

User Interface Design

01418321 System Analysis and Design
Chalothon Chootong (Ph.D.)

Department of Computer Science and Information, Faculty of
Science at Sriracha, Kasetsart University Sriracha Campus

chootong.c@ku.th





Human Factor of Interaction Software

Introduction

- ▷ Main focus of HCI is UID
- ▷ What is involved in UI?
 - H/W
 - Behavior of S/W
 - Supporting Documentation

Who Build Interface?

▷ A team of specialists (ideally)

- Graphic designers
- Interaction / interface designers
- Technical writers
- Marketers
- Test engineers
- Software engineers
- customers

Human-Computer Interaction (HCI)

▷ Human

- The end-user of a program
- The others in the organization

▷ Computer

- The machine the program runs on
- Often split between clients & servers

▷ Interaction

- The user tells the computer what they want
- The computer communicates results

Why study User Interfaces?

- ▷ Major part of work for “real” programs
 - Approximately 50%
- ▷ You will work on “real” software
 - Intended for people other than yourself
- ▷ Bad user interfaces cost
- ▷ User interfaces hard to get right
 - People are unpredictable

Goals of System Engineering

Steps for user-interface engineering

1. Task analysis to ensure proper functionality
 - Define tasks and subtasks
 - Functionality must be complete
2. Reliability, availability, security and data integrity
 - Commands must function as specified
 - Data displayed must reflect the actual database
 - Error free
 - High availability
 - Ensure the user's privacy.

Goals of System Engineering

3. Standardization, integration, consistency, and portability

4. Schedules and budgets

- Late products can make a company ineffective or uncompetitive.



Usability

What is Usability?

- ▷ Learnability
- ▷ Efficiency
- ▷ Memorability
- ▷ Error
- ▷ Satisfaction

Learnability

- ▷ Easy to learn
- ▷ How long does it take for typical member of the community to learn relevant task?

Efficiency

- ▷ How long does it take to carry out the benchmark set of task?
- ▷ A high level of productivity is possible
- ▷ Speed of performance

Memorability

- ▷ Easy to remember
- ▷ Frequency of use and ease of learning help make for better user retention
- ▷ Retention over time

Errors

- ▷ Low error rate
- ▷ How many and what kinds of errors are commonly made during typical applications?

Satisfaction

- ▷ Pleasant to use
- ▷ Allow for user feedback via interviews, free-from comments and satisfaction scales
- ▷ Questionnaires in 1-5 or 1-7 rating scales
 - ▷ 1 = strongly disagree
 - ▷ 2 = partly disagree
 - ▷ 3 = neither agree nor disagree
 - ▷ 4 = partly agree
 - ▷ 5 = strongly agree

