Fluid Design

Rehan Butt
Master of Tangible Interaction Design
Candidate

Carnegie Mellon University
Pittsburgh, PA

May 2018

Table of Contents

Terminology

Flexibility
Ability to be easily modified while being easily adaptable to the context

Modularity
Employing standard units as the basis of design in order to create more complex systems

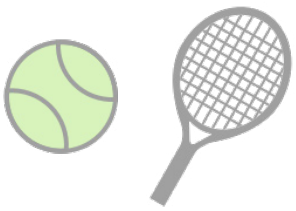
Accessibility
The quality of being easy to obtain or use or be understood

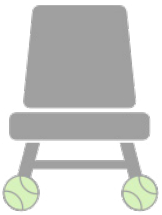
Substitutable
Something that acts or serves in place of another

Module
A standard part or unit that can be used to construct more complex structures

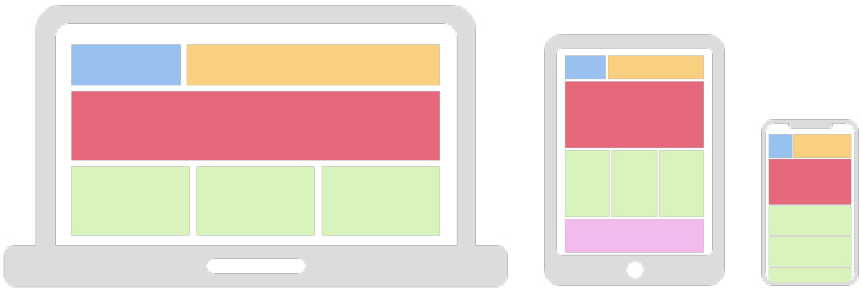
High Fidelity
Designs that include all details necessary for the interactions

Affordance
The qualities of an object that define its possible uses

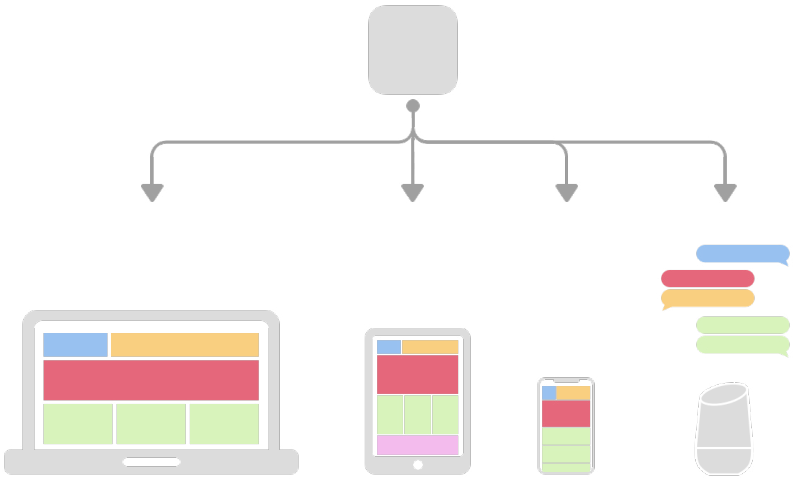




Responsive Design
Creating screen based interfaces that adapts to a wide range of displays



Fluid Design
Creating experiences agnostic of platform, including screen and non-screen interfaces



Abstract

As a user and a designer, I found the currently available design tools to be limiting as they silo designs for each platform, leading the designers to replicate their work for different output platforms. Generally, this approach leads to inconsistencies in user experience and possible errors on part of the designers. My thesis, “Fluid Design” is an investigation into designing digital experiences that expand on responsive design to account for non-screen based interfaces such as voice.

During my research, I have endeavored to evaluate the pros and cons of existing tools, conducted workshops with designers to better understand their needs, conducted and collected survey results. As a result of the findings, I am proposing a new design tool that looks to aid designers in creating the overall experience versus designing for a particular platform. Whilst the need for such a tool was agreed and accepted by designers, designing experiences across different platforms presented challenges that at times were difficult to overcome.

Also, the issue in introducing new design tools revolves around the speed of technological changes. By the time a new tool is understood and deployed for design, some of the functions might have become obsolete. I believe that building flexibility will extend the life cycle of these tools to address future developments.

Introduction

“As McCarthy and Wright write, “We don’t just use or admire technology; we live with it...technology is deeply embedded into our ordinary everyday experience” [13]” [3]

“The massive growth of mobile and tablet devices has compelled both enterprises and individual developers to create UIs viewable on multiple devices. The solution is to create responsive designs [31, 34], which adapt to device environments, e.g., different form-factors. They provide optimal viewing experience across devices by allowing users to read and navigate a page easily and minimizing the effort spent on window resizing, panning, and scrolling.”

Responsive design has become common phrase in the design community these days. The designers and their organizations create experiences across a growing number of devices we use, but the basic definition limits the scope of devices to phones, tablets and desktop or laptop interfaces. What about all the other devices like voice assistants; Amazon’s Alexa, Apple’s Siri or Google Assistant, or smart watches and smart fridges. How can we expand the definition of responsive design to account for these different interactions and for future interaction still to come and how we can design for them? In order to do so I propose a new design tool that looks to aid designers in creating experiences to this more expansive definition over designing for a particular platform. In a similar way to how Bruce Lee describes water in the following quote: “You put water into a cup it becomes the cup. You put water into a bottle it becomes the bottle. You put it in a teapot, it becomes the teapot.” The water being the application or the experience and the container being the platform, such as a phone, watch or voice assistant.

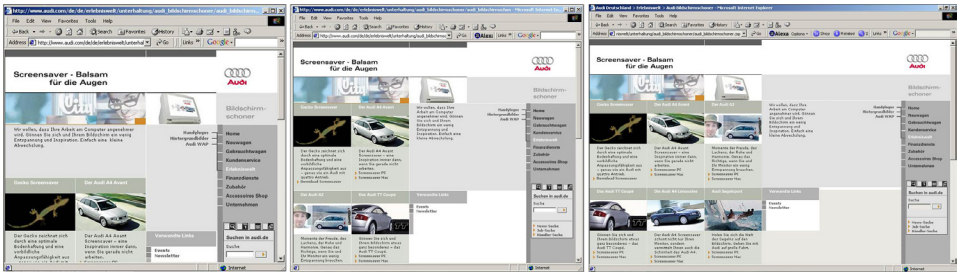
In order to provide context for my proposal I collected background materials, built prototypes, reflected on past design experiences, ran a workshop, designed the proposed tool and tested it with user to see if it worked as intended. Throughout this process adjustments were made based on user feedback. The need for such a tool was agreed on by the user groups, except creation of such a tool proved challenging taking into account different experiences across different platforms, some still to come.

Background

During my research for this project, I have collected materials that support the need for a new kind of design tool around an expanded definition of responsive design. I will discuss my findings organized in three sections; world mimicry, Collaboration and flexibility (material).

History of Responsive Design

“Developing fixed-size Web pages is a fundamentally flawed practice. Not only does it result in Web pages that remain at a constant size regardless of the user’s browser size, but it fails to take advantage of the medium’s flexibility.” James Kalback wrote in January of 2002. (Kalbach, James) He goes on to say designing to some predefined screen resolution does not work even if everyone had the same screen resolution set, because of the way people work. Some people have several windows open, others keep windows smaller than their screen’s resolution. “When developing Web sites we should accommodate a continuum of unpredictable human behaviors… Attempting to control exactly what the user sees is futile—the final product varies due to a wide number of client-end factors. The impulse to prescribe all aspects of layout is a leftover ritual from print media, where designers carefully position each page element.” (Kalbach, James)



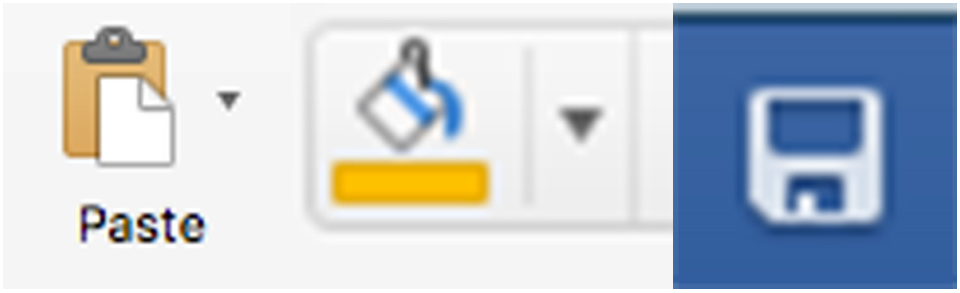
First responsive site Audi site (Kalbach, Jim. “The First Responsive Design Website: Audi (circa 2002).”)

The need for responsive design came out of the growing number of devices, where mobile browsing would overtake desktop browsing. Input devices were changing from T9 keypads and mouse and keyboard to touch or game controllers and many more devices were connected to the internet. And working to facilitate a unique site for each of these devices feels “like a zero sum game”. (Marcotte) Creating a unique experience for each device is far too laborious, but we could treat each device as a facet to the same experience. “We can design for an optimal viewing

experience but embed standards-based technologies into our designs to make them not only more flexible, but more adaptive to the media that

World Mimicry

renders them." (Marcotte)
Mimicking parts of the physical world in digital experiences aids the users understanding. As stated by Wiltse "Graphical user interfaces (GUIs) use metaphors like the desktop, handles, sliders, and file folders that allow users to draw on familiar mental models." (Wiltse) These metaphors allow the users to get past understanding how certain digital experience may work and focus on using it, whether a consumer facing interaction or a design tool. Mimicry works as it leverages our existing understanding such as the save icon being represented by a floppy disk or using a paint bucket icon to representing changing the color of an object. Given this shared understanding, designers and their tools should leverage mimicry for ease of use of their experiences.



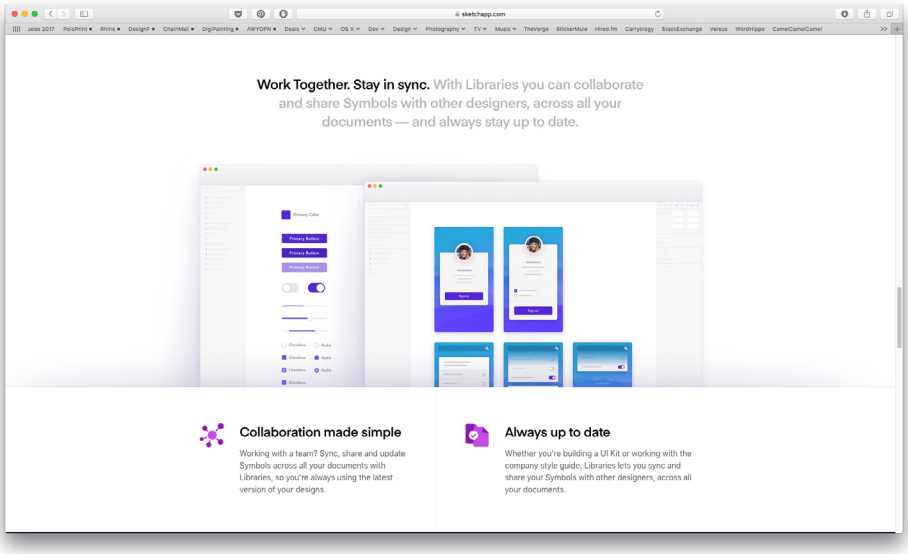
Microsoft Word screenshot: Paste icon represented by a clipboard; color function represented by a paint bucket; save represented by a floppy disk. All mimicking aspects of the physical world within a digital interface.

However, removing the hurdles of understanding a product or experience does not necessarily mean it will be used as expected "discussing the user experience, state that designers can create "situations" or "levers" that people can interact with, but they cannot design an outcome for a user to experience [5]. Such interpretations suggest that the authors acknowledge the uniqueness of the individual in making sense of the world" (Muise) This is where the designers' knowledge and training should come in, verses them spending time understanding how to use a tool or finding the right tool for the job. The designers' knowledge should be deployed to create a robust system in order to account for many of the user interactions and flows. By using this idea of mimicry in the proposed design tool we also aid the designers in their understanding of the tool and in turn aid in the understanding of the designed experience to the end user.

Collaboration

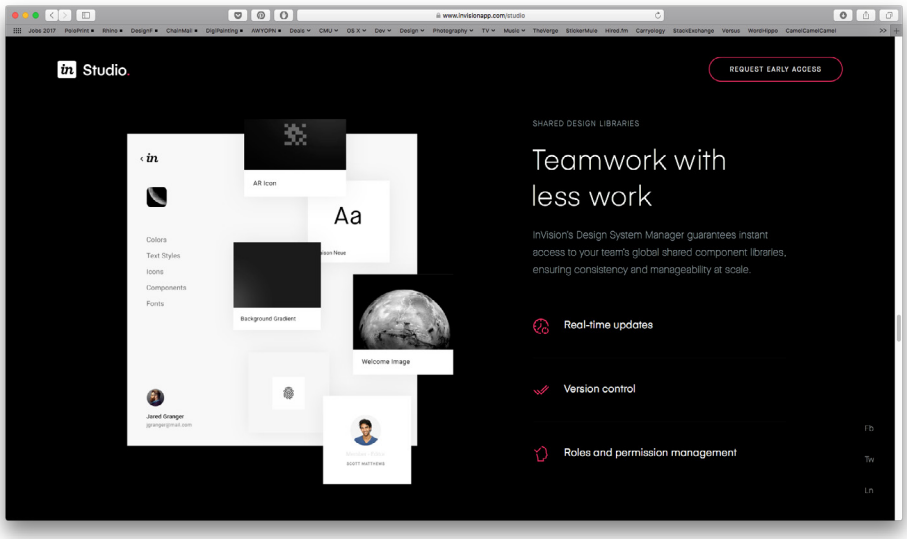
Many of the best designs come from the collective work and thoughts of teams of people. “So I think that the best thing that we did about the design...is that we forced people to talk to each other” (participant 1). Creating interactions that allowed for conversations to take place.” (Muise) Social interactions between individuals helps capture the experiences, by sharing and conversing about what the users have learned and bring forward relevant past experiences. Keeping this in mind, how do we bring collaboration to digital design tools? “The important aspects of group design are the communication and documentation of plans, and the coordination of software versions as group members begin to work on complementary components of the design.”(Rosson) I would like to focus my work on the second part of this quote where it refers to collaboration in design, helping designers design together when they move to the computer. Looking at how software can aid team members stay aligned as the design process transitions from initial discussions and white boarding sessions to computers and higher fidelity output.

Existing tools can be drawn on a spectrum of two axis: level of fidelity and whether they allow for real-time collaboration between team members. Sketch, Adobe XD, inVision Studio and Figma all allow for the highest fidelity of output when it comes to user interface (UI) design, but only one allows for any level of real-time collaboration outside of updating shared components or symbols which are generally elements of a style guide. Tools that define collaboration as more than a shared component library are also referred to as collaborative real-time editors.



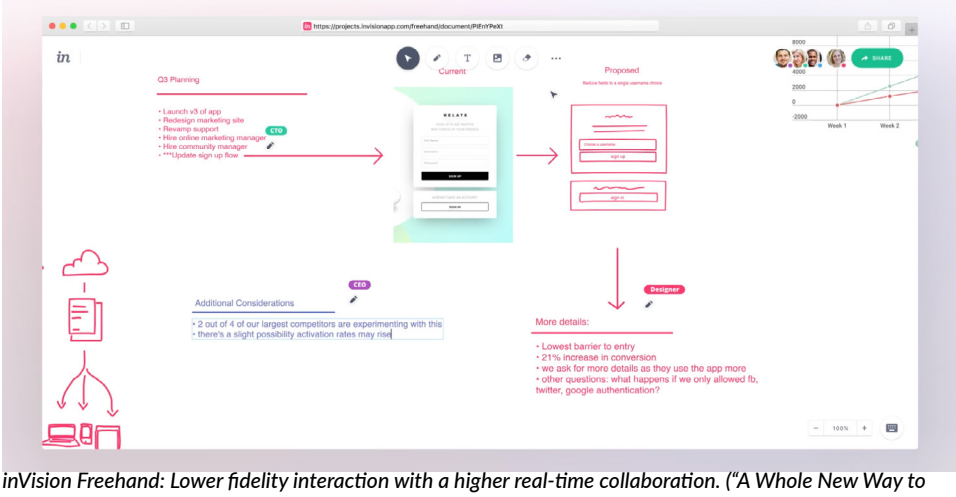
Sketchapp.com screenshot: Their definition of collaboration limited to the sharing of components; rather than working simultaneously with other team members (“The Digital Design Toolkit”)

Real-time collaborative editors allow for several people to edit a file simultaneously. Sketch and inVision Studio fail to offer simultaneous editing and rather promote “libraries” and “the Design System Manager” as their answer to collaboration. These libraries and design system managers are used to keep colors, fonts, icons and UI elements such as buttons in sync between the team, falling short of real-time collaboration.



inVision Studio Splash page screenshot: Their definition of collaboration again is limited to the sharing of components and styles such as color fonts and icons; rather than working simultaneously with other team members. (“The World’s Most Powerful Screen Design Tool.”)

inVision does offer another product in their suite called Freehand. It allows for commenting and marking up mockups from Sketch or Photoshop using typed text and a pencil tool for drawing, increasing the real-time collaboration as the fidelity decreases. The intend of Freehand is to also include other team members outside of the design discipline, like managers, developers and project leads. “Simple tools, unlimited creativity: Freehand is built with intuitive tools-like Draw, Write, Sketch and Comment-and functionality that make joining in easy and fun.” (“A Whole New Way to Collaborate Creatively.”) In order to allow all stakeholders to join in on the conversation they have limited the functionality to common interactions. “Good design is good for business, and great ideas come from anywhere in your organization.” (“A Whole New Way to Collaborate Creatively.”)



inVision Freehand: Lower fidelity interaction with a higher real-time collaboration. (“A Whole New Way to Collaborate Creatively.”)

During their demonstration, Adobe XD has shown what they would like collaboration to be. They imagine multiple users to be simultaneously editing a single file but limited to the artboard level. Meaning that if user one

is editing artboard one, user two cannot edit the same artboard. Paul Gubbay from Adobe says they have this artboard limitation as “we don’t want designers coming in and stepping on each other’s toes” (“Adobe MAX 2016. Day 1 Keynote (Chapter 3).”) Whether an artboard level lock is the right level to prevent user interference is yet to be tested. Does an artboard level separation allow for enough intermingling of ideas or would it need to be at a lower level such as an object level?

Figma sees real-time collaboration as not only the sharing of style guide elements but also, as in a similar case to Adobe XD’s demo, a simultaneous editing space for multiple designers to work. (“The Collaborative Interface Design Tool.”) Figma calls their approach real-time collaboration multiplayer editing. Figma takes a different approach than Adobe when it comes to a level of cross interaction. They restrict user overlap to the object level over the artboard level. “You don’t have to worry about editing the wrong version, coordinating with others to avoid overwriting their work, or dealing with conflicts from your files living inside a separate syncing service.” (Wallace) Wallace notes that their multiplayer editing was also their answer to version control, as there is only one live file being used at all times by the whole team when using real-time simultaneous collaboration.

In some cases, due to the lack of real-time collaboration of these high-fidelity tools, teams revert to lower fidelity tools in order to get the level of collaboration they need. “we use Google Drawings to create wireframes because it allows us all to work in the same file at the same time and to instantly see changes made.” (“Collaboration on Same Document.”) That being said these are not strictly design tools but tools that are more beneficial for the task of collaborating.

	UX Quality	Simultaneous Realtime Editing	Collaborate Real-time Feedback	Fidelity (If a viewer, view quality)	Cost (per user)
Modelo	Ok	None	Yes	Medium	N/A
Sketch	Excellent	None	None	High	$100
RedPen	Good	None	Yes	High	Free - $90 Monthly
Zeplin	Good	1-Directional	None	High	Free - $100 Monthly
Pixelapse	Ok	None	Yes	High	Free- $69 Monthly
Realtime Board	Ok	Yes	Yes	Medium	Free - $50 Monthly
Screen Sharing	Excellent	1-Directional	Yes	Medium	Free
Adobe CC	Excellent	None	None	High	$9 - $80 Monthly
Google Apps*	Great	Yes	Yes	Low	Free
Dropbox Paper	Good	Yes	Yes	Low	Free - $15 Monthly
Wake.IO	Ok	None	Yes	High	$12 Monthly

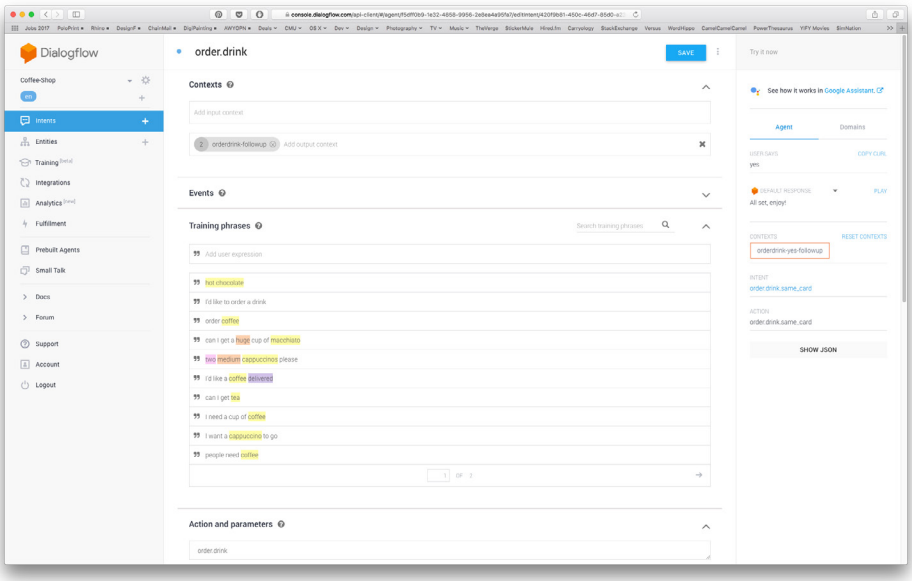
Tool Comparison Chart

Overall there are many types of collaboration among design tools whether a sharing of design libraries or real-time collaboration. It is important

to have a high-fidelity design that allows for collaboration in order to capture audiences that are reverting to lower fidelity tools such as Google Drawings or InVision Freehand later transitioning to a secondary higher fidelity tool such as Sketch, InVision Studio or Figma. My tool will need to address collaboration.

