CSE 4451: HUMAN-COMPUTER INTERACTION

Class 14: Inspection & Heuristics

Inspection-based methods

- We have cut prototyping to its minimum
  - Sketches, storyboards, paper prototypes
  - Rapid exploration of potential ideas

- But we need evaluation to guide improvement
  - Can be relatively slow and expensive
  - Study participants can be scarce
  - Can waste participants on obvious problems

Inspection-based methods

- Simulate study participants
  - Instead of actual participants, use inspection to quickly and cheaply identify likely problems
  - Evaluate the prototype without users (e.g., Heuristic Evaluation)
  - Guess how users could use the interfaces
- Inspection methods are rational, not empirical

Heuristic Evaluation

Evaluate a digital artifact through **a set of usability principles** or well-accepted **rules of golden design**, known as “**heuristics.**”

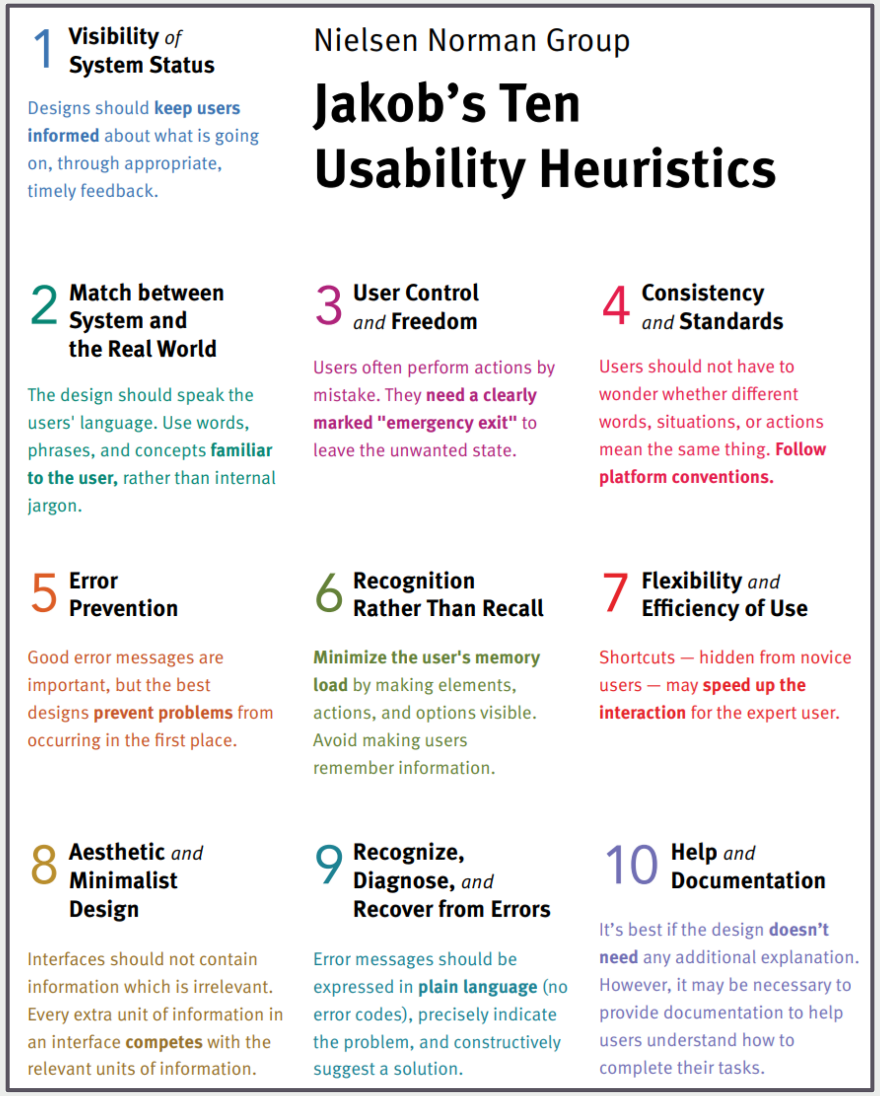
- A HE helps evaluate the overall design;
- The more heuristics you consider, the more comprehensive your evaluation will be;
- Focus/choose heuristics, which are relevant to your project;

Nielsen’s 10 Heuristics

- Too few can be unhelpful, too many can be overwhelming

- Nielsen seeks to create a small set
  - Collects 249 usability problems
  - Collects 101 usability heuristics
  - Rates how well heuristics explain problems
  - Factor analysis to identify key heuristics

Nielsen's 10



Nielsen Norman Group

Jakob's Ten Usability Heuristics

1 **Visibility** *of* **System Status**

Designs should **keep users informed** about what is going on, through appropriate, timely feedback.