Likert scale Example

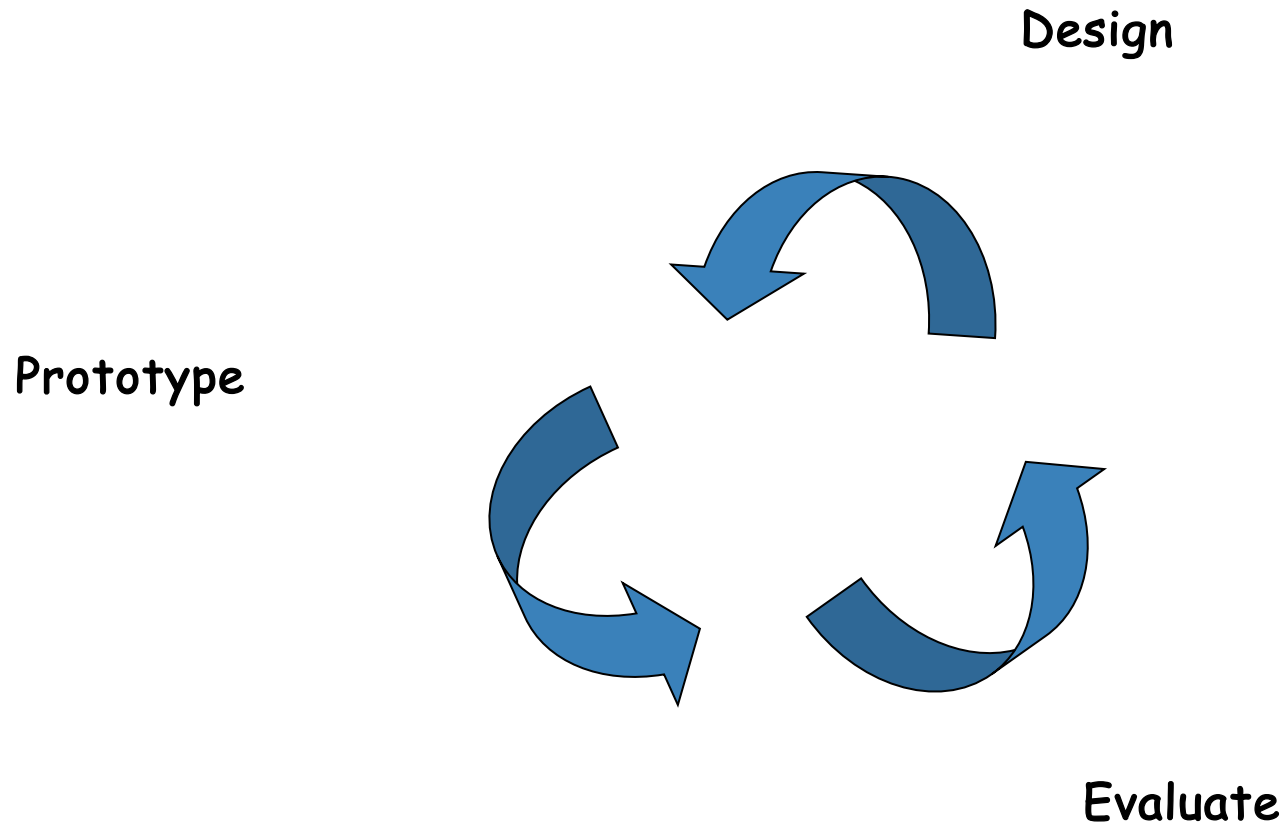
Please indicate the degree to which you agree or disagree with the following statements about the system:

- “It was very easy to learn how to use this system.”
- “Using this system was a very frustrating experience.”
- “I feel that this system allows me to achieve very high productivity.”
- “I worry that many of the things I did with this system may have been wrong.”
- “This system can do all the things I think I would need.”
- “This system is very pleasant to work with.”

Keys to Designing & Building Successful Interfaces

- ▷ Design cycle
- ▷ Customer-centered design
- ▷ Task analysis & contextual inquiry
- ▷ Rapid prototyping
- ▷ Evaluation
- ▷ Programming
- ▷ Iteration

Interface Design Cycle



User-centered Design

▷ “Know the customer”

▷ “User analysis”

○Potential / Primary

○Secondary

Task Analysis & Contextual Inquiry

- ▷ Observe existing work practices
- ▷ Create scenarios of actual use
- ▷ Try-out new ideas before building software

Rapid Prototyping

- ▷ Build a mock-up of design
- ▷ Low fidelity
 - Paper sketches
 - Cut , copy , paste
 - Video segments
- ▷ Interactive prototyping tools
 - HTML , Visual Basic , Adobe XD, Photoshop , etc.

Evaluation

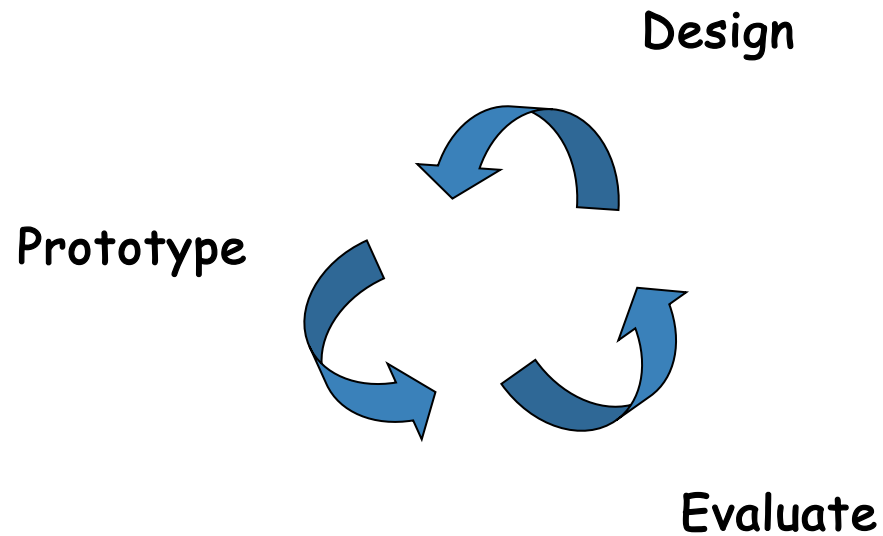
- ▷ Test with real customers(participants)
- ▷ Build models
- ▷ Low-cost techniques
 - Expert evaluation
 - Walkthroughs