Voice Tools

Previous sections outlined research primarily for design tools involving on-screen experiences. With the increasing adoption of voice, it is imperative that the design tools are able to address experiences on this platform. Though several tools are readily available for on-screen experience designing, voice design tools are not as common. In this section I will discuss some of the more commonly used voice design tools and frameworks, including Dialogflow, Sayspring, Apple's Siri Guidelines, Google's Conversation UI Guideline and Amazon's Alexa Voice Design Guide.

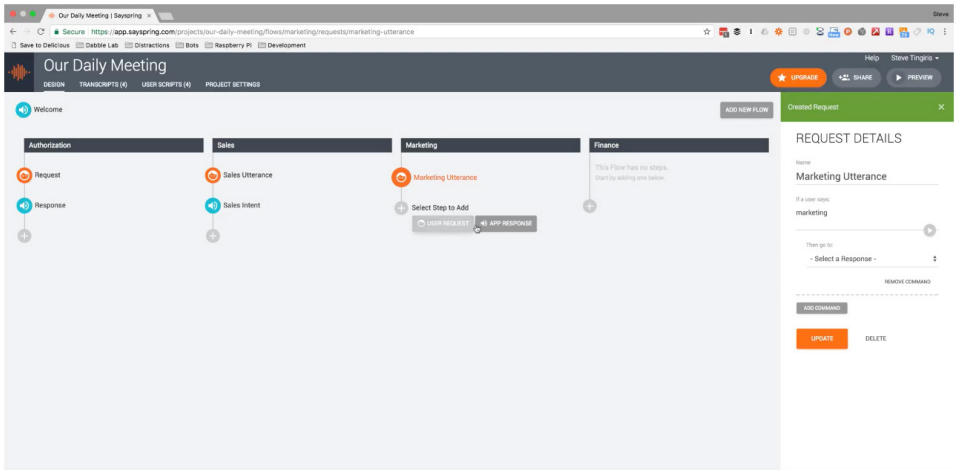


Dialogflow User Interface ("Dialogflow.")

Dialogflow formerly api.ai, "provides tools to developers building apps ("Actions") for the Google Assistant." ("Dialogflow.") Dialogflow handles Natural Language Processing the idea of "keyword matching, understanding human speech to derive intent and meaning". (Dhir) In other words it can support language processing of existing applications or help users create chatbots. In order to actually build an experience, it required far more details than what is generally accomplished during the design process. Dialogflow is made up of three main components: intents, contexts and entities. Intents are mappings between what a user says and what action the application takes. "Contexts represent the current context of a

user's request." ("Docs Basics.") In other words it brings clarity to general terminality through previous dialog. For example, if a user asks "Turn the living room lights on" followed by "turn them off" the system would know that "them" refers to the "living room lights". And lastly entities allow for extracting parameters from the users input.

Dialogflow is an interesting tool as it reduces the barrier to entry for producing a chatbot application for testing or production. It removes some of the development hurdles and helps developers and designers be very detailed with the types of inputs and outputs but fails to create a common vocabulary that maps cleanly with common conversational components while displaying the entire conversation in a single view. Dialogflow is a prototyping and production tool rather than a design tool and has a decent learning curve in creating interactions.



Sayspring User Interface ("Prototyping Amazon Alexa Skills with Sayspring.")

Sayspring is another voice tool that is more geared towards the pre-development phase. "Sayspring enables designers to create voice-enabled apps without code ahead of handing over projects to development." "what we are not building is the Squarespace of voice, the idea is how do we remove all the technical limitations of working with the voice medium so we can facilitate a proper design process for design teams and product teams." (Webster, Mark. "What Ever Designer Should Know About Voice.") Sayspring fits much closer to a design tool and the work I'm pursing around creating a tool that can help design for different platforms including voice. Adobe also realizes that voice designs tools are needed and as of April 16, 2018 Sayspring is part of Adobe. "Webster, Mark. "Sayspring Is Now Part of Adobe.""

There are three main components of Sayspring; flows, user requests and app responses. Flows contain users request and app response components. Sayspring allows you to build flows and create responses with example content. It also allows you to test the voice interaction

directly in the tool. I would be interested to know if it allows for more robust flows but unfortunately was not able to test Sayspring due to its limited invitation only access. “We’re moving beyond the keyboard and mouse and even our touch screens to using something that is even more natural — our voice — to interact with technology. Voice tech is growing fast, and we strongly believe it must become an integral part of Adobe’s portfolio moving forward.” (“Adobe Bets on Voice with Sayspring Acquisition.”)

Apple, Google and Amazon all provide guidelines for using their respective voice interfaces; Siri, Assistant and Alexa. Apple provides a list of interactions that can be used to integrate your application with Siri. Unfortunately it is quite a limited list and Apple manages all the interactions of Siri integration. “Don’t create an interface that appears interactive. Your interface can’t respond to gestures—other than a tap, which opens your app—or other events when displayed within Siri, so avoid displaying imagery or shapes that appear interactive.” (Apple Inc.) That being said they do note that interactions should “focus choices that reduce the possibility of additional prompting”, “take people directly to the expected destination” and a more specific guideline around purchasing through Siri “For a purchase with multiple pricing levels, don’t default to the most expensive. At the point where a user is making a payment, don’t charge extra fees without informing them.” (Apple Inc.)

Supported Interactions

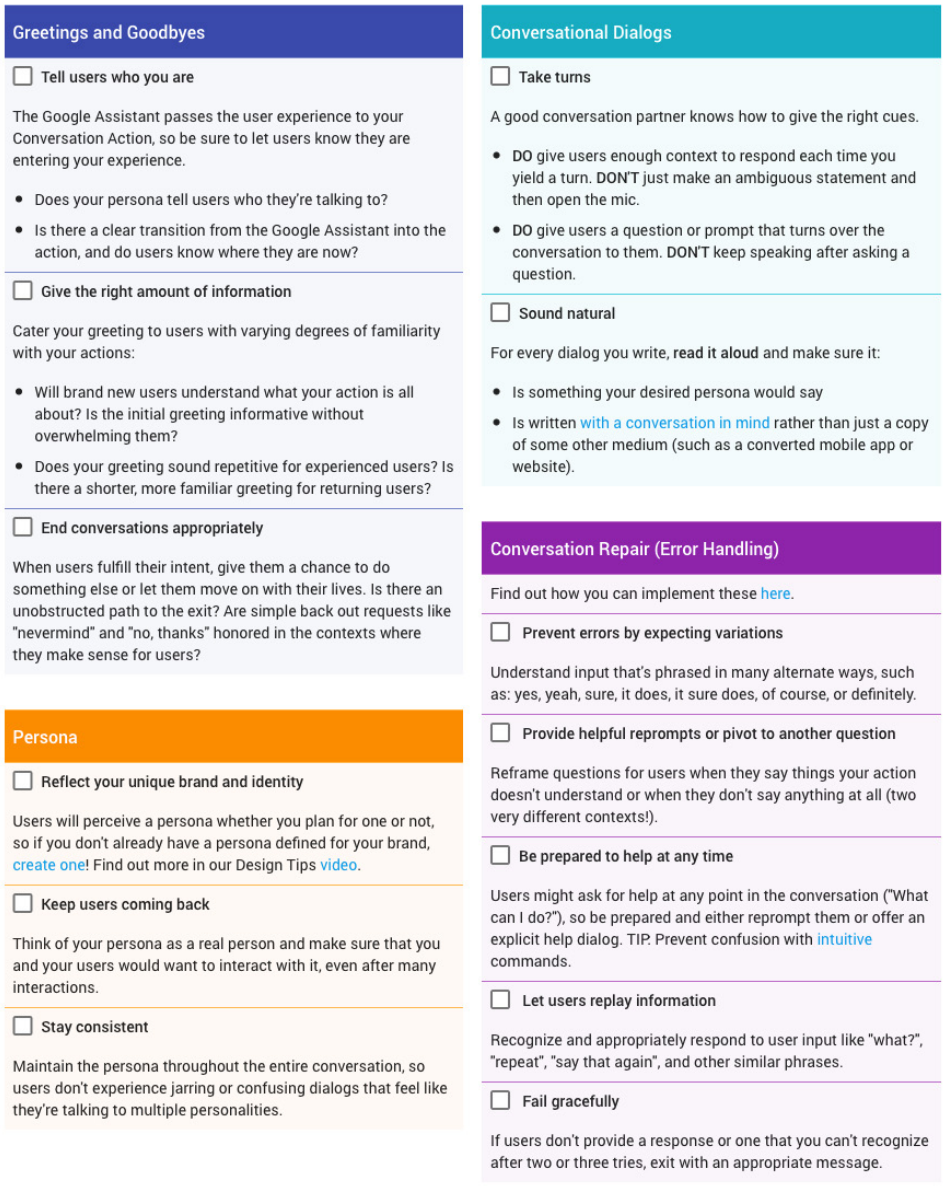
iOS apps that offer the following services can integrate with Siri.

Service	Supported Siri interactions
Audio and video calling	Initiate calls. Search the call history.
CarPlay integration	Activate and save a driver's settings. Change the car’s audio source. Change the car’s climate. Change the car’s defroster settings. Change the car’s seat settings. Change the car’s radio station.
Fitness activities	Start, pause, resume, end, and cancel workouts.
Lists and notes	Create to-do lists and items. Search for to-do lists and items. Mark to-do list items as complete. Create reminders based on a date, time, and/or location. Create notes. Search for notes. Modify notes.
Messaging	Send messages. Read received messages. Search for messages.
Payments	Send payments. Request payments. Pay bills. Search for bills. Search for and view account information, including balances, points, and miles. Transfer money between accounts.
Photo management	Search for photos and show them in the app.
Ride booking	Book rides. Provide ride status information.
Vehicle integration	Activate hazard lights or honk the horn. Lock and unlock the doors. Check the current fuel or power level.
Visual codes	Show a visual code, like a QR code or bar code.

Siri Supported Interaction Image (Apple Inc.)

Google has a dedicated conversational UI guideline. They start by sharing

their “building blocks” for a conversation: “Turn-taking”, “Threading”, “leveraging the inherent efficiency of language” and “anticipating variable user behavior”. From here Google moves you through their design principles for voice, all based on the idea that voice should be simple, clear and only add to the conversation when it benefits the user. Google has a far more robust guideline than Apple as to how they would like the integration with Google Assistant and share their constraints and asks. They nicely sum up all of their design recommendations in this “Design Checklist”.



Greetings and Goodbyes

☐ **Tell users who you are**

The Google Assistant passes the user experience to your Conversation Action, so be sure to let users know they are entering your experience.

- Does your persona tell users who they're talking to?
- Is there a clear transition from the Google Assistant into the action, and do users know where they are now?

☐ **Give the right amount of information**

Cater your greeting to users with varying degrees of familiarity with your actions:

- Will brand new users understand what your action is all about? Is the initial greeting informative without overwhelming them?
- Does your greeting sound repetitive for experienced users? Is there a shorter, more familiar greeting for returning users?

☐ **End conversations appropriately**

When users fulfill their intent, give them a chance to do something else or let them move on with their lives. Is there an unobstructed path to the exit? Are simple back out requests like "nevermind" and "no, thanks" honored in the contexts where they make sense for users?

Persona

☐ **Reflect your unique brand and identity**

Users will perceive a persona whether you plan for one or not, so if you don't already have a persona defined for your brand, create one! Find out more in our Design Tips video.

☐ **Keep users coming back**

Think of your persona as a real person and make sure that you and your users would want to interact with it, even after many interactions.

☐ **Stay consistent**

Maintain the persona throughout the entire conversation, so users don't experience jarring or confusing dialogs that feel like they're talking to multiple personalities.

Conversational Dialogs

☐ **Take turns**

A good conversation partner knows how to give the right cues.

- DO give users enough context to respond each time you yield a turn. DON'T just make an ambiguous statement and then open the mic.
- DO give users a question or prompt that turns over the conversation to them. DON'T keep speaking after asking a question.

☐ **Sound natural**

For every dialog you write, **read it aloud** and make sure it:

- Is something your desired persona would say
- Is written with a conversation in mind rather than just a copy of some other medium (such as a converted mobile app or website).

Conversation Repair (Error Handling)

Find out how you can implement these here.

☐ **Prevent errors by expecting variations**

Understand input that's phrased in many alternate ways, such as: yes, yeah, sure, it does, it sure does, of course, or definitely.

☐ **Provide helpful reprompts or pivot to another question**

Reframe questions for users when they say things your action doesn't understand or when they don't say anything at all (two very different contexts!).

☐ **Be prepared to help at any time**

Users might ask for help at any point in the conversation ("What can I do?"), so be prepared and either reprompt them or offer an explicit help dialog. TIP: Prevent confusion with intuitive commands.

☐ **Let users replay information**

Recognize and appropriately respond to user input like "what?", "repeat", "say that again", and other similar phrases.

☐ **Fail gracefully**

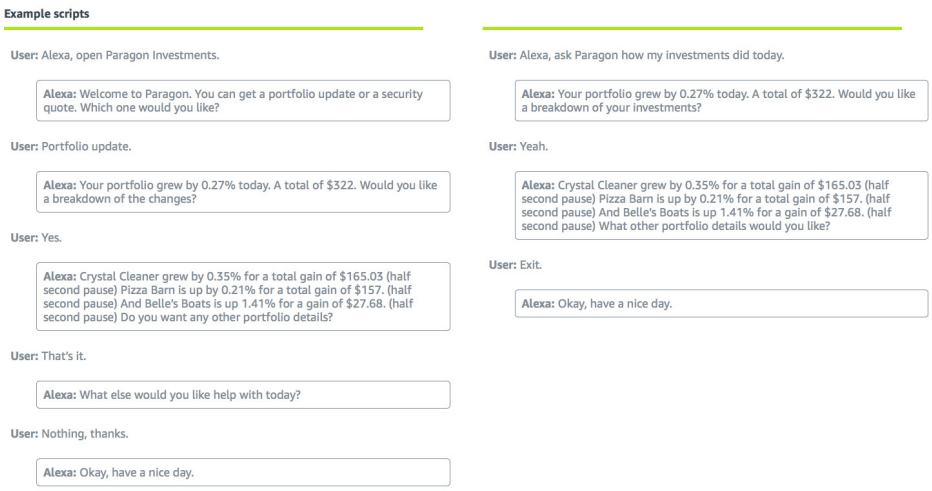
If users don't provide a response or one that you can't recognize after two or three tries, exit with an appropriate message.

Google Design Checklist (“The Conversational UI and Why It Matters | Actions on Google | Google Developers.”)

Lastly Amazon shares their Voice Design Guide, which has many similarities to that of Google’s. Both Google and Amazon guide you through their process for designing a voice experience, starting with establishing the purpose for the experience, writing out a script, translating that script into a flow and lastly a “get ready to build” step that looks at translating the flow into intents and utterances for a developer to work with.

Google and Amazon differ as Google asks you to think about the persona or voice of your experience where Amazon does not as it will use their voice, Alexa. Google has several voice options to choice from; two male and two female voice options for several of the languages they support. This functionality could later expand and adapt to the needs of future voice applications.

The section above summarizes review findings of Dialog Flow and Sayspring interfaces in designing the voice experience along with the benefits and shortcomings of these tools. The three most common voice platform currently used; Apple's Siri, Amazon's Alexa and Google's Assistant, were reviewed for voice experience creation. Siri's guidelines were found to be limited with apple controlling the interface, whereas Google and Amazon provided more robust and



Amazon Example Scripts ("Design Process.")

Flexibility (Material)

detailed guidelines for the designers. Google provides more options that will be helpful in developing future voice applications and experiences.

"discussing the user experience, state that designers can create "situations" or "levers" that people can interact with, but they cannot design an outcome for a user to experience (Zhang). Such interpretations suggest that the authors acknowledge the uniqueness of the individual in making sense of the world" (Muise) This passage also speaks to the need for flexibility as the users' flow is indeterminate. Flexibility in how the experience is designed, allowing for the user to accomplish their task without frustration and difficulty getting in the way due to this lack of flexibility.

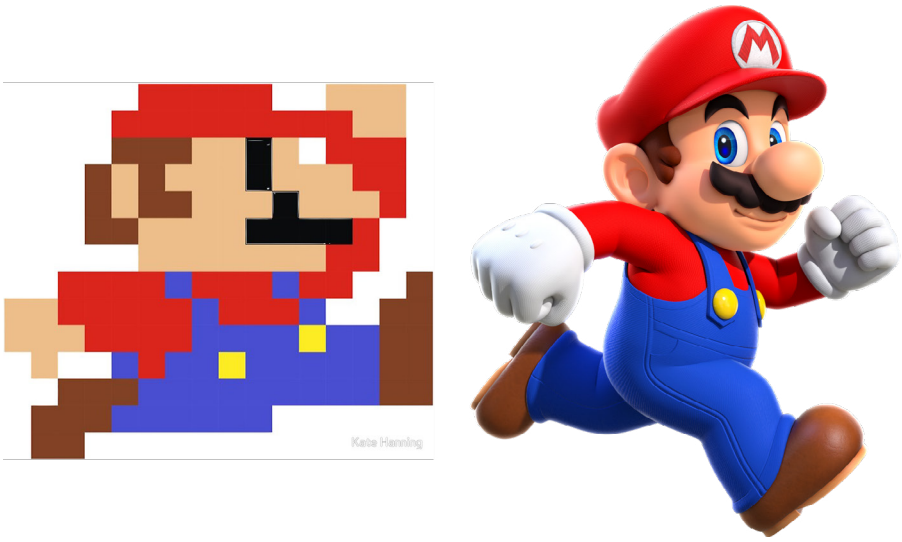
Flexibility starts from accessibility. The perceived ease of access allows users to create new uses for an artifact. "it is somewhat unsurprising that a lower number of digital objects were repaired compared to those that were mechanical. This comparison gives evidence of a much larger phenomenon where not only mechanical objects are found to be more 'repairable' than digital objects, they also lend themselves to be more reusable." (Maestri) This idea holds true when physical and digital

artifacts are compared. Physical objects are easier to visualize giving an illusion that they are more functionally transparent, unlike a digital product where layers of software are hidden underneath and require technical expertise to comprehend. Soft good products such as bags are a great example of flexibility, as these are made of fabrics and are easily manipulated adapting to different circumstances. The other aspect that makes soft goods easy to access is a result of their abundance in our daily lives, giving us a sense of understanding of the material. In Maestri's "Understanding Repair as a Creative Process of Everyday Design" she draws a connection between flexibility and reparability where flexibility in material leads to ease of repair: "Key material attributes of repair: flexible materials, substitutable materials, salvageable materials. [4]" There is an interesting concept that starts to emerge from these attributes of flexible and substitutable leading to being repairable. "I have an old climbing rope that could be labeled as 'broken'. It's not suitable as a safety device while climbing anymore but I have woven it into a door mat" (Maestri) This is a great example of how a flexible material has led to modifying within a different context of a doormat from a climbing instrument. The question becomes what is the digital equivalent of this rope example? Something so flexible it can substitute itself on different platforms within a digital context such as on a screen and within a voice assistant, in a way a transformative object. What would one of these materials look like in a digital design tool and how do these flexible modules work together to create a design system?

Rope shifting context and uses

One criticism to introducing a new design tool is whether it will hold up to the test of time as said in "Past, Present and Future of User Interface Software Tools": "Moving Targets: It is difficult to build tools without having significant experience with, and understanding of, the tasks they support. However, the rapid development of new interface technology, and new interface techniques, can make it difficult for tools to keep pace. By the time a new user interface implementation task is understood well enough to produce good tools, the task may have become less important,

or even obsolete.” (Myers) Currently available design tools do not offer a flexible material to this extent. Building a flexible material or object within a digital tool would alleviate the issue of limited usability and would hold up to the test of time for future design conventions and new platforms still to be developed. A secondary question that arises is; to what depth do these materials exist, at the pixel level or something larger? “how to appropriately modularize the software into smaller parts, while still providing significant capabilities and integration to users.” (Myers) I plan to address this within my work as well as looking at the appropriate scale or depth for a flexible material to exist. Creating a module at the pixel level would not add value to the designer but if the flexibility is only at the largest scale it becomes inflexible. It is synonymous to a render at 8-bit vs 64-bit where the level of detail at sixty-four is far higher because more components are being used as part of the design.



Two Mario images illustrating an increased level of detail with smaller components (“8-Bit Mario Nintendo Jumping.”) (Super Mario Run)

Process & Methodology

This section of the thesis covers the methods used to build the designs for Fluid Design. A human centered design approach was used in order to gain insights about the UI/UX design process, specifically when designing for more than one platform. This work largely informs my proposed design tool while address several questions from my background research. During this process, I created several paper prototypes, revisited previous works and ran a workshop to gain insights from designers.

Definition

One of the first things undertaken was to establish a baseline understanding of responsive design amongst the design community. I sent out surveys, used online forums and had several conversations with designers to gather feedback. Whilst there were a whole range of answers regarding understanding and definition of responsive design, including, environment considerations, and input method ie. touch vs mouse, most of the responses addressed one consistent aspect “an interface that responds to the changing screen size”. Even though this could have been taken as the general understanding of the term “responsive”, it was nonetheless important to have a clear understanding and agreement for it will be the basis of my research. This established a baseline and an alignment of understanding about responsive design between myself and the design community I would be working for.

There were a few other interesting definitions outside of this aspect of design; “a responsive design is made to fit all types of screens depending on size (small phone screens to 49” TV screens), rotation (if you turn your phone, design should adapt) and way of use (a touchscreen will not be the same as a mouse or keyboard).” In this definition the idea of rotation and “way of use” start to incorporate the user’s context. In case of rotation is this a moveable, handheld device or a stationary machine such as a desktop, whereas in the “way of use” is it a touch screen or a mouse and keyboard interaction. Getting to the fact of some of the affordances of

these context changes: touch interfaces do not allow for a hover interaction where a mouse interaction does, or person of a touch interface is defined by your finger were the mouse cursors can allow for more precision that would not occur with the finger. In additional a mouse allows for 3 controls where your finger only has one (Nielsen, Jakob. “Mouse vs. Fingers as Input Device.”). Below is a chart displaying even more of the affordances of a mouse vs finger as an input device.

