UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

SOFTWARE ENGINEERING – LECTURE 17

Guidelines and Principles in Human-Computer Interaction

Prof. Luigi Libero Lucio Starace

luigiliberolucio.starace@unina.it

https://luistar.github.io

https://www.docenti.unina.it/luigiliberolucio.starace

Guidelines and Principles in HCI

- Guidelines and principles for good UI design have been established over the years
- These guidelines are applicable to most interactive systems
- Derived from experience and refined over decades
- No pretense of completeness or universality
  - Require validation and tuning for specific design domains
- Nonetheless, useful for students and practitioners

Guidelines and Principles in HCI

Many authors presented more or less popular sets of guidelines:

- Eight Golden Rules by Ben Shneiderman [1]
- Ten Usability Heuristics by Nielsen and Molich [2]
- Twenty usability principles studied by Holcomb and Tharp [3]
- Eight design principles for successful guessing by Polson and Lewis [4]
- Nineteen artifact claims analysis questions by Carroll and Rosson [5]

For additional heuristics check out Reference [6]

Guidelines and Principles in HCI

- Guidelines and principles can be useful:
  - To **guide** the design phase
  - To **evaluate** a UI to find usability problems (we’ll see in a few lectures)
- Overlaps exists between the different sets of guidelines/principles
  - Recurring themes: Prevent Errors, Minimize Memory Load, …
- In today’s lecture, we’re going to focus on the usability heuristics by Nielsen-Molich and Shneiderman’s golden rules
  - We’ll go over the union of both sets

Shneiderman’s Eight Golden Rules

1. Strive for Consistency
2. Seek Universal Usability
3. Offer Informative Feedback
4. Design Dialogs to Yield Closure
5. Prevent Errors
6. Permit Easy Reversal of Actions
7. Keep Users in Control
8. Reduce Short-term Memory Load

Ben Shneiderman

Nielsen-Molich Ten Usability Heuristics

1. Simple and Natural Dialogue
2. Speak the User’s Language
3. Minimize User Memory Load
4. Consistency
5. Feedback
6. Clearly Marked Exits
7. Shortcuts
8. Good error messages
9. Prevent Errors
10. Help and Documentation

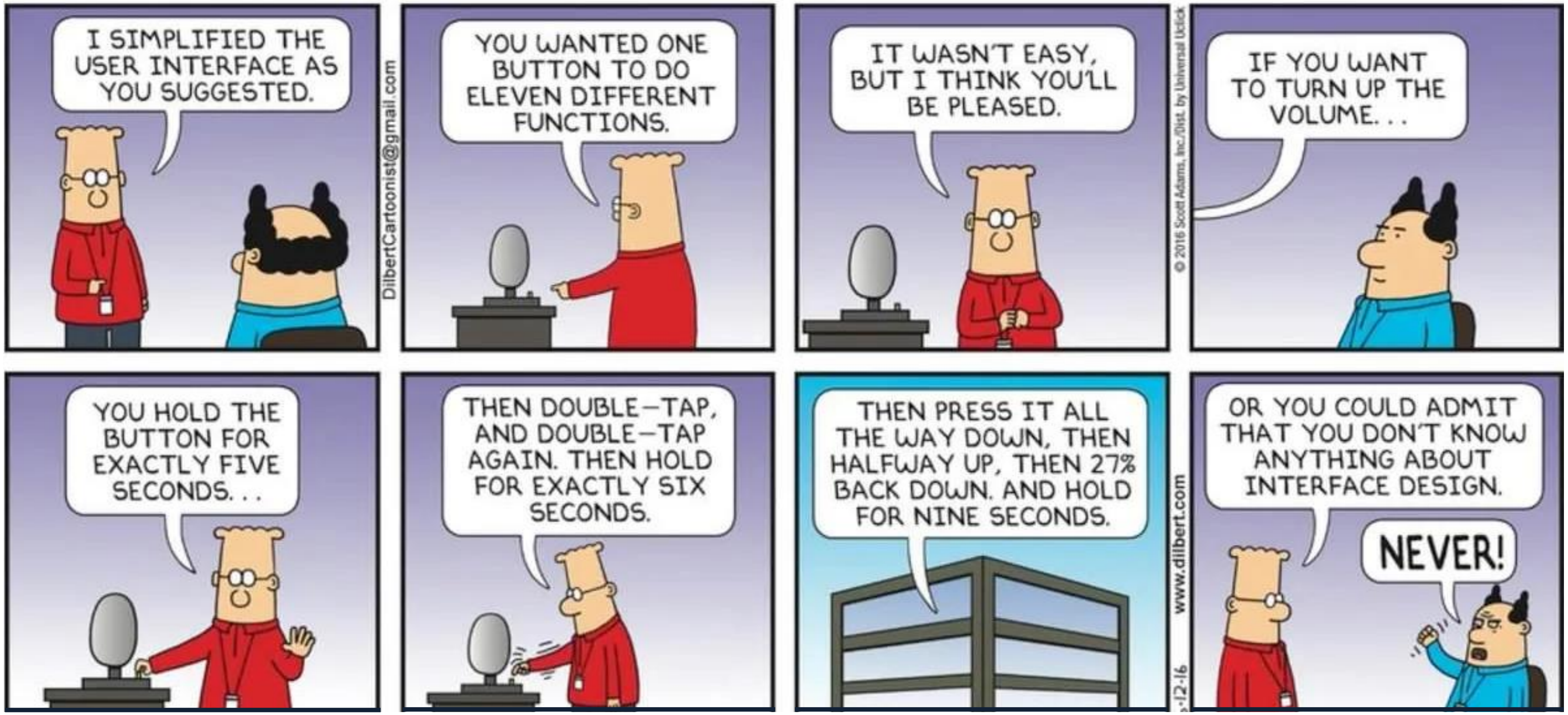
Jakob Nielsen
(pic from nngroup.com)

Rolph Molich
(pic from dialogdesign.dk)

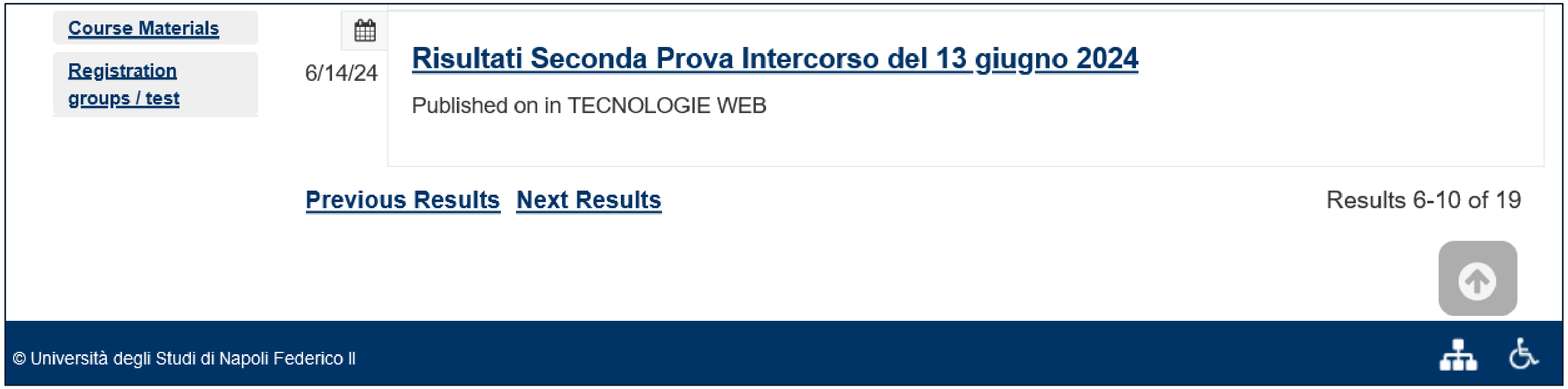
Simple and Natural Dialogue

- UI should be as **simple** as possible (but no simpler!)
  - **Less is more**: Every additional feature/item of information is one more thing to learn, possibly misunderstand, and search through when looking for the thing we want
  - Novice users can get overwhelmed by too much information
  - Hick's Law!
- UI should match the user's task as **naturally** as possible (**mappings** and **metaphors**!)
  - The digital version of a form is organized in the same way as the paper version
  - A compass app functions much like an actual compass

Less is more, until less is less


I SIMPLIFIED THE USER INTERFACE AS YOU SUGGESTED.
YOU WANTED ONE BUTTON TO DO ELEVEN DIFFERENT FUNCTIONS.
IT WASN'T EASY, BUT I THINK YOU'LL BE PLEASED.
IF YOU WANT TO TURN UP THE VOLUME. . .
YOU HOLD THE BUTTON FOR EXACTLY FIVE SECONDS. . .
THEN DOUBLE-TAP, AND DOUBLE-TAP AGAIN. THEN HOLD FOR EXACTLY SIX SECONDS.
THEN PRESS IT ALL THE WAY DOWN, THEN HALFWAY UP, THEN 27% BACK DOWN. AND HOLD FOR NINE SECONDS.
OR YOU COULD ADMIT THAT YOU DON'T KNOW ANYTHING ABOUT INTERFACE DESIGN.
NEVER!
DilbertCartoonist@gmail.com
© 2016 Scott Adams, Inc./Dist. by Universal Uclick
www.dilbert.com

Simple and (un)natural dialogue: example


Course Materials

Registration groups / test

6/14/24

Risultati Seconda Prova Intercorso del 13 giugno 2024

Published on in TECNOLOGIE WEB

Previous Results Next Results

Results 6-10 of 19

© Università degli Studi di Napoli Federico II

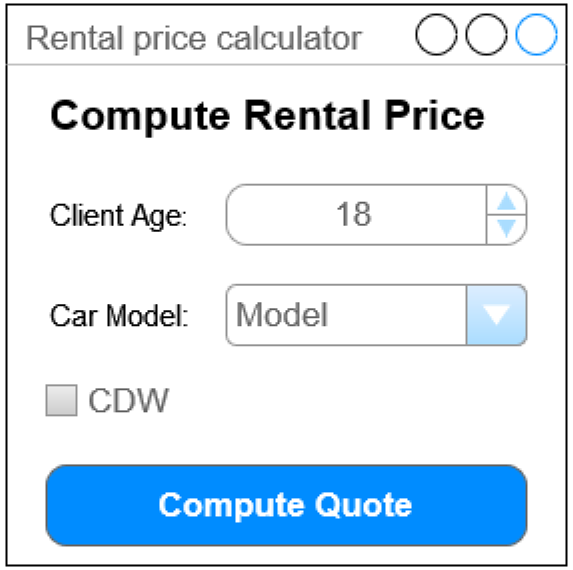
- In the docenti.unina platform, the news published by a professor are sorted (most recent first) and paginated
  - **Previous Results** leads to more recent news
  - **Next Results** leads to older news

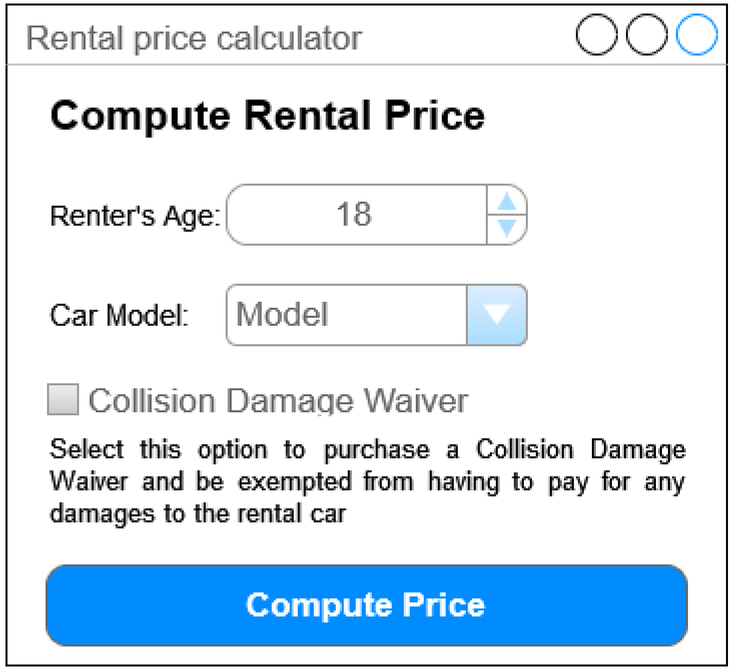
Speak the user’s language

Dialogue should be expressed in words, phrases and concepts **familiar to the user** rather than in system-oriented terms (**human-centered design**!)

- Dialogues should be in the user’s native language (**localization**)
  - Not limited to text, but also non-verbal elements like icons!
- Beware of which words you use
  - When designing for the general public, use words everyone can understand, with their standard meaning
  - When designing for a user group with its own specialized, domain-specific terminology, make use of the specialized terms

Speak the user's language

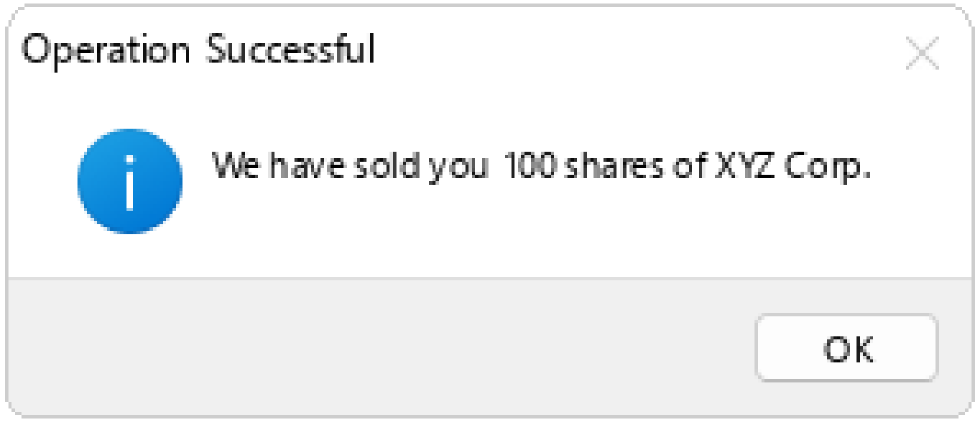




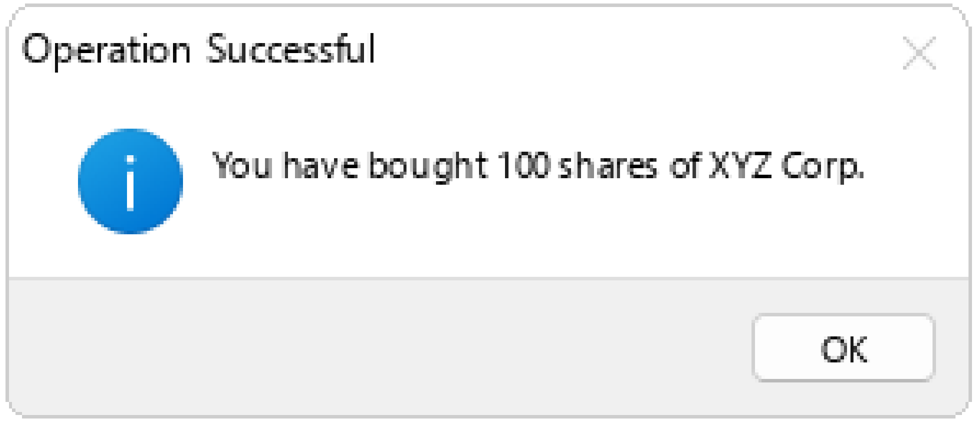
Select this option to purchase a Collision Damage Waiver and be exempted from having to pay for any damages to the rental car

Speak the user’s language

Speaking the user’s language also involves viewing interactions from the user’s perspective


Operation Successful

We have sold you 100 shares of XYZ Corp.