Programming

- ▷ Toolkits
- ▷ UI Builders
- ▷ Event models
- ▷ Input / Output models
- ▷ Etc.

Iteration

▷ At every stage



Categories of User

- ▷ **Novices User** of System vs. **Expert User** of System
- ▷ **Minimal** computer Experience vs. **Extensive** Computer Experience
- ▷ **Ignorant** about Domain vs. **Knowledgeable** about Domain
- ▷ Difference between individual user is very high

Usability Slogans

- ▷ Your best guess is not good enough
- ▷ The user is always right
- ▷ The user is not always right
- ▷ Users are not designers
- ▷ Designers are not users
- ▷ Less is more
- ▷ Detail matter



Usability Heuristics

Outline

- ▷ Simple and Natural Dialogue
- ▷ Speak the users' Language
- ▷ Minimize user memory load
- ▷ Consistency
- ▷ Feedback
- ▷ Clearly marked exits

Outline

- ▷ Shortcuts
- ▷ Good Error Messages
- ▷ Prevent Errors
- ▷ Help and Documentation
- ▷ Heuristic Evaluation

Simple and Natural Dialogue

- ▷ Simplified as much as possible
- ▷ Match the natural of users' task as much as possible to simplified the mapping between computer concepts and user concepts
- ▷ Present only enough information
- ▷ Put related information close together
- ▷ Object and action should match the users' sequence
- ▷ Give user the control of sequence
- ▷ User attention begins from top/left
- ▷ UPPERCASE is about 10% slower to read

Simple and Natural Dialogue

▷ For color screen design

- Don't over-do it: limit small # of consistency colors
- Make sure it can be used without the colors for color blind
- Use color to categorize , differentiate and highlight , not to give information (quantitative)

Speak the User's Language

- ▷ Terminology based on users' language
- ▷ Dialogues in users' native language
- ▷ Include nonverbal element—icons
- ▷ View interactions from the users' perspective
 - “We have sold you 100 shares of xyz corp”
 - “You have bought 100 shares of xyz corp”
- ▷ Simply ask users the words and concept they want, sometimes get verbal disagreement

Minimize User Memory Load

- ▷ Recognition rather than remember
- ▷ Describe required format when ask for the input
- ▷ Small # of rules
 - Large # of rules : learn & remember all rules
 - No rule : learn everything & predict the behavior
- ▷ Generic command
 - Copy , cut , paste

Consistency

- ▷ Same command or same action → same effect
- ▷ Same information in the same location, same format
- ▷ Follow standard
- ▷ Not just consistency in screen design but also in task and functionality

Feedback

- ▷ Continuously give feedback to users
- ▷ Not just negative feedback, give positive feedback, partial feedback
- ▷ Give detailed feedback
- ▷ Different types of feedback need different degrees of persistence
- ▷ Information feedback incase of system failure

Feedback

▷ Response time

▷ Basic advice

- 0.1 sec → react instantaneously no feedback is needed
- 1.0 sec → notice delay but no interruption
- 10 sec → limit for keeping the user's attention

Clearly marked Exits

- ▷ Give easy way to get out of situations
 - Cancel
 - Undo
 - Restart
- ▷ Basic principle: users will make errors
- ▷ Visible to user

Shortcuts

- ▷ For experts to get fast access to actions
 - Abbreviations
 - Function keys
 - Right-clicking menu
- ▷ Type ahead – typing the next input before the computer is ready to accept it
- ▷ Most recently used (MRU)
- ▷ System provided default values