	Mouse	Fingers
Precision	High	Low (“fat-finger problem”)
Number of points specified	1	usually 1 2–3 with multi-touch
Number of controls	3: left/right button, scroll wheel	1
Homing time?	Yes	No
Signal states	Hover, mouse-down, mouse-up	Finger-down, finger-up
Accelerated movements	Yes	No
Suitable for use with huge screens (30-inch or more)	Yes, because of acceleration	No: arm fatigue
Visible pointer/cursor	Yes	No
Obscures view of screen	No, thus allowing for continuous visual feedback	Yes
Suitable for mobile	No	Yes: nothing extra to carry around
Ease of learning	Fairly easy	Virtually no learning time
Direct engagement with screen and "fun" to use	No: an indirect pointing device	Yes
Accessibility support	Yes	No

Mouse vs Finger superiority across several dimensions (Nielsen, Jakob. “Mouse vs. Fingers as Input Device.”)

Continuing on this idea of the user’s context: “Responds to a different environment, so if there is different lighting condition it may change, responsive design for Waze so when it’s night it changes to a black background”. In this case the individual sees responsive design as a response to the environment, such as lighting or time of day, which definitely goes beyond changing screen sizes. The third example really brings to attention a design aspect regarding seamless change, to the given hardware or platform, and the software components such as the operating system. “An interface that responds to the changing hardware or software that’s being used whether that’s screen size or operating system, seamless transition”.

I have decided to focus my efforts on Fluid Design as being seamless across different hardware or software platforms. I will endeavor to create a design environment that not only seamlessly handles the hardware transition and the operating system but also between mobile and voice platforms of that device. When thinking about expanding on responsive design into Fluid Design on the basis of platform differences, I wanted to choose two distinct and existing platforms to use for benchmarking. I decided to use mobile and voice, leveraging a screen based interface and a voice based interface. The choice to use a mobile interface and voice interface comes from the likelihood that you have had many interactions with them. As well as wanting to include a screen interface something UI/

UX designers are familiar with and voice, a non-screen based interface, which fewer designers may be familiar with designing for but is still critical to the overall landscape of platforms that exist today as well as looking forward.

Chatbot

With the idea of benchmarking on a mobile interface and a voice interface I reflected on a chatbot I had developed previously to help identify some of the potential issues that I may run into through this process of designing platform agnostic. Through this reflection I was able to list issues that I would need to keep in mind when proposing a new design tool. There are aspects of a chatbot experience similar to voice experiences, but there are also several differences including the context in which they are used. Chatbots require users to type and interface with a screen device where voice experiences do not. (Botanalytics) Chatbots also allow the user to recall previous messages more easily through the chat history where voice does not store a visual history of the conversation.

In this regard, I will be covering tone, personality, error handling and clarity as the main topics when discussing the chatbot. Tone and personality are very much linked and are critical in a chat based experience, as text is the main user interface element. (Dudley) Whether fun and colloquial or more formal, the way the copy is written helps the user understand how they may interact, setting an expectation. In my example my audience was design inclined individuals as the chatbot was expanding the color vocabulary of the user. Given this audience and service the chatbot was providing I wanted to keep the tone fun and used simple terminology, colloquial phrases and emojis. Some of the phrases I used are as follows: “Awesome sauce! Give me one second.” “Coming right up!” “Enjoy your Inspiration!” “Happy Coloring!”

The next aspect that is particularly important in chat based experiences is error handling, in other words being able to handle unexpected answers in the conversation. The reason this is a bigger concern in a chat environment is because the chatbot environment is free form in comparison to a visual

based interface. The keyboard allows for any number of responses to be typed versus a visual interface where there may only be a couple of buttons which are binary interactions. Either you are engaged or you are not with the function triggered by the buttons.

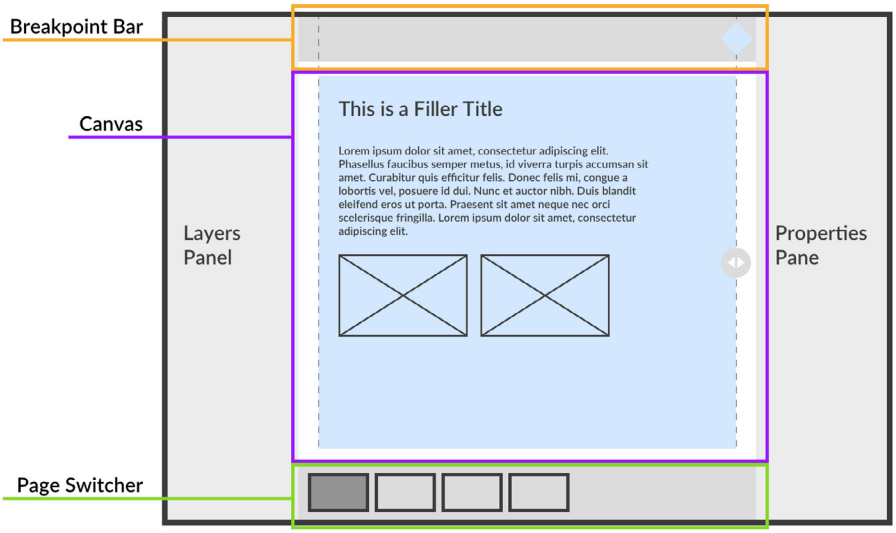
This leads to the next topic of clarity, if the chatbot interactions are well defined and communicated there is less of a need to create error handling cases. Creating a help prompt as well as an introduction prompt on how to interact can lead the user down the right path. Another example of error handling and clarity would be to inform the user on how they may have to respond to a particular question. A simple yes or no question may elicit a response of “sounds good” but the designer may not have considered that response on the first iteration as my chatbot did, only handling “yes” or “no”. This would be fairly easy for the designer to add in. The tool could potentially also aid in suggesting possible answers, leading to more error-proof experiences.

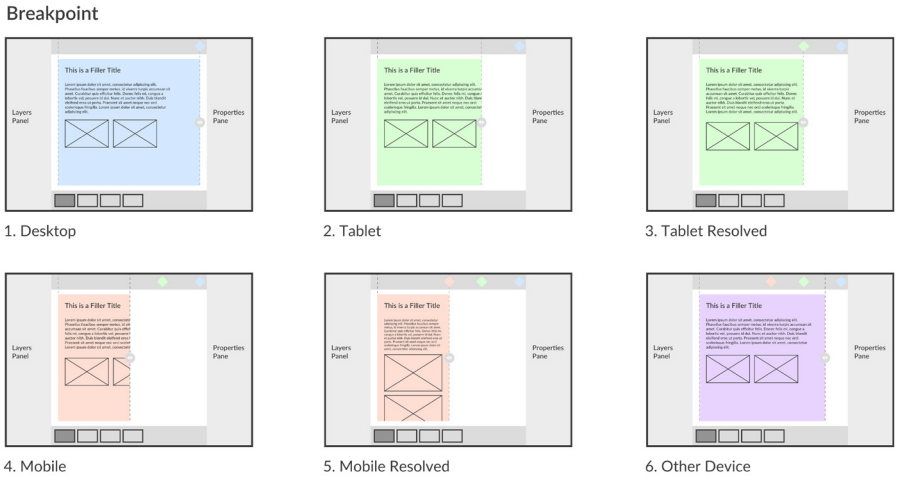
Prototypes

“It’s [responsive design] always going to be a problem when we’re designing using static pixel tools instead of declaring relationships between things”. This quote from one of the designers I interviewed, starts to get at what is lacking in our current set of design tools. We are using tools that mainly work in static ways whereas the environments used for these experiences are very dynamic. So why is this the case? Shouldn’t our design tools be as flexible as the platforms and environments they are deployed in? I address this issue by introducing a new tool feature while using many functions from popular screen design tools.

I have created two prototypes and labeled them “Breakdown” and “Autosize” based on elements that they represent. In this first paper prototype, “Breakpoint”, I explore the concept of a breakpoint that is used commonly on the web within the framing of a design tool. Breakpoints refer to the point at which your site’s content will provide the user with the best possible layout to view your site.

To the left is an illustration discussing the main sections of the interface, which includes a layers panel, breakpoint bar, canvas, page switcher and properties pane. The layers panel, canvas and properties pane mimic many similarities to existing screen design tools such as Sketch. What is unique in this prototype is the addition of the breakpoint bar and the page switcher. The page switcher switches between several different pages in an experience, such as a home page, about page and contact page. The following image helps to show how the breakpoint bar interacts with the designer. Let's imagine you are designing a single page that should be displayed across several different screen sizes, desktop, tablet and mobile. You start designing the page at a desktop size, containing a title, text and two images and then want to transition to design this same page on tablet.

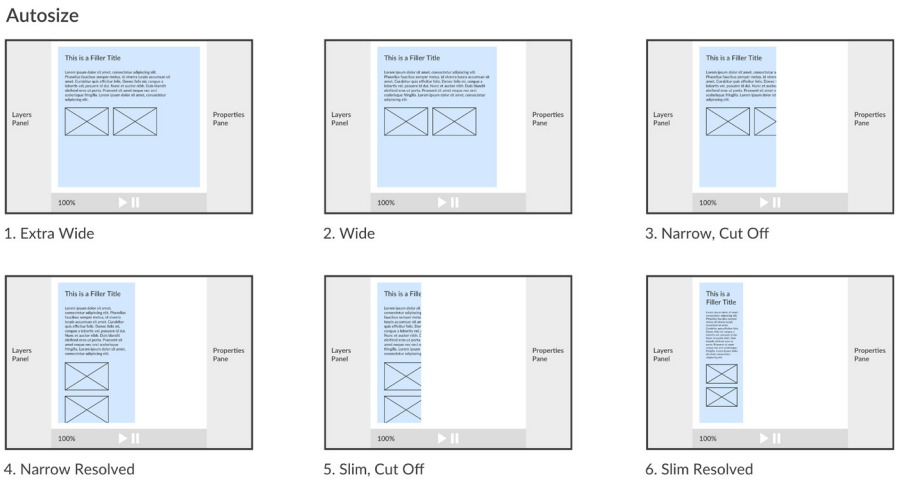




If you were using an existing tool such as Sketch you would duplicate the artboard and resize it to the desired tablet width, rearranging the content as you see fit. But the issue with this approach is that any content updates have to be done across two artboards or even more if you had many more device specific versions you wanted to configure. In the breakpoint version you simply resize the canvas to the desired width and edit the content as you see fit, taking us to the third step illustrated in the image below. Once elements are moved around at a specific width a marker is dropped within the breakpoint bar to show that changes will occur at that width. This process can continue for as many devices as are desired.

This doesn't take into consideration the non-screen based interfaces or even aspects that exist in CSS grids where items are percentage based. Percentage sizing could easily be added to the properties pane to pull in a more natural transition from device to device that CSS grids allows for.

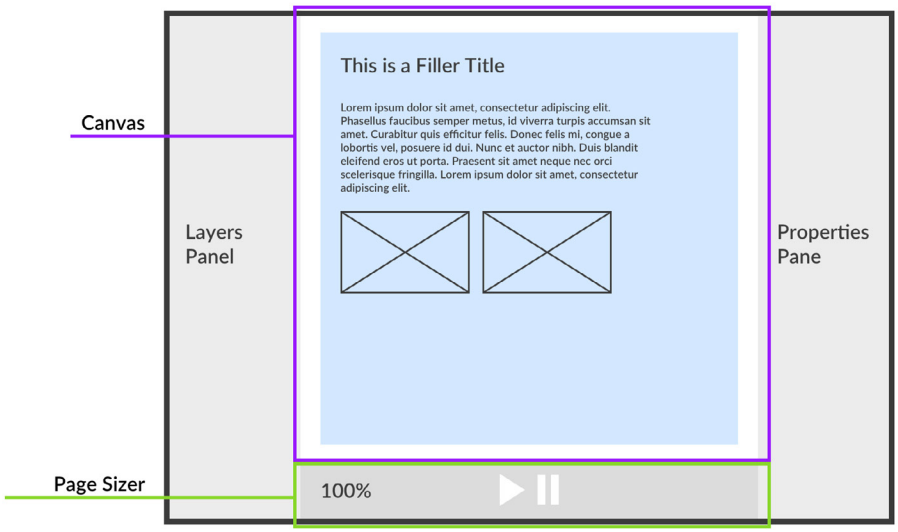
This prototype called "Autosize" builds on the previous prototype, "Breakpoint", as it puts the screen resizing function on the tool rather than the designer expanding on expected screen sizes users will be viewing the experience on. Meaning most designers may only design for mobile, tablet and desktop but in this case since the system is adjusting the screen size many more screen sizes are created. This capability avoids several design flaws when testing across a small number of screens or devices.



Later on in my research process I was informed that Axure RP actually has a very similar feature to my "Breakpoint" prototype that they call "Adaptive Views". "An adaptive view is a version of a page designed for a specific screen size. Changes to a widget or page property can affect one view, several views, or all of a page's views." ("Adaptive Views.") Feeding into this idea that designers should match the conditions of the final environment as best as possible early on in the process. Also supported by the following "In a responsive design process, the old black and white PDF can quickly become a cumbersome, misleading, throw-away deliverable. Which is why creating wireframes in HTML and CSS can be a great move for you and the project stakeholders." (Griffin)

The illustration below showcases the key components of the "Autosize" prototype interface. "Autosize" uses several conventions from existing screen design tools like Sketch starting with the layers panel, canvas and the properties pane. The uniqueness of this prototype involves the use of the canvas and the "Page Sizer". The "Page Sizer" contains two key functions; a view size percentage counter and a play / pause function.

When the tool in playing, screen width adjusts automatically, exposing the designer to potentially more devices then they would otherwise design for, as in the previous image between step one and two. As “Autosize” continues to adjust the size some content may start to get cut off, which is where the designer would pause and adjust the content accordingly to that screen size proceeding to steps three and four as illustrated in the “Autosize” flow image above. The same workflow follows for steps five and six in this exercise.



The issue with both of my prototypes is that they only take into consideration screen based interfaces and only screen size rather than more environmental considerations such as touch vs mouse and keyboard. As these prototypes do not take into consideration non-screen based interfaces such as voice, I went back to the drawing board and revisited a typical design process to guide my next steps. Interaction flows as being a good place to start as they are clear in terms of what the user should be doing but are still abstract because they leave out the platform they work on.

To investigate interaction flows more, I set up a workshop. The goal of the workshop was to address one of my big questions from my background research, what is the digital equivalent of the flexible rope which can work in several different contexts? Looking at how designers consider designing an experience agnostic of a platform, I started the process by setting the scene for the participants, sharing the assumptions I will be making and defining terminology I will be using. I have planned several activities to help the participants think about Fluid Design and working in this platform agnostic way. The first set of activities were to create interaction flows of a movie ticket purchasing experience first for mobile, second assuming a voice platform and lastly for a Fluid experience, each

lasting ten minutes. These interactions flows were done with sharpie and post-its with some pre-defined parameters. The movie ticket purchase experience would need to capture a city, theater, movie and movie time and could optionally capture assigned seating information, the movie type (IMAX / 3D / standard), number of tickets, type of tickets (Adult, Child, Senior Student), payment information and membership login. Following these interaction flows wireframes were drawn up from the third interaction flow, the Fluid experience, thinking about how that would propagate to both a mobile and voice platform. This activity also lasted ten minutes and was carried out with sharpie and paper. Following the wireframing activity, they were tested amongst the group to ascertain if the Fluid interaction flow actually could be propagated to several platforms or if it failed to capture a certain part of the experience. Asking questions such as how clear is the flow? Is there error handling when propagated to a voice platform? What is missing from these wireframes? How was this process of designing? What did you discover through this process?

My Process

From the answers and insights collected during the workshop I will implement the flexible material by designing several common experiences such as ridesharing, food ordering and movie purchasing and evaluating them using the ten usability heuristics for interface design by Nielsen. These designs will be evaluated by several third party evaluators who are considered either an expert or novice based on the recommendation of Nielsen. (Nielsen, Jakob, and Rolf Molich. “Heuristic Evaluation of User Interfaces.”) The heuristics are as follows:

“*Visibility of system status:*
The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

Match between system and the real world:
The system should speak the users’ language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

User control and freedom:
Users often choose system functions by mistake and will need a clearly marked “emergency exit” to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

Consistency and standards:
Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

Error prevention:
Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action. (Read full article on preventing user errors.)

Recognition rather than recall:
Minimize the user’s memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate. (Read full article on recognition vs. recall in UX.)

Flexibility and efficiency of use:
Accelerators — unseen by the novice user — may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

Aesthetic and minimalist design:
Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

Help users recognize, diagnose, and recover from errors:
Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

Help and documentation:
Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large."

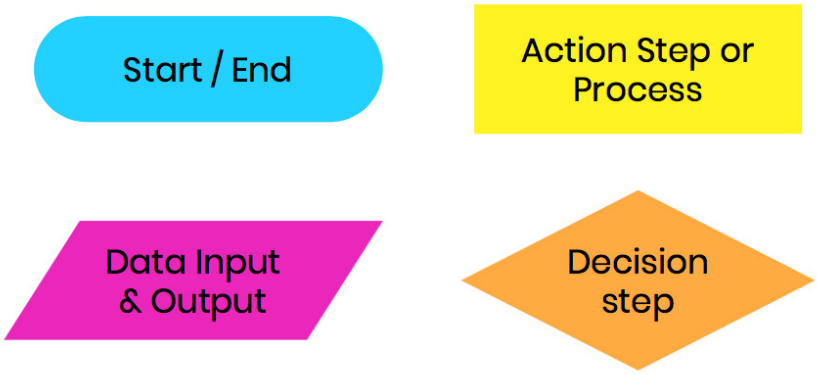
Evaluation

This section covers the outcomes and insights gained from different stages of my thesis; the workshop, designing using the Fluid material for different experiences and the proposed Fluid Design tool.

Workshop

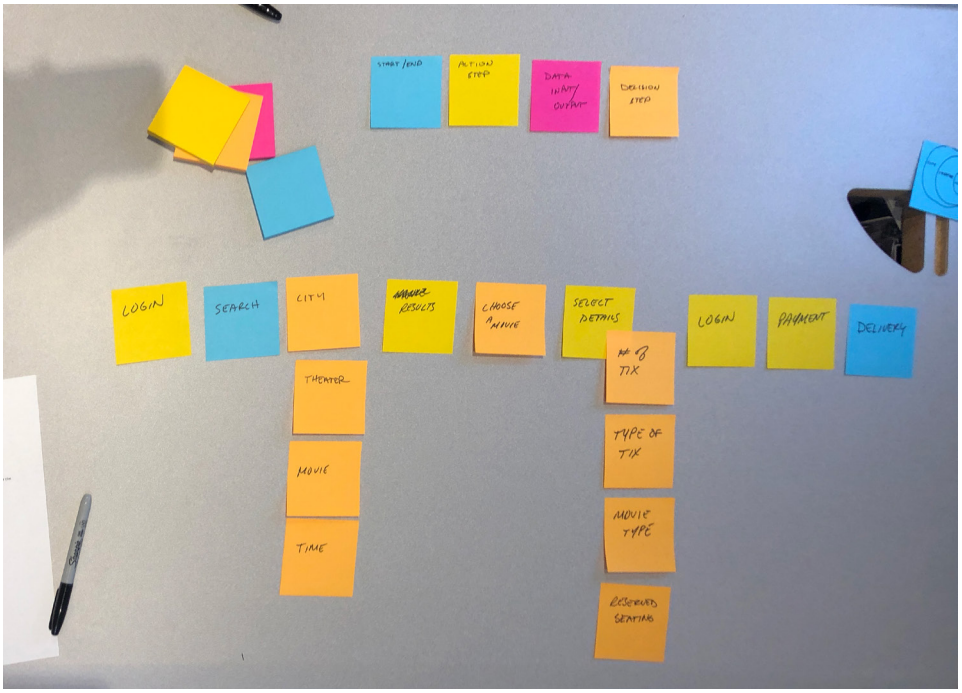
In this section I describe the outcomes from each phase of the workshop. We began by reviewing my assumptions and some of the background materials. The next step involved developing flows for different platforms. Throughout this process the benefits and drawbacks were discussed and outlined.

The first phase of the workshop involved sharing my assumptions around Fluid Design and some of the background material to set the stage for active discussion. Some of the outcomes of this discussion were; that responsive design explores the “use of a design pattern or system where the medium is less important” (Abele) and the idea of “diversity of platform and generality of the experience”. (Zepeda) Design pattern was defined by one of the participants as, reusable components or elements of the design that could be deployed across several actions, generally tied to the style guide or visual language for the experience. This also relates to the second comment regarding platform diversity and generality of the experience, as sharing components of an experience leads to design patterns and libraries, that allow for a scaled capability across many devices.

