Bay Center

(30 mins)

Part 0: Intro / Warmup (2-3 mins)

Build rapport.

Part 1: Context of use (5 mins)

What apps do you have on your smartphone? What apps do you use frequently? What apps had you downloaded but then not use at all? Why?

Where do you feel comfortable using your smartphone? What situations would you feel uncomfortable using your smartphone?

Part 2: Open-ended questions about daily tasks (10 mins)

Focus on big issues:

- How do you get to where you want to go (transportation)
    - Smartphone apps
- How do you communicate with your friends/workers (communication)
    - Possible answers: text, emails, voices, social networks
- How do you check the news/weather/get information (information)
- How do you manage your time (scheduling)
- What do you do for leisure (leisure: music, games, videos)
- How do you manage your files and images (File management)?
    - If local: hard drive, smartphone, If online: Google Drive, Dropbox
- How do you manage your bank accounts / personal finance?
- How do you do complex math calculations?
- How do you find new places to explore? (Discovery)

Follow up questions:

- Use smartphone apps: How do you feel? What's the process, can you demo?
    - What means? Voiceover? Siri/Google Now? Other interaction?
- Don't use smartphone: Why not? Have you ever tried to do the task using a smartphone app? What didn't work or made you stop using it? If you haven't tried, why not?

Blue Sky Question: What are some things you wish you could do on your smartphone that you can't do currently? (get at values/opportunities)

Part 3: Contextual Inquiry on how Interviewee uses single app (5-10 mins)

Part 4: Debrief (2-3 mins)

Appendix C

Scenario:

1. iPhone has new voice-over features. You want to try it out! You heard that the app Uber is a convenient way to book a texi. You just downloaded the app and want to try it out using the new voice-over.
2. To turn on the new voice over feature, first turn off voice over and turn it back on.
3. Before using the app, think of a location you want to be picked up and a location you want to go to.

User testing plan:

1. Tell use the scenario.
2. Asks her to use uber with our interface.
3. Tell her to think aloud and voice any confusions or suggestions as they come up.
4. Ask her to use the iphone voice over feature to navigate Uber.
5. Ask for her reaction on our interface

6. If time permits, ask her to make up some interactions for an app she uses frequently.

Roles:

1. iPhone: Wenqin
2. Conductor: Safei
3. Observer: Pi-Tan

Recording:

1. Ask user for picture, video, recording permission.
2. Observer take a short video, picture.
3. Conductor records an audio of the entire interview.

Appendix D

Heuristic Evaluation Results and Fixes

Violations of "1. Visibility of System Status"

N/A, tech constraint

User doesn't know when does the ""Speech recognition engine error"" happens.

severity - 3.

Reason of severity: While Speech recognition engine is the core of the app, this error message uncaught will cause user great confusion.

Unresolved, interaction issue not able to fix

In the long procedural interaction of getting a ride, there is no sound clue of where the user is.

severity - 3.

Reason of severity: Because it matters for every interaction loop, however, it doesn't disable the user from completing the task but causes confusion.

Violations of "3. User Control and Freedom"

Fixed, now app allows users to start over

Cannot exit and cannot go back.

Severity - 2

Reason of severity: I can always leave the app, so although it happens a lot, but it is still ok.

N/A, app feature specific interaction

No way to change car types.

Severity - 2

Reason of severity: Most people (I'm assuming) only use the cheapest Uber, and don't change cars often, so this is not a frequent problem. Because defaults to cheapest, doesn't impact users too negatively.

Unresolved, interaction issue not able to fix

It doesn't allow user freedom to exit a loop, undo/redo, escape or go back.

Severity - 4

Reason of severity: Because this impacts the usability a lot.

Fixed, backend will continuously adapt to new commands.

Violations of "7. Flexibility and Ease of Use"

Violated, because it doesn't afford user-specified customized commands or customized actions.

Severity - 3

This is happening for every interaction loop, however, it doesn't impact the functionality of the overall interaction.

Fixed, design now accepts multiple user commands.

The app doesn't implement the simplified sentence-based commands, such as "Set Home Address" or "Pick me form here".

Severity - 2.

Reason for severity: This is not very important.

Fixable, will address in the future.

Violation of "8. Aesthetic and Minimalist Design"

Questions are simple, but no aesthetics or personality elements are considered in the design.

Severity - 1

Reason for severity: not crucial to usability

N/A, tech constraint.