2 **Match between System and the Real World**

The design should speak the users' language. Use words, phrases, and concepts **familiar to the user,** rather than internal jargon.

3 **User Control** *and* **Freedom**

Users often perform actions by mistake. They **need a clearly marked "emergency exit"** to leave the unwanted state.

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**

Good error messages are important, but the best designs **prevent problems** from occurring in the first place.

6 **Recognition Rather Than Recall**

Minimize the user's memory load by making elements, actions, and options visible. Avoid making users remember information.

7 **Flexibility** *and* **Efficiency of Use**

Shortcuts — hidden from novice users — may **speed up the interaction** for the expert user.

8 **Aesthetic** *and* **Minimalist Design**

Interfaces should not contain information which is irrelevant. Every extra unit of information in an interface **competes** with the relevant units of information.

9 **Recognize, Diagnose,** *and* **Recover from Errors**

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

10 **Help** *and* **Documentation**

It's best if the design **doesn't need** any additional explanation. However, it may be necessary to provide documentation to help users understand how to complete their tasks.

1. Visibility of system status
2. Match between the system and 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. Help recognize, diagnose, and recover from errors
10. Help and documentation

1. Visibility

- Visibility of system status
  - The system should always **keep people informed** about what is going on, through appropriate **feedback** within a reasonable time.

- Anytime a person is wondering what state the system is in, or the result of some action, this is a visibility violation.

2. Real World Match

- Match between system and the real world
  - The system should speak a person’s language, with words, phrases, and concepts familiar to the person, rather than system-oriented terms.

  - Follow real-world conventions, making information appear in a natural and logical order.

- Refers to word and language choice, mental model, metaphor

3. User in control

- User control and freedom
  - People often choose system functions by mistake and will need a clearly marked “emergency exit” to leave the unwanted state without having to go through an extended dialog.

- Support undo and redo.

- Not just for navigation exits, but for getting out of any situation or state

4. Consistency

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

- Internal consistency is consistency throughout the same product.

- External consistency is consistency with other products in its class.

5. Error prevention

- Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present people with a confirmation option before they commit to the action.

- Try to commit errors and see how they are handled. Could they have been presented?

6. Recognition not Recall

- Recognition rather than recall
  - Minimize a person’s memory load by making objects, actions, and options visible. A person should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be clearly visible or easily retrievable whenever appropriate.

- People should never carry a memory load

- Problems with affordances might go here
  - Hidden affordance: remember where to act

7. Flexibility and Efficiency

- Flexibility and efficiency of use
  - Accelerators, while unseen by novices, may often speed up the interaction for experts such that the system can cater to both the inexperienced and experienced use. Allow people to tailor frequent interactions.

- Concerns anywhere users have repetitive actions that must be done manually. Also, concerns allowing multiple ways to do things.

8. Aesthetic design

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

- Not just about “ugliness”. About clutter, overload of visual field, visual noise, distracting animations

8. Aesthetic design

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

- Not just about “ugliness”. About clutter, overload of visual field, visual noise, distracting animations

9. Error recovery

- 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.

- Error prevention is about preventing errors before they occur. This is about after they occur

10. Help

- Help and documentation
  - Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on a person’s task, list concrete steps to be carried out, and not be too large.

- This does not mean that a person must be able to ask for help on every single item.

Heuristic Evaluation Process

- Evaluators go through interface several times
  - Inspect various dialog events
  - Compare with list of usability principles

- Usability principles
- Nielsen’s heuristics
- Supplemental list of category-specific heuristics (competitive analysis of testing existing products)

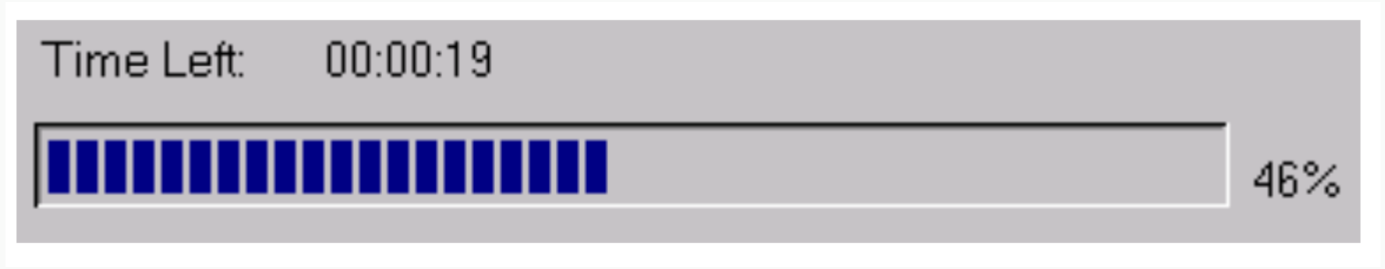
- Use violations to redesign/fix problems

