LE/EECS 4443: Mobile User Interfaces (LAB)

Week 2: Good Design

Shogo Toyonaga[1]

[1]Lassonde School of Engineering
York University

January 8, 2025

Table of Contents

Introduction

Remark

Note: There are no demos to accompany this weeks slides. Rather, this weeks recitation acts as a recap with some additional information to help contextualize the need for good mobile user interface design.

Good UI Design

A good design will:

1. Achieve goals within constraints
   - Form Factor
   - Hardware & Software Limitations
   - Cultural Differences (Heuristics)
   - Accessibility
2. Balance between form & function
   - Cognitive Load
   - Fatigue
3. Be justifiable
   - Frameworks (Standards)
   - Literature (Documentation)

Good UI Design

Some established frameworks that you can use or reference...

1. Shneidermans Eight Golden Rules
2. Nielsen and Molich's 10 User Interface Design Guidelines
3. Android Developers Design Recommendations

Note: There are specialized frameworks for certain use-cases. Do some research and find a framework that best aligns with your domain of interest, target audience, and constraints.

Good UI Design

Good designs use an **iterative** approach consisting of:

1. Wireframes
2. High-Fidelity Prototypes (Figma)
3. User Consultation
4. User Testing

Read More:
https://www.nngroup.com/articles/parallel-and-iterative-design/

Good UI Design

In the context of your assignment, project, and future work:

1. Ensure that your app and UI support a broad range of different devices & screen sizes.

   Note: You can see how your app will look on different devices via, "Reference Devices" in the design view of your layout (XML)

2. Ensure that the UI acts how you intend it to with Android UI Tests (Espresso Tests)

Avoid: Dark Patterns

1. **Bait & Switch:** Coerce an action or user behaviour under false pretenses. Once the premise has been achieved, switch the consequent.
2. **Trick Question:** Inserting a payload into a set of commonly known steps (actions) while the user is immersed in thought.
3. **Roach Motel:** Hide inconvenient options from the user.
4. **Misdirection:** Using ambiguous and confusing text to deter users from choosing the inconvenient option.

Dark Patterns

Here are some short & interesting papers to read if you are intrigued by dark patterns:

1. UI Dark Patterns & Where to Find Them: A Study on Mobile Applications & User Perception
2. What Makes a Dark Pattern... Dark? Design Attributes, Normative Considerations, and Measurement Methods
3. The Dark (Patterns) Side of UX Design
4. Beyond Dark Patterns: A Concept-Based Framework for Ethical Software Design

Good Software Design

Good software will:

1. Adhere to well-defined requirements (frameworks)
   - Recall: Model-View Controller
   - Recall: Software Engineering Design Patterns (LE/EECS 3311)
2. Be manageable / maintainable
   - Abstractions (e.g., Encapsulation)
   - Decomposition
   - Hierarchy (Top-Down, Bottom-Up?)
3. Function in the face of uncertainty (Crash Recovery)

Good Software Design

Some other useful references...

1. Android Developers Guide to App Architecture
2. App Performance Analysis with the Android Studio Profiler

Conclusion

Conclusion

Thank You For Your Attention!
Questions?