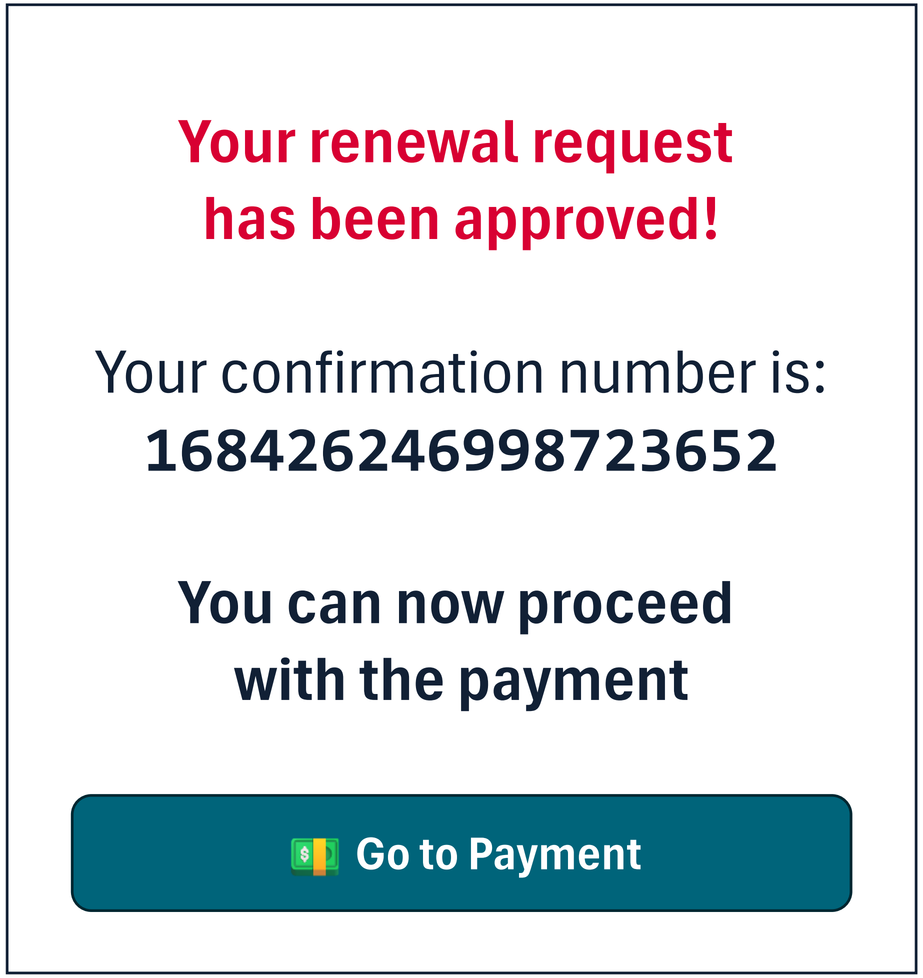

Operation Successful

You have bought 100 shares of XYZ Corp.

Minimize User Memory Load

- Computer momories are very effective at precisely remembering things
- The human working memory not so much! (Remember MHP?)
- The UI we design should take over the burden of memory from the user as much as possible
- How to do that?
  - Recognition is better than recall
  - When asking users to provide inputs, describe the required format and provide and example. Explicitly state the ranges and of legal inputs (if any)

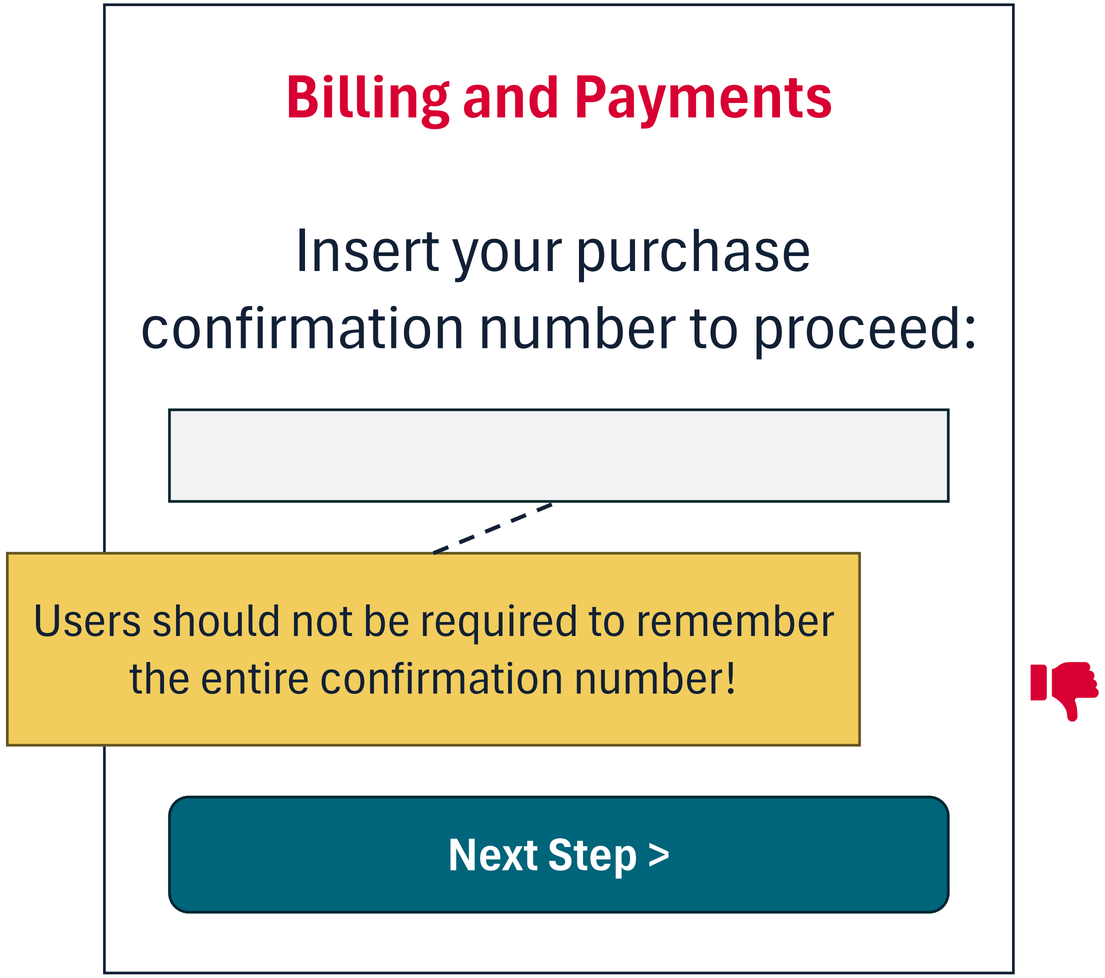
Minimize User Memory Load: Examples



Your renewal request has been approved!

Your confirmation number is: **16842624698723652**

You can now proceed with the payment



Billing and Payments

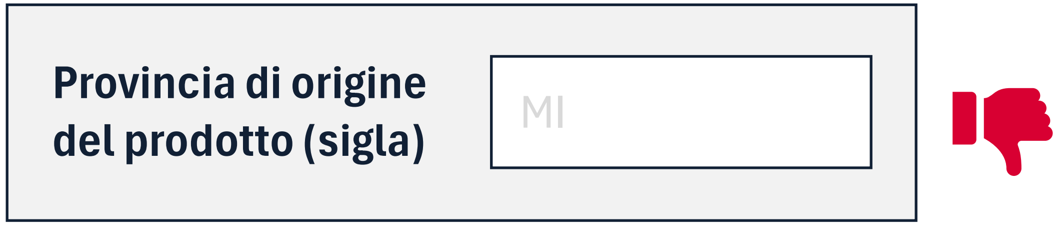
Insert your purchase confirmation number to proceed:

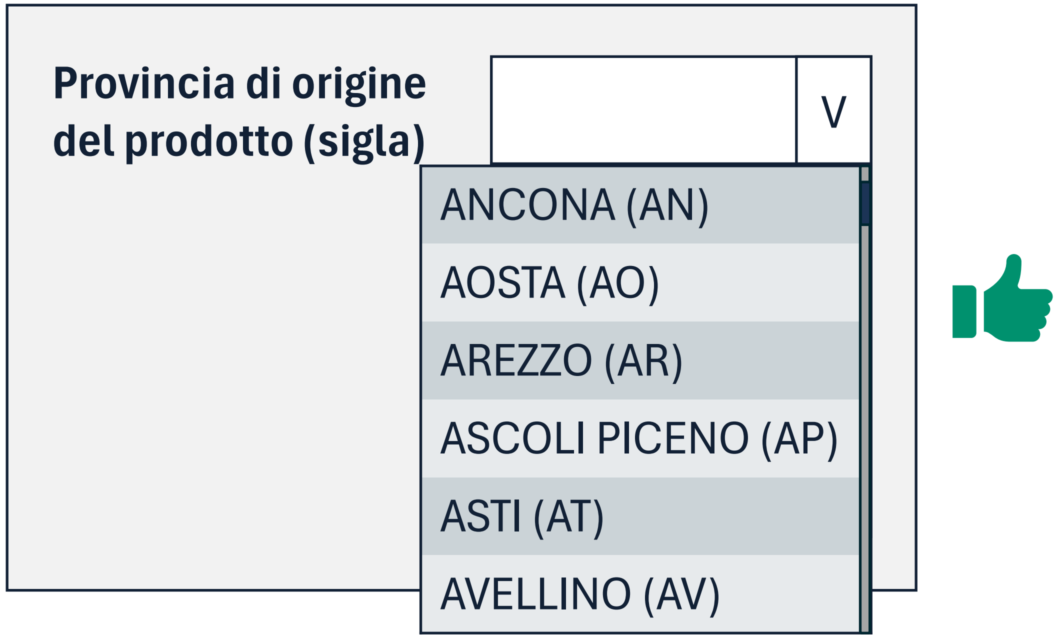
Users should not be required to remember the entire confirmation number!

Minimize User Memory Load: Examples

Users need to insert the two-letter code for one of Italy's provinces

- Accepting such input via a text field requires the users to remember the code of the provide they want to select
  - What is the code for Lecce?
  - And for Lecco?
- Using a dropdown list showing the full names and codes for all takes over the burden of having to remember the codes for the 110 italian provinces!


Provincia di origine del prodotto (sigla)


Provincia di origine del prodotto (sigla)

Consistency

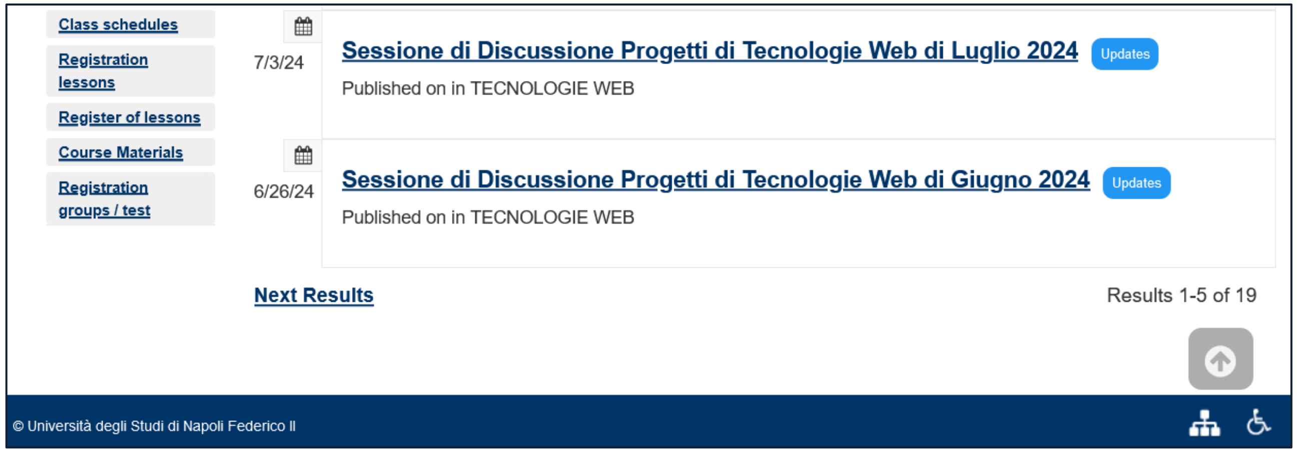
To be usable, a system should exhibit **internal** and **external consistency**

- **Internal Consistency** (within the product itself or a family of products)
  - Consistent sequences of actions should be required in similar situations
    - Deleting a customer and deleting a supplier should require a similar sequence of actions
  - The same information should be presented in the same way and in the same location on all screens
- **External Consistency** (with established conventions)

Internal (In)consistency: Examples

In the docenti.unina platform, the (sic) **Next Results** control changes position