Heuristic Evaluation Process: Example

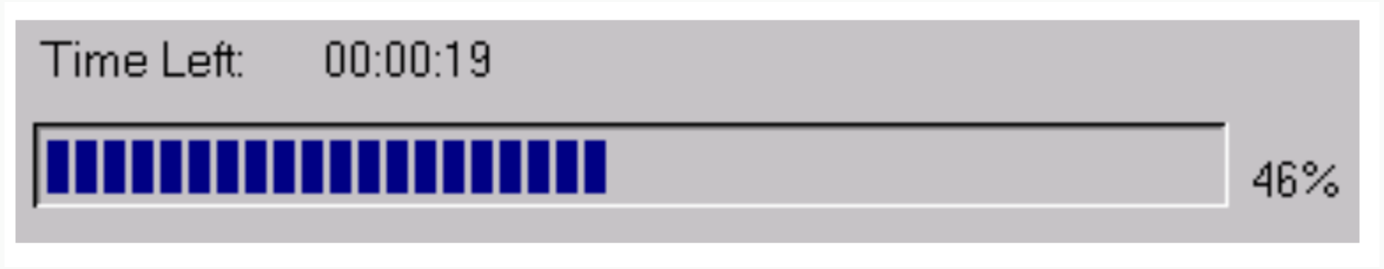
- Can’t copy information from one window to another
  - **Violates** “Minimize memory load” (H6)
  - **Fix**: allow copying

- Typography uses different fonts in 3 dialog boxes
  - **Violates** “Consistency and standards” (H4)
  - Slows users down, probably won’t be found by usability testing
  - **Fix**: pick a single format for the entire interface

Examples

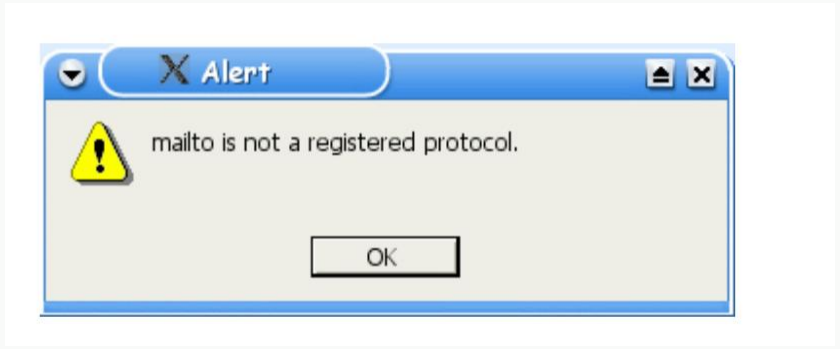


Examples



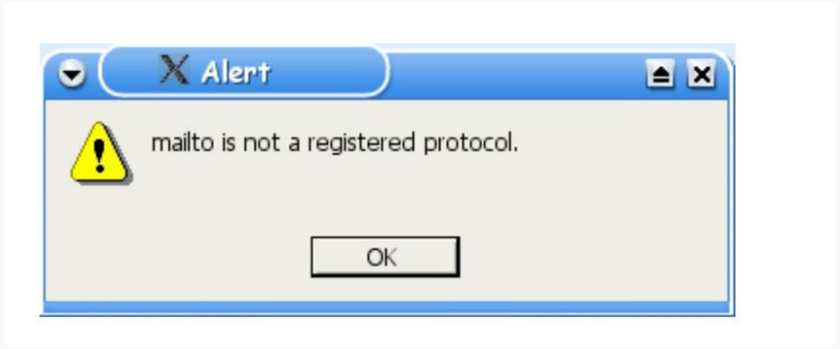
Visibility of system status
• Pay attention to response time
• 0.1 sec: no special indicators needed (why?)
• 1.0 sec: person tends to lose track of data
• 10 sec: maximum duration if person to stay focused
Longer delays require progress bars

Examples



mailto is not a registered protocol.