Violations of "9. Help Users Recognise, Diagnose, and Recover from Errors"

Not repeating the instruction (red blocks in the diagram below, only showing up once in the flow), not showing the error message. When the input fails, the "Speech recognition engine error" does not have special alert sound. And user doesn't know what happened if the query takes longer than usual. He/she might think there is this "Speech recognition engine error" and tap the input again.

Severity 4.

Reason for severity: This happens frequently with the prototype, and the input failure "Speech recognition engine error" is very frustrating and makes it difficult to complete a task and unpleasant to use the app .

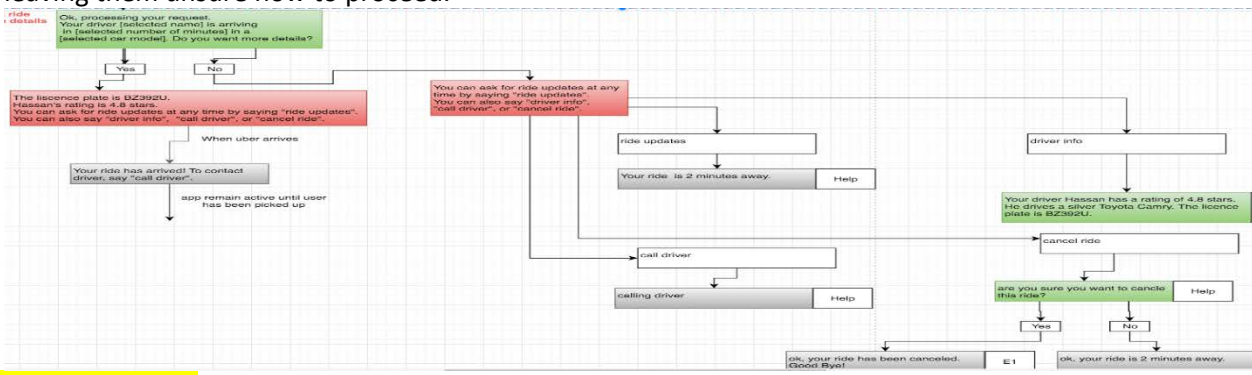
Fixable, will address in the future.

No way to repeat questions.

Severity - 3

Reason for severity: If users are distracted while listening, no way to repeat question. can happen frequently, and impact users negatively by making them confused and

leaving them unsure how to proceed.



N/A, the system can handle this.

It only allows the user to repeat input address when the system gets it wrong, with little instruction about what's the correct input format or what's the options of inputs the user could choose.

Severity - 4

Reason for severity: Because this is important and will impact the usability of every single interaction loop, however the user might still be able to figure out the visual-based error message and instruction embedded in the original app.

Violations of "10. Help and Documentation"

Partially fixed, added contextual help

No way to hear/repeat/discover available options.

Severity - 3

Reason for severity: Users may not know/remember available options. No way to figure out without guess and check (aka not discoverable), which can get tedious and frustration and negatively impact the user experience.

Fixed, added tutorial

Though it provides context-related help command, but it doesn't provide an initial tutorial or onboarding steps to inform the user that those shortcuts like "help" command exist.

Severity - 4

Reason for severity: Because all the shortcut commands such as "help" are designed to improve the usability of the interaction, however, if the user could not know they exist in the beginning, those shortcut commands are not accessible for them.

Appendix E

User #	Gender	Occupation	VoiceOver Proficiency	Age	Cellphone Type	Task 1	Task 2&3	Task 3	Task 4
1	female	-	High	Mid-30s	Android	Y	Y	Y	Y
2	male	-	Low	Late 40s	Feature Phone	Y	N	N	-
3	female	-	Non-user	Late 60s	-	Y	Y	Y	Y
4	female	Volunteer	Non-user	Late 60s	-	Y	Y	Y	Y
5	female		High			Y	Y	Y	Y

Appendix F

Test Guide for Wizard-of-Oz Voice UI Usability Testing
[You could use this test guide as a way to take notes for each task and each question.]