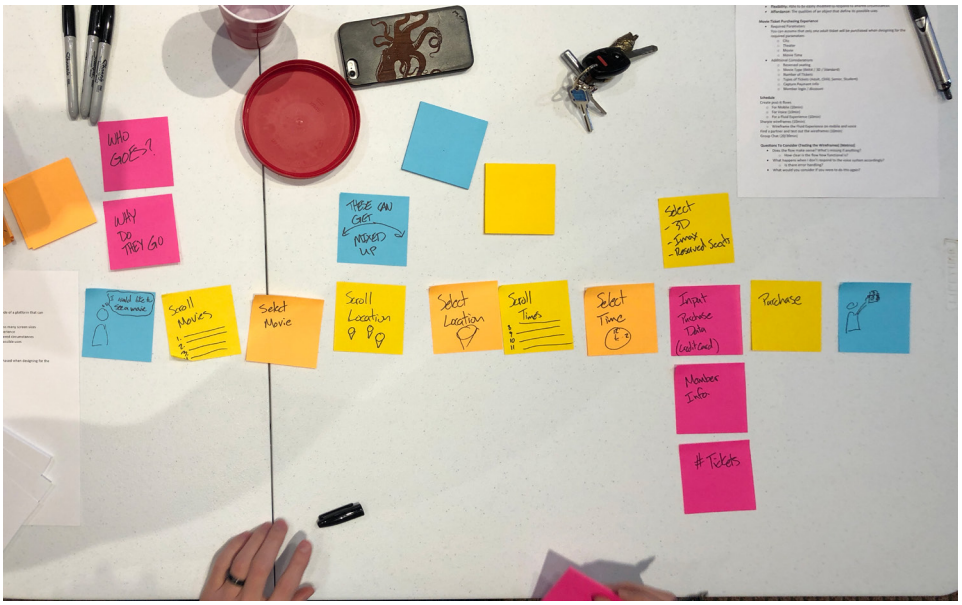


After the discussion involving my assumptions and my point of view, we started creating post-it flows around a movie ticket

buying experience beginning with a flow for a mobile platform. This exercise was pretty straightforward for the participants except one aspect of the flow. I had previously defined colors that represented different functions in the process such as orange color representing a decision, whereas a pink post-it would signify data input or output. This level of separation of an interaction in this low-fidelity setting did not match the mental models for the participants and resulted in distracting them from creating a clearly laid out flow. As a result the color designation was abandoned, “creating groups or categories for the functions is far more important than the color labeling” (Abele)



On the left are two of the post-it flows that were created for the mobile experience during this exercise. The first image shows a grouping of actions and starts to unveil that a particular sequence in completing a task is not necessarily important, as logging in at the beginning or towards the end of the experience are both acceptable, shown by the duplicate login post-its.



The second flow speaks more to specifically being on a screen device as it mentions actions such as scrolling. The second flow also starts by asking questions like why the user might be wanting to buy a movie ticket: “Why do they go[?]”. This indicates how the designer is thinking through this process of creating a flow for a given experience. Interestingly enough both flows start to

group actions into vertical stacks where these actions are all related to an overall action such as “select details” containing the elements: “number of tix [tickets]”, “type of tix [tickets]”, “movie type” and “reserved seating”.

After completing the mobile post-it flows we created the flows for a voice platform. This lead to a conversation about the affordance of a voice platform. “I am more likely to buy when using voice and browse on my phone.” As was mentioned having a browsing experience by voice was not ideal and it would be better suited for a screen interaction, where simpler requests work best on voice using existing platforms. “Payment on voice would be strange for a first time [purchase]” This quote speaks to the difficulty of trying to provide more sophisticated information on voice, such as entering payment information that captures credit card number, name and expiration date as well as the shipping and billing address and possibly discount codes and shipping options, and how it is not a great experience for the user. Which resulted in one participant making a comment around handing off the experience between voice and mobile: “why don’t I use both platforms together, voice and mobile [used together] could be a great experience?” (Zepeda)

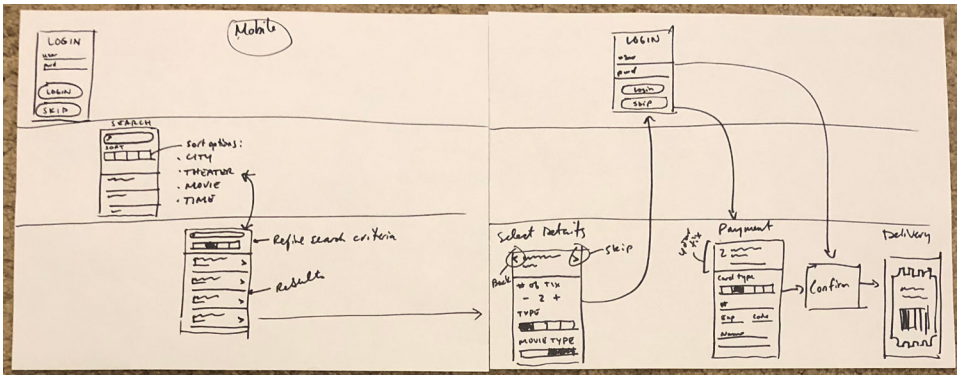
During our review of voice interfaces, there was a discussion around whether to expect a complex request up front or to scaffold it for the user. “If I feed it [the voice assistant] more info it should work, but not all current systems are equipped to do so, it should take all the information at once or scaffold it so the user knows what to give.” It was concluded that the existing systems don’t handle complex requests well today, but it would not be something out of scope as technology advancements are introduced in the coming years.

Our third activity involved participants creating a post-it flow for a Fluid experience, thinking of designing a flow independent of a platform. This initiated conversation about the overlap and differences between voice and screen-based interfaces. “You can bounce [around] a bit more between information on mobile [than with voice]” (Abele) Reflecting on this quote the group said that the user has to retain all the information necessary for an experience when using it with a voice platform, making it somewhat restrictive. This statement is also supported by one of the usability heuristics of Nielsen, recognition rather than recall, where the experience should “minimize the user’s memory load by making objects,

actions and options visible." This is a far more difficult challenge when using voice especially for content rich experiences.

Another overlap discussed was that "voice is the same [as a mobile interface], but in the form of questions". I would add that this holds true at the level of fidelity we were working, post-it flows. These platforms do start to diverge at a higher level of fidelity as was mentioned in the previous quote about "bouncing" around more easily on screen. They also diverge in part due to voice's more linear narrative, "voice doesn't have the same level of hierarchy that mobile does, it is more of a linear experience."

Following the post-it flow exercises participants created wireframes for mobile and voice from their Fluid post-it flows. One of the outcomes is included in the following graphic.

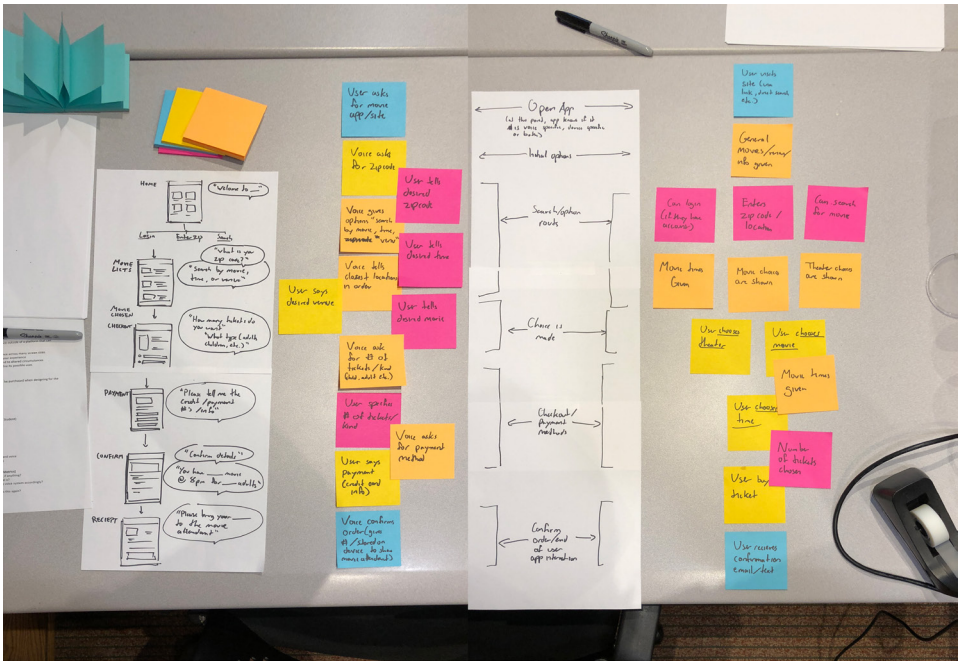


Depth Wireframe

This mobile wireframe showcases "depth" as an important aspect of a user's flow. Where for some actions the order in which they are carried out is irrelevant, such as logging in. The user could log in at the beginning of the experience or prior to paying, as indicated in the image above. The relevant part being that each of these actions are necessary for the experience. The aspect of sequence being non-critical as long as each action is completed, it leads to free standing action blocks in the final tool and shows an expansion of the stacks of actions previously described. On voice this order agnostic idea could carry through by the level of detail that a user passes along to a voice application "search by voice criteria, 'I want 2 tickets for x movie at 10pm'- which jumps you down and up an experience" combining certain actions into one command.

As we wrapped up the workshop the participants reflected on the process. The main objective being to have the participants answer what is the digital equivalent of the flexible material. One of the insights around creating interaction flows was "if you can get to a level when you can group functions, that's magic!" If the designer can be clear enough to

categorize actions it will greatly aid in next steps as you increase the fidelity of the experience.



Another participant asked “how do you wireframe [a] voice [experience]?”, leading me to think through what a simple voice flow tool set would look like when I implement the Fluid Design process, that could easily be used for Wizard-of-Oz testing. One of the last comments made was regarding a multi-platform design “more seamless is the way to go!” being able to handoff an experience from one platform to another using each platform for what it can do best. The conversation concluded with the group addressing what they thought the Fluid Material should be; a core action, an action that must exist on each platform in order to complete the experience.

Fluid Experiences

In this section I discuss why I choice the three experience I did. The Fluid Design process that emerged, while designing these experiences, how they were viewed through user testing and the takeaways that would influence the design process of the Fluid Design tool.

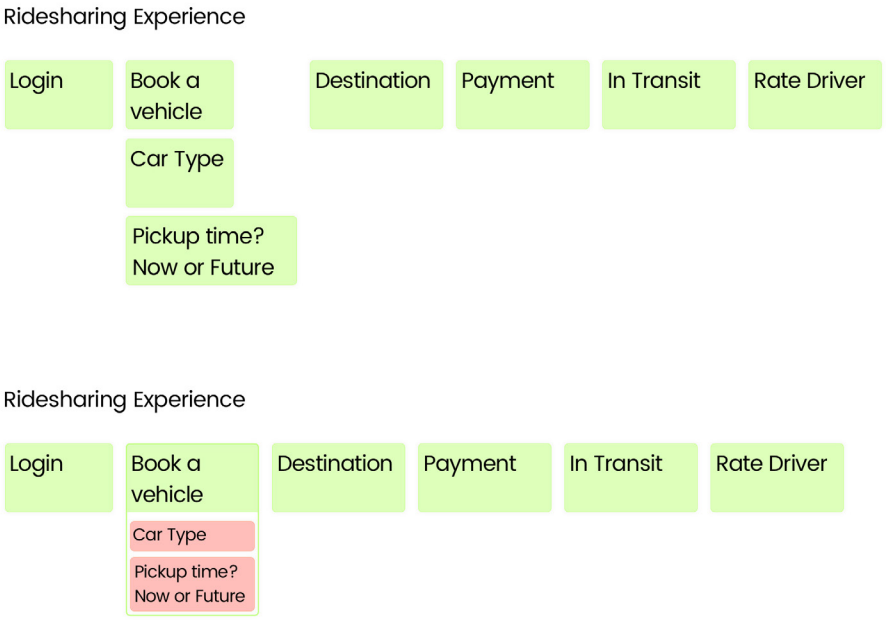
Designed Scenarios

Once I had completed the workshop and captured the insights, the next step was to implement the Fluid Material and the process of designing an interaction flow that is propagated to several platforms, in this case a mobile and voice platform. As the Fluid Material was seen as a core action, I wanted to make sure to choose three distinct experiences that

would evaluate whether or not this process of designing works. I chose to design a rideshare experience, movie ticket purchasing experience and a pizza ordering experience. Each of these experiences were chosen to create a range of constraints on the Fluid Design process. The ridesharing experience tests how the Fluid Material acts in a mostly system driven experience with minimal user input as compared to the other experiences. The movie ticket experience was chosen to test a more baseline number of options and interactions, while the pizza ordering experience lives on the opposite side, having an overwhelming number of options and input parameters. The idea with all of these experiences was to test a design, after it was propagated to their respective platform, to evaluate if it still made sense and was a usable and practical experience.

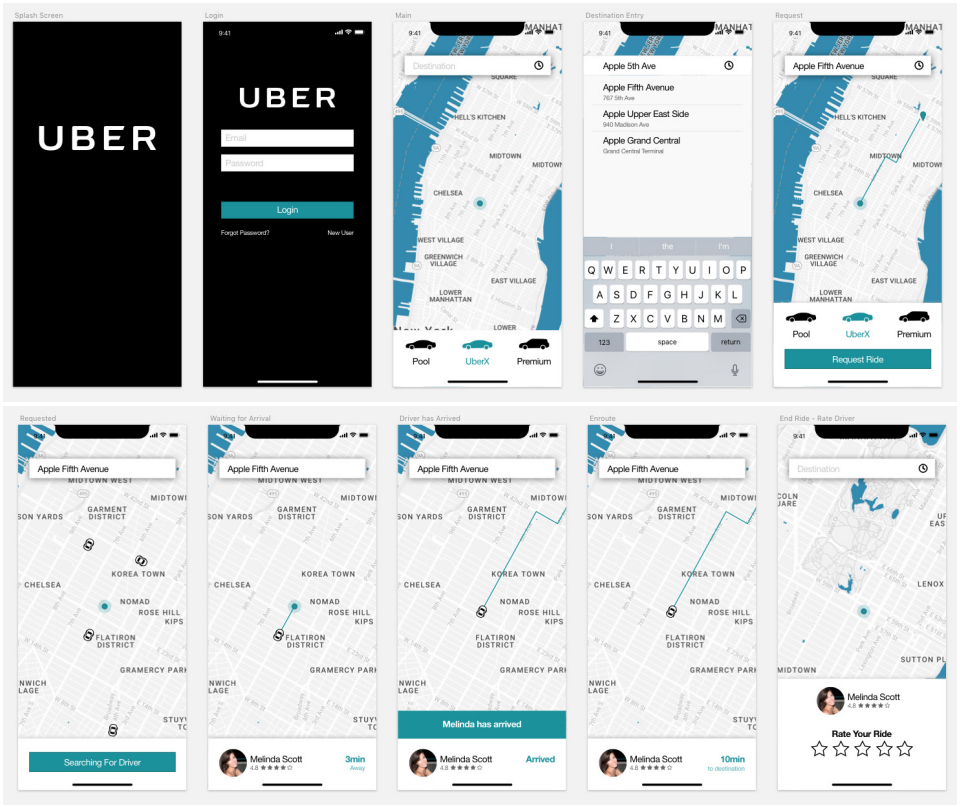
Fluid Design Process

I started by first creating an interaction flow of the experience. Writing down all the core actions followed by grouping them into stacks as shown below.

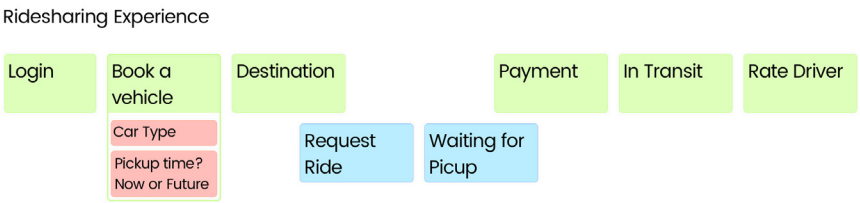


Once I was satisfied with core actions, I started to propagate them to a mobile interface and a voice interface. The mobile design process was simpler as I, like many others, am familiar with designing for screens using conventions implemented by design tools such as symbols and layer styles to have a cohesive visual language. I realized while working on the mobile propagation for the ridesharing experience that I had missed two core actions and went back to add them in, “requesting the ride” and “waiting for pickup” (show in blue below). In this process I accidently tested another aspect of Fluid Design, switching between

different levels of abstractions throughout the design process, leading to an important feature of the final tool, the ease of transitioning between design phases.



When it came to voice, this was more challenging as a common pre-defined process integrated into a design tool does not exist. In order to create a voice interface design process, I looked to chatbot conversation flow diagrams such as the following one by Arctouch as well as the Amazon Alexa Voice Design Guide. The Arctouch conversation flow was a good starting point, but as it was designed for a chat experience and some of the aspects would not translate, such as a carousel, image or an emoji. From the Amazon Alexa Voice Design Guide I captured the following:



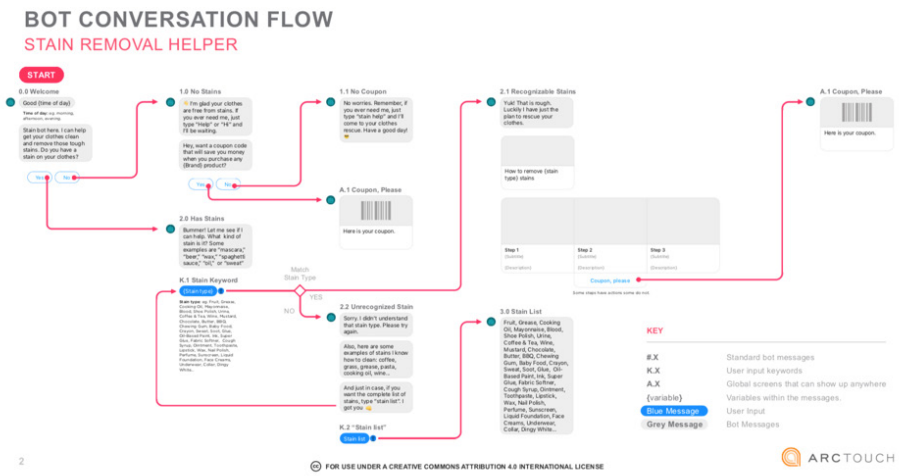
"Outline the shortest route to completion:
The shortest route to completion is generally when the user gives all information and slots at once, an account is already linked if relevant, and other prerequisites are satisfied in a single invocation of the skill."

"Outline alternate paths and decision trees:
Often, what the user says doesn't include all information necessary to complete the request. In the flow, identify alternate pathways and user decisions."

“Outline behind-the-scenes decisions the system logic will have to make:
Identify behind-the-scenes system decisions, for example with new or returning users. A background system check might change the flow a user follows.” (“Design Process.”)

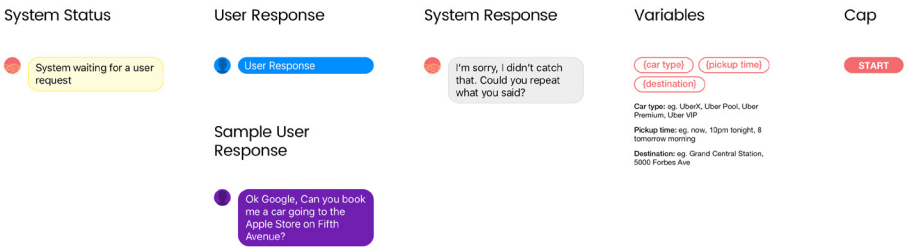
I later devised a voice flow that captures system status, user responses, system responses, variables, options, direction of flow as well as the start and end of an experience or a cap. While testing the tool later on I realize that this diagram failed to convey several other voice user interface elements such as tone and accent.

I continued with the design for the other two experiences getting more and more comfortable with the process myself. After completing the propagation for each of these three experiences to both mobile and voice platforms, I tested the experiences on both platforms with several participants. The participants were asked to evaluate the experiences using the ten usability heuristics of Nielsen. The tests included a clickable prototype for the mobile version and I acted as the voice agent while testing



ArcTouch Bot Conversation Flow (Hall)

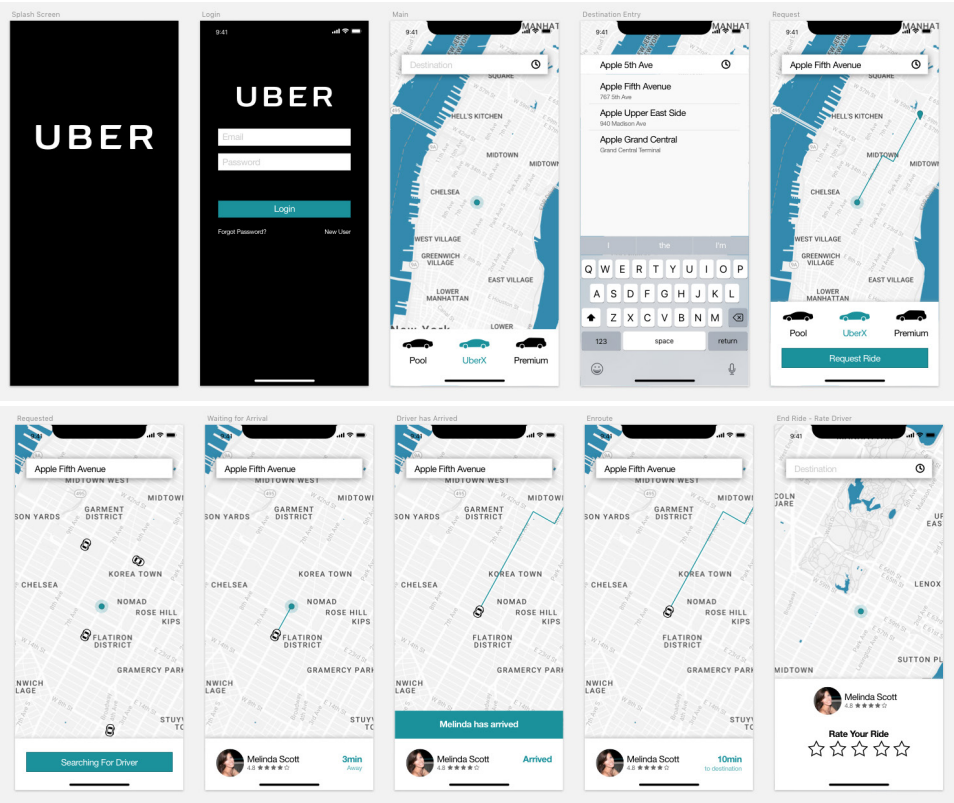
Ridesharing voice flow



Original voice flow elements

the voice interaction. Overall the feedback was consistent across the platforms which indicated that the Fluid Process had worked, but that I may have missed certain interactions when designing.

