

## UX FOUNDATION: MULTIDEVICE DESIGN:

### 1. Overview of multidevice strategies

- **Design for the people first**, evaluate the key tasks on each device.
- Understand the goals of your users on the device.
- **Design for an ecosystem**: users may move fluidly across devices.
- Determine how the user experience will interact with each device.
- **Design a coherent experience**: adhere a common interaction patterns and platform-specific guidelines.
- Apply consistent branding across devices.
- **Context of use**: understanding of the conditions and environment in which the user interacts with your product.
- e design does not take into account TV layouts or watches as of yet. More specifically, responsive web design refers to a similar or the same web experience between desktop, tablet, and smartphone. We use breakpoints in the layout to determine at which point the content will be displayed for each device.
- Responsive design allows for easy access for content. Users can access content from any device they want and without having to download an app. And lastly, there are some search engine benefits. While not user experience related, it is helpful for your site to ensure that it's listed in search engine ranking if using a single URL. Smartphone searching is a key activity and it helps if your site is ready for mobile, when somebody is looking for your product or service.
- Native applications have design guidelines for the specific platform they were developed on. For example: Apple, Google and Windows all have design guidelines for interaction patterns and visual design to help unify the experience and create a consistent and familiar experience for the user regardless of which app is used on the respective platforms. Knowing if the design is going to be native will help you as a designer to formulate your design using the existing platform paradigms. Google's material design patterns use density independent pixels to define layouts that might be scaled to various resolutions across devices.
- Native apps are accessed from app stores.
- Hybrid applications take an approach to app development that includes a mix of using web-based technologies and native technologies or interactions.
- Google has compiled a list of screen sizes and resolutions for Android, its native platform, as well as Apple's native platforms of iOS, OSX, and some devices for the Windows platform.

### 2. Multidevice Design Consideration

- Touch interfaces include mobile devices and tablets and can include desktops as well.
- Users interact with touch interfaces using gestures. Some of the key gestures that are present on both iOS and Android are tap or touch, double tap, drag, flick, swipe, pinch, touch and hold, or long press. In addition to this list there are a few more platform-

specific gestures that cover other use cases as well. For some gestures there may be some discoverability issues. So if your task is important, make sure your UI is clear in how users can engage with your experience by being explicit. Since gestures are how your users will interact with your app, keep them intuitive. Use animation. Animation is useful for reinforcing state changes and for giving users feedback that something just happened, like adding an item to a cart or acknowledging that something was deleted.

- In recent years, smartwatches have become more mainstream and the market for smartwatches is growing. Apple has a watch that uses the WatchOS platform and Google has partnered with various watch brands to run its Android Wear platform. There are a few different watch platforms, but Apple and Google are the most popular. Watches have a small area in which the user can interact with and watches run the risk of being more disruptive to the user than a phone because it's worn on the user's body and will be with the user throughout the day. Some things to consider when designing for watches. Highly focused tasks. Apps for watches should be highly focused on a singular task. Given the nature of the small screen, there isn't a lot of affordance to accommodate a complex task and it can be tedious to scroll or interact with multiple views and screens for information on a small device. Legibility. Given the small screen, the text should be both legible and minimal.
- Designing for TV presents some unique challenges. Typically, when watching TV, people are about 10 feet away from the television screen. So, sometimes, you might hear designing for TVs referred to as 10-foot UI. Typical activities are to watch content, or for gaming. The primary method of interacting with the television is through a remote control or a game pad. Most TV and game controllers have a D-pad directional controller which consists of four directions. Up, down, left, and right which are used to navigate the screen. In addition to the traditional remote, most Smart TVs and devices for TV have an app you can pair with your phone with the TV, wirelessly using Bluetooth to act as a remote. Some considerations, when approaching a design for TV - legibility of content. While sitting 10 feet away, you'll want to ensure there isn't a lot of text on the screen and that the text is large enough to see from a distance. Using light text on a dark background increases contrast, making it easier to read the text at a distance. Clear navigation - when navigating content with a remote control, gaming control, or an app, it's essential that the navigation pattern on the screen is clear and mapped to the proper controls on the device.
- When designing for voice, keep in mind you are interacting only with voice. Feedback, errors, and content should be clearly defined.
- Conversation should be brief.
- There should be consistency between devices.
- Same content across devices.
- Responsive web design

### **3. Planning a Multidevice Experience**

- Understand your users need.
- Talk to your users.
- Experience with other apps.
- Contextual inquiry.

#### **4. Designing your mobile experience in sketch**

- Section 4 is about demonstrating that how to make an art board with sketch.
- It also shows how to make a playlist screen.
- She also demonstrates making a music player screen.

#### **5. Designing for additional devices in sketch**

- section 5 starts with demonstrating of creating layout for watches in sketch.
- It also shows layout demos for tv and voice.
- or mobile devices and watch, you can prototype these experiences in a lightweight format using InVision.

#### **6. Frameworks and guidelines**

- Bootstrap is a responsive framework.
- Foundation is another responsive framework.
- CMS systems power the content for your website and have a blog that allows for commenting, image loads, video, and other types of content. Popular CMSs have customized layouts and themes. Themes allow you to change the look of the site quickly. WordPress is one of the more popular CMSs and has a lot of themes that support a responsive layout. Unlike Bootstrap or Foundation, there is no coding required. However, it does have its own ecosystem of plugins and widgets, but you can opt for an out-of-box solution, which gives you the flexibility to upload your own content and the CMS itself handles the responsiveness.
- take a closer look at this design documentation around Material Design. When working in cross-native platforms and through various devices, it's important to use these design documents as a guide to make sure that you're adhering to platform standards. However, there are times when you may want to deviate from the standard look and feel, but that's a choice that you'll make depending on the project you're working on. Typically, you'll be using the documentation to get a sense of the options you have available to you from a component standpoint, as well as other specs you might need to understand the platform design paradigms.
- Apple has its own documentations for its products as well, which they call Human Interface Guidelines. Let's take a look at these iOS guidelines. Like the material design document, we immediately see the designing principles of the system. Navigating to the App Architecture section, there's some guidance on how to structure your app's navigation, and some links to different view components.