- In the first page (top figure), it's the leftmost control
- In the subsequent pages (bottom figure), it's shifted to the right



Class schedules
Registration lessons
Register of lessons
Course Materials
Registration groups / test

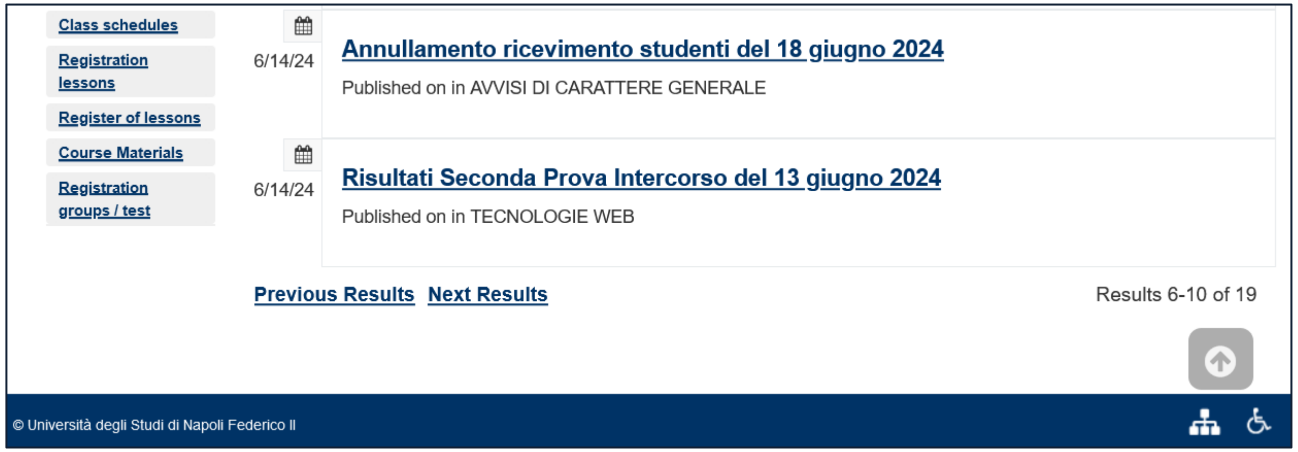
7/3/24 Sessione di Discussione Progetti di Tecnologie Web di Luglio 2024 Updates
Published on in TECNOLOGIE WEB

6/26/24 Sessione di Discussione Progetti di Tecnologie Web di Giugno 2024 Updates
Published on in TECNOLOGIE WEB

Next Results

Results 1-5 of 19

© Università degli Studi di Napoli Federico II



Class schedules
Registration lessons
Register of lessons
Course Materials
Registration groups / test

6/14/24 Annullamento ricevimento studenti del 18 giugno 2024
Published on in AVVISI DI CARATTERE GENERALE

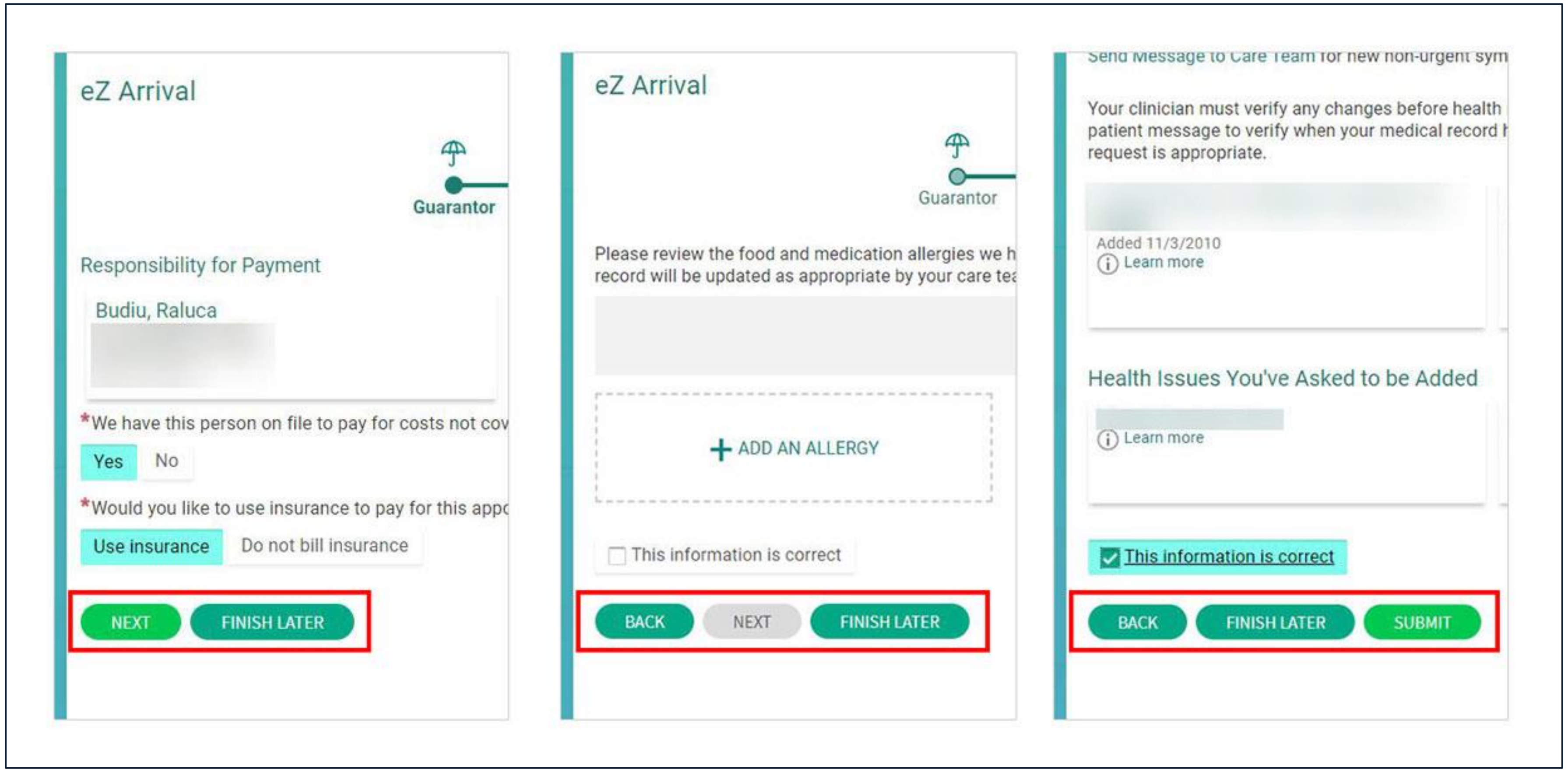
6/14/24 Risultati Seconda Prova Intercorso del 13 giugno 2024
Published on in TECNOLOGIE WEB

Previous Results Next Results

Results 6-10 of 19

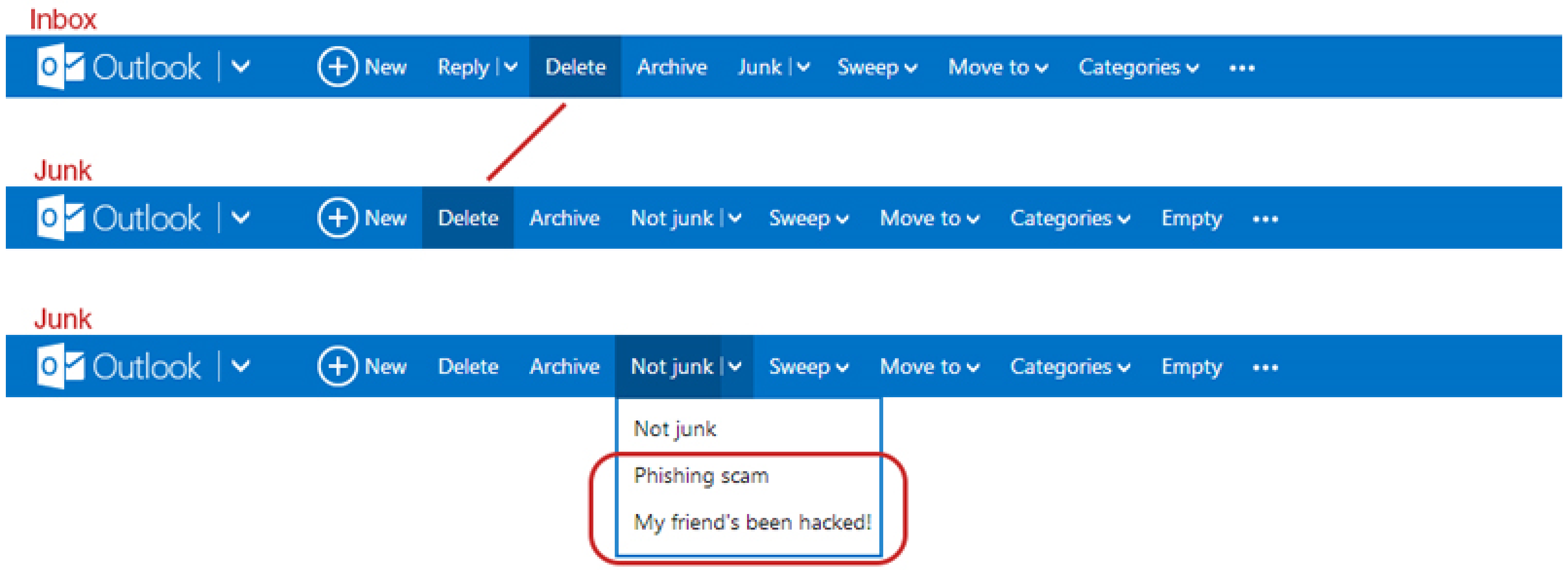
© Università degli Studi di Napoli Federico II

Internal (In)consistency: Examples


eZ Arrival
Guarantor
Responsibility for Payment
Budiu, Raluca
*We have this person on file to pay for costs not cov
Yes
No
*Would you like to use insurance to pay for this appo
Use insurance
Do not bill insurance
NEXT
FINISH LATER
eZ Arrival
Guarantor
Please review the food and medication allergies we h
record will be updated as appropriate by your care tea
ADD AN ALLERGY
This information is correct
BACK
NEXT
FINISH LATER
Send Message to Care Team for new non-urgent sym
Your clinician must verify any changes before health
patient message to verify when your medical record h
request is appropriate.
Added 11/3/2010
Learn more
Health Issues You've Asked to be Added
Learn more
This information is correct
BACK
FINISH LATER
SUBMIT

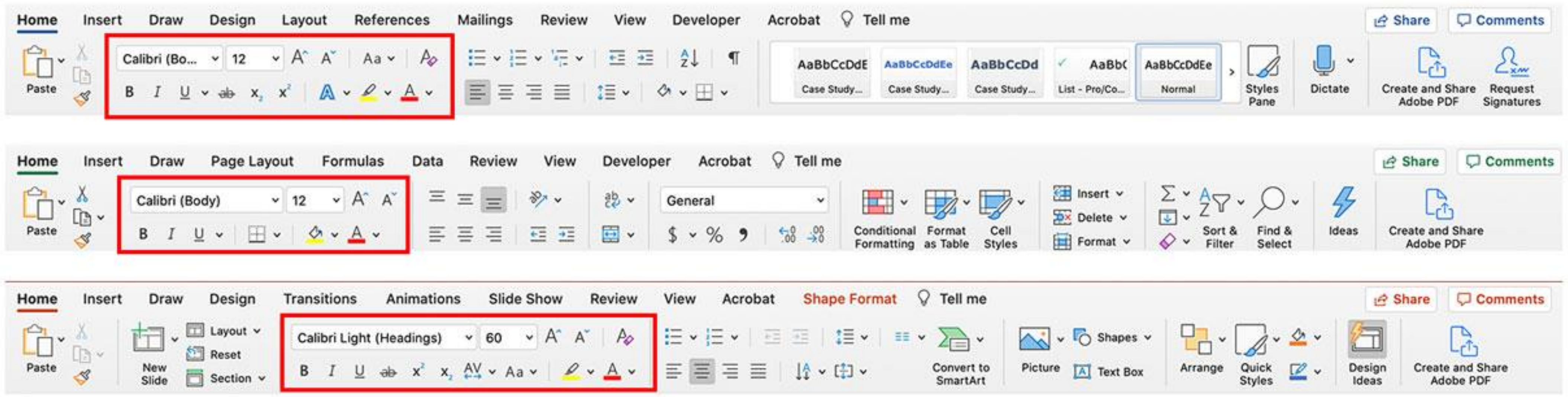
Sutter Health appointment checkin process, from https://www.nngroup.com/articles/consistency-and-standards/