User feedback



"The interface is simple and clear" This participant commented on the aesthetic and minimalist design heuristic also noting that for most of the actions appropriate information was displayed with the exception of displaying a price during the request step. This participant also wanted to see their account information and profile, in order to represent a fuller experience outside of ordering a car as an existing customer, which was the defined task for this exercise. "The application functions in a logical order and provides good feedback" (O'Toole), referring to the Match between system and the real-world heuristic, where "information [should] appear in a natural and logical order." Clear feedback also refers to the visibility of system status heuristic. Another participant said "it is pretty straightforward and in line with my expectations" while she also mentioned that "an exit or escape once the ride has been ordered needs to be added" referring to the heuristic on user control and freedom, "users often choose system functions by mistake and will need a clearly marked 'emergency-exit'". (Nielsen, Jabok. "10 Heuristics for User Interface Design.")

When evaluating the same ridesharing experience on voice, some of the comments made by the users are summarized in this paragraph. One

of the first reactions was that “speaking is far more natural” matching the system with the real world. (Nielsen, Jabok. “10 Heuristics for User Interface Design.”) This same participant wanted to know if they could request a status update “Can I ask ‘where is my uber?’” which did not exist in the flow shown above, causing the visibility of system status to lack to a satisfactory level. Another tester also mentioned a similar request wanting the system to confirm that a driver was being located rather than just enroute. The last system confirmation missing that was noted was for “what vehicle I should be looking for?” (Connolly)

“There is a good level of error prevention since the system is confirming requests” such as with the system response of “would you like me to request your ride to {destination}?”

Another participant noted “what if there are no cars or surge pricing, will it confirm the increase in price?” This was a design error on my part design for a more narrowed scope then users expected as well as I did not take it into consideration for this experience on mobile platform either.

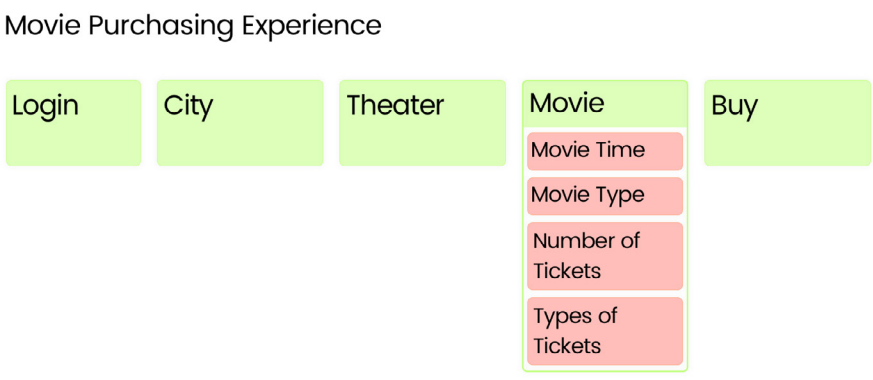
Ridesharing voice flow

The most interesting response from the voice test was “can I check the status on my phone after the request from voice?” (Naidu) Which follows from a comment made during the workshop about being able to handoff the experience between platforms. The idea of handoff was definitely a feature that needed to be incorporated in my tool.

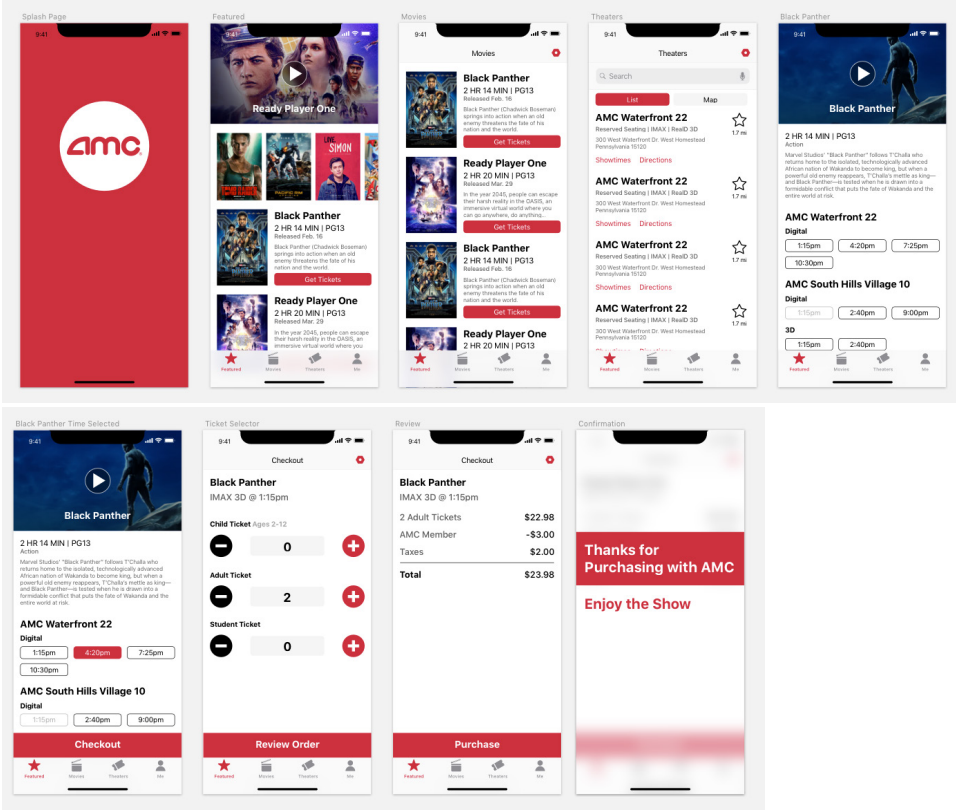
The Ridesharing experience tested something that is mostly system driven with little input from the user. Designing this experience was easier for on screen inputs vs the voice platform due to my familiarity with screen based design. The feedback was overall positive, a clean interface that was easy to use, matching the user expectations. There

were several interactions where users wanted more details, such as the ability to see surge pricing and profile information. All of this feedback could be addressed in further iterations.

For the movie ordering experience much of the comments were quite similar to the ridesharing experience: a clean interface, easy to use and understand, consistent design language. Many of the negatives shared regarding this design were related to visibility of system status and recognition rather than recall. Such as “the prices should be shown on the ticket selector page”, “the theater choice should be shown when selecting the tickets” (O’Toole) reducing the memory required on preceding steps or “what card is being charged?” should be displayed on the checkout page. Lastly in the confirmation page several participants wanted to know where the tickets went, whether or not they were emailed to them or available to view in app. All of these comments identified design aspects that I as a designer missed to incorporate in the experience.



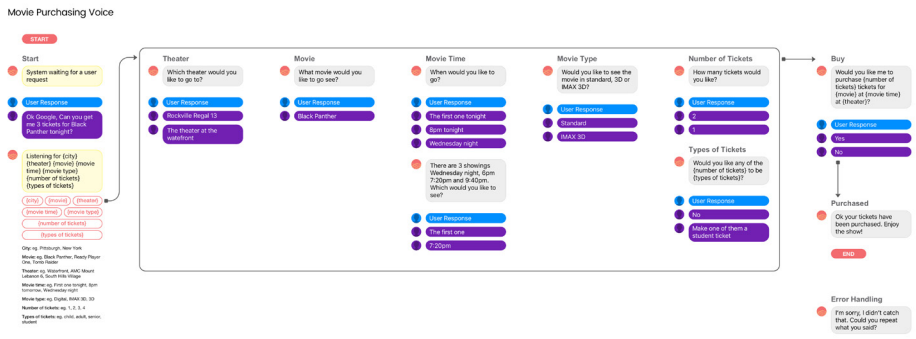
Movie purchasing interaction flow



Movie purchasing mobile experience

For the voice experience, most of the feedback was identical to that of the mobile interface. Participants wanted to know which card was being charged, how much the tickets were, and where the tickets would be accessible, whether email or some other way. The symmetry of responses for both the mobile and

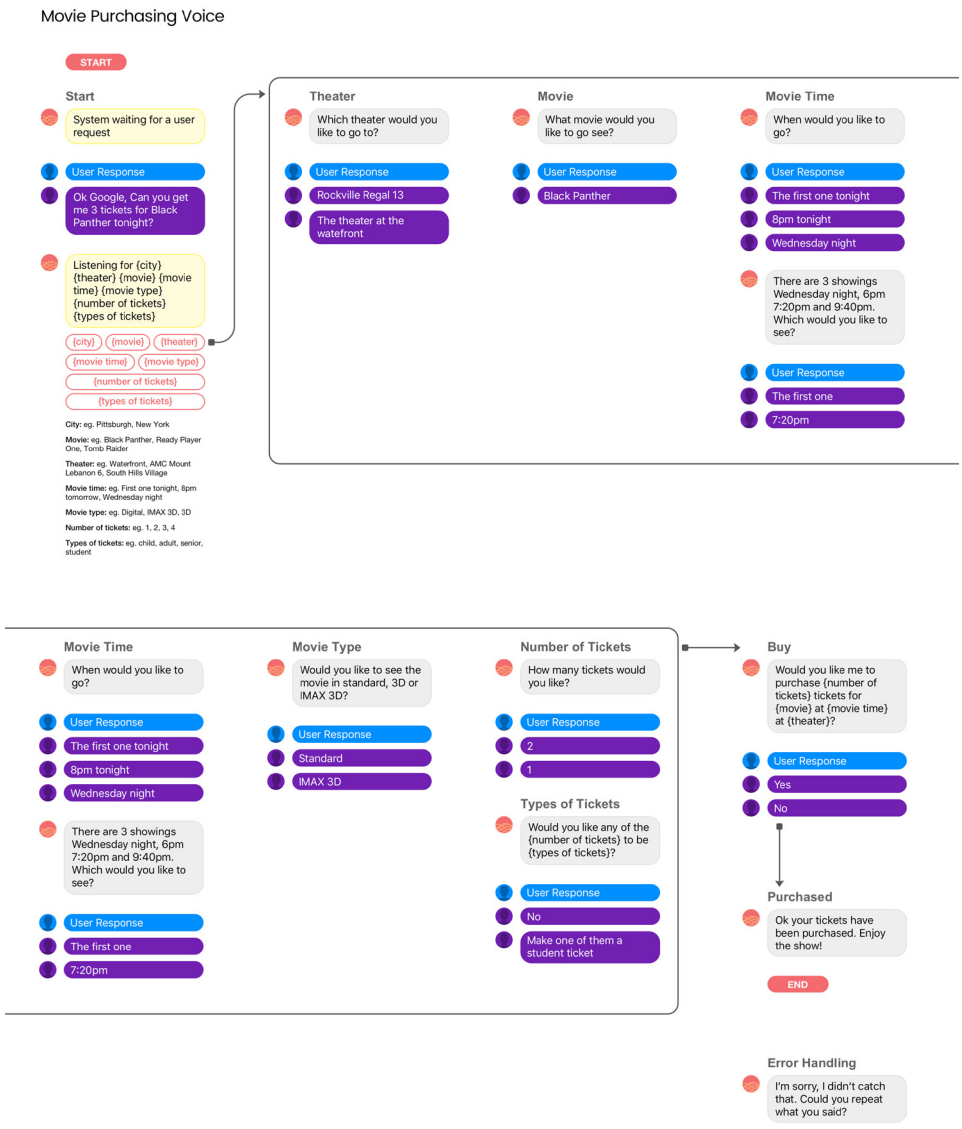
voice experience indicated that the Fluid Design process indeed worked, but I as the designer of these experiences missed some interactions that users would want. This could be ironed out by revising the interaction flows of each experience and subsequently updating the designs for each platform.



Movie purchasing voice flow

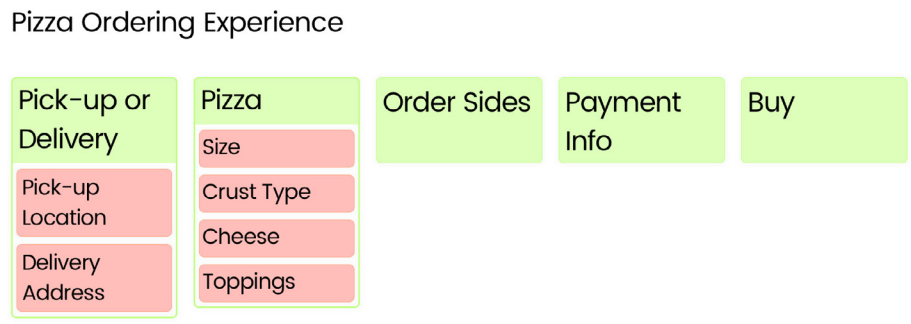
The designs created for the movie ticket purchasing experience though similar to that of the ridesharing experience pointed some of the specific elements that need to be addressed both in on screen as well voice platforms. The items included; clarity of location, payment method and where the user would receive the tickets once purchased.

Moving on to the last of the three experiences tested, was the pizza ordering experience. Following are some of the points the testers mentioned when speaking about the mobile version. “Not clear, the cart page looks too much like an order item” (Connolly) referencing the consistency and standards heuristics, where distinct functions should look distinct not to confuse users, in this case between reviewing an order and selecting an item.



Movie purchasing voice flow zoomed in

Similarly, an order confirmation feedback interaction would need to be added incorporating the following: “it would be nice to have an ETA [estimated time of arrival] for delivery or pickup time”. And finally, the users mentioned that they wanted a simple way to get back and forth to the cart: “how can I get to and from the cart in order to add more?” (Naidu)



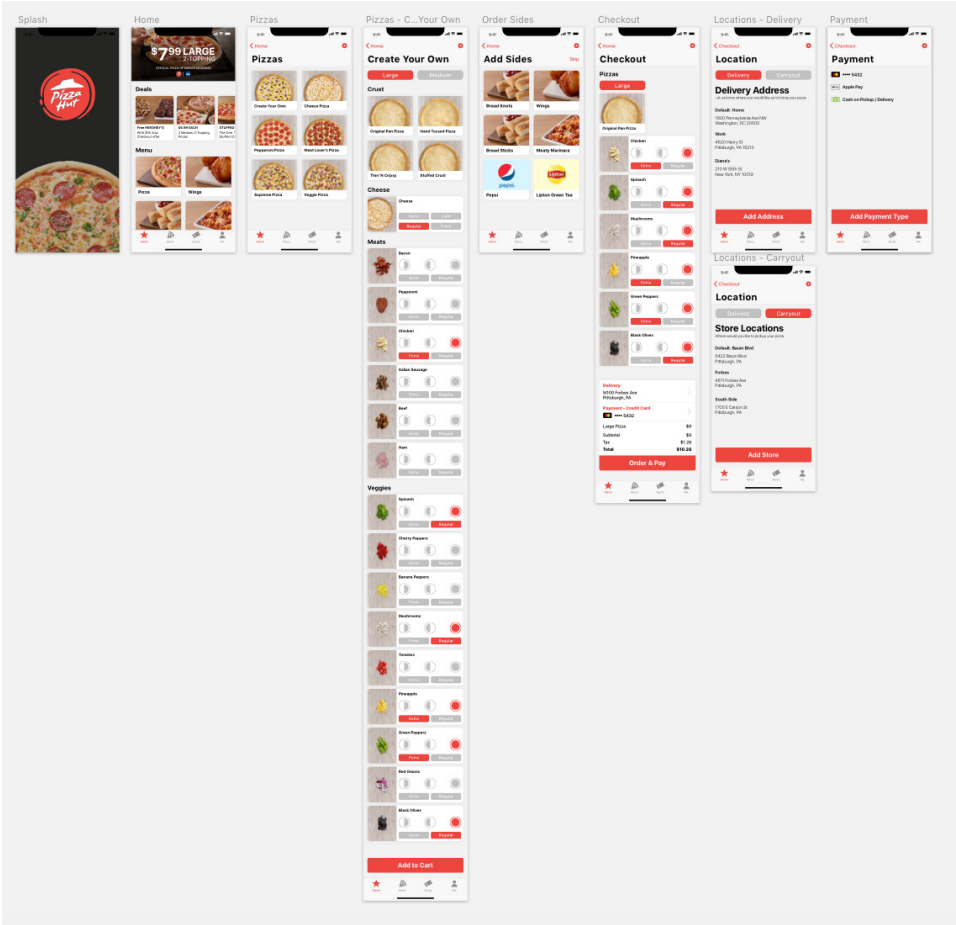
Pizza ordering interaction flow

The pizza ordering experience for voice brought up many of the current limitations with using a voice only experience.

“I don’t think I would order a pizza like this”

“much slower than on the phone unless it was to repeat order, the process was too long”

“there would be too many options for me to remember if the system listed them all out, would be better to order the ‘usual’”



Pizza ordering mobile interface

All three of the quotes above bring up that voice is not ideal for a new pizza order task highlighting the issue of recognition rather than recall and flexibility and efficiency of use heuristics.

The pizza ordering experience tested an option rich interaction which became easier on a mobile platform vs voice. Even though improvements had to be incorporated for the mobile platform we start to see a

divergence of optimal usages between these platform as voice was deemed slow and not ideal for this experience.

Much of the feedback indicates that there should be more stacks for actions in the interaction flow to address many of the issues that were brought up. As these issues revolved around some of the secondary actions. And that it would be easy to go back and forth after collecting feedback to update the interaction flow which in turn would lead to updated the experience on their respective platforms.

Voice as a standalone is better suited for stored user preferences such as a favorite pizza order. In order to design for voice in a way that satisfies the Nielsen usability heuristics a hybrid or handoff experience could be designed which allows the user to switch between platforms within an experience.

Pizza ordering voice flow

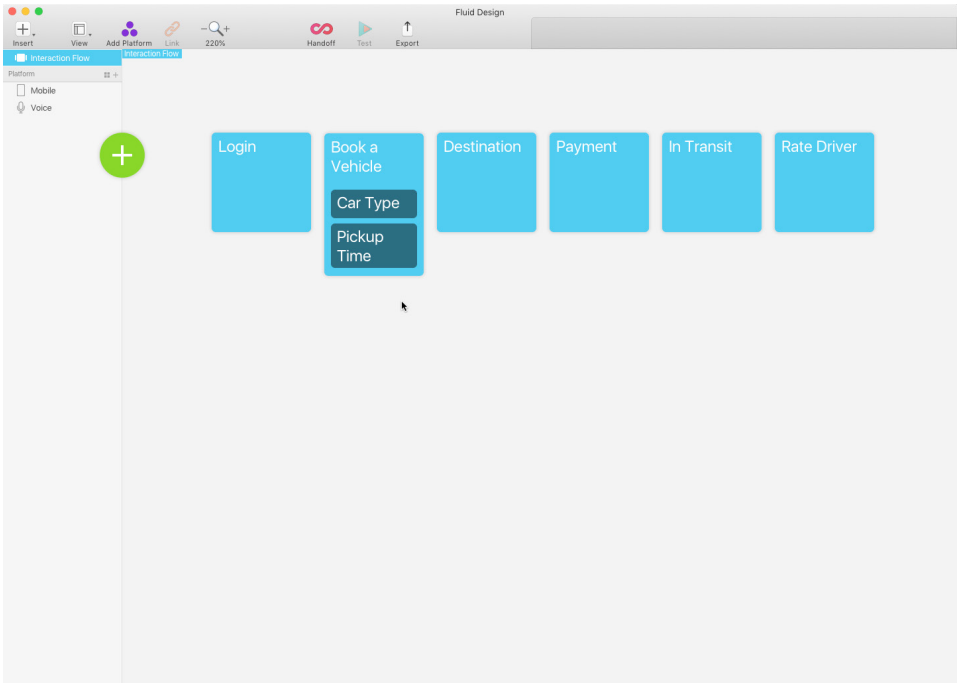
Pizza ordering voice flow zoomed in

Fluid Design Tool

Taking into account the feedback collected from the workshop, the Fluid Design process and the designed experiences, I have designed and tested a Fluid Design tool that would aid in thinking and working in a fluid, platform agnostic way. Part of designing the tool I used a participatory design method with two fellow designers. (Martin) The research surfaced several required features for the tool, to be included: high fidelity output across platforms, working between levels of abstraction seamlessly, simple design environment for each of the platforms and allow for a hybrid or handoff experience between platforms.

Sketching out the Fluid Design Tool

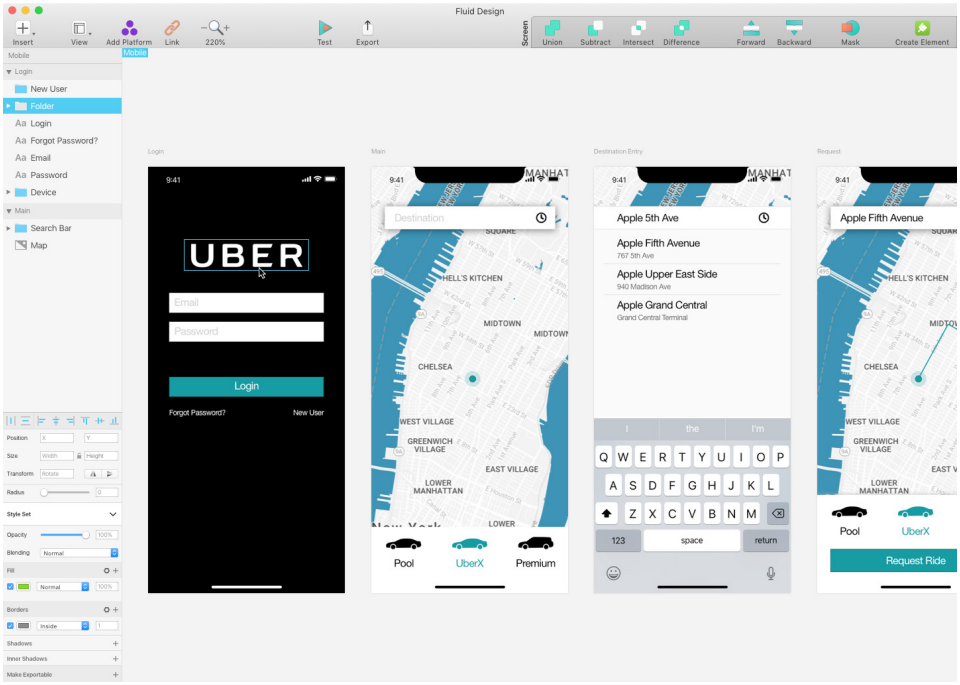
The idea behind the interaction flow interface was to keep it simple as the workshop participants experienced an unnecessary complex system for generating interaction flow, mapping certain colors to certain action types. Taking this into consideration as well as the comment around grouping actions being ideal for proceed into higher fidelity. The interface contains a add button, which adds an action block to the canvas for the user to type within



Interaction Flow Viewport

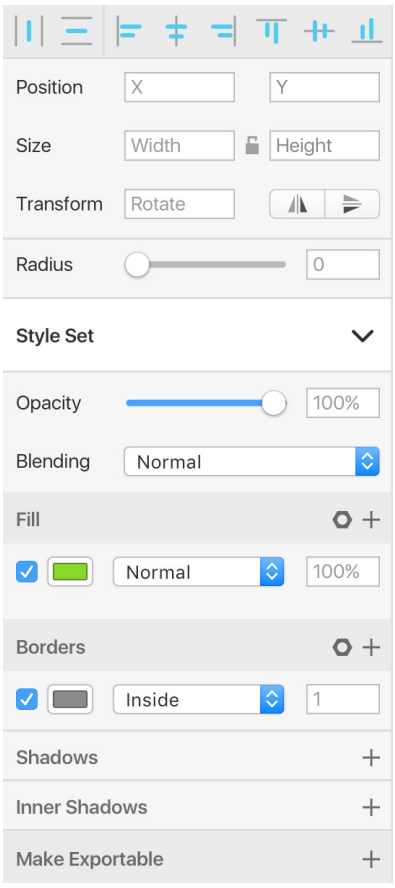
and either leave as a standalone block or group into a stack as with the “car type” and “pickup time” blocks.