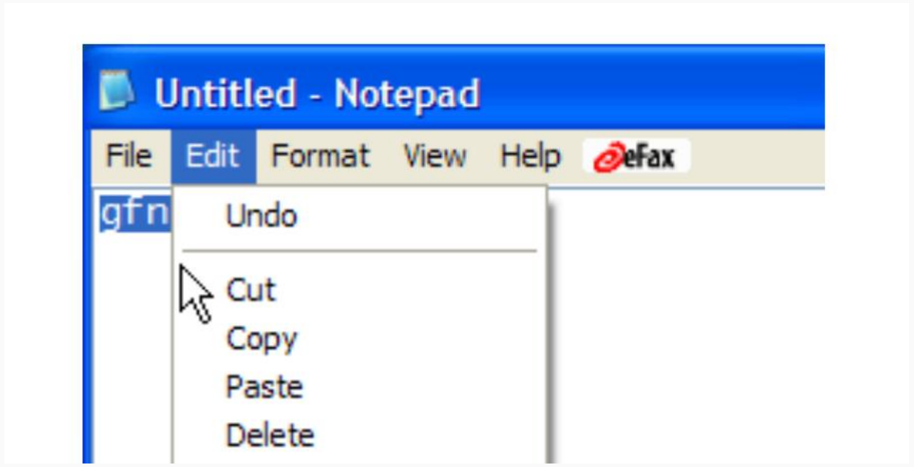
Examples


Alert

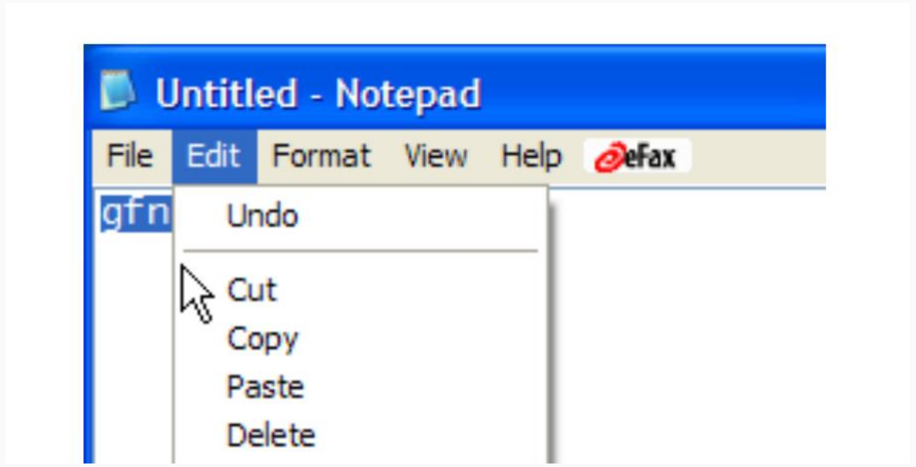
mailto is not a registered protocol.

Mailto? protocol?
Match system to real world
Speak the person’s language

Examples


Untitled - Notepad
File
Edit
Format
View
Help
eFax
gfn
Undo
Cut
Copy
Paste
Delete

Examples



Flexibility and efficiency of use
Accelerators for experts (e.g., keyboard shortcuts)
Allow tailoring of frequent actions (e.g., macros)

How to perform heuristic evaluation

- At least two passes for each evaluator
  - First to get feel for flow and scope of system
  - Second to focus on specific elements

- If system is walk-up-and-use or evaluators are domain experts, no assistance needed
  - Otherwise might supply evaluators with scenarios

- Each evaluator produces list of problems
  - Explain why with reference to heuristic
  - Be specific & list each problem separately

How to perform heuristic evaluation

1. [H4 consistency] [Severity 3] [Fix 4]

The interface used the string “Save” on the first screen for saving the person’s file, but used the string “Write File” on the second screen. People may be confused by this different terminology for the same function.

Fix: Change the second screen to “Save”.

- At least two passes for each evaluator
  - First to get feel for flow and scope of system
  - Second to focus on specific elements
- If system is walk-up-and-use or evaluators are domain experts, no assistance needed
  - Otherwise might supply evaluators with scenarios
- Each evaluator produces list of problems
  - Explain why with reference to heuristic
  - Be specific & list each problem separately

How to perform heuristic evaluation

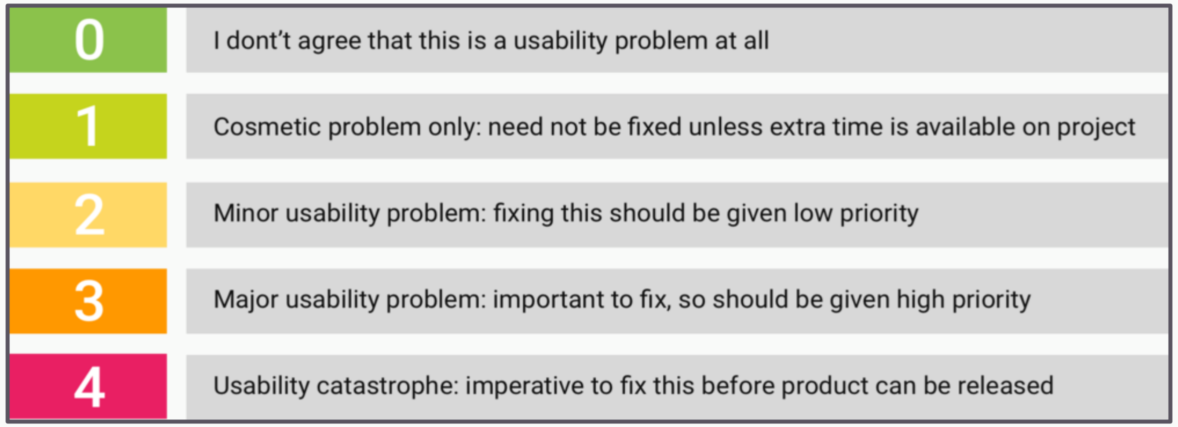
Are the problems…

- … **critically impeding** the user / **preventing** the user from **succeeding** in their main task?

