User Interface Design
COMS 4170 · Spring 2019

Goals
1. Build websites that suit the needs and abilities of users.
2. When the needs and abilities of users are uncertain, design systems by learning from iteration.

INSTRUCTOR
Prof. Lydia Chilton
OH: Wednesday 5:30-6:30, CEPSR 612

Please contact staff through Piazza only

TAS
Angelina Wang OH: TBA, TBA
Daniel Li OH: TBA, TBA
Eleanor Murgua OH: TBA, TBA
Katie Pfleger OH: TBA, TBA
Melanie Sawyer OH: TBA, TBA

WEEKLY SCHEDULE
Lecture
Monday, Wednesday
4:10–5:25pm
451 CSB
I’ve been teaching Web Dev & UI for 11 years

TA’d AI courses

6.470 IAP

Web Programming

2010 Winners

First Place - $4000

Joseph Lin

Davide Whitley, Jong-moon Kim

Second Place - $5000

ML Review

Raymond Ho

Third Place - $4000

Lambda Fitness

Ryan Ko, Cal Geoghegan, Jacob Bredthauer

MIT
2008 - 2010

Univ Washington
2012 - 2013

Stanford
2014 - 2016

Columbia
2017 - now

User Interface Design

HCI design studio
CS 247 - Spring 2017

Advanced Web Design Studio
COMS 6998 - Fall 2018

Columbia University
4170 Staff

• Prof. Chilton
  • Office hours: Wednesdays 5:30-6:30 in CEPSR 612

• TAs:
  • Angelina Lam
  • Daniel Li
  • Eleanor Murguia
  • Katie Pfleger
  • Melanie Sawyer

• My goal is to learn all of your names.
Why are user interfaces important?
1613 – 1940s

**Computers**: people who performed calculations
Computers: Tools for Calculation and Symbolic Manipulation
Computers: tools to augment human cognition

Vannevar Bush’s vision of computers

1945
1963: First Graphical User Interface
Ivan Sutherland’s CAD software, Sketchpad
1968: Interaction devices for computer use. Douglas Engelbart’s mouse
Computers: Tools to augment human intelligence.

To augment human intelligence, computers must suit the needs and abilities of people.

Computer-centric interface

Human-centric interface
For physical products, users did not get to experience the usability of the product until after they bought it.

For desktop software, users call expensive support centers, but the costs aren’t “charged” to the software engineers, so they have no motivation to ship great UIs.

On the Web, users experience the usability of a site before they have committed to using it and before they buy it.

UI is now the primary “selling point” of software

Goals of COMS 4170

1. Build websites that suit the needs and abilities of users.

2. When the needs and abilities of users are unclear, design systems by learning from iteration and experimentation.
Grading Buckets

- **A** >= 90%
- 90% > **B** >= 80%
- 80% > **C** >= 70%
- 70% > **D** >= 60%
- **F** > 60%
Grade breakdown

• **Weekly Homework: 70%**
  • 15 homework assignments
  • Each homework worth 5% of grade
  • We will drop your lowest HW grade

• **Participation: 15%**
  • Come to every class and speak up
  • Every class is worth ~0.5% of your grade
  • We will drop your two lowest participation grades.

• **Individual Final Project: 15%**

• No final exam
How we measure participation

• Speak up once during class
• After class, post to piazza two things:
  • what you said (just to remind us)
  • In your own words, write down one thing you learned or remember
• Due by 6pm
Late Policy for Homework

• Assignments are due Friday at 4pm
  • There is a small grace period, which we will not announce.
  • Assume it is 10 minutes.
• Assignments turned in up to 24 hours late get 10% deducted (Sat 4pm)
• Assignments turned in up to 48 hours late get 20% deducted (Sun 4pm)
• Assignments turned in up to 72 hours late get 30% deducted (Mon 4pm)
• After 4pm, work cannot be accepted because we will discuss solutions in class.
• If you are ill or have other difficulties,
  • Email Prof Chilton before the class/due date to let us know.
  • Provide note from a doctor or advising dean
  • Email me a plan for when you will submit the work
  • It can’t be later than 72 hours (Monday 4pm)
Participation make up policy for excused absences

• Email Prof Chilton before the class
• Provide note from a doctor or advising dean
• Write a 1-page summary of the key points of the lecture
• Bring it to a staff member during office hours to go over it.
Attendance is crucial to understanding the material.

Final Grade (participation portion omitted)
Please don’t underestimate this class

Simple, functional design is deceptively difficult
Why is participation 15% of my grade?
Human memory is tree-structured
New knowledge gets appended to the tree.
Where does new knowledge get appended?
To where nodes of tree are currently active.
By guessing about new knowledge before it is presented, you warm up the right place for it in memory.

**Generation**: Guessing before you hear the answer
Once you hear the new knowledge, you want to connect it to connect other to other knowledge so it will trigger when relevant.

**Elaboration**: Relating new knowledge to old topics.
Guess about the new knowledge. Must take risks, you will probably be (partially) wrong.

Relate new knowledge to old topics. This aspect of participation is about providing insights.
Learning from mistakes is good

Tell us about a time that you were wrong about something and learned something from it.

Long answer text
Learning from mistakes is good

Tell us about a time that you were wrong about something and learned something from it.

Long answer text

You are here because you expressed an insight about a time you learned from a mistake.