What does high fidelity look like for screen and voice based designs? For screen designs I took many queues from existing tools such as Sketch and inVision Studio as these are the industry standards. Below is an image of the tool for screen designing. It encompasses a layers and properties pane on the left, a toolbar at the top with screen specific functions and the main canvas area.



Screen design interaction

Zooming in we can inspect the properties panel more closely. Starting from the top we have “alignment/distribute”functions, X/Y positioning, size, rotation and flipping functions, radius of corners, a “Style Set” allow users to configure and store styles to use across many objects, object opacity and blending mode, fill colors, border colors, shadows and inner shadows and lastly an export function.



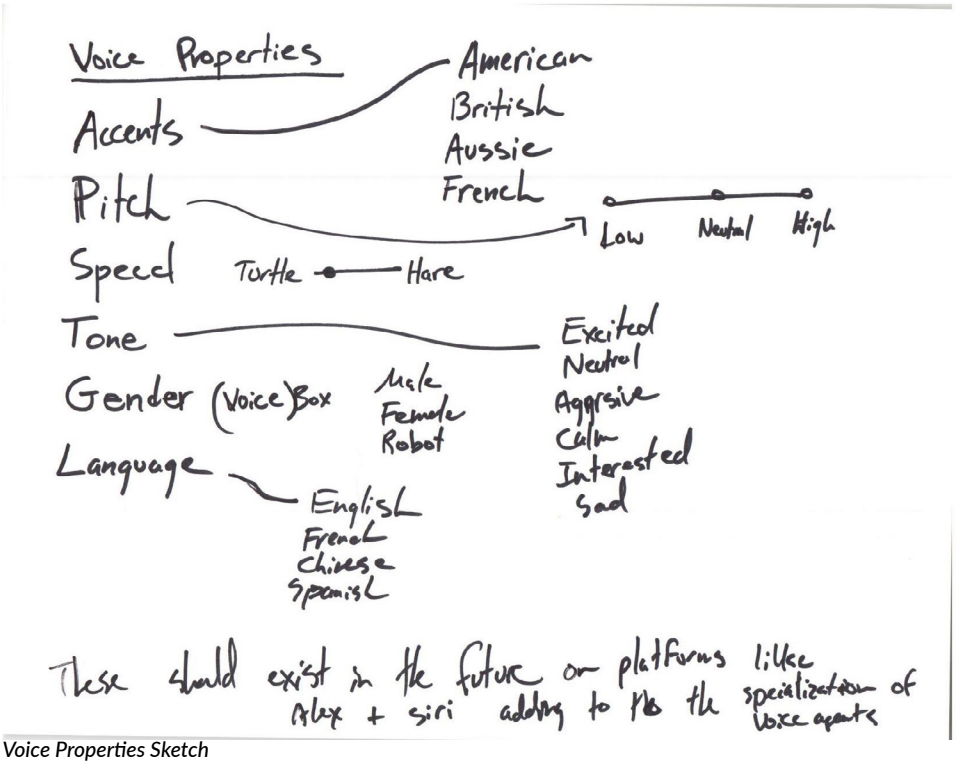
Screen design properties panel

It was more interesting thinking through what a high-fidelity design looked like for a voice UI. Current tools such as DialogFlow look at the flow of operations for a voice interaction but not

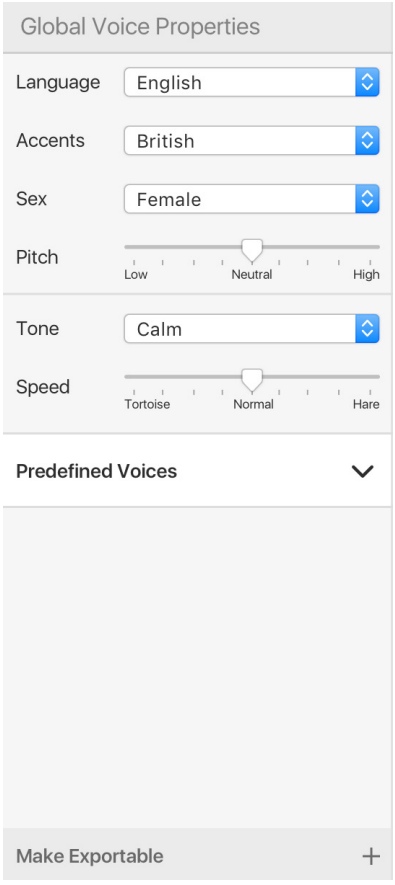
necessary allow for the styling of the voice as it is a voice prototyping tool. In order to come up with the appropriate properties for voice I used a co-design or participatory design method with a fellow designer as well as insights gathered from interviews. Landing on the following properties language, accent, sex, pitch, tone and speed of speech.

My co-designer was asking whether the category should be “sex” or “gender” which lead into a small investigation on how these words are defined and how they differ from one another. “sex tends to refer to biological differences, while gender refers to cultural or social ones.” (“Definition of Gender by Oxford Dictionaries.”)

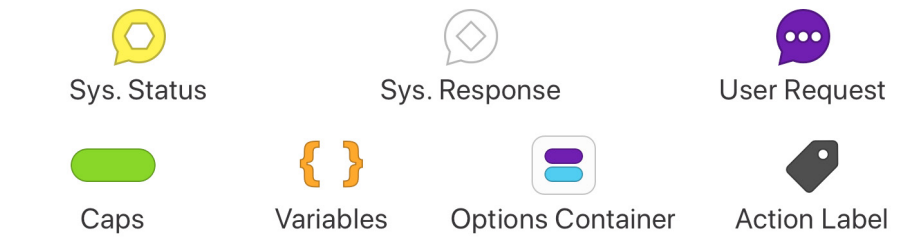
“These voice properties are very interesting. I would play with them all day, [it would be] so much fun” (Calzada) a user tester mentioned as he had not seen this level of detail or fidelity when designing for a voice tool and was really curious what all the possible options were in each of the dropdowns. He wanted to try out many of them and really see how these best fit the experience.



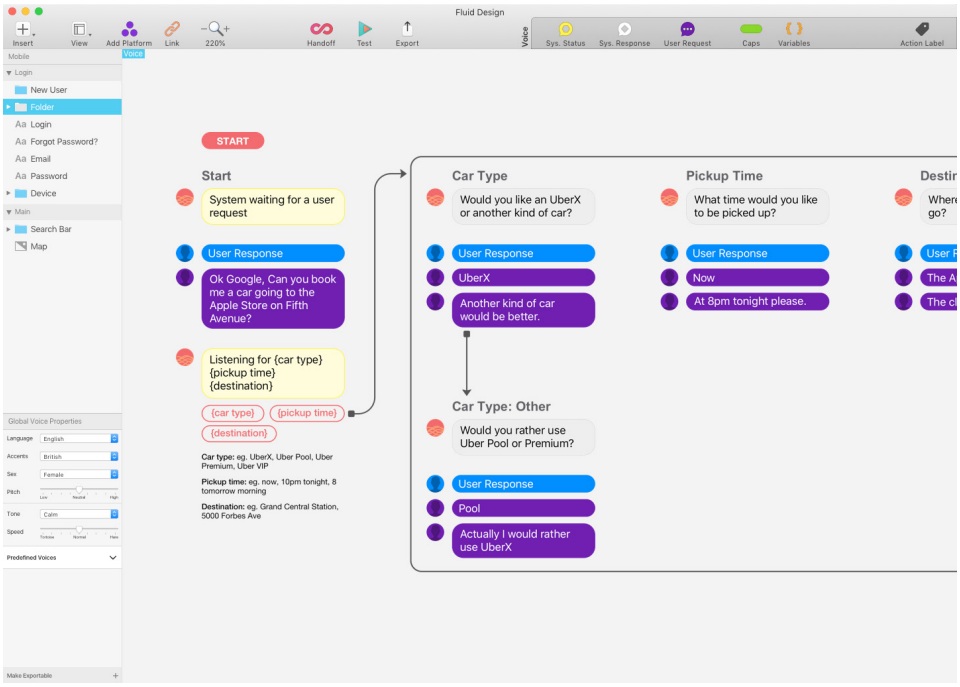
Voice Properties Sketch



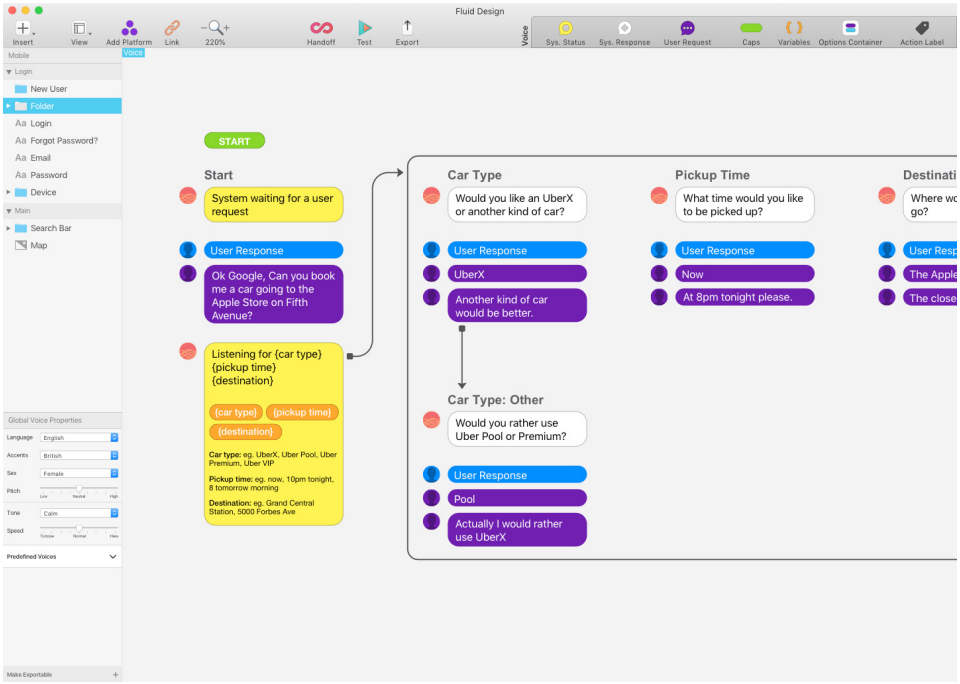
The previous design shown on the next page, was also the system I used when designing using the Fluid Design Process. It lacks clarity in addressing items such as whether only the “{pickup time}” variable linked to the next step or if it was all of the variables and how/if the variables were connected to the system status bubble shown above it. Color was expressed as a positive by some of the designers as it clearly separated different types of elements. Another participant noted that they liked having the ability to include several user examples at each step, in order to clarify the diversity of possible answers. The “options container” was also created as a separate element in order to match the workflow that one of the designers had mentioned. In order to keep on task of creating the entire experience, he wanted to create a container for all the options prior to thinking through the options. This will facilitate a process starting at the more central issues and actions in the experience then diving deeper to the secondary actions. All of these comments were taken into account leading to the revised interface show below.



Voice tool set



Voice UI previous design



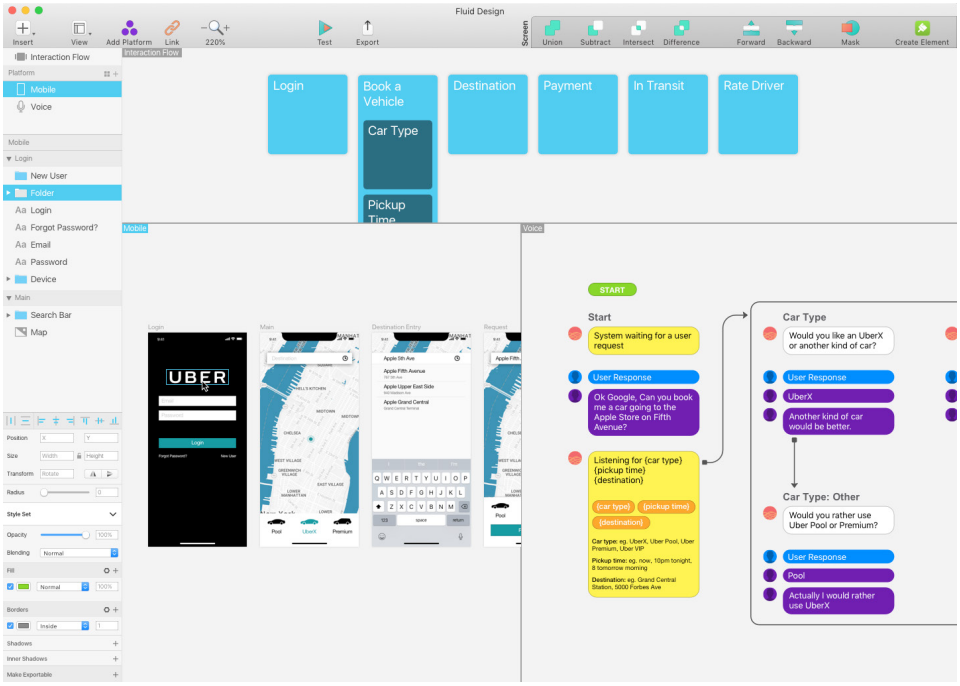
Voice UI current design

Working Between Platforms

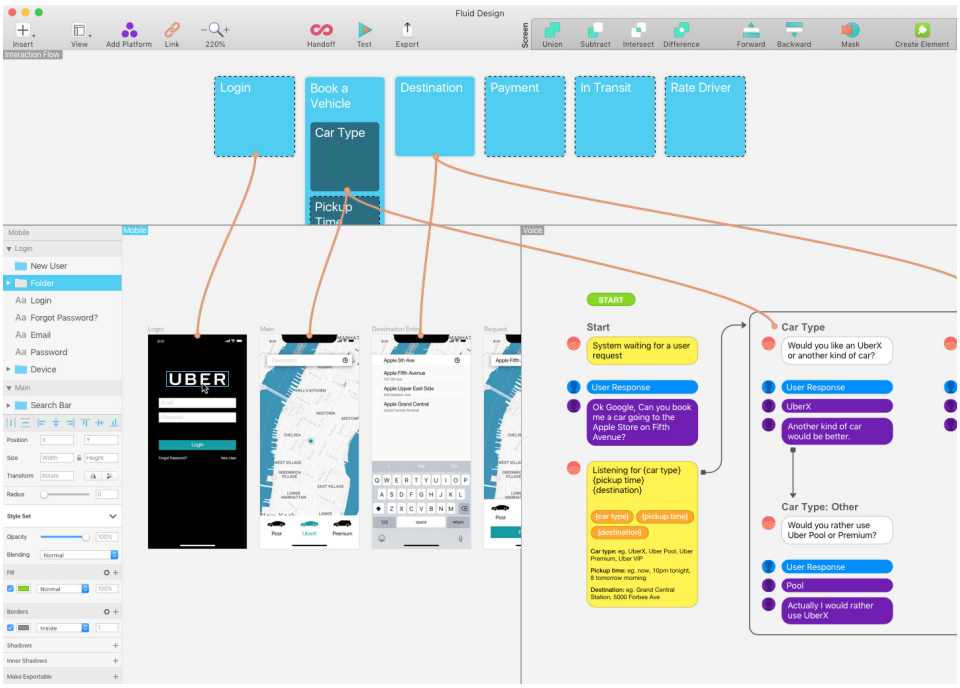
Working at different levels of abstraction and between different platforms was another critical experience that my tool had to facilitate. One participant mentioned to have some way to see everything in one view. In order to achieve this, I took a queue from Rhino3D's, a computer-aided design application, viewport interface. ("The Rhino Window.")

I also wanted to aid in linking these pieces together, different platforms and the interaction flow, to ensure that each one of the core actions were propagated. This was the outcome from the workshop that the core actions were required for the experience to be deemed completed. One user tester mentioned he was very happy to see that all these versions of an experience could be tied together in the tool and not just on an abstract level within the designers' heads. "This is all connected! Yes!"

"Handoff is very sophisticated and could be very powerful for the experience" Handoff is this idea that you can create a multi-platform experience, Such as logging into an rideshare application on your phone, then asking a voice assistant to call you are car giving it the destination and other request information and going back to



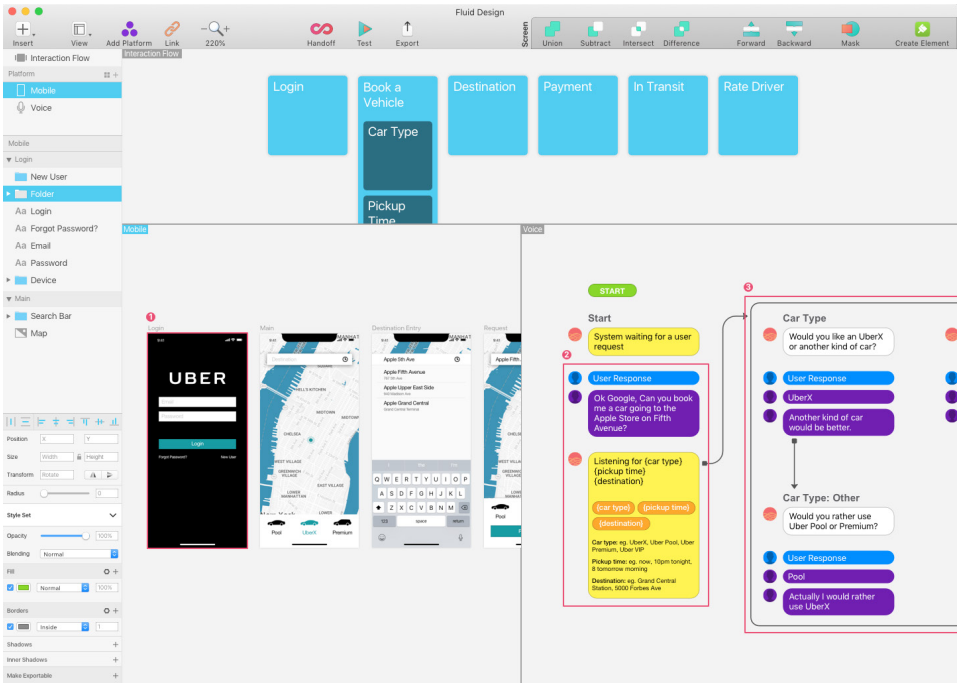
Main workspace of Fluid Design



Main workspace showcasing the linking feature

your phone checking the current location of the driver.

“I [as the user] want to know through some feedback that I’m engaged in this handoff experience on another platform, would help with error handling as well.” (Calzada) This was an interesting point that one of the user testers mentioned. He wanted to make sure that users were informed that they were engaged in a handoff experience with some kind of feedback, such as a modal or a notification on the mobile device informing the user this experience is also currently being controlled by voice or another device.



Main workspace showcasing the handoff feature

Thinking more about the handoff experience I explored Paper Signals a Google Voice Experiment. The idea of Paper Signals are “build-it-yourself objects that you control with your voice” (Paper Signals) Creating physical objects that react to some data point which is triggered by voice. Such as tracking if it is going to rain in your city. This is an interesting project as it limits the physical feedback to one dimension and ties in nicely with Nielsen’s recognition rather than recall heuristic. Allowing a user to collect a certain data point such as the weather from a simple visual.

Handoff experience diagram

During this stage of my research I began by conducting a workshop and shared by assumption and background materials. Flows were developed with the help of users to outline various tasks agnostic of platforms and observations and comments recorded. The next stage was to develop three experiences on two different platforms; mobile and voice. Throughout this development process flows were tested. Mobile experience was easier to conceptualize and develop as several existing tools are available. Voice platform was more difficult to develop and presented several issue that are unique to voice users. Even though there were adjustments made, conclusion derived from this exercise is that the Fluid Design indeed works and can be used to deliver experiences across different platforms.

Paper Signals weather (Paper Signals)

Conclusion

My research was to evaluate the current design tools in light of a growing number of distinct device interaction, such as on mobile and voice platforms. It further highlighted the need for a comprehensive design environment that not only takes into account the on-screen experiences but also the emerging voice platform and myriad of applications that this platform can offer. During this research I identified the challenges designing for different platforms not only from users' standpoint but also from the designer's lens. Identifying a clear need for an all-encompassing design tool, I proposed Fluid Design that will allow for platform agnostic designing. Fluid Design is a look at an expanded view on responsive design to account for non-screen-based interfaces and shows rich potential for a platform-agnostic high fidelity design tool.

During this research I compiled and reviewed background materials relating to responsive design; understanding evolution and applicability in the current environment. In order to involve and achieve optimum design results, collaboration tools and the limitations were discussed and evaluated. Designer feedback was collected through surveys and a workshop with clear understanding of desired features and optimum efficiency for a consistent user experience. Prototypes were developed across different experiences and different platforms testing out the Fluid Design process. This process further affirmed the need for a platform agnostic design tool.