Introduction:
Welcome! The goal of this test is to help us understand how you interact with the new voice user interface designed by our fellow students at UC Berkeley School of Information, and help us identify potential problems with the Voice UI. During the test you will perform 4 tasks. Each task is defined with a specific goal. Since we are not the designers nor the developers of this voice UI app, please feel free to express your confusion or frustration after each task if there is any, since we will not be offended and we want to learn about your true feeling about this app. No prior knowledge about how the voice UI works is required for this test. Please complete the tasks based on your knowledge about smartphone voice interactions in general.
Also, after conducting each task, I will ask for your feedback for each task. This will help us understand what you were thinking when you conduct each task. I will ask for your general feedback and ask you a set of questions after each task.
Thank you again for participating in our test. Could I record the testing process for research purpose only? I'll delete it after summarizing findings from this testing… Thanks! Before we testing the app, think of a location you want to be picked up and a location you want to go to. In the process, feel free to think aloud and say any confusions or suggestions as they come up.

And now, if you are ready, let's start testing this Uber voice app! After "Ding!" I'll switch to the computer voice to play the role of the Uber voice app system, when each task ends, I will notify you with another "Ding!" so that you'll know that I'm switching back to a human being and will ask you questions for feedback.
Your goal for Task 1: Specify your pick up location with Uber Voice
You can say "Open Uber" or "request a ride" to start the Uber voice app!
"Ding!", and follow the flow chart for interactions.
"Ding!" - Questions to ask the participant:

1) What do you generally think about performing this task? Is there any problem you encountered?
2) What you are doing by each voice input? (if a specific voice input from the participant seems involving any confusion or problem, make sure to bring it up and ask the participant) And what's the reason behind each voice input?
3) Is there anything unclear, confusing or making you frustrated or making errors in the process?
4) Is it easy for you to conduct this task?
5) How do you feel about the number of voice inputs required for this task? (too many? Acceptable?)
6) Is it satisfactory to use this app to conduct this task? Why?

Your goal for Task 2 & 3: Select a ride

(our research goal for Task 2 & 3: to compare usability and gain user feedback on voice commands or gesture interactions to go through a list of information)

"Ding!", and follow the flow chart for interactions.

"Ding!" - Questions to ask the participant (tailored to compare Task 2 & 3):

1) What do you generally think about selecting a ride with the two different ways? Which one do you prefer? Is there any problem you encountered?
2) What you are doing by each voice input? (if a specific voice input from the participant seems involving any confusion or problem, make sure to bring it up and ask the participant) And what's the reason behind each voice input?
3) Is there anything unclear, confusing or making you frustrated or making errors in the process? Which method is clearer or less frustrating?
4) Is it easy for you to conduct this task? Which method is easier?
5) How do you feel about the number of voice inputs required for this task? (too many? Acceptable?) Which method is fewer steps?
6) Is it satisfactory to use this app to conduct this task? Which method is more satisfactory? Why?

Your goal for Task 4: Check ride updates and ride details

"Ding!", and follow the flow chart for interactions.

"Ding!" - Questions to ask the participant:

1) What do you generally think about performing this task? Is there any problem you encountered?
2) What you are doing by each voice input? (if a specific voice input from the participant seems involving any confusion or problem, make sure to bring it up and ask the participant) And what's the reason behind each voice input?
3) Is there anything unclear, confusing or making you frustrated or making errors in the process?
4) Is it easy for you to conduct this task?
5) How do you feel about the number of voice inputs required for this task? (too many? Acceptable?)
6) Is it satisfactory to use this app to conduct this task? Why?

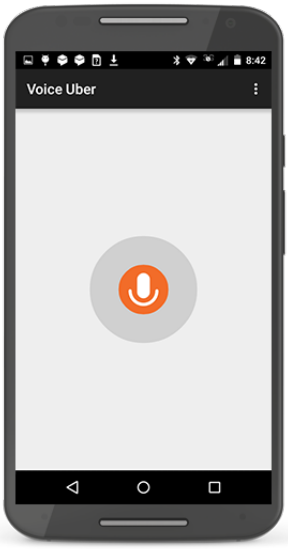
Reflection Questions after Finishing All the Tasks

1) What was the most frustrating thing about your experience?
2) What other ideas do you have about how it could be improved?
3) What did you like about it?
4) Are you likely to recommend this Voice UI to a friend or colleague (no, maybe, yes)?

Wrap-up:

Thank you for participating in this test. You have completed all the tasks. We really appreciate your input. Your participation will be very helpful for us to provide feedback to improve the design of the voice user interface.

Appendix G



APK available for download at
https://drive.google.com/file/d/0B0XsudvRRRiGbFVqcVNnU2xwUTQwLUxhdGpMOW44Z0Vkc0pV/view?usp=sharing