

## Week 7: UI

# UI

- The user interface (UI) is the space where interactions between humans and machines occur.
- UI consists of two major parts:
  - visual design, which conveys the look and feel of a product; and
  - interaction design, which is the functional and logical organization of elements.

# UI designers and SW Engineers

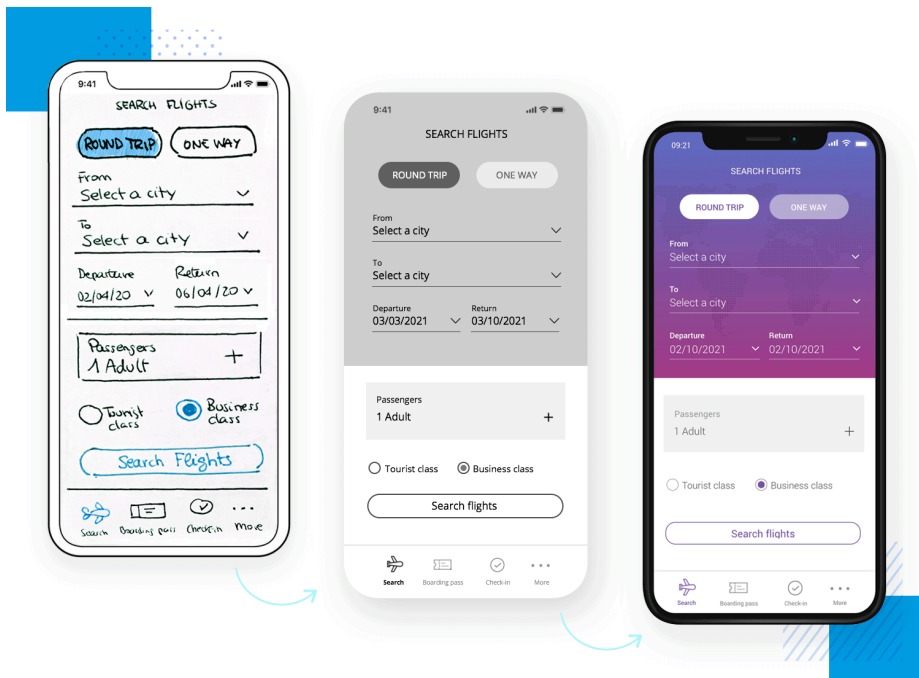
- UI designer:
  - focuses on the visual appearance and user interaction of an application
  - conducts usability testing.
- SWE will:
  - build the actual product
  - Front-end engineers will build the web interface
- UI design and front-end dev are sometimes combined in a UI Dev role.

# UI prototyping

- What?
  - a simulation of the final interaction(s) between the user and the interface.
  - it can simulate an entire app or just a single interaction.
- Why?
  - Ensure the design concept works as intended
  - Determine if people are able use a product.