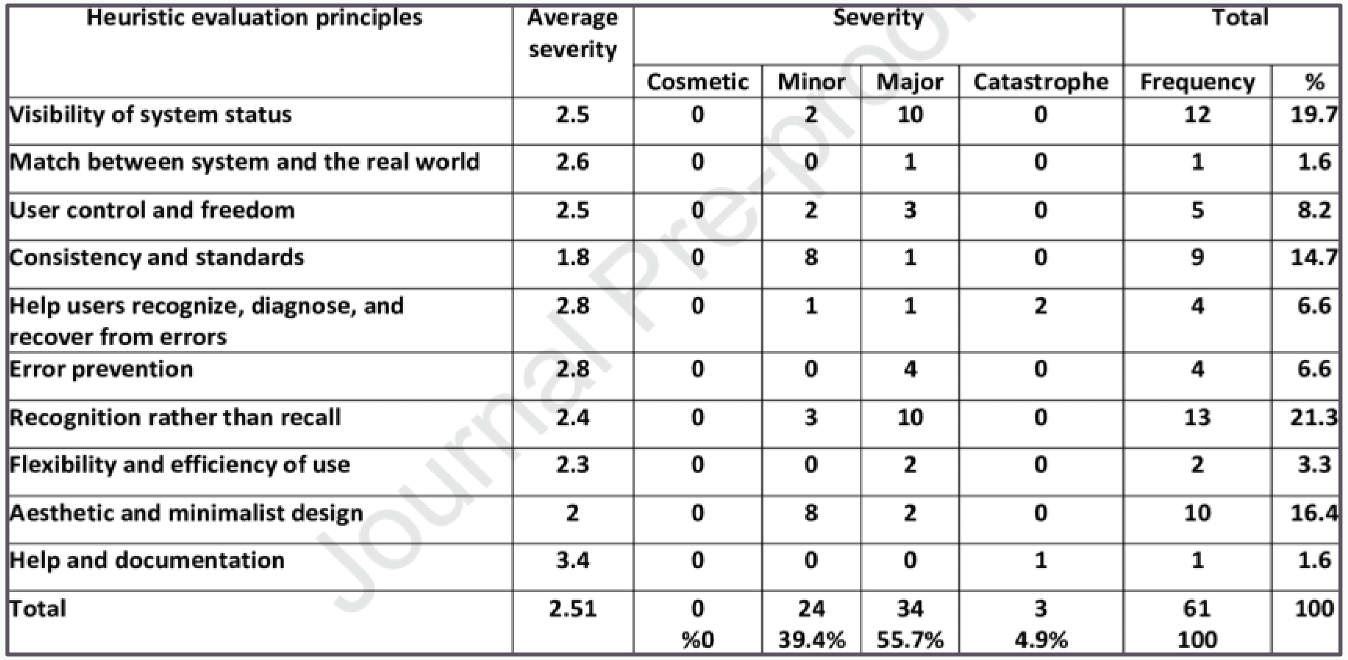
- … **breaking the flow** and the user can't recover?

- … **slowing down the user**, but still allowing them to complete the task?

A Rating Scale for Severity of Problems



0	I dont't agree that this is a usability problem at all
1	Cosmetic problem only: need not be fixed unless extra time is available on project
2	Minor usability problem: fixing this should be given low priority
3	Major usability problem: important to fix, so should be given high priority
4	Usability catastrophe: imperative to fix this before product can be released



Heuristic evaluation principles	Average severity	Severity				Total	
		Cosmetic	Minor	Major	Catastrophe	Frequency	%
Visibility of system status	2.5	0	2	10	0	12	19.7
Match between system and the real world	2.6	0	0	1	0	1	1.6
User control and freedom	2.5	0	2	3	0	5	8.2
Consistency and standards	1.8	0	8	1	0	9	14.7
Help users recognize, diagnose, and recover from errors	2.8	0	1	1	2	4	6.6
Error prevention	2.8	0	0	4	0	4	6.6
Recognition rather than recall	2.4	0	3	10	0	13	21.3
Flexibility and efficiency of use	2.3	0	0	2	0	2	3.3
Aesthetic and minimalist design	2	0	8	2	0	10	16.4
Help and documentation	3.4	0	0	0	1	1	1.6
Total	2.51	0 %0	24 39.4%	34 55.7%	3 4.9%	61 100	100