Good Error Messages

- ▷ Clear
- ▷ Precise
- ▷ Constructive
- ▷ Polite
- ▷ Multiple-level message

Prevent Errors

- ▷ Better than good error messages
- ▷ Avoid modes
 - Different ways of interpreting same actions
 - Spring – loaded modes – hold down a special key
- ▷ Avoid having too similar commands
 - Case-independent search

Help and Documentation

- ▷ Most users do not read manuals
- ▷ When they need to read, they're in troubles
- ▷ Task-oriented search
- ▷ Lookup tools
- ▷ Online versus printed documents
- ▷ Hypertext

Heuristic Evaluation

- ▷ Goal : to find the usability problems
- ▷ Each evaluator individually & independently inspect
- ▷ Heuristic evaluation versus user testing
- ▷ Outcome : list of usability problems
- ▷ Does not provide a systematic way of generating fixes to the problems



UI Design system

Creating a UI Design System

Moodboards & Colour Palette

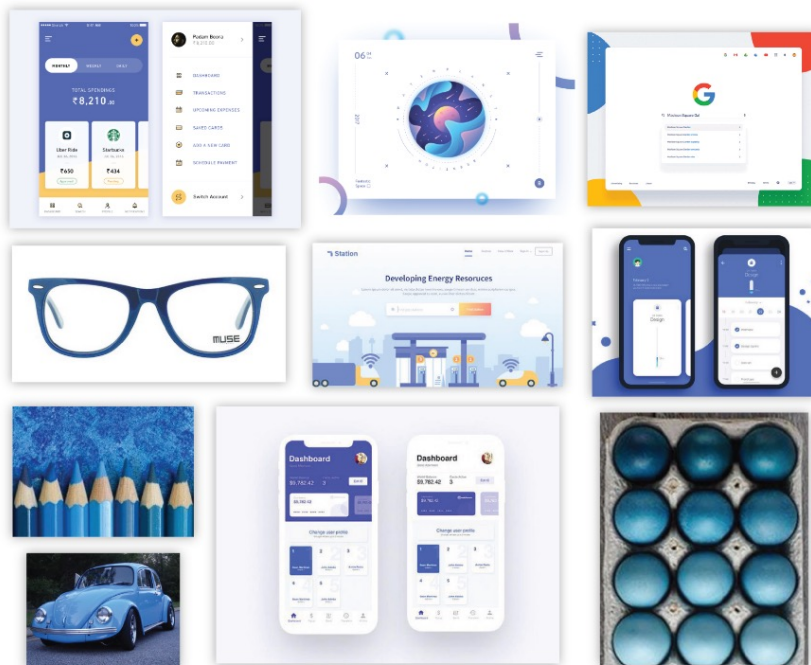
- A very light grey for backgrounds
 - A slightly darker grey for borders, lines, strokes or dividers.
 - A medium grey for subheadings and supporting body copy.
 - A dark grey for main headings, body copy and backgrounds
- Ideally, you'd want to keep tints and tones of your primary colours they may come in handy.



Nutshell

UI Design System - 1

Mood-Board



Colour Palette

Primary Colour



Azure - Blue

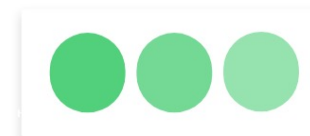
#5680F9

Secondary Colour



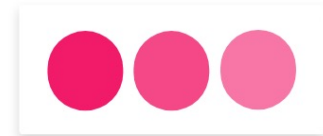
#4C36DD

Success Colour



#5680F9

Failure Colour



#5680F9

Border Radius and shadows

- ▷ let's start setting up the **border radius** and **shadows** for your UI.
- ▷ Border radius set's the tone for your UI Cards, Tables and changes the overall look and feel of your UI.
- ▷ If your radius is more rounded it may come off as **cute** or more **friendly**

The suggestion is to always stay in-between 2–8 px.