Internal (In)consistency: Examples



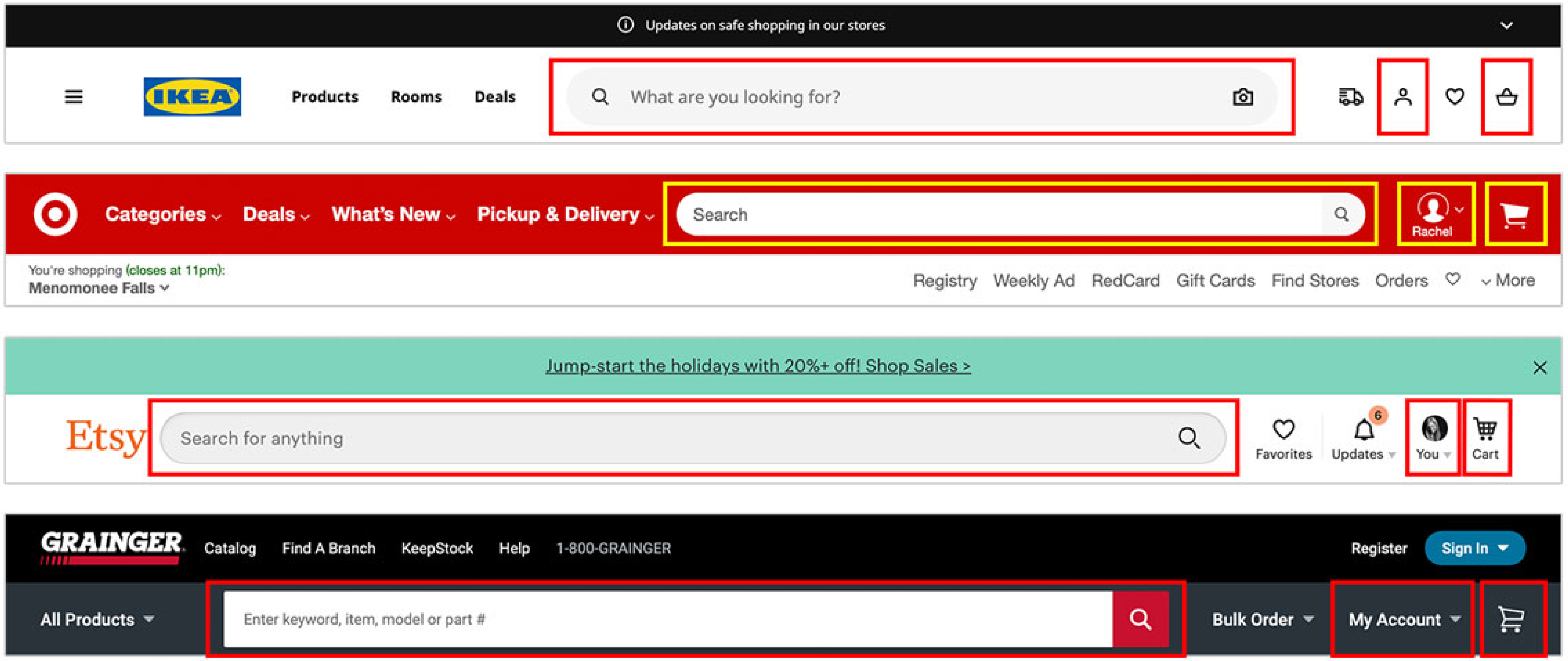
Outlook.com new UI Inconsistencies, from https://www.elwinlee.com/blog/outlook-coms-ui-inconsistency/

Internal Consistency



Internal consistency in the Microsoft Office Suite. From top to bottom: Word, Excel, PowerPoint
https://www.nngroup.com/articles/consistency-and-standards/

External Consistency



Navigation in different e-commerce websites.
https://www.nngroup.com/articles/consistency-and-standards/

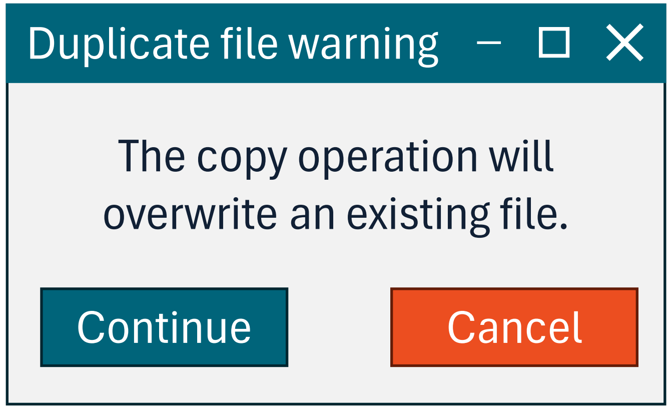
Feedback

The system should **continuously** inform the user about what it’s doing and how it’s interpreting the user’s input

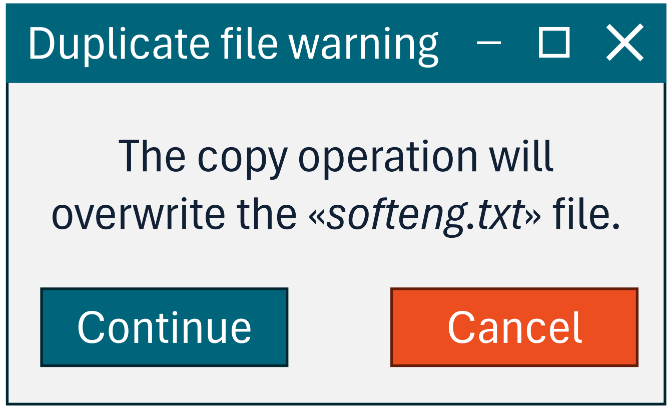
- Not only when errors occur
- Positive feedback is as important as negative feedback
- Whenever possible, give feedback also in case of system failure
- The worst possible feedback is no feedback at all!

Feedback

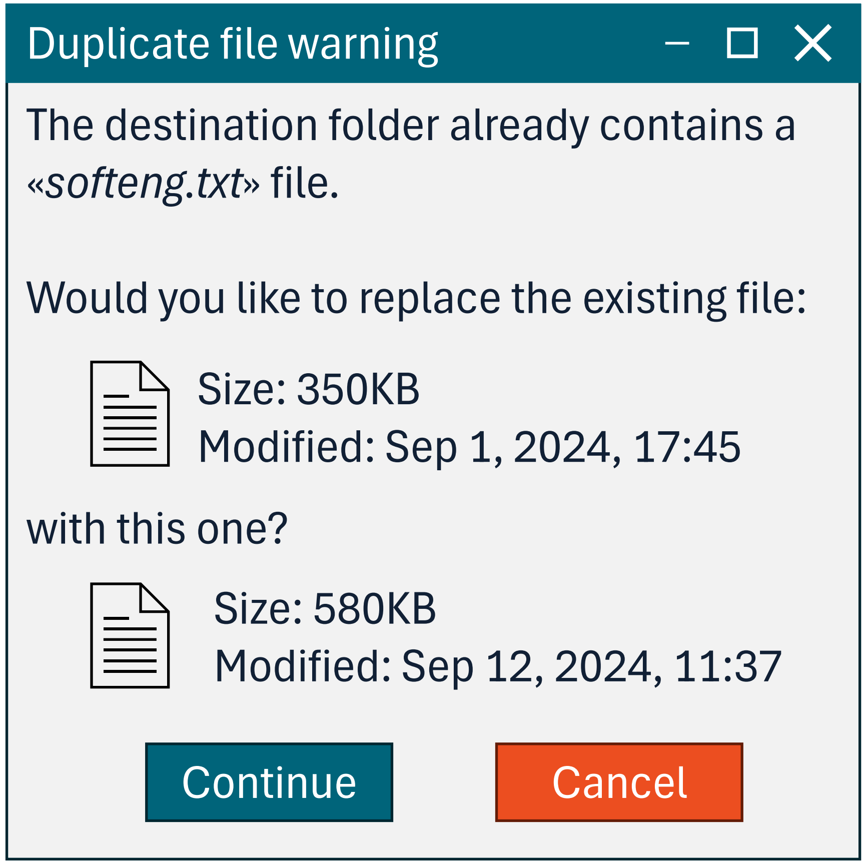
- Feedback should not be too abstract and general


Duplicate file warning

The copy operation will overwrite an existing file.


Duplicate file warning

The copy operation will overwrite the «*softeng.txt*» file.


Duplicate file warning

The destination folder already contains a «*softeng.txt*» file.

Would you like to replace the existing file:

Size: 350KB
Modified: Sep 1, 2024, 17:45

with this one?

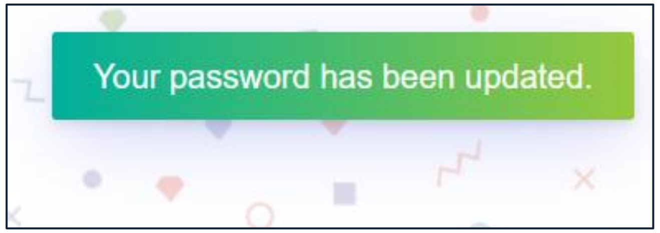
Size: 580KB
Modified: Sep 12, 2024, 11:37

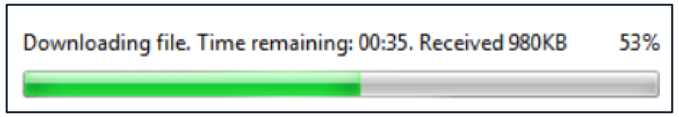
✗ 👎 ✓ 👍

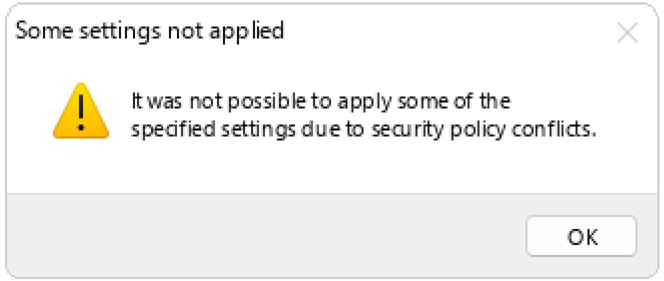
Feedback Persistence

Different types of feedback may require different **levels of persistence**

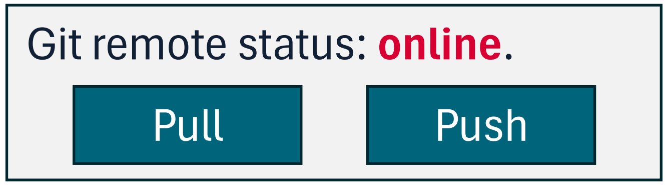
- Some feedbacks are relevant only for the duration of a certain phenomenon or are mere confirmations that an operation was performed.
  - May disappear automatically (e.g.: Toast messages)
- Others (especially warnings or errors) may require an explicit acknowledgement by the user.
- Others may require high persistence and be a permanent part of the UI


Your password has been updated.


Downloading file. Time remaining: 00:35. Received 980KB 53%


Some settings not applied

It was not possible to apply some of the specified settings due to security policy conflicts.


Git remote status: online.

Feedback and System Response Times

Feedback is crucial when systems have longer response times

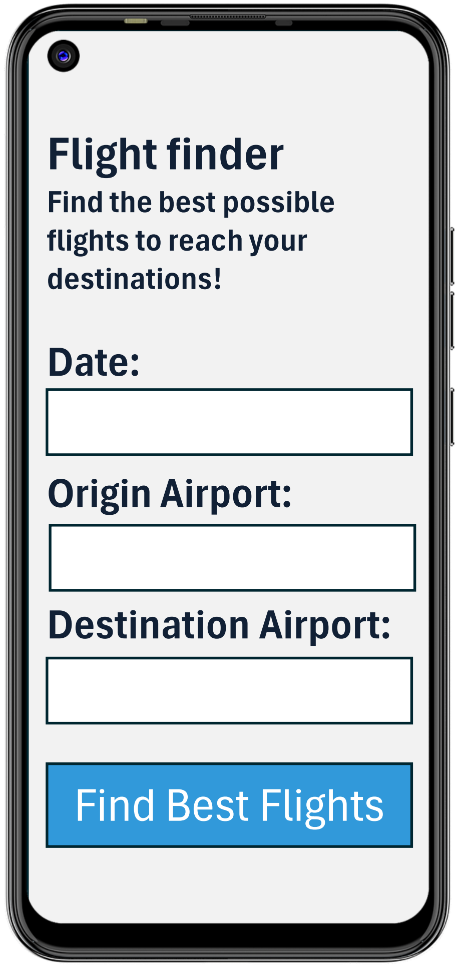
- **Less than 0.1 seconds**: reactions are perceived as instantaneous
  - No feedback required except to display the result or confirm the outcome
- **Less than 1 second**: user’s flow of thought stays uninterrupted
  - No special feedback is required (but no feeling of instantaneous reactions)
- **10 seconds**: limit for keeping the user’s attention focused
  - Feedback is crucial for delays longer than 10 seconds
  - Provide an estimation of when the task will be completed (users will want to do something else while they wait)
  - Possibly update the progress indicator frequently

Labor Perception Bias

The Labor Perception Bias: People **trust** and **value** things **more** when they perceive the underlying work

- Everyone dislikes waiting
- But if users have high expectations (e.g.: dealing with money, backup or migration of important data, data analysis and reporting, …), they can become skeptical if the waiting time is too short!
  - Adding a labor screen right after a key action can improve the User Experience
  - Sometimes, "benevolent deceptions" (e.g.: fake loading times) are added by designers: https://www.theatlantic.com/technology/archive/2017/02/why-some-apps-use-fake-progress-bars/517233/

Labor Perception Bias



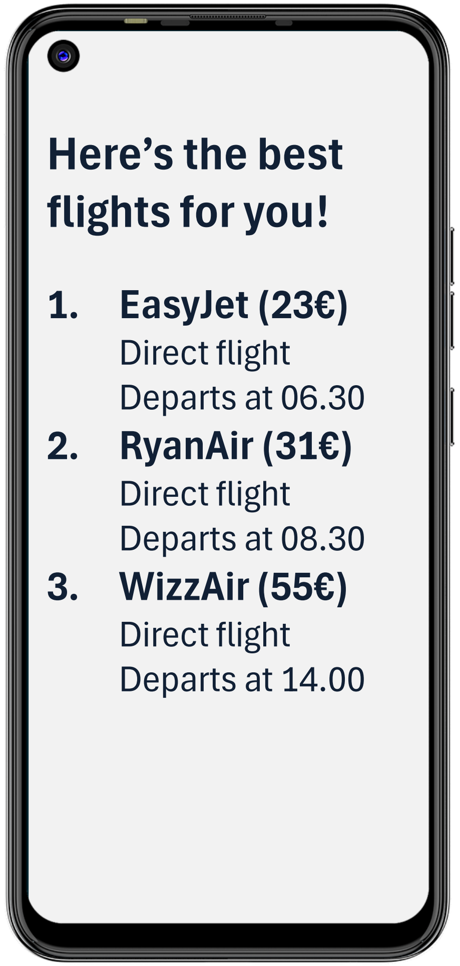
Flight finder

Find the best possible flights to reach your destinations!