Fluid Design, through its dedicated interaction flow tool, screen design tools and voice design tools, is helpful in designing across different platforms, but showed the limitations when designing in a platform agnostic fashion. User and designer feedback indicated that not all experiences should be propagated to their fullest extent on all platforms. Option rich experiences, for example, are not ideal for voice, but can be handled more effectively on a screen-based interface, leading to the idea of a handoff experience: experiences that will bridge multiple platforms. Having robust high-fidelity design tools for distinct platform types; screen and voice, will facilitate the rich handoff experiences. Outside of the tool it also exposes a process of creating robust low fidelity experiences and their transition to a higher fidelity output and the interactions that

occur through updating and propagating design decisions at one level of abstraction to another and back again. Fluid Design helps to address what the future of our design tools may look like.

Future Works

Given further exploration of Fluid Design, I would look to produce higher-fidelity interactive prototypes for more nuanced user testing. Expand the tool to also handle more context aware situations. Allowing designers to specify designs such as a dark mode in low light situations.
Prototyping would also be inquired for both screen and voice interfaces, by including motion design and sound design elements. Lastly, in order to make sure Fluid Design could be used effectively in an industry's technology team interactions around the designer / developer handoffs would need to be investigated and designed.

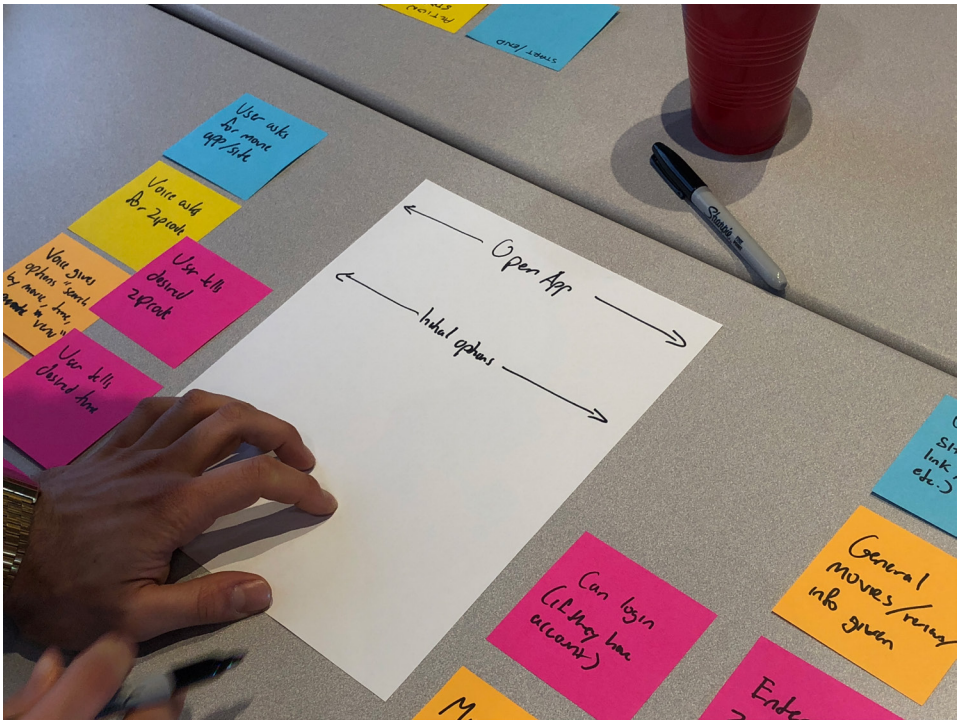
Though the flexibility built into this tool will make it useful for a while, we need to be watchful and test how Fluid Design would work for future platforms, not yet conceived. Does it still hold up as designed or are their additional aspects that would influence design changes in the tool?

All in all Fluid Design addresses some of the questions around designing platform-agnostic experience, but could afford further study with more details and nuances for designers and their users.

Appendix

Workshop participant walking us through his design

Workshop process



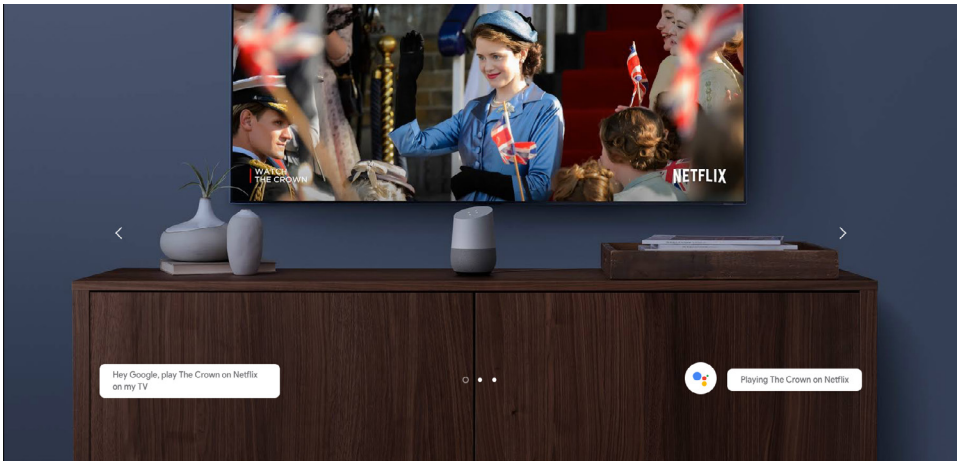
Workshop participant connecting the mobile and voice flows

Design the Experience

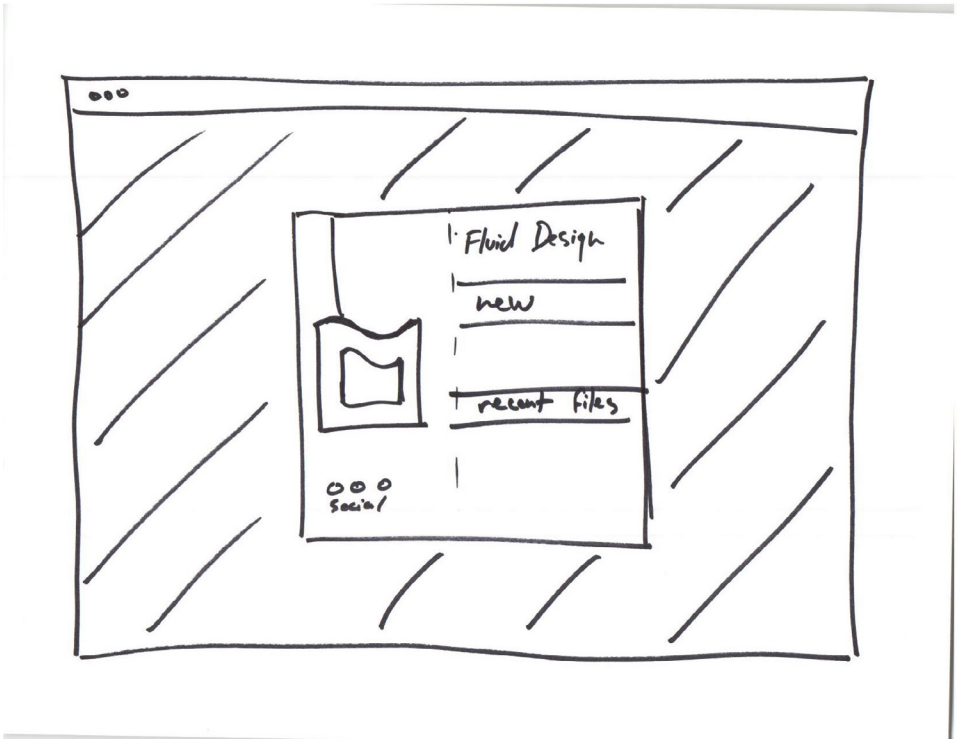
Skinning the Design

UI	Voice
Rounded Corners	Accent
Colors	Male / Female Voice
Motion / animation	Tone

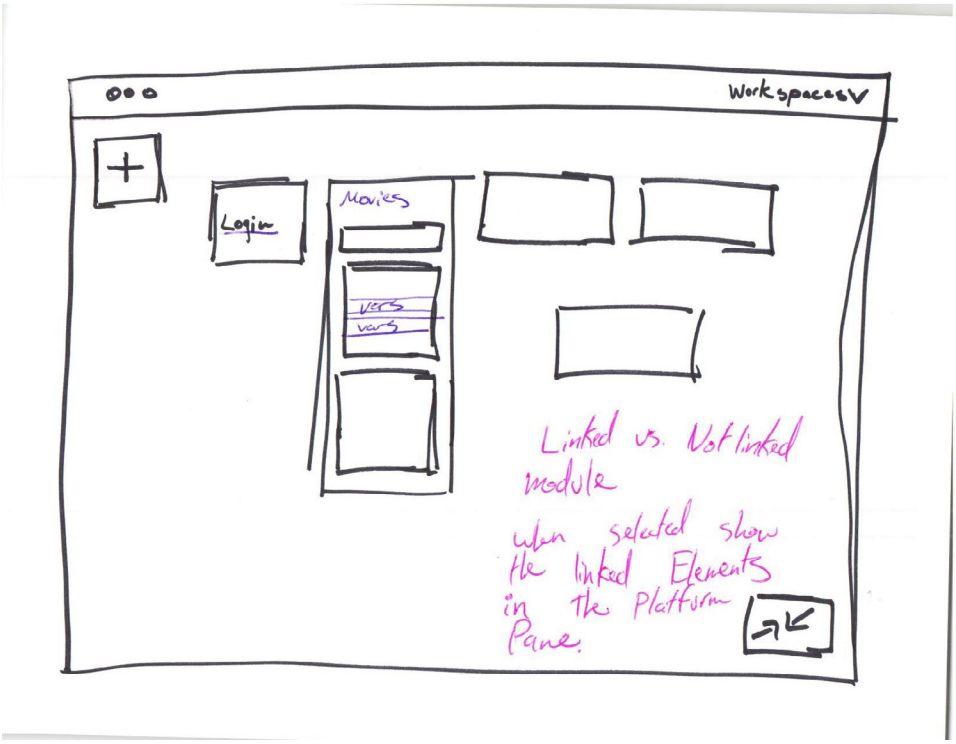
Showcasing the separation of experience and user interface



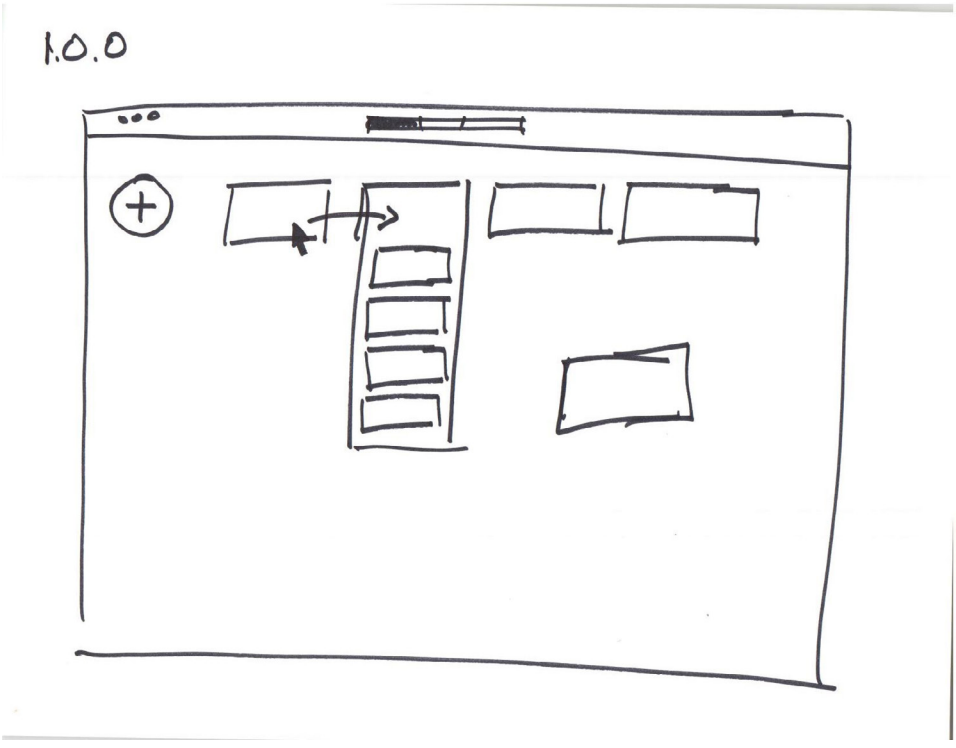
Google Home handoff experience ("Google Home")