Phases of Heuristic Evaluation

- Pre-evaluation and training
  - Give expert evaluators needed domain knowledge & information on the scenario
- Evaluation
  - Individuals evaluate interface and make lists of problems
- Severity rating
  - Determine how severe each problem is
- Aggregation
  - Group meets and aggregates problems (w/ ratings)
- Debriefing
  - Discuss the outcome with the design team

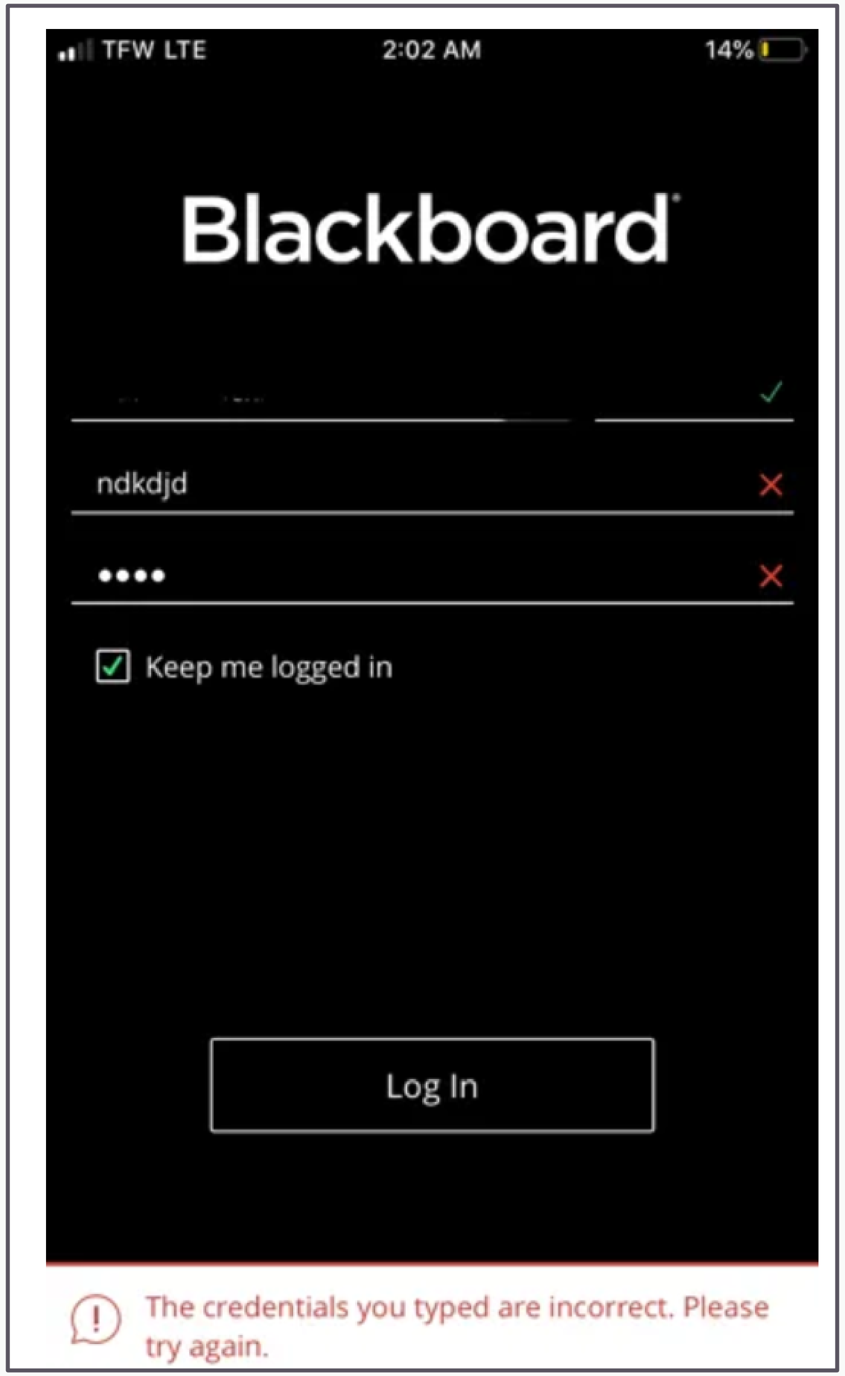
What is wrong here?

What heuristics are violated?

→ #1 Visibility of System Status (not clear what is going on)
→ #5 Error prevention (the error message is not telling us what the problem is)

What's the severity of the problem?