Date:

Origin Airport:

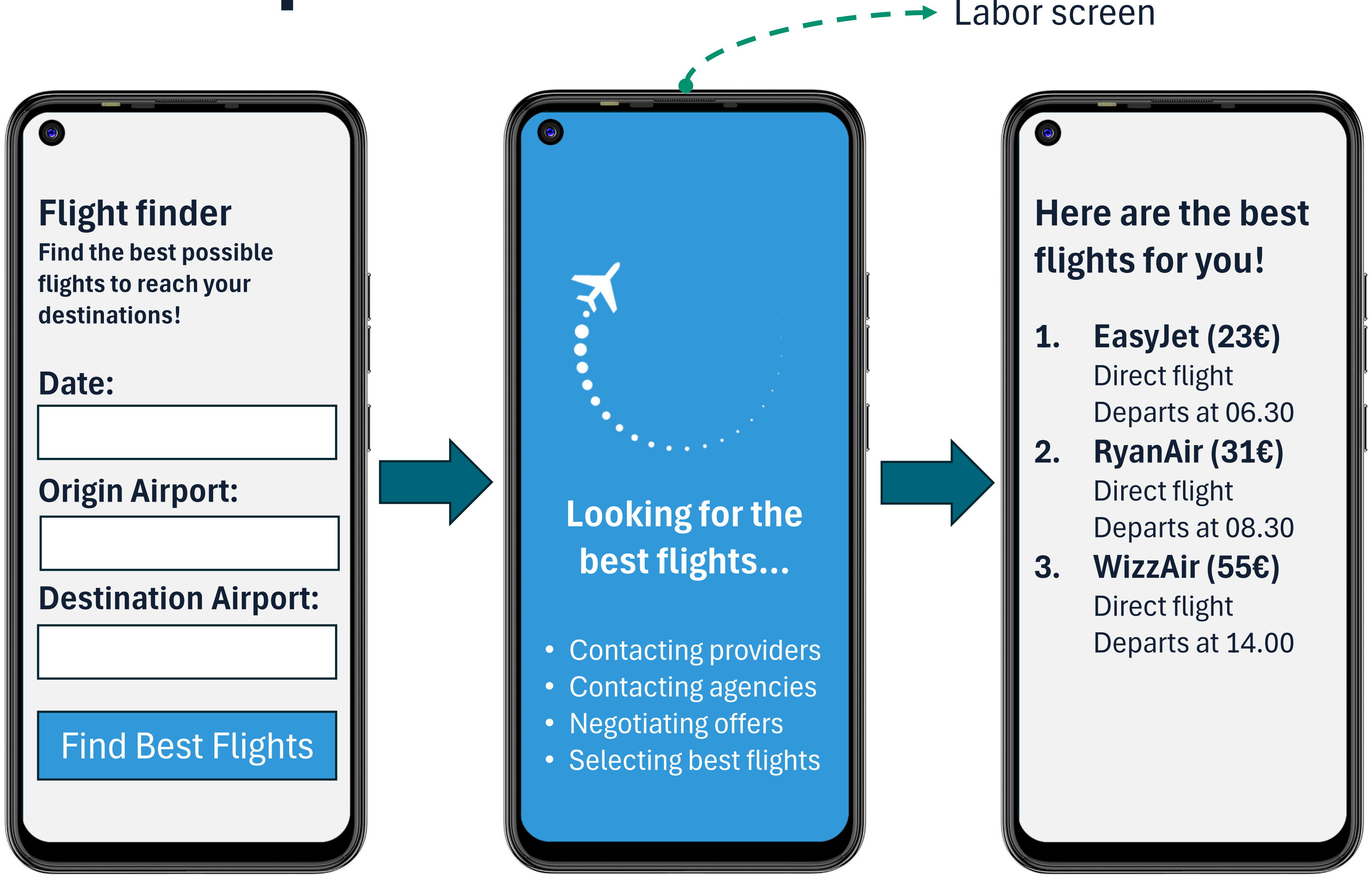
Destination Airport:



Here's the best flights for you!

1. **EasyJet (23€)**
   Direct flight
   Departs at 06.30
2. **RyanAir (31€)**
   Direct flight
   Departs at 08.30
3. **WizzAir (55€)**
   Direct flight
   Departs at 14.00

Labor Perception Bias



Flight finder

Find the best possible flights to reach your destinations!

Date:

Origin Airport:

Destination Airport:

Find Best Flights

Looking for the best flights...

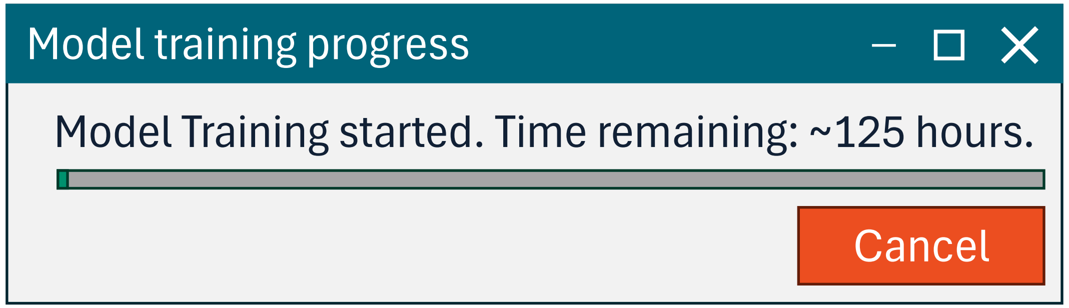
- Contacting providers
- Contacting agencies
- Negotiating offers
- Selecting best flights

Here are the best flights for you!

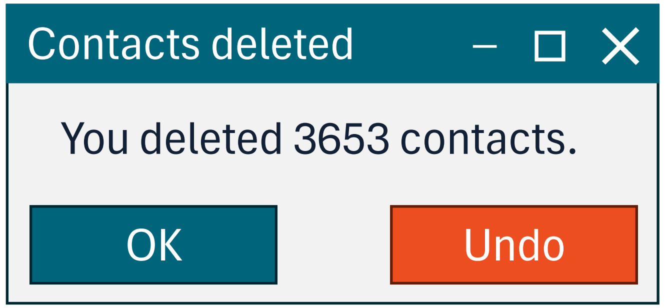
1. **EasyJet (23€)**
   Direct flight
   Departs at 06.30
2. **RyanAir (31€)**
   Direct flight
   Departs at 08.30
3. **WizzAir (55€)**
   Direct flight
   Departs at 14.00

Clearly Marked Exits andnd Reversal of Actions

- Users want to feel in control of the interaction
- Users will still make errors when using the system, no matter what
- The system should offer an easy way out of most situations
  - When the system cannot complete the action within 10 seconds, users should be able to interrupt the operation and cancel the action
  - In operations with side effects, exists can be provided by supporting an «**Undo**» facility that reverts the system to the previous state

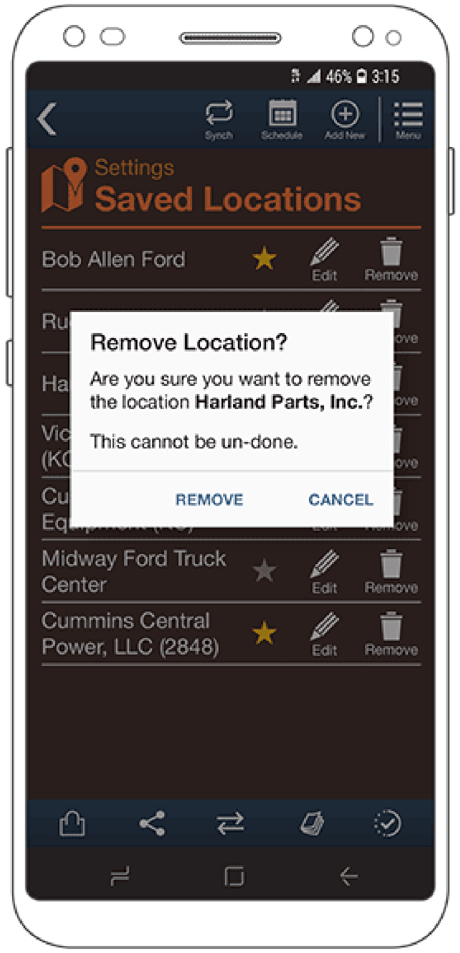


Model Training started. Time remaining: ~125 hours.



You deleted 3653 contacts.

Reversal of Actions: Example

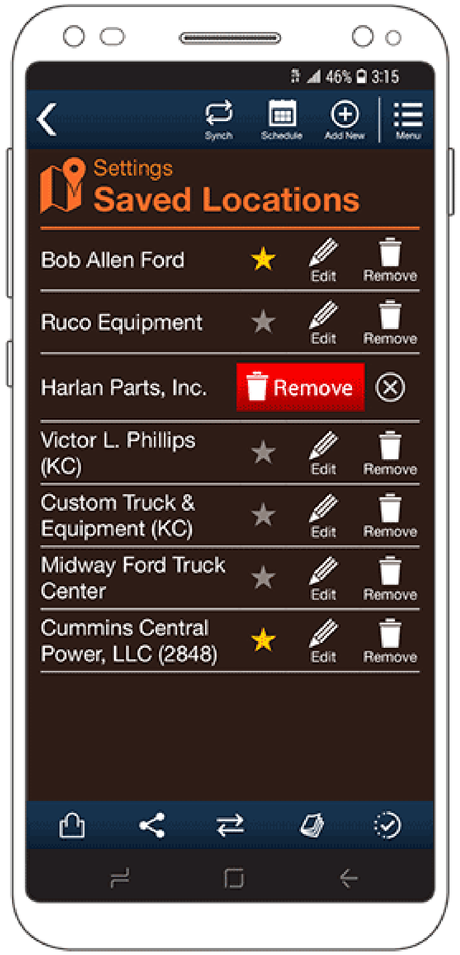


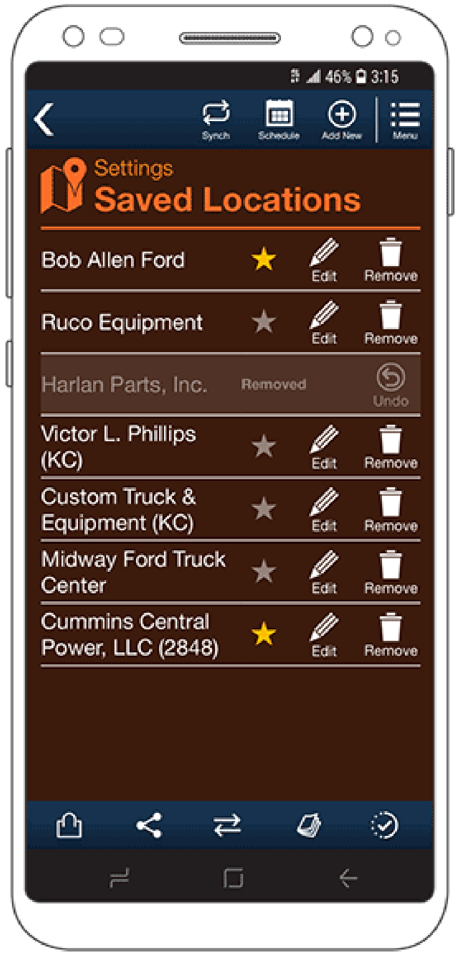
Remove Location?

Are you sure you want to remove the location **Harland Parts, Inc.**?

This cannot be un-done.

REMOVE CANCEL



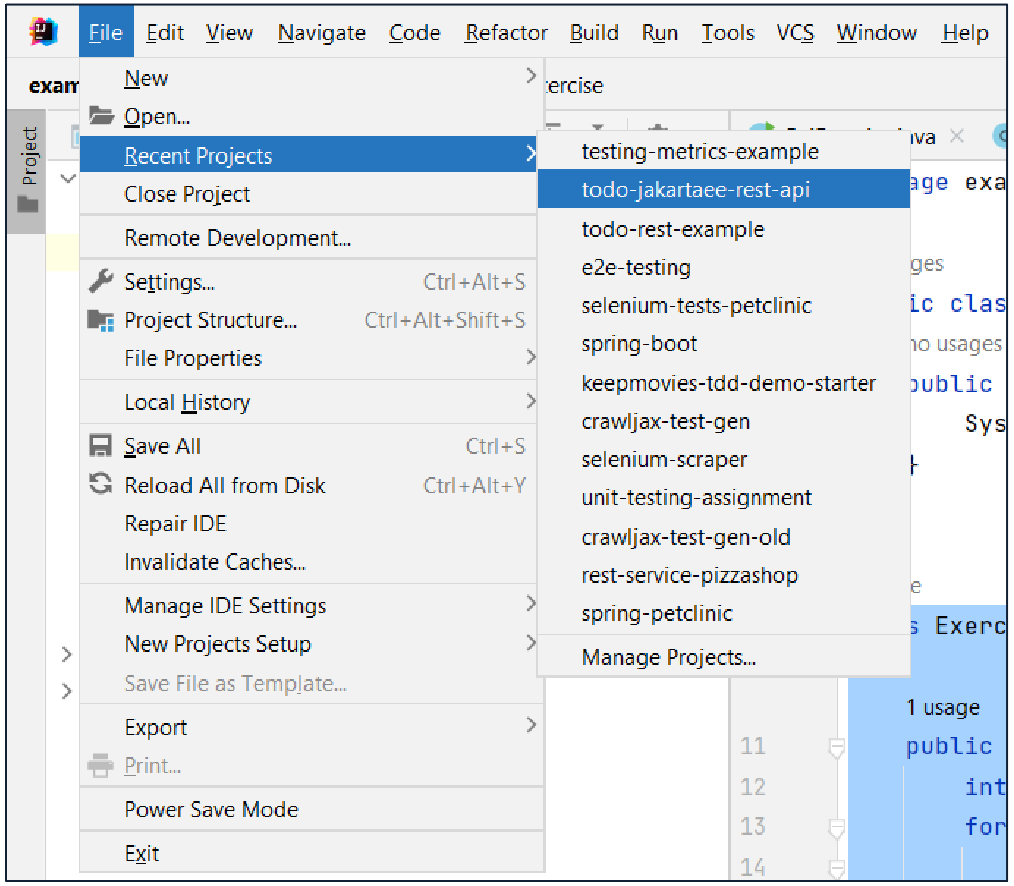


https://www.uxmatters.com/mt/archives/2020/03/are-you-sure-versus-undo-design-and-technology.php