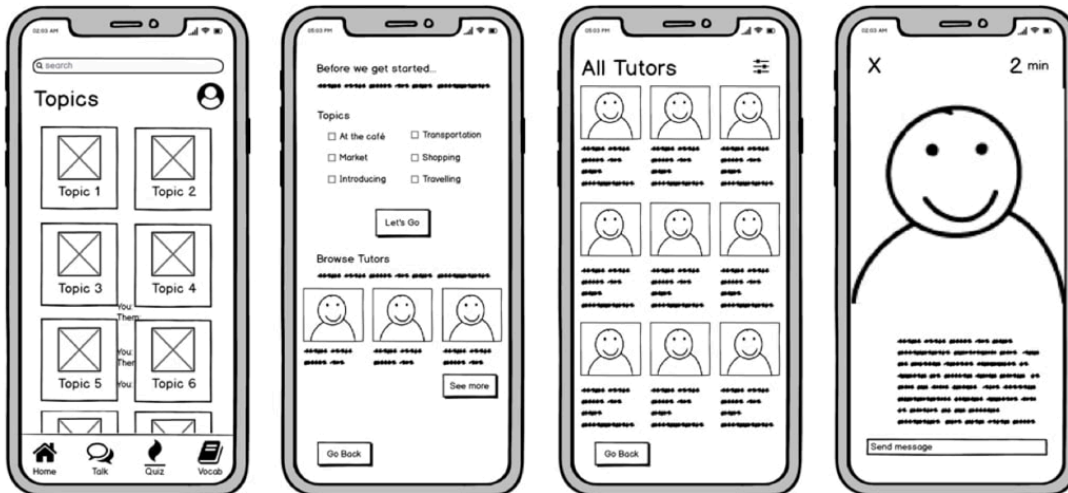
# Fidelity

- The fidelity of a prototype refers to its level of detail and realism.
  - Low-fidelity
  - High-fidelity



# Low-fidelity prototypes

- low-fi prototypes are primarily about interaction design
- paper prototypes: hand drawings of different screens that represent user interfaces of a product
- wireframes: visual representations of a web page or app interface, stripped down to its bare bones



# Usability testing

1. define goals
2. create prototype
3. recruit participants
4. prepare the testing environment
5. run the test
6. analyze results

# Usability testing with lo-fi prototypes

- can be conducted with non-interactive prototypes
- One person will be the facilitator ('computer') that's helping the test participant walk through the design



# Interactive prototpyes - no code

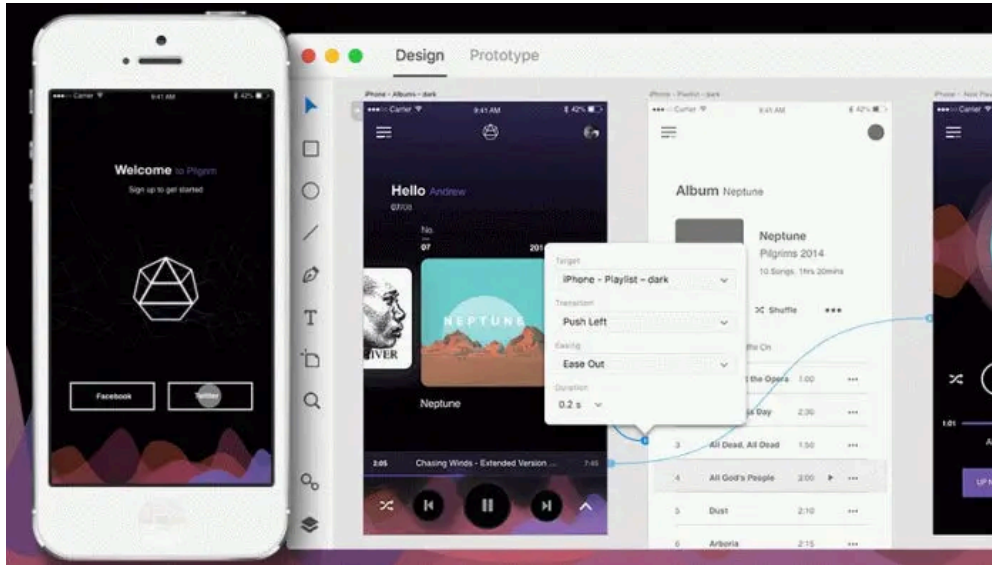
- basic interactive prototype can be made in a presentation software like PowerPoint
- more complex prototypes can be made in specialize software like Figma.
- Testing doesn't require a facilitator

# Interactive prototypes - with code

- you can also make prototypes with code
- For web-based interface, this means (at least) HTML
- pros: you can connect this prototype to a backend (web server) and use it for functional testing
- cons: it's more work than no-code prototyping, and may end up being redone.

# High-fidelity prototypes - no code

- appear and function as similar as possible to the actual product that will ship.
- usually done using a specialized tool like Figma or Adobe XD.



# High-fidelity - with code

- For web-based interface, this means CSS and (usually) JavaScript.
- CSS frameworks like Bootstrap:
  - provide a collection of pre-designed components,
  - allowing developers to quickly build interfaces
- Pros: getting very close to an actual product
- Cons: significant effort, may need to be redone

# Iterative development

- Start with lo-fi prototypes, early in the process (right after writing stories)
- iterate with usability testing,
- increase interactivity and fidelity as you go