→ **4 | Usability Catastrophe:** If users can't login, they can't use the system.


TFW LTE 2:02 AM 14%

Blackboard

ndkdjd

••••

✓ Keep me logged in

Log In

The credentials you typed are incorrect. Please try again.

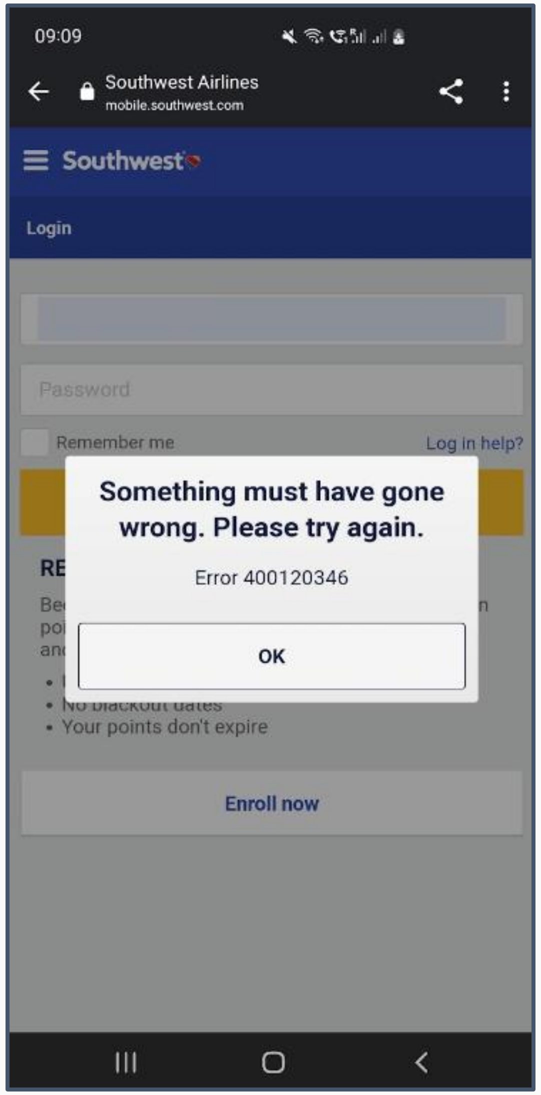
What is wrong here?

What heuristics are violated?

→ #9: Help users recognize, diagnose, and recover from errors (what is Error 400120346?)

What's the severity of the problem?

→ **4 | Usability Catastrophe:** If users can't login, they can't use the system.


09:09
Southwest Airlines
mobile.southwest.com
Southwest
Login
Password
Remember me
Log in help?

Something must have gone wrong. Please try again.

Error 400120346

OK

• No blackout dates
• Your points don't expire

Enroll now

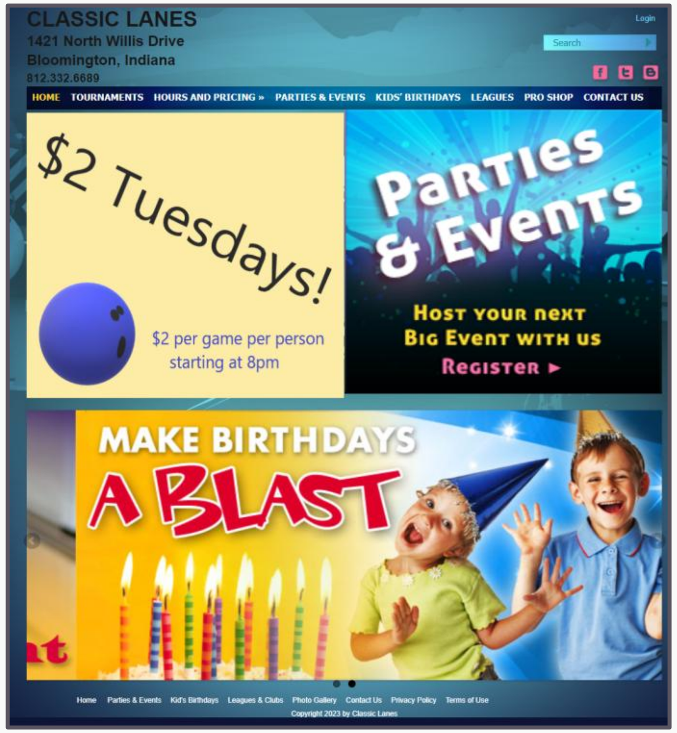
What is wrong here?