Shortcuts

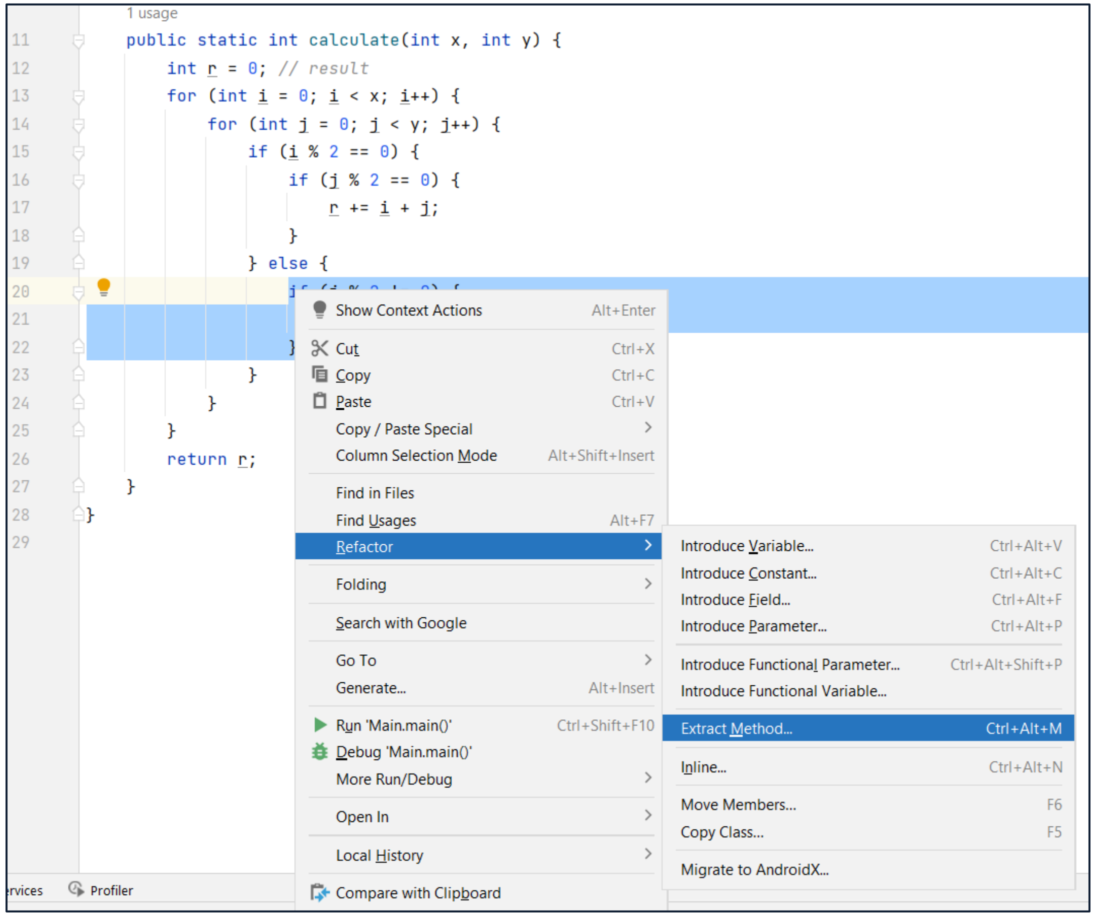
- Generally, operating a UI should require the knowledge of just a few rules
- Experienced users should also be able to perform frequent actions using shortcuts and accelerators
  - Function keys or command keys that package an entire command in a keypress
  - Double click on an object to perform the most common action on that object
  - Having specific buttons to access important functions directly from those parts of the dialogue where they may be more frequently needed
  - Re-using the past interaction history (rapidly perform the same commands)
  - Providing good default values in form, when possible

Shortcuts



Recent Projects accelerator in IntelliJ IDEA

Shortcuts

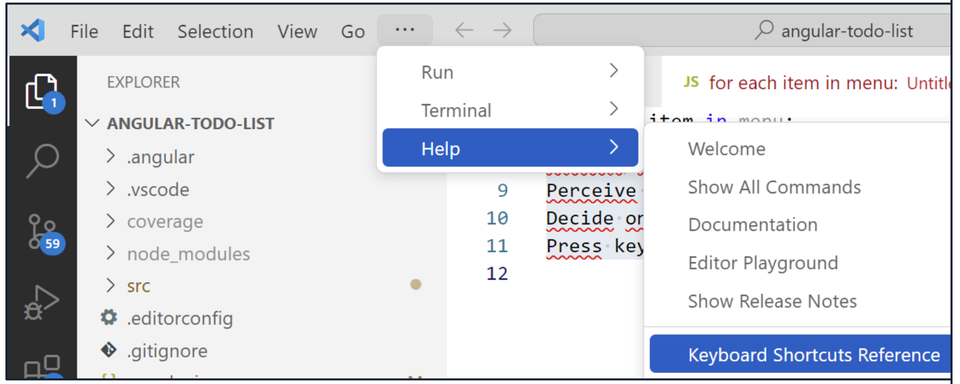


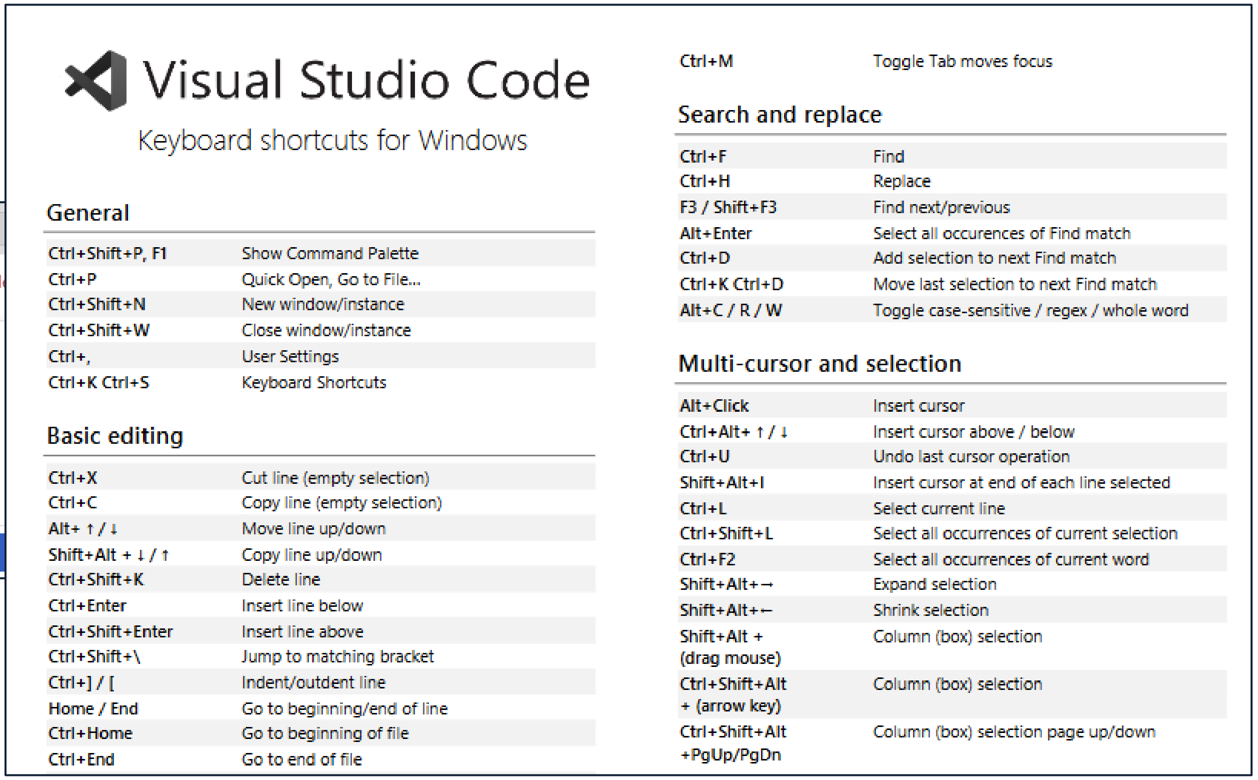
```
1 usage
public static int calculate(int x, int y) {
    int r = 0; // result
    for (int i = 0; i < x; i++) {
        for (int j = 0; j < y; j++) {
            if (i % 2 == 0) {
                if (j % 2 == 0) {
                    r += i + j;
                }
            } else {

                }
            }
        }
    }
    return r;
}
}
```

Show Context Actions Alt+Enter
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Copy / Paste Special
Column Selection Mode Alt+Shift+Insert
Find in Files
Find Usages Alt+F7
Refactor
Folding
Search with Google
Go To
Generate... Alt+Insert
Run 'Main.main()' Ctrl+Shift+F10
Debug 'Main.main()'
More Run/Debug
Open In
Local History
Compare with Clipboard
Introduce Variable... Ctrl+Alt+V
Introduce Constant... Ctrl+Alt+C
Introduce Field... Ctrl+Alt+F
Introduce Parameter... Ctrl+Alt+P
Introduce Functional Parameter... Ctrl+Alt+Shift+P
Introduce Functional Variable...
Extract Method... Ctrl+Alt+M
Inline... Ctrl+Alt+N
Move Members... F6
Copy Class... F5
Migrate to AndroidX...
rvices
Profiler

Keyboard shortcuts also shown in context menus in IntelliJ IDEA

Shortcuts





Visual Studio Code

Keyboard shortcuts for Windows

General

Ctrl+Shift+P, F1	Show Command Palette
Ctrl+P	Quick Open, Go to File...
Ctrl+Shift+N	New window/instance
Ctrl+Shift+W	Close window/instance
Ctrl+,	User Settings
Ctrl+K Ctrl+S	Keyboard Shortcuts

Basic editing

Ctrl+X	Cut line (empty selection)
Ctrl+C	Copy line (empty selection)
Alt+ ↑ / ↓	Move line up/down
Shift+Alt + ↓ / ↑	Copy line up/down
Ctrl+Shift+K	Delete line
Ctrl+Enter	Insert line below
Ctrl+Shift+Enter	Insert line above
Ctrl+Shift+\	Jump to matching bracket
Ctrl+] / [	Indent/outdent line
Home / End	Go to beginning/end of line
Ctrl+Home	Go to beginning of file
Ctrl+End	Go to end of file

Ctrl+M	Toggle Tab moves focus

Search and replace

Ctrl+F	Find
Ctrl+H	Replace
F3 / Shift+F3	Find next/previous
Alt+Enter	Select all occurences of Find match
Ctrl+D	Add selection to next Find match
Ctrl+K Ctrl+D	Move last selection to next Find match
Alt+C / R / W	Toggle case-sensitive / regex / whole word

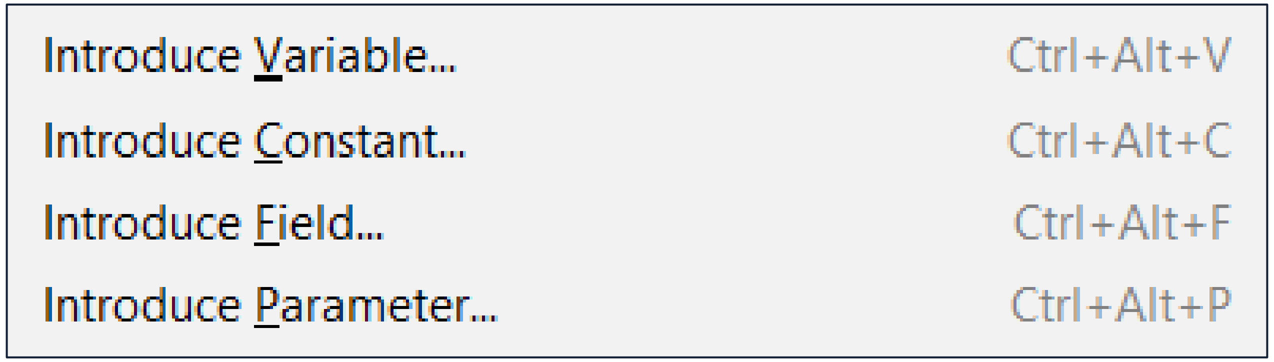
Multi-cursor and selection

Alt+Click	Insert cursor
Ctrl+Alt+ ↑ / ↓	Insert cursor above / below
Ctrl+U	Undo last cursor operation
Shift+Alt+I	Insert cursor at end of each line selected
Ctrl+L	Select current line
Ctrl+Shift+L	Select all occurrences of current selection
Ctrl+F2	Select all occurrences of current word
Shift+Alt+→	Expand selection
Shift+Alt+←	Shrink selection
Shift+Alt + (drag mouse)	Column (box) selection
Ctrl+Shift+Alt + (arrow key)	Column (box) selection
Ctrl+Shift+Alt +PgUp/PgDn	Column (box) selection page up/down

Keyboard Shortcuts reference in VS Code

Shortcuts Guidelines

- Keyboard shortcuts should also be **learnable** and **memorable**



Introduce Variable...	Ctrl+Alt+V
Introduce Constant...	Ctrl+Alt+C
Introduce Field...	Ctrl+Alt+F
Introduce Parameter...	Ctrl+Alt+P

- You can't just use any unique combinations of keys!
- If users need to check the reference everytime they want to use a shortcut, then it is no shortcut at all!