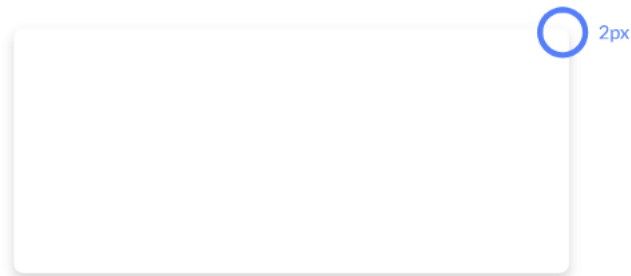


Nutshell

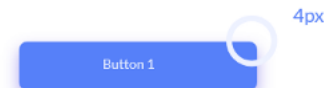
UI Design System - 1

Border Radii

Cards



Buttons



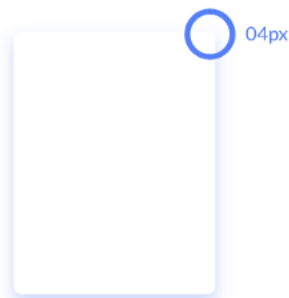
Smaller Buttons



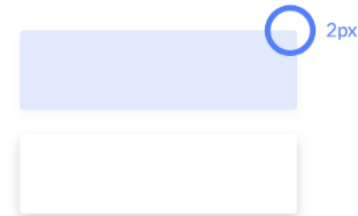
UI Faces



Drop-down



Vertical Tabs



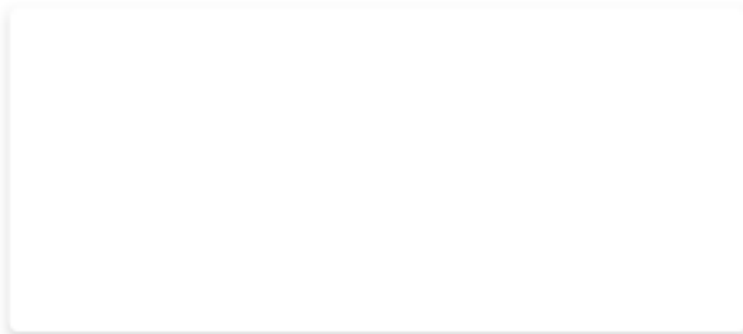


Shadow

Top Nav



Cards



Buttons



Smaller Buttons



Drop-Down



Type scale

- ▷ Default (1em) for standard text that will appear in many places throughout our marketing site, UI etc. 16px is the default browser font size so let's run with that.
- ▷ A slightly larger size for large body copy in a blog for example.
- ▷ A couple of larger sizes for headings and sub-headings.
- ▷ Very large size for section titles.
- ▷ A ridiculously large size maybe for prices on a pricing page for example.
- ▷ We will also need some smaller sizes for smaller body copy, input hints and other secondary text.



Visual TypeScale

Font Selection

Lato Lato Lato *Lato*
Lato Lato Lato *Lato*

Visual TypeScale / Major third

Headings - TypeScale

3.052em (48.83px)

Subheadings - TypeScale

2.441em (39.06px)

Taglines

1.953em (31.25px)

Heading for pages

1.563em (25.00px)

Card / Headings

1.25em (20.00px)

Body

1em (16.00px)

Sub-body in popovers

0.8em (12.80px)

Badges, Tags

0.64em (10.24px)

Icons

- ▷ Make your Icons **uniform**, ensure they use a **similar style guide** to one another.
- ▷ For starters, you could just use Icon set packs which are free to use so you don't spend much time on making icons.



Buttons, Sliders & Progress Bars

- ▷ **Buttons** allow users to take actions, and **make** choices, with a single tap.
- ▷ They **should** be easily findable and identifiable while clearly indicating the action they allow a user to complete.
- ▷ There are various types of buttons.

- Buttons
- Radio Buttons
- Checkboxes
- Navigation Buttons

Button Design



Sliders



On



Off

Sliders, Progress Circles help you define these small elements to make your UI look even more uniform.

