Topics

• Peer evals
• UI Sketches

Homework

• Quiz 4 next Tues
• HW5 out
Context Diagrams describe the system “boundaries”

- what is inside (“in scope”) vs outside (“out of scope”)
- things out of scope that you interface with
- meant to be high-level (coarse grain)
- subsystems, environments, interactions

Context Diagram Examples ...

![Diagram showing labeled inputs and outputs with actors outside of the current system connected to a central system.](image-url)
Good context diagrams highlight items outside the boundary

- interactions & dependencies to other systems/subsystems
- usually important or non-obvious systems/subsystems
- sometimes to people (e.g., for approvals, etc.)

Focus on what exists that your project will use versus build

- All about what you wont implement (but need)
- DBMS, Authentication systems, GPS/Map Displays, ...
Design in Software Engineering ...

Throughout a software project, we do lots of “design” work

- e.g., the MVP, feature set, user interface / interaction model, data models, architecture, software, tests

Design is a process involving various steps, including ...

- Idea generation (“ideation”)
- Idea testing, prototyping, and refinement
- Idea selection and communication

Keys to keep in mind when doing design work ...

- Go for quantity (“quantity breeds quality”) ... alternatives
- Withhold judgement ... analyze & critique ideas later
- Combine and improve

User stories help design & capture system features ...

- communicate to customer, scope project, carry out sprints

But hard to see how system should work with stories alone

- Can make it hard to know if set of stories is complete
Lightweight “prototyping” is a tool to help extend/refine stories ...

- **Basic Idea**: use lightweight materials to “mockup” system

### Two flavors of UI design (mockups)

1. To better understand the MVP ... *should be your current focus*
   - Rough sketch of basic application “screens” and “pages”
   - How they roughly work together
   - **Focus on content & flow, not layout & details**

2. Detailed design of the UI to build the system
   - Details of each “screen” or “page”, how layed out
   - Colors, format, widgets, messages, ...
   - Professional “look & feel”

### We’ll talk about two types of mockups

- **UI sketches** ... freehand drawing of UI “screens”
- **UI storyboards** ... flow diagram of how screens “connect”
A UI Sketch is a high-level drawing of a part of the UI

- Each sketch is usually a single “screen” (or “page”)
- E.g.: one page in a web site or one screen on a mobile app

In a UI Sketch start simple then iteratively add detail ...

(1). At first focus on basic elements needed (button, text, etc)
   - Don’t worry too much about placement
   - Develop alternatives (if unsure which pages are needed)

(2). Create more detailed sketches
   - Add in missing elements
   - Consider different ways to organize elements
   - Consider different basic widgets

(3). Create alternatives, get feedback, refine until you converge

Some tips for creating UI Sketches ...

(a). Use “throw away” materials (don’t get attached)
(b). Consider whiteboard first (easy to change & chuck)
(c). Move to paper & pencil (also easy to change & chuck)
(d). Goal is to converge to a basic design (not a pretty picture)
(e). Once you converge, move to pen & paper (if needed)

UI Sketch Examples ...
UI Storyboards connect the different screens together

- Describe the “flow” between screens/pages (like a “map”)
- Which screen is the “landing” page (home screen)
- What screens can be accessed from which other screens

Good Storyboards emphasize common navigation paths

- The “typical” paths through the screens users will take
- E.g.: Search for items ⇒ View item ⇒ Select item ⇒ Add to cart ⇒ Checkout

Tips for UI Storyboards

(a). Use throw-away materials (whiteboard, paper/pencil)
(b). Focus on common paths through the app
(c). Each path should have a specific point/goal (for the user)
(d). You want to make the typical paths “easy” ...

UI Storyboard Examples ...
From UI Sketches to Formal Mockups/Prototypes

(1). Start with sketches and storyboards
   – Basic info on each screen & flow between screens

(2). Create refined sketches and storyboards
   – Add more details and organization

(3). Create detailed diagrams for look and feel ("wireframes")
   – Take longer to develop

(4). Create functional prototype mockups
   – “Dummy” web forms, stubbed out UI screens, etc.

• In 1-3, generate many alternatives!
• In each step, get feedback, refine, & repeat until converge!

Wireframe Examples ...
Some general UI Design Tips ...

- Use “standard” UI components & design (where possible)
  
  Follow the “law of least surprise"
  
  Look at (and borrow from) similar and popular UIs

- Get lots of feedback and often
  
  Have users try out your sketches on paper!

- Optimize for frequent cases
  
  E.g., minimize number of clicks for common functions

- Aim for self explanatory
  
  Try out on your peers, friends, etc.
Architecture Diagrams

Architecture vs (detailed) design

- architecture focused on subsystems, technology choices
- detailed design focused on classes, interfaces, interactions

Context (Diagram) ⇒ Architecture (Diagrams) ⇒ Detailed Design (Diagrams)

Organizing into subsystems

- Names for major components / responsibilities
- Basic flow of data and/or delegation
- What specific technologies will be used

Just like UI design, good to sketch architectures

- saves time
- prevents “over architecting”
- lets you quickly converge as a team

Like any diagram or model, presents an abstraction ...

- helps team have a shared understanding
- use same terminology
- start to break up into well-defined areas of work
Can start with sticky notes or 3x5 cards ...

- identify subsystems (e.g., major responsibilities)
- think about creating logical boundaries
- think about how they interact / what they need
- think about how external services / technologies fit

Examples ...