Error Messages

Error messages are critical for usability

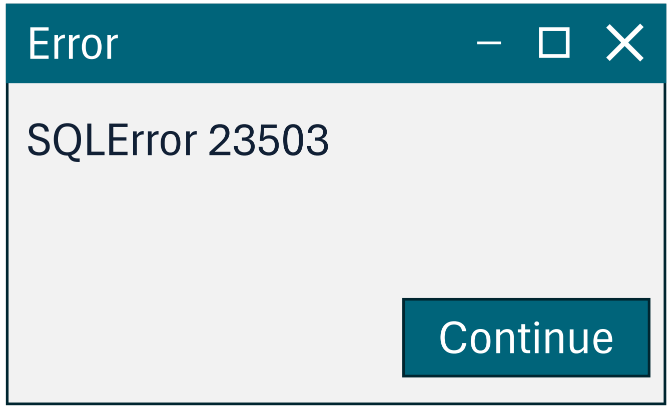
- They represent situations in which users are in trouble and might be unsable to use the system to achieve their goals
- They present opportunities to help users understand the system better
  - Users are generally more motivated to pay attention to the content of error messages

Good Error Messages

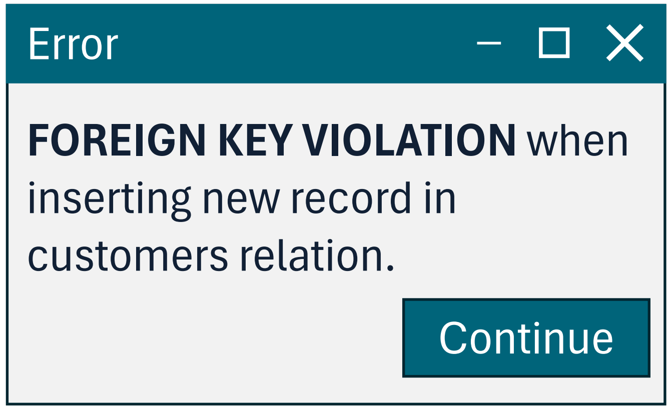
According to Shneiderman, error messages should follow four rules:

1. They should be phrased in **clear language** and **avoid obscure codes**
2. They should be **precise** rather than vague or general
3. They should **constructively help** the users solve the problem
4. They should be **polite** and **not intimidate or blame** the user

Good Error Messages: Clear Language

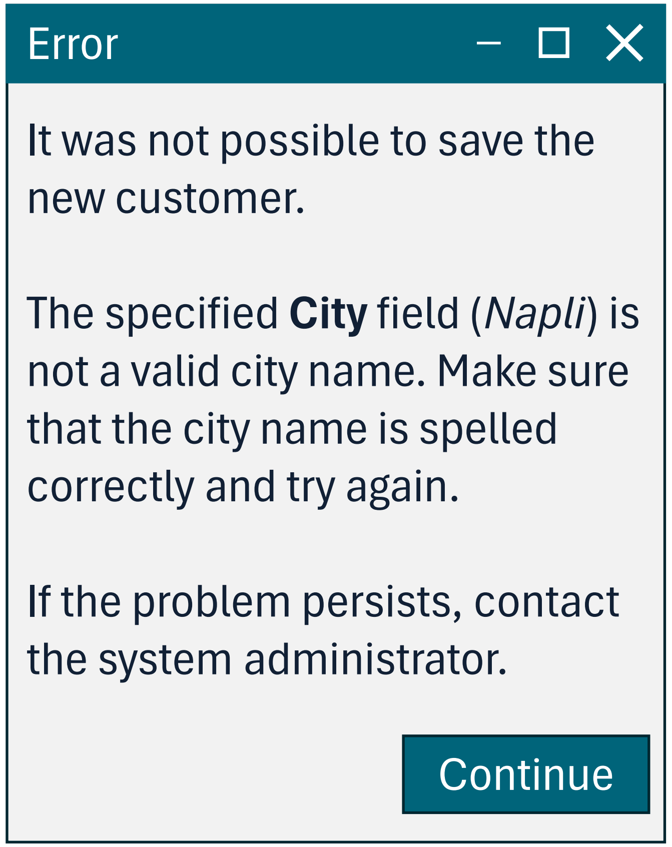

Error

SQLError 23503


Error

FOREIGN KEY VIOLATION when inserting new record in customers relation.

👎


Error

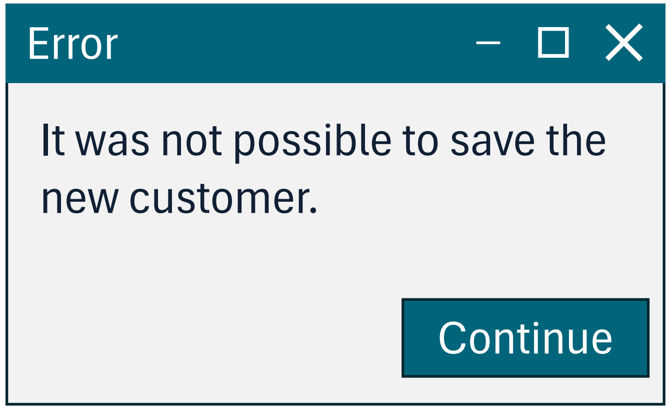
It was not possible to save the new customer.

The specified **City** field (*Napli*) is not a valid city name. Make sure that the city name is spelled correctly and try again.

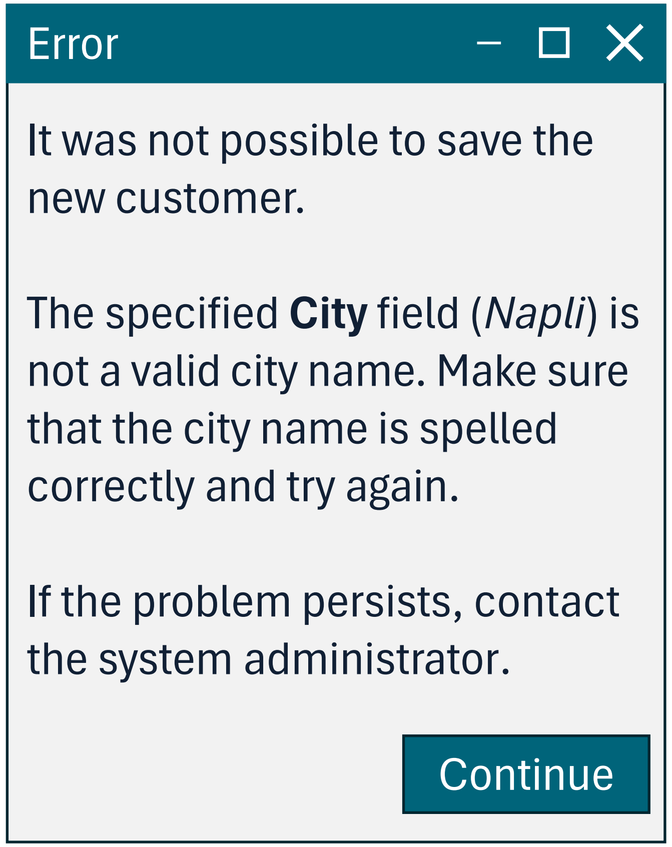
If the problem persists, contact the system administrator.

👍

Good Error Message: Precise and Not General


Error

It was not possible to save the new customer.

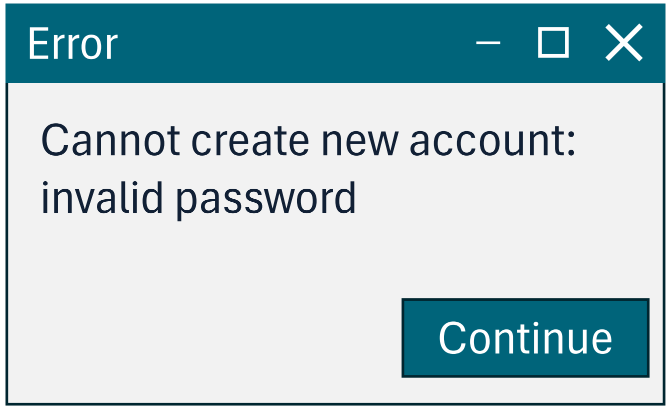

Error

It was not possible to save the new customer.

The specified **City** field (*Napli*) is not a valid city name. Make sure that the city name is spelled correctly and try again.

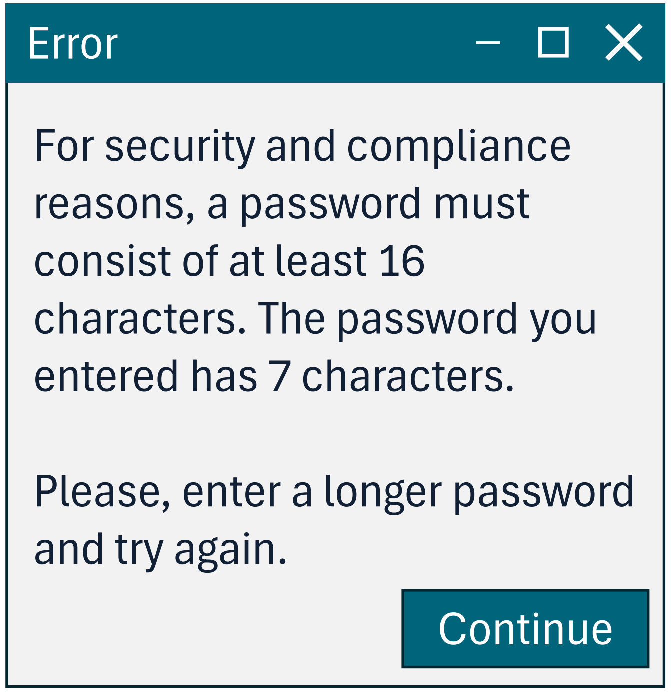
If the problem persists, contact the system administrator.

Good Error Messages: Be Constructive



Cannot create new account: invalid password

👎



For security and compliance reasons, a password must consist of at least 16 characters. The password you entered has 7 characters.

Please, enter a longer password and try again.

👍

Good Error Messages: Be Polite

Do not intimidate or blame the user

- Users already feel bad not being able to achieve their goals, no need to rub it in!
- Avoid abusive terms
  - «ILLEGAL USER ACTION!», «JOB ABORTED», «PROCESS KILLED», «FATAL ISSUE»
- Try to phrase error message so as to suggest that the problem is really the system's fault
  - In a way it is, since good UI design might have prevented that error!

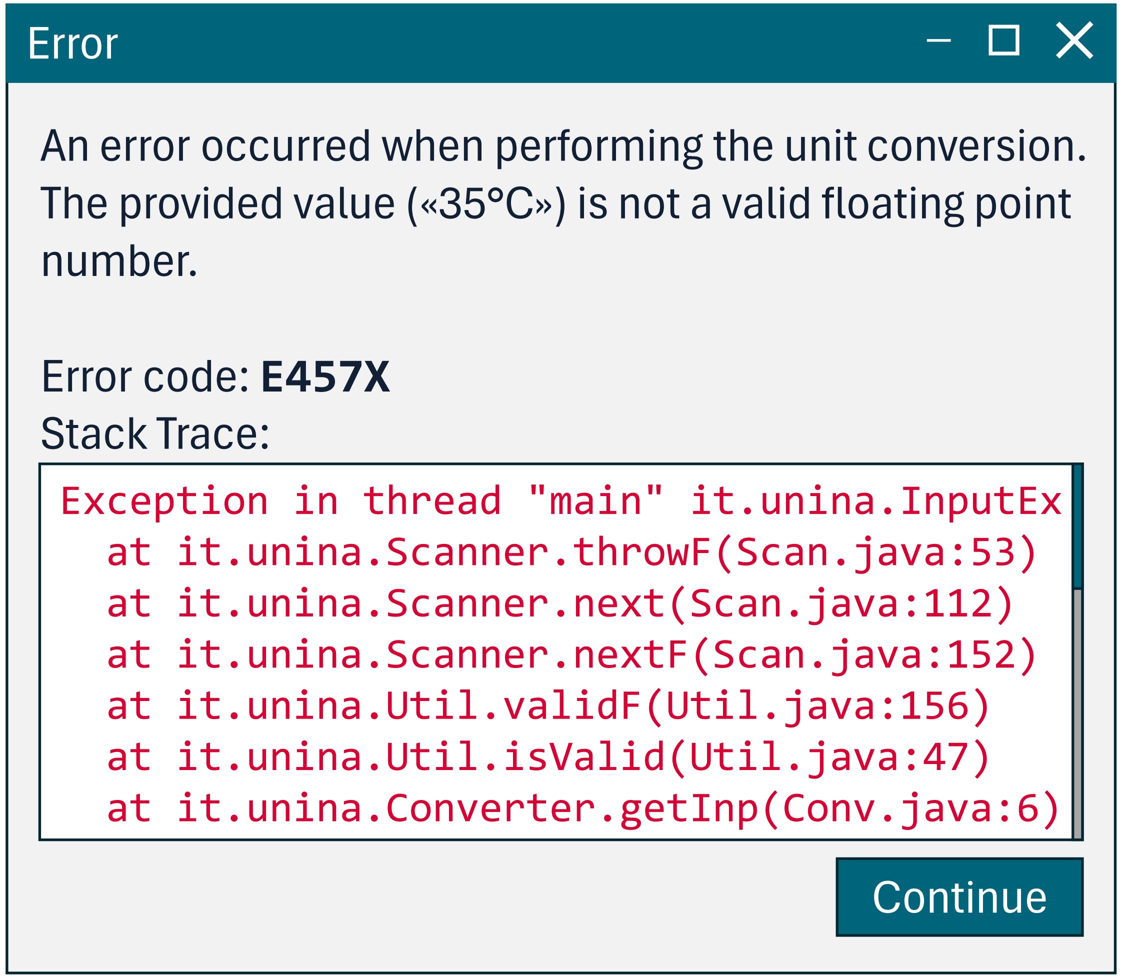
```
$> Foo()
Fatal error! You are
trying to execute a
function (Foo) that you
have not previoysly
defined!
$>
```

```
$> Foo()
I don't know how to Foo.
Are you sure the Foo
function has been defined?
$>
```

Good Error Messages: Multiple Levels

Error messages are useful for users, but also for technical staff operating the system

- Users typically do not understand technical details (e.g.: specific error codes or stack traces), but technical staff may need those details for troubleshooting
- Error messages may need to contain both



An error occurred when performing the unit conversion. The provided value («35°C») is not a valid floating point number.

Error code: **E457X**

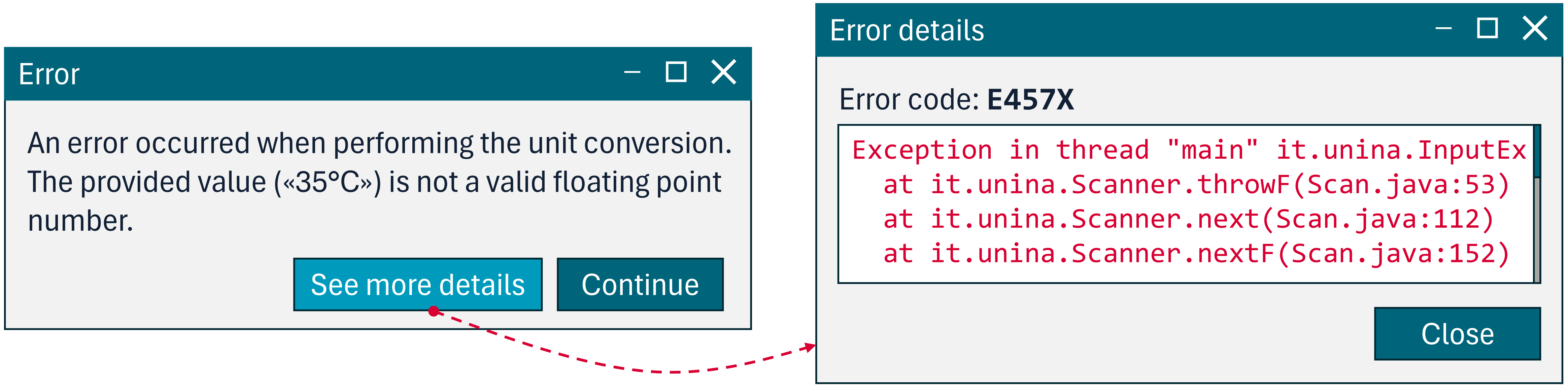
Stack Trace:

```
Exception in thread "main" it.unina.InputEx
  at it.unina.Scanner.throwF(Scan.java:53)
  at it.unina.Scanner.next(Scan.java:112)
  at it.unina.Scanner.nextF(Scan.java:152)
  at it.unina.Util.validF(Util.java:156)
  at it.unina.Util.isValid(Util.java:47)
  at it.unina.Converter.getInp(Conv.java:6)
```

Good Error Messages: Multiple Levels

It is often preferrable to separate the views of the different levels

- Regular users are not intimidated by strange-looking messages
- Tech Staff can access the troubleshooting information
- Error dialogs may also include hyperlinks to a support website



An error occurred when performing the unit conversion. The provided value («35°C») is not a valid floating point number.