Search



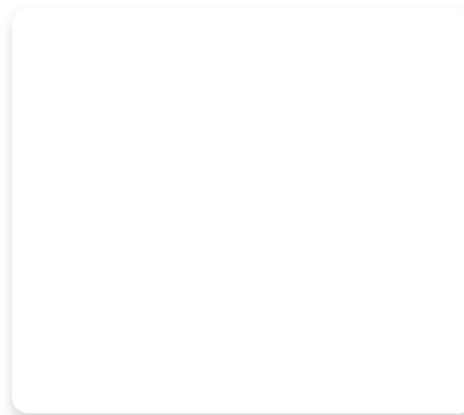
Search on Nutshell

Progress Bar

Loading systems



Progress Circle



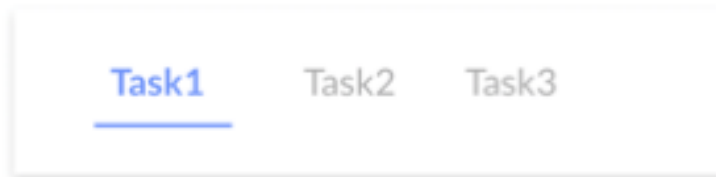
Dropdowns, Tags and Popups (Complex elements)



UI Design System - 7

Tabs

Horizontal Tabs



Vertical Tabs



Tags & Badges

Tags



Rounded Tags



Removeable Tags



Badges



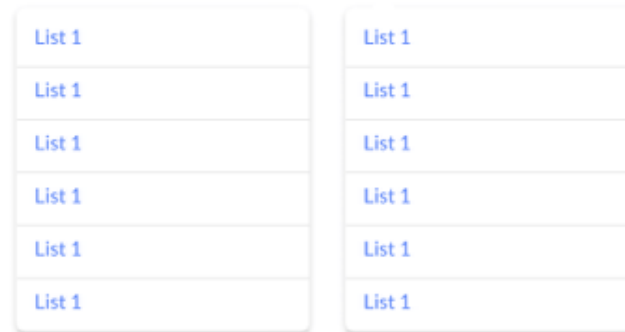
On Icons



On Nav Bar



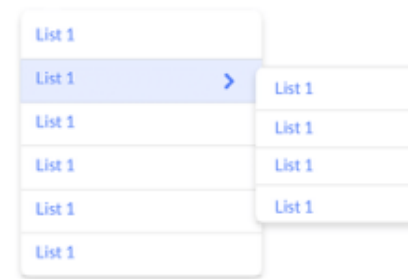
Drop-down



Icon Drop Down



Nested List



Filter

▼

Stops

Nonstop

✓

1 Stop

✓

2+ Stops

✓

Duration

0h 35m – 28h 00m

Cabin

Economy

✓

Business

☐

RESET FILTERS

!

No results, try refining your search.

Don't.

Since red is a brand color, don't also use it to convey an error state.

Filter

Stops

Nonstop

1 Stop

2+ Stops

☒

☒

☒

Duration

0h 35m – 28h 00m

Cabin

Economy

Business

☒

☐

RESET FILTERS

!

No results, try refining your search.

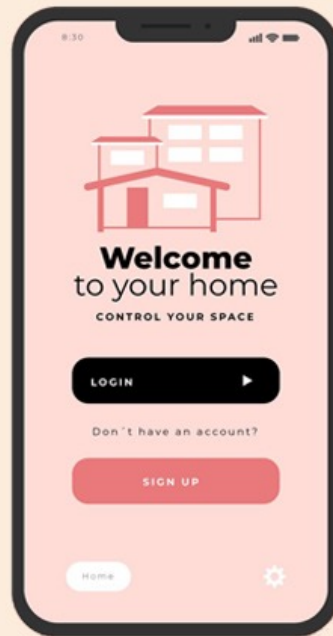
Do.

Choose alternative alert colors that don't use brand coloring.

bemyfriend

UI

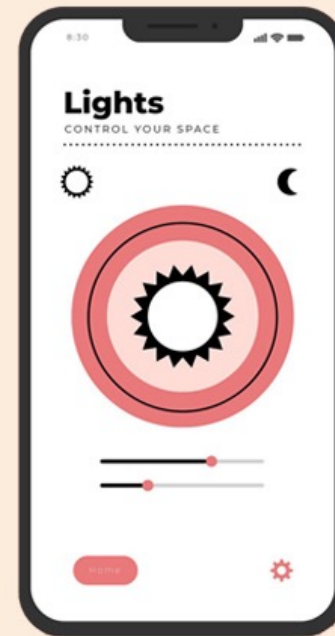
User Interface



ดึงดูด



สวยงาม



ใช้ง่าย

Thanks!

Any questions?