You were admitted to the this class because you were able to express an insight from a time you made a mistake.
1. Visibility of system status

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
1. Visibility of system status

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
1. Visibility of system status

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
2. Match between system and the real world
The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms.
2. Match between system and the real world
The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms.
2. Match between system and the real world

The system should speak the users' language, using everyday words and concepts familiar to the user, rather than system-oriented terms.
2. **Violation:** Match between system and the real world

The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms.

"I'd spell **crea**t with an e."
3. User control and freedom (Navigation)

Users often choose system functions by mistake and will need easy ways to fix the mistakes. Support undo and redo.
3. User control and freedom (Navigation)

Users often choose system functions by mistake and will need easy ways to fix the mistakes. Support undo and redo.
4. Consistency and standards

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.
4. Consistency and standards

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.
5. Error prevention

Even better than good error messages is a careful design which prevents a problem from occurring in the first place.
5. Error prevention

Even better than good error messages is a careful design which prevents a problem from occurring in the first place.
5. **Violation:** Error prevention

Even better than good error messages is a careful design which prevents a problem from occurring in the first place.
6. Recognition rather than recall

Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another.
6. Recognition rather than recall
Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another.
6. **Violation:** Recognition rather than recall

Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another.
7. Flexibility and efficiency of use

Accelerators — unseen by the novice user — may often speed up the interaction for the expert. Allow users to tailor frequent actions.
7. Flexibility and efficiency of use

Accelerators — unseen by the novice user — may often speed up the interaction for the expert. Allow users to tailor frequent actions.
7. Flexibility and efficiency of use

Accelerators — unseen by the novice user — may often speed up the interaction for the expert. Allow users to tailor frequent actions.
8. Aesthetic and minimalist design

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.
8. Aesthetic and minimalist design

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.
8. **Violation: Aesthetic and minimalist design**

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.
9. Help users recognize, diagnose, and recover from errors

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
9. **Violation** Help users recognize, diagnose, and recover from errors

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
10. Help and documentation

Documentation should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

---

Why is it faster to process a sorted array than an unsorted array?

Here is a piece of C++ code that seems very peculiar. For some strange reason, sorting the data miraculously makes the code almost six times faster.

```cpp
#include <algorithm>
#include <ctime>
#include <iostream>

int main()
{
    // Generate data
    const unsigned arraySize = 32768;
    int data[arraySize];

    for (unsigned c = 0; c < arraySize; ++c)
        data[c] = std::rand() % 256;

    // !!! With this, the next loop runs faster
    std::sort(data, data + arraySize);
    // Test
```
10. **Violation**: Help and documentation:

Documentation should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.
10. **Violation**: Help and documentation
Nielsen’s 10 Usability Heuristics

1. Visibility of system status
2. Match the real world
3. User control and freedom
4. Consistency and Standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Recover from Errors
10. Help and documentation
QUIZ 1 of 3

1. Visibility of system status
2. Match the real world
3. User control and freedom
4. Consistency and Standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Recover from Errors
10. Help and documentation
QUIZ 1 of 3

1. Visibility of system status
2. **Match the real world**
3. User control and freedom
4. Consistency and Standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Recover from Errors
10. Help and documentation
QUIZ 2 of 3

1. Visibility of system status
2. Match the real world
3. User control and freedom
4. Consistency and Standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Recover from Errors
10. Help and documentation
QUIZ 2 of 3

1. Visibility of system status
2. Match the real world
3. User control and freedom
4. Consistency and Standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Recover from Errors
10. Help and documentation
QUIZ 3 of 3

1. Visibility of system status
2. Match the real world
3. User control and freedom
4. Consistency and Standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Recover from Errors
10. Help and documentation
1. Visibility of system status
2. Match the real world
3. User control and freedom
4. Consistency and Standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Recover from Errors
10. Help and documentation
Post on Piazza right after class!

• In reply to the post
  • Today I Said And Learned (TISAL)
    • Said what question you answered.
      • Example: “On slide 44 about why UNIX commands don’t match the real world, I said ‘creat’ doesn’t have an e at the end.”
    • Say one thing you learned or remembered.
      • Example: “I learned that airline boarding passes can be vastly improved to have more aesthetic and minimalist design. They can be redesigned to help people find the key information when and where they need it during the stressful airport experience.”
Homework 1

• Due Friday Jan 25\textsuperscript{th} @ 4:00 PM.
  • Find \textbf{two} examples of web or mobile applications that \textit{positively} exhibit one of the usability heuristics
  • Find \textbf{two} examples of web or mobile applications that \textit{negatively} exhibit one of the usability heuristics
    • How would you fix it?
  • Questions about class policy

The homework is posted on the class website and \textbf{Piazza}
Turn in homework on \textbf{Courseworks}.
(The assignment contains specific turn-in instructions)
User Interface Design
COMS 4170 · Spring 2019

Goals
1. Build websites that suit the needs and abilities of users.
2. When the needs and abilities of users are uncertain, design systems by learning from iteration.

INSTRUCTOR
Prof. Lydia Chilton
OH: Wednesday 5:30-6:30, CEPSR 612

Please contact staff through Piazza only

TAS
Angelina Wang OH: TBA, TBA
Daniel Li OH: TBA, TBA
Eleanor Murguia OH: TBA, TBA
Katie Pfleger OH: TBA, TBA
Melanie Sawyer OH: TBA, TBA

WEEKLY SCHEDULE
Lecture
Monday, Wednesday
4:10-5:25pm
451 CSB