Error code: **E457X**

```
Exception in thread "main" it.unina.InputEx
  at it.unina.Scanner.throwF(Scan.java:53)
  at it.unina.Scanner.next(Scan.java:112)
  at it.unina.Scanner.nextF(Scan.java:152)
```

:(

Your PC ran into a problem and needs to restart. We're just collecting some error info, and then we'll restart for you.

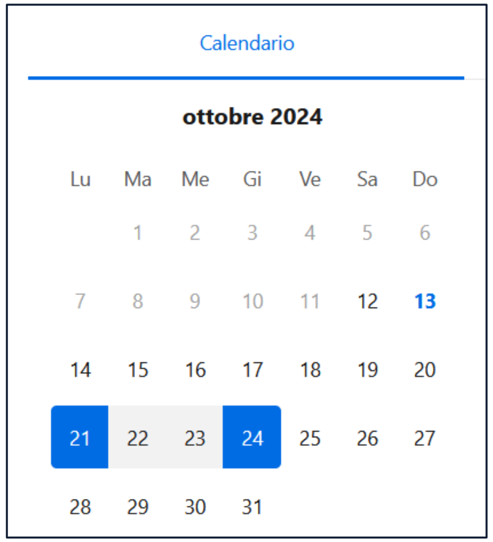
15% complete

For more information about this issue and possible fixes, visit
http://windows.com/stopcode

If you call a support person, give them this info:
Stop code: SOFTWARE_ENGINEERING_LECTURE

Prevent Errors

- Even better than to have good error messages, is to avoid errors!
- Try to avoid putting users in error-prone situations
  - If the user is asked to type the name of a city, there is a risk of spelling errors
  - If users need to insert a date range in the future, formatted in a specific way, there is a risk users will not format the date correctly, or insert a date that is not in the future
- Design the UI to avoid (or minimize) such errors
  - This is not only good for usability! It also likely means less work to formalize use cases and less code to manage error situations!



Date picker on Booking.com

Types of Errors

According to Don Norman, two categories of errors exist:

- **Slips:** user intends to perform an action, but ends up doing another
  - Pressing the «Enter» key instead of the «Backspace» key
  - Clicking on the «Minimize» button instead of the «Maximize» one
- **Mistakes:** user forms a inappropriate goals for the current problem/task
  - The manager of an e-commerce website wants to delete all items from a certain category. He believes that deleting the category will also delete all associated items. Items are actually implicitly moved to the «Other» category.

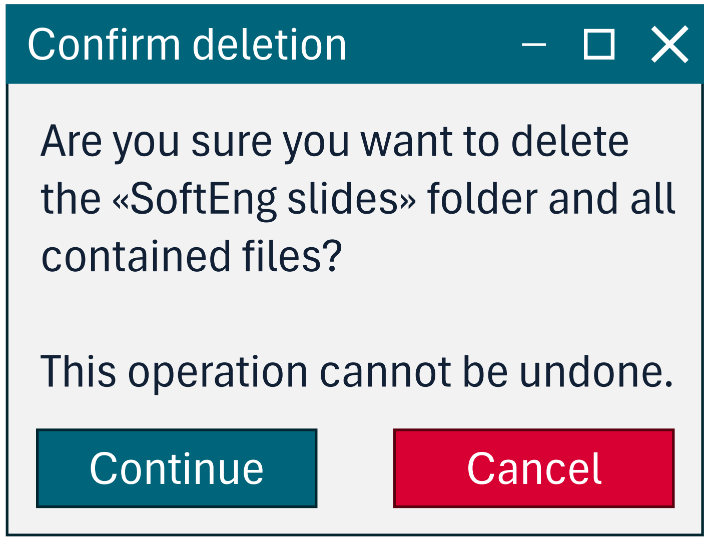
Types of Errors: Slips

- If users form appropriate goals, but mess up the execution, they've made a slip
- Slips typically result from automatic behaviour
- Slips are more frequently in skilled behaviour (users pay more attention when they are still learning to use a system)
- Slips are often «**capture errors**»
  - When two sequences of action have a common prefix, and one sequence is used way more often than the other, users find themselves inconscioulsy following the more frequent sequence, even if they wanted to perform the less frequent one

Types of Errors: Slips

- Slips are the reason that allowing also for easy reversal of actions is generally preferrable to just relying on a confirmation dialog

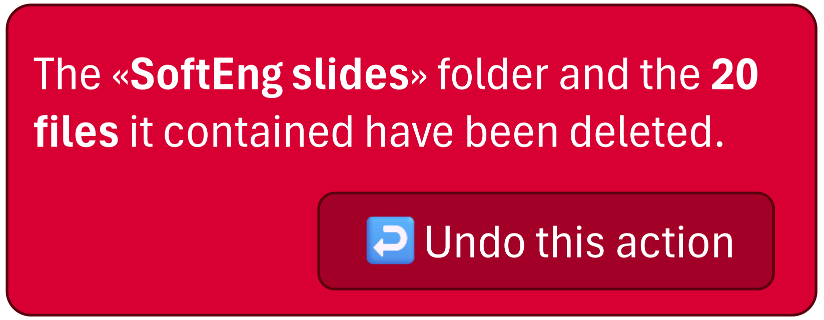
Users may click «continue» out of habit, since most of the times they intend to delete the correct directory



Confirm deletion

Are you sure you want to delete the «SoftEng slides» folder and all contained files?

This operation cannot be undone.



The «**SoftEng slides**» folder and the **20 files** it contained have been deleted.

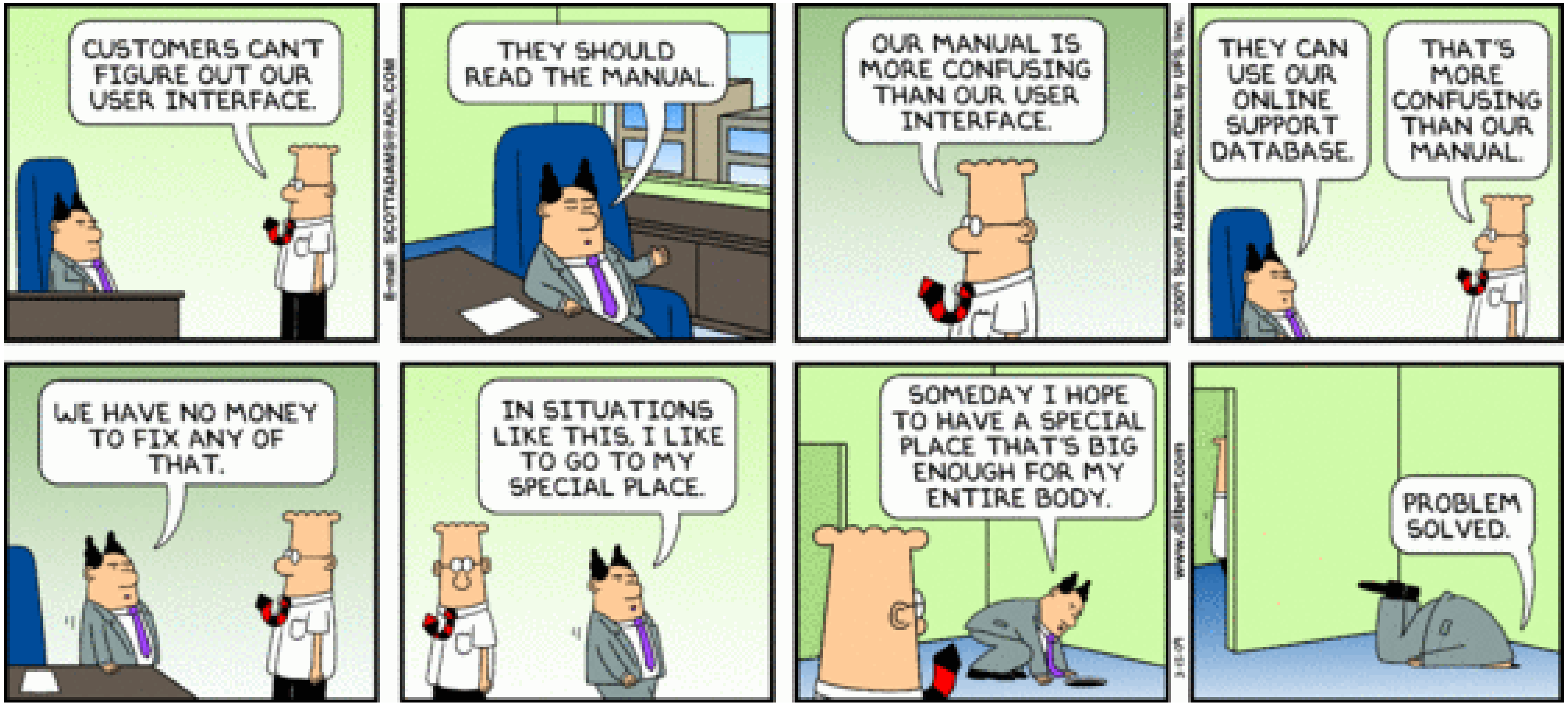
Types of Errors: Mistakes

- Mistakes are way more critical
- They often derive from the users having formed an incorrect mental model of the system
- They can also be way more difficult to detect (and thus more dangerous!)
- Think of the example from the previous slide:
    - The manager of an e-commerce website wants to delete all items from a certain category. He believes that deleting the category will also delete all associated items. Items are actually implicitly moved to the «Other» category.
    - When will the manager notice?

Help and Documentation

- Ideally, a system should be so easy to use that no further help or documentation is needed to supplement the UI
- This goal unfortunately cannot always be met. Apart from true walk-up-and-use systems, most UIs have enough functions to warrant a manual and possibly an help system
  - A manual could also be used by regular users to acquire higher levels of expertise and increase their productivity
- **Note**: having a nice manual and help system does not reduce usability requirements!
  - «*It's all explained in the manual!*» is not a good excuse for bad UI design!

Help and Documentation


CUSTOMERS CAN'T FIGURE OUT OUR USER INTERFACE.
THEY SHOULD READ THE MANUAL.
OUR MANUAL IS MORE CONFUSING THAN OUR USER INTERFACE.
THEY CAN USE OUR ONLINE SUPPORT DATABASE.
THAT'S MORE CONFUSING THAN OUR MANUAL.
WE HAVE NO MONEY TO FIX ANY OF THAT.
IN SITUATIONS LIKE THIS, I LIKE TO GO TO MY SPECIAL PLACE.
SOMEDAY I HOPE TO HAVE A SPECIAL PLACE THAT'S BIG ENOUGH FOR MY ENTIRE BODY.
PROBLEM SOLVED.
E-mail: SCOTTADAMS@AOL.COM
© 2005 Scott Adams, Inc./Dist. by UFS, Inc.
www.dilbert.com

The Fundamental Truth about User Manuals

- Users **do not read** user manuals
- They prefer spending time in activities that make them feel productive
- They typically start using the system without having read the instructions
- Corollary to the fundamental truth about user manuals
  - If users do want to read the manual, they are probably in some kind of panic and need immediate help
  - Online manuals with task-oriented lookup and custom search functions are particularly useful in these cases

Seeking Universal Usability

Seek usability for **everyone**! You should be mindful of:

- Novice vs Expert users differences, Age ranges, Disabilities, International variations

Designing for everyone does not mean ending up with a product that is overall less effective

- Often many categories of users can reap benefits of design considerations made to accomodate the needs of a specific category
- Think of curb ramps! (curb cut effect)

A curb ramp

Design Dialogs to Yield Closure

- Sequences of actions should be organized into groups with a beginning, middle, and end.
- Informative feedback at the completion of a group should give users the satisfaction of accomplishment, a sense of relief, an indicator to prepare for the next group of actions
- For example, e-commerce websites move customers through a series of clear steps
  - Add items to cart
  - Specify payment method, address for delivery, etc…
  - Payment

READINGS AND REFERENCES

[1] Shneiderman, B., & Plaisant, C. (2010). *Designing the user interface: strategies for effective human-computer interaction*. Pearson Education.

[2] Molich, R., & Nielsen, J. (1990). Improving a human-computer dialogue. *Communications of the ACM*, *33*(3), 338-348. https://dl.acm.org/doi/10.1145/77481.77486

[3] Holcomb, R., & Tharp, A. L. (1991). What users say about software usability. *International Journal of Human-Computer Interaction*, *3*(1), 49-78. https://doi.org/10.1080/10447319109525996

READINGS AND REFERENCES

[4] Polson, P. G., & Lewis, C. H. (1990). Theory-based design for easily learned interfaces. *Human–Computer Interaction*, *5*(2-3), 191-220. https://doi.org/10.1080/07370024.1990.9667154

[5] Carroll, J. M., & Rosson, M. B. (1992). Getting around the task-artifact cycle: How to make claims and design by scenario. *ACM Transactions on Information Systems (TOIS)*, *10*(2), 181-212. https://dl.acm.org/doi/abs/10.1145/146802.146834

[6] Nielsen, J. (1994, April). Enhancing the explanatory power of usability heuristics. In Proceedings of the SIGCHI conference on Human Factors in Computing Systems (pp. 152-158). https://dl.acm.org/doi/10.1145/191666.191729