ATLS 4320: Advanced Mobile Application Development

# Week 2: Designing Intuitive User Experiences and Navigation

## Designing Intuitive User Experiences

### UX in App Design

- Good UX design will become more and more important as our apps become more and more complex. It's essential that our apps still feel simple and easy to use regardless of how complex they actually are under the hood.
  - Not easy to do
  - There are certain guidelines to adhere to but a lot of it is left up to the designer
  - Since we'll be both developer and designer for our projects, we have a unique opportunity to quickly integrate UX at a very deep level
  - A lot of this comes down to the decisions we make before we even start thinking about development or visual design
- Bad Example of Mobile UX
  - Co—Star: aside from their ridiculous content, they also have some structural/navigation issues
    - Content is not divided at all — in a single home page which links to many sub pages
    - Astrology is apparently super complicated and Co—Star does nothing to make this better despite their status of "fun and trendy"
    - Multiple ways to access the same information
    - Very hard to navigate around to the "most important" stuff whatever that may be
- Good Example
  - Uber
    - I don't love the company but they are always mentioned in mobile UX conversations
    - Everything is present on the home screen
    - Tabs for the two main sections of content
    - All of more detailed aspects are hidden in the hamburger menu
      - Not a huge fan of this but there's only so much you can do with the size constraints of mobile only
- Mike Stern: WWDC 2017 Essential Design Principles: https://developer.apple.com/videos/play/wwdc2017/802/
  - Essentially Norman's design concepts but applied to iOS
  - Should be familiar, but this is a pretty good review/summary
  - Get us in the headspace to think about how our users will interact with our apps
  - Last video of the iOS unit — sorry they both came so early!

### Navigation

https://developer.apple.com/ios/human-interface-guidelines/app-architecture/navigation/
App navigation should feel natural and familiar. It should complement the content and purpose of the app. In iOS there are 3 main navigation types. Before you decide which type of navigation to use, it is important to nail down all of the specifics of your information architecture. Consider the following:

- What actions will the user be able to take within your app?
- What kind of data will you be displaying in your app?

Can be helpful to use a textual outline to answer these questions and show how they fit together in as much detail as possible and use this outline to determine what your navigation will look like. Let's look at each type of navigation and then work through an example.

### Flat Navigation

Tab bars switch between content views
https://developer.apple.com/design/human-interface-guidelines/ios/bars/tab-bars/

- Provide a flattened hierarchy for your app
- Provide access to several peer information categories or modes at once
- Content is not passed between tabs
- Strictly for navigation
    - Don't confuse with toolbars to perform actions
- Clock, Music, App store

Page Controls are another way to implement flat navigation
https://developer.apple.com/design/human-interface-guidelines/ios/controls/page-controls/

- Always shows the number of pages at the bottom
- All pages are "peers" and show the same type of content
- Weather app is an example

### Hierarchical Navigation

Navigation controllers w/table make one choice per screen until the desired destination is reached. To go to another destination, you must retrace your steps or start over from the beginning and make different choices.

- Use table views to present hierarchical content
  https://developer.apple.com/design/human-interface-guidelines/ios/views/tables/
- Navigation bars enable navigation through hierarchical content
  https://developer.apple.com/design/human-interface-guidelines/ios/bars/navigation-bars/
- Settings, mail, contacts
- Possible to use navigation bar without a table view and vice versa
    - Calendar uses nav bar but doesn't necessarily use table view (though it does use table views for some content like schedule)

### Content-Driven or Experience-Driven Navigation

Move freely through content, or the content itself defines the navigation.

- The content itself defines the navigation
- Games, books, other immersive apps

## Demo IA

Let's use the clock app to reverse engineer an information architecture and learn a bit more about why it utilizes the navigation type that it does.

### Information Architecture

- World Clock
    - View list of clocks at various locations around the world
        - Name
        - Time Difference
        - Current Time
    - Delete a clock from the list
    - Add a clock to the list
        - Select from list of available locations
    - Reorder list

- Alarm
    - View list of alarms
    - Toggle alarm
    - Add alarm
        - Time
        - Repeat options
        - Label
        - Sound
        - Snooze toggle
    - Edit alarm
        - Same as add alarm
- Bedtime
- Stopwatch
    - View time
    - Start/Stop
    - Lap/Reset
- Timer
    - Set time
    - Start/cancel
    - Set sound

## Analysis

Each top level node is completely separate from the others and unaffected by actions in other nodes. The entire architecture is largely action based and there really isn’t much of a hierarchy. For this reason, a tab bar controller was used to provide the main navigation