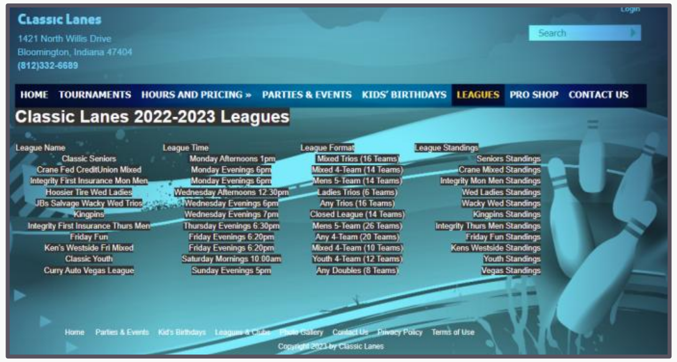
What heuristics are violated?

→ #8: Aesthetic and minimalist design (Font-Background Contrast, switching font color, unreadable text, redundant menus, etc.)

What's the severity of the problem?

→ **2 | Minor Usability Problem:** Fixing this has low priority.


CLASSIC LANES
1421 North Willis Drive
Bloomington, Indiana
812.332.6689
Login
Search
HOME TOURNAMENTS HOURS AND PRICING » PARTIES & EVENTS KIDS' BIRTHDAYS LEAGUES PRO SHOP CONTACT US
$2 Tuesdays!
$2 per game per person starting at 8pm
Parties & Events
Host your next Big Event with us
Register ►
MAKE BIRTHDAYS A BLAST
Home Parties & Events Kid's Birthdays Leagues & Clubs Photo Gallery Contact Us Privacy Policy Terms of Use
Copyright 2023 by Classic Lanes


Classic Lanes
1421 North Willis Drive
Bloomington, Indiana 47404
(812)332-6689
Login
Search
HOME TOURNAMENTS HOURS AND PRICING » PARTIES & EVENTS KIDS' BIRTHDAYS LEAGUES PRO SHOP CONTACT US
Home Parties & Events Kid's Birthdays Leagues & Clubs Photo Gallery Contact Us Privacy Policy Terms of Use
Copyright 2023 by Classic Lanes

Classic Lanes 2022-2023 Leagues

League Name	League Time	League Format	League Standings
Classic Seniors	Monday Afternoons 1pm	Mixed Trios (16 Teams)	Seniors Standings
Crane Fed CreditUnion Mixed	Monday Evenings 6pm	Mixed 4-Team (14 Teams)	Crane Mixed Standings
Integrity First Insurance Mon Men	Monday Evenings 6pm	Mens 5-Team (14 Teams)	Integrity Mon Men Standings
Hoosier Tire Wed Ladies	Wednesday Afternoons 12:30pm	Ladies Trios (6 Teams)	Wed Ladies Standings
JBs Salvage Wacky Wed Trios	Wednesday Evenings 6pm	Any Trios (16 Teams)	Wacky Wed Standings
Kingpins	Wednesday Evenings 7pm	Closed League (14 Teams)	Kingpins Standings
Integrity First Insurance Thurs Men	Thursday Evenings 6:30pm	Mens 5-Team (26 Teams)	Integrity Thurs Men Standings
Friday Fun	Friday Evenings 6:20pm	Any 4-Team (20 Teams)	Friday Fun Standings
Ken's Westside Fri Mixed	Friday Evenings 6:20pm	Mixed 4-Team (10 Teams)	Kens Westside Standings
Classic Youth	Saturday Mornings 10:00am	Youth 4-Team (12 Teams)	Youth Standings
Curry Auto Vegas League	Sunday Evenings 5pm	Any Doubles (8 Teams)	Vegas Standings

Cognitive Walkthrough

- Evaluation method based on:
  - A person works through an interface in an exploratory manner
  - A person has goals
  - The person is applying means-ends reasoning to work out how to accomplish these goals

- Evaluation by an expert, who goes through a task while simulating this cognitive process

Cognitive walkthrough: need four things

- Person description, including level of experience and any assumptions made by the designer
- System description (e.g., paper prototype)
- Task description, specifying the task the expert has to carry out, from a person’s point of view
- Action sequence describing the system display and the actions needed to complete the task. One system display and one action together are one step.

Believability

- Will the person be trying to produce whether effect the action has?

- Will the person be able to nice that the correct action is available?

- Once the person finds the correct action at the interface, will they know that it is the right one for the effect they are trying to produce?

- After the action is taken, will the person understand the feedback given?

Inspection vs Usability Testing

- Inspection
  - Is much faster
  - Does not require interpreting participant actions
  - May miss problems or find false positives

- Usability testing
  - More accurate, by definition
  - Account for actual people and tasks

Inspection methods	User testing
Evaluate the prototype **without users** (e.g. Heuristic Evaluation) ***Guess*** how users could use the interfaces	Evaluate the prototype **with users** ***Observe*** how users use the interfaces