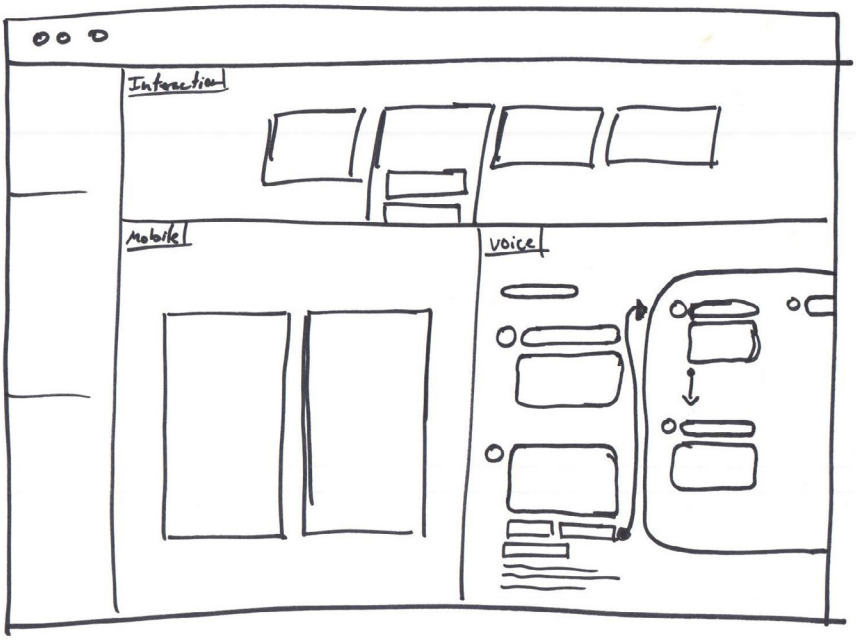
Welcome Screen Sketch



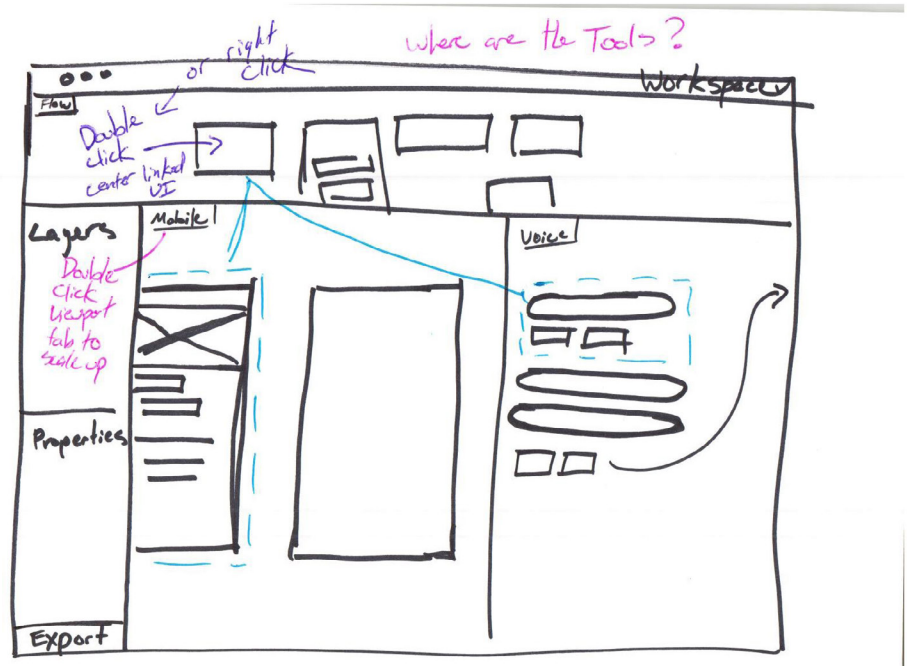
Interaction flow UI sketch



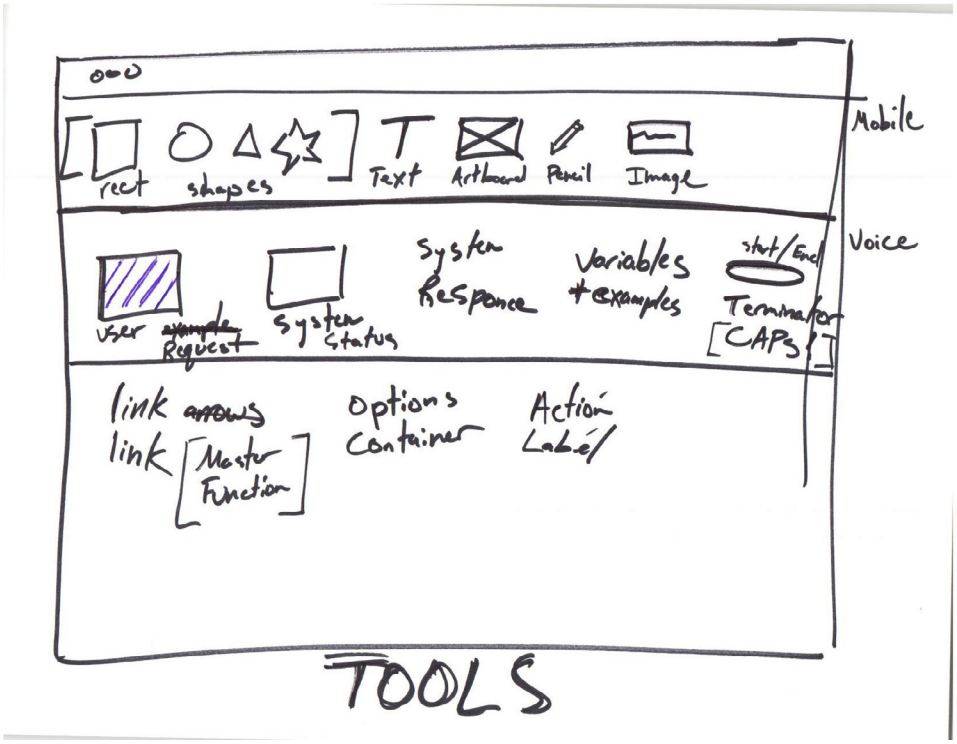
Interaction flow UI interaction sketch



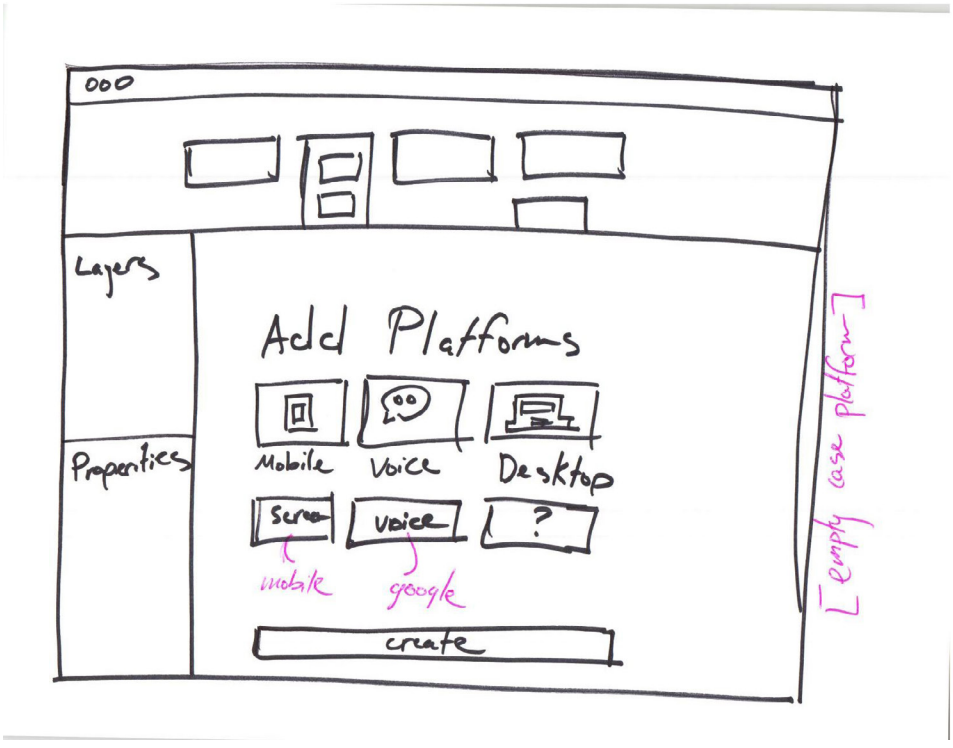
Main interface sketch



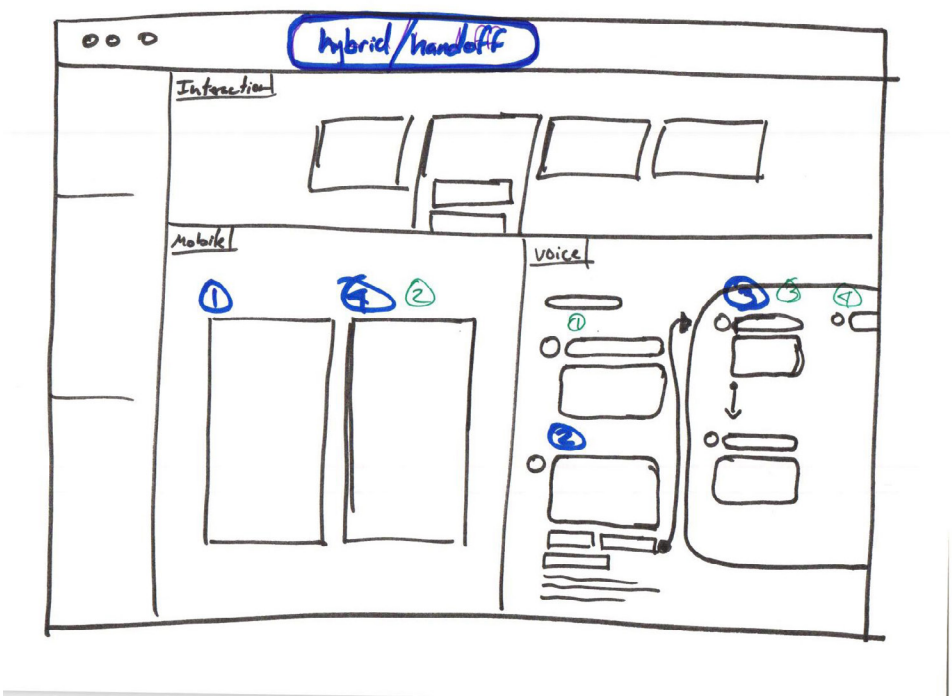
Main interface interactions sketch



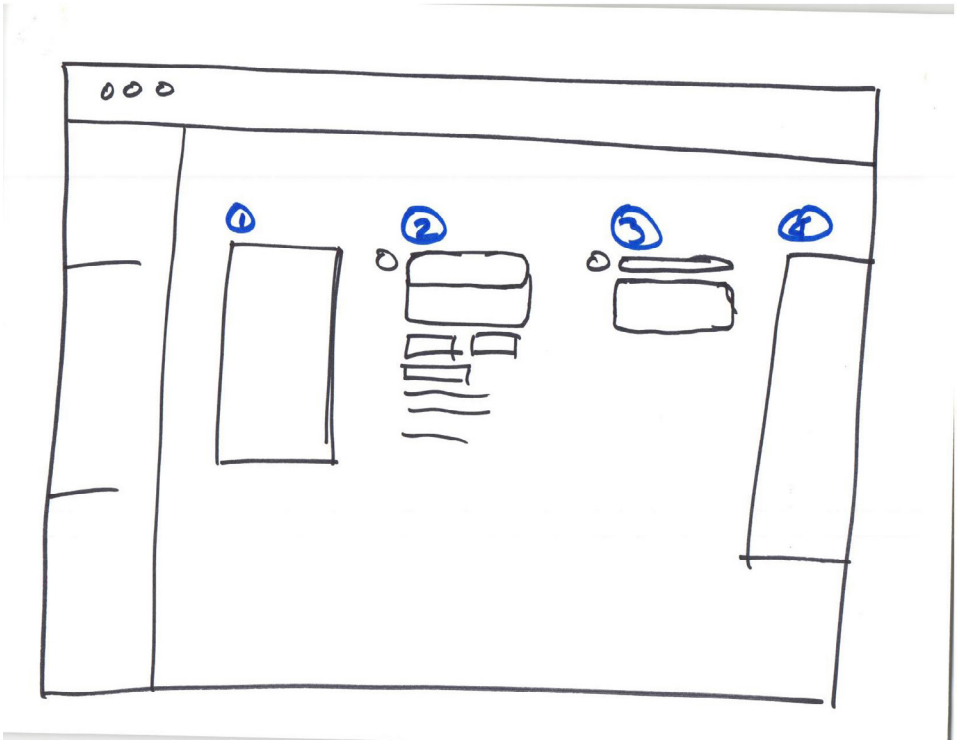
Screen and voice toolset sketch



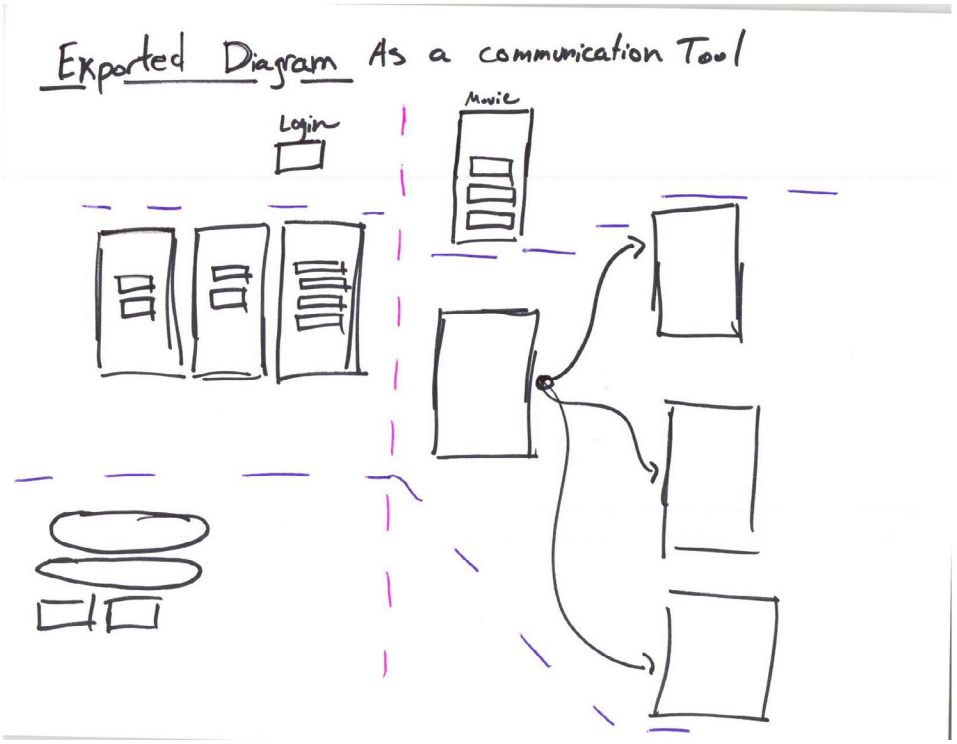
Add platform UI sketch



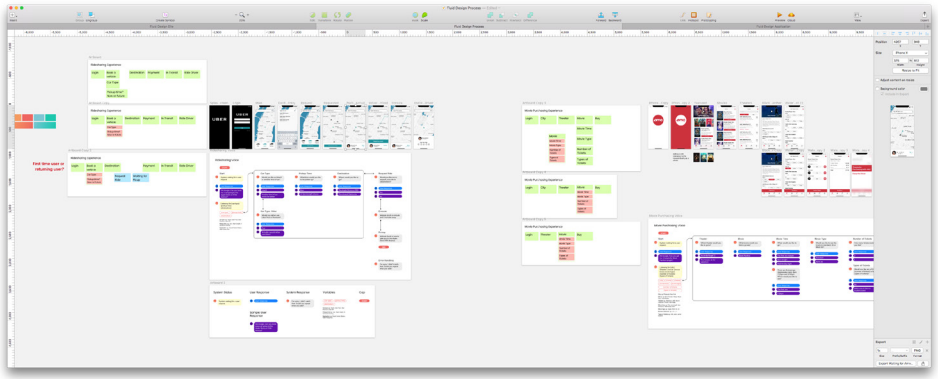
Handoff interaction sketch



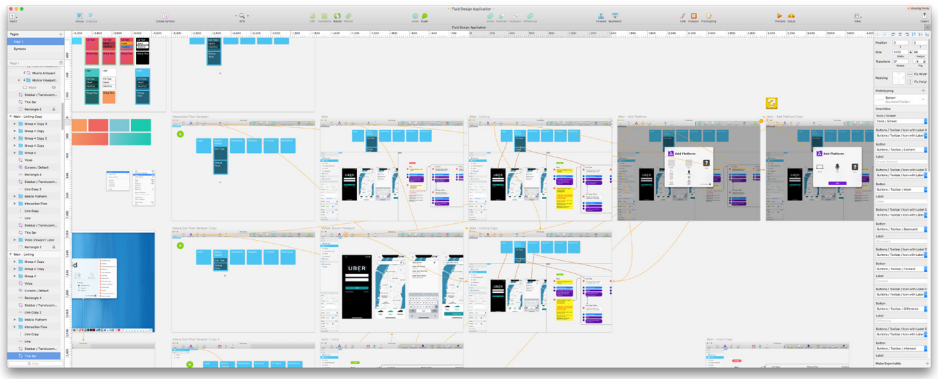
Handoff diagram sketch



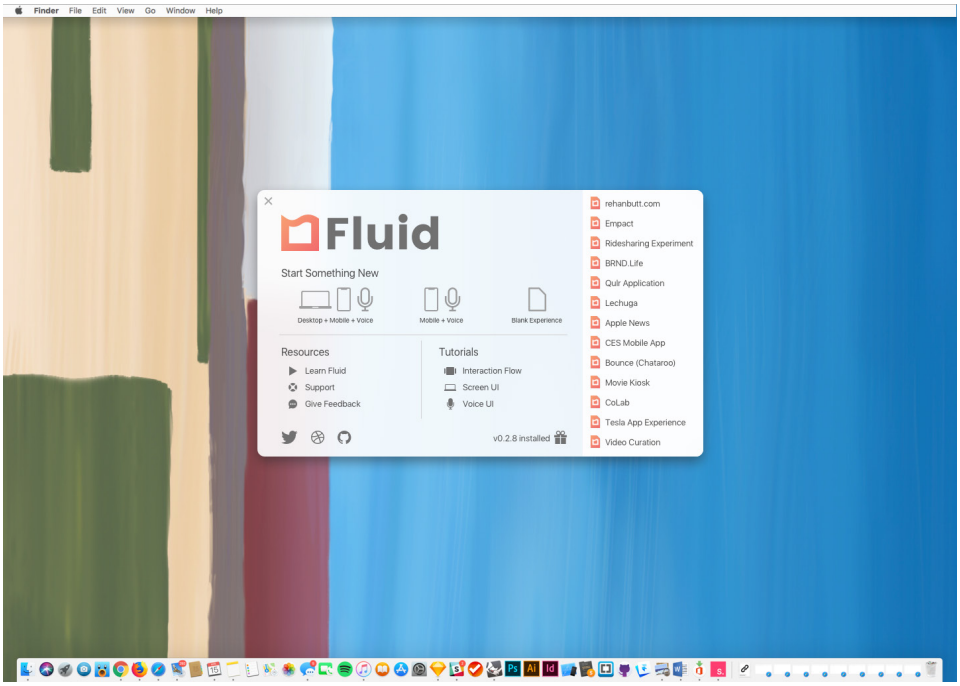
Exported Fluid diagram



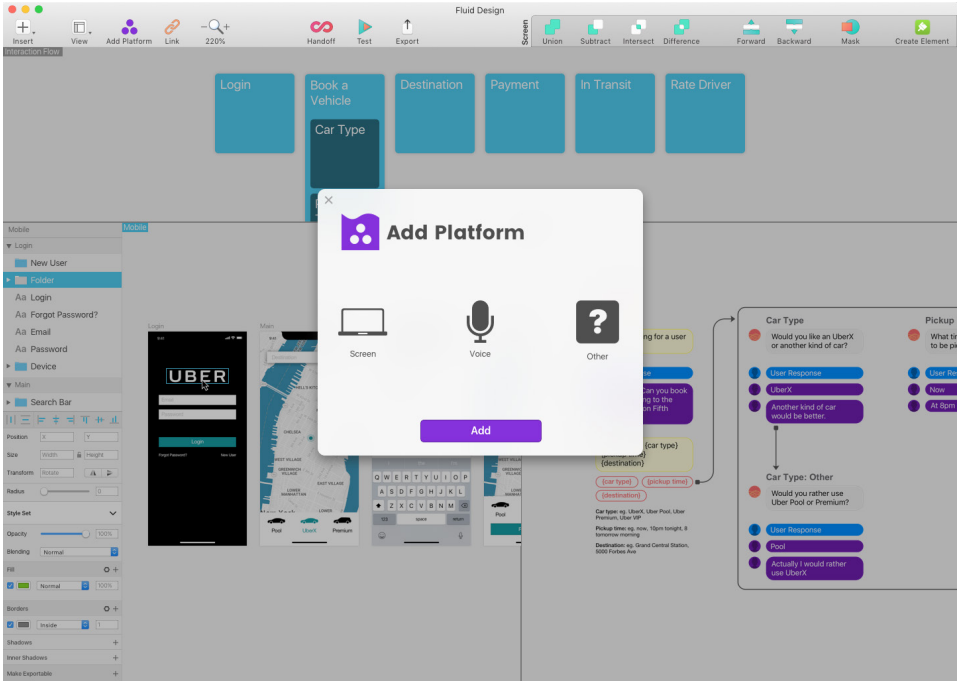
Designing the Fluid Experiences



Designing the Fluid Design interface



Fluid Design welcome screen



Fluid Design add platform prompt

References

"8-Bit Mario Nintendo Jumping." Redbubble, Astropop, www.redbubble.com/people/astropop/works/16078603-8-bit-mario-nintendo-jumping.

Abele, Nick. Personal Interview. 24 March 2018

"Adaptive Views." Adaptive Views, www.axure.com/support/reference/adaptive-views.

"Adobe Bets on Voice with Sayspring Acquisition." Adobe Blog, Adobe Communications Team, 16 Apr. 2018, theblog.adobe.com/adobe-bets-on-voice/.

"Adobe MAX 2016. Day 1 Keynote (Chapter 3)." Youtube, Adobe Creative Cloud,3Nov.2016,www.youtube.com/watch?v=dn_9u1qtQ8s&t=1412s.

Aljowaysir, Nouf. Personal Interview. 10 February 2018

Apple Inc. "Siri." Siri - System Capabilities - IOS Human Interface Guidelines, developer.apple.com/ios/human-interface-guidelines/system-capabilities/siri/.

"A Whole New Way to Collaborate Creatively." Freehand by InVision | Design Better, Faster-Together, www.invisionapp.com/feature/freehand.

Botanalytics. "Voice Chatbots vs. Messenger Bots: Everything You Need to Know." Voice Enabled Chatbots Vs. Messenger Bots: Everything You Need To Know, 21 Feb. 2018, botanalytics.co/blog/2018/02/21/voice-chatbots-vs-messenger-bots/.

"Building a Chatbot with Dialogflow and Google Cloud Platform." Youtube, Google Cloud Platform, 9 Feb. 2018, www.youtube.com/watch?v=5r4AAIfe4Rw.

Calzada, Jonathan. Personal Interview. 16 April 2018

"Collaboration on Same Document." Adobe XD User Voice, 21 Mar. 2016, adobexd.uservoice.com/forums/353007-adobe-xd-feature-requests/suggestions/13061826-collaboration-on-same-document-real-time-or-check.

Connolly, Diana. Personal Interview. 4 April 2018

Crawford, Kaleb. Personal Interview. 3 December 2017

Crowel, Charles R., et al. "Gendered Voice and Robot Entities: Perceptions and Reactions of Male and Female Subjects." 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009, doi:10.1109/iros.2009.5354204.

"Definition of Gender by Oxford Dictionaries." Oxford Dictionaries | English, Oxford Dictionaries, en.oxforddictionaries.com/definition/gender.

"Design Process." Amazon Alexa Voice Design Guide, developer.amazon.com/designing-for-voice/design-process.

Dhir, Kevin. "Introduction to DialogFlow." YouTube, 1 Dec. 2017, www.youtube.com/watch?v=DkZmVLHoCLo.

"Dialogflow." Dialogflow, dialogflow.com/.

"Dialogflow." Wikipedia, Wikimedia Foundation, 10 Apr. 2018, en.wikipedia.org/wiki/Dialogflow.

"Docs Basics." Dialogflow, dialogflow.com/docs.

Dudley, Ian. "Perspectives: How to Make a Good Chatbot." Nielsen, 21 July 2017, www.nielsen.com/eu/en/insights/news/2017/perspectives-how-to-make-a-good-chatbot.html.

Geerts, David. “Comparing Voice Chat and Text Chat in a Communication Tool for Interactive Television.” Proceedings of the 4th Nordic Conference on Human-Computer Interaction Changing Roles - NordiCHI ‘06, 2006, doi:10.1145/1182475.1182537.

Griffin, Matt. “Start Coding with Wireframes.” A List Apart, A List Apart, 18 Feb. 2014, alistapart.com/column/start-coding-with-wireframes.

“Google Home.” Google Store, Google, store.google.com/product/google_home.

Hall, Anthony. “Bot UX: 10 Steps to Creating a Great User Experience.”

ArcTouch, 31 July 2016, arctouch.com/blog/bot-ux/.

Huang, Chung-Ching, and Erik Stolterman. “How Interactive Artifacts ‘Change’ over Time.” Proceedings of the 7th Nordic Conference on Human-Computer Interaction Making Sense Through Design - NordiCHI ‘12, 2012, doi:10.1145/2399016.2399086.

Huang, Chung-Ching, and Erik Stolterman. “Temporality in Interaction Design.” Proceedings of the 2011 Conference on Designing Pleasurable Products and Interfaces - DPPI ‘11, 2011, doi:10.1145/2347504.2347572.

Jung, Heekyoung, and Erik Stolterman. “Material Probe.” Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction - TEI ‘11, 2011, doi:10.1145/1935701.1935731.

Kalbach, James. “The Myth of 800x600.” Dr. Dobb’s, 1 Jan. 2002, www.drdobbs.com/the-myth-of-800x600/184412392.

Kalbach, Jim. “The First Responsive Design Website: Audi (circa 2002).” EXPERIENCING INFORMATION, 14 Dec. 2015, experiencinginformation.com/2012/07/22/the-first-responsive-design-website-audi-circa-2002/.

Maestri, Leah, and Ron Wakkary. “Understanding Repair as a Creative Process of Everyday Design.” Proceedings of the 8th ACM Conference on Creativity and Cognition - C&C ‘11, 2011, doi:10.1145/2069618.2069633.

Marcotte, Ethan. “Responsive Web Design.” A List Apart The Full, A List Apart, 24 Jan. 2013, alistapart.com/article/responsive-web-design.

Martin, Bella, and Bruce M. Hanington. Universal Methods of Design: 100 Ways to Research Complex Problems, Develop Innovative Ideas, and Design Effective Solutions. Rockport Publishers, 2012.

Muise, Kevin, and Ron Wakkary. “Bridging Designers’ Intentions to Outcomes with Constructivism.” Proceedings of the 8th ACM Conference on Designing Interactive Systems - DIS ‘10, 2010, doi:10.1145/1858171.1858229.

Myers, Brad, et al. “Past, Present, and Future of User Interface Software Tools.” ACM Transactions on Computer-Human Interaction, vol. 7, no. 1, 2000, pp. 3–28., doi:10.1145/344949.344959.

Nielsen, Jabok. “10 Heuristics for User Interface Design.” Nielsen Norman Group, 1 Jan. 1995, www.nngroup.com/articles/ten-usability-heuristics/.

Nielsen, Jakob. “Mouse vs. Fingers as Input Device.” Nielsen Norman Group, 10 Apr. 2012, www.nngroup.com/articles/mouse-vs-fingers-input-device/.

Nielsen, Jakob, and Rolf Molich. “Heuristic Evaluation of User Interfaces.” Proceedings of the SIGCHI Conference on Human Factors in Computing Systems Empowering People - CHI ‘90, 1990, doi:10.1145/97243.97281.

Naidu, Rubini. Personal Interview. 19 April 2018

Odom, William, et al. “Understanding Why We Preserve Some Things and Discard Others in the Context of Interaction Design.” Proceedings of the 27th International Conference on Human Factors in Computing Systems - CHI 09, 2009, doi:10.1145/1518701.1518862.

Olsen, Dan R., and Scott R. Klemmer. “The Future of User Interface Design Tools.” CHI ‘05 Extended Abstracts on Human Factors in Computing Systems - CHI ‘05, 2005, doi:10.1145/1056808.1057124.

O’Toole, Erin. Personal Interview. 20 March 2018

Paper Signals, Google, papersignals.withgoogle.com/.

“Prototyping Amazon Alexa Skills with Sayspring.” YouTube, Dabble Lab, 13 Oct. 2017, www.youtube.com/watch?v=6SQBbdFmW3o&t=3s.

Rosson, Mary Beth, et al. “The Designer as User: Building Requirements for Design Tools from Design Practice.” Communications of the ACM, vol. 31, no. 11, 1988, pp. 1288–1298., doi:10.1145/50087.50090.

Saffer, Dan. Microinteractions. O’Reilly, 2014.

Super Mario Run. Nintendo, supermariorun.com/en/index.html.

Stolterman, Erik, and James Pierce. “Design Tools in Practice.” Proceedings of the Designing Interactive Systems Conference on - DIS ‘12, 2012, doi:10.1145/2317956.2317961.

“The Collaborative Interface Design Tool.” Figma, www.figma.com/features.

“The Conversational UI and Why It Matters | Actions on Google | Google Developers.” Google, Google, developers.google.com/actions/design/.

“The Digital Design Toolkit.” Sketch, sketchapp.com/.

"The Rhino Window." Rhino 3-D Modeling, docs.mcneel.com/rhino/5/help/en-us/user_interface/rhino_window.htm.

"The World's Most Powerful Screen Design Tool." InVision Studio | Screen Design. Redesigned., www.invisionapp.com/studio.

Topdjian, Albert. Personal Interview. 14 April 2018

Wallace, Evan. "Multiplayer Editing in Figma – Figma Design." Figma Design, Figma Design, 28 Sept. 2016, blog.figma.com/multiplayer-editing-in-figma-8f8076c6c3a6.

Webster, Mark. "Sayspring Is Now Part of Adobe." Sayspring, 16 Apr. 2018, www.sayspring.com/blog/sayspring-is-now-part-of-adobe/.

Webster, Mark. "What Ever Designer Should Know About Voice." FirstMark's Design Driven, 3 August 2017, New York, New York

Wiltse, Heather, and Erik Stolterman. "Architectures of Interaction." Proceedings of the 6th Nordic Conference on Human-Computer Interaction Extending Boundaries - NordiCHI '10, 2010, doi:10.1145/1868914.1869038.

Zepeda, Brandon. Personal Interview. 24 March 2018

Zhang, Xiao, and Ron Wakkary. "Understanding the Role of Designers' Personal Experiences in Interaction Design Practice." Proceedings of the 2014 Conference on Designing Interactive Systems - DIS '14, 2014, doi:10.1145/2598510.2598556.