Narrative

Software Design and Development In Practice

Narrative Workshop Series

jkolko@wonderfulnarrative.com

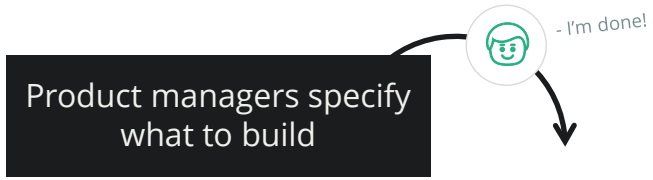
Overview

Warning: software development is often really dogmatic.

Common approaches to design and development coordination

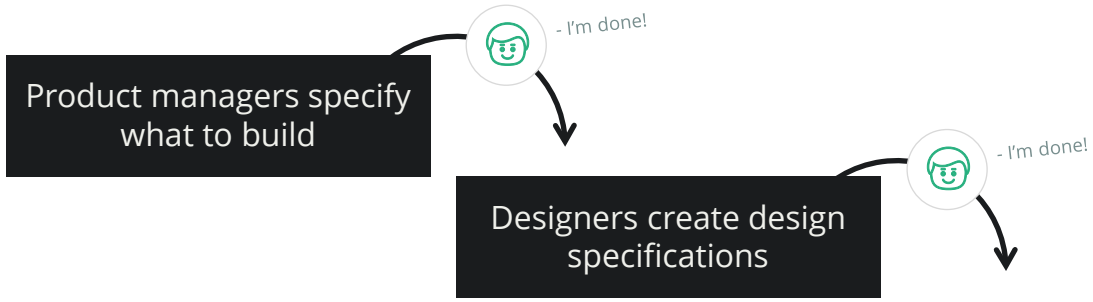
Common approaches to design and development coordination

Document and deliver (“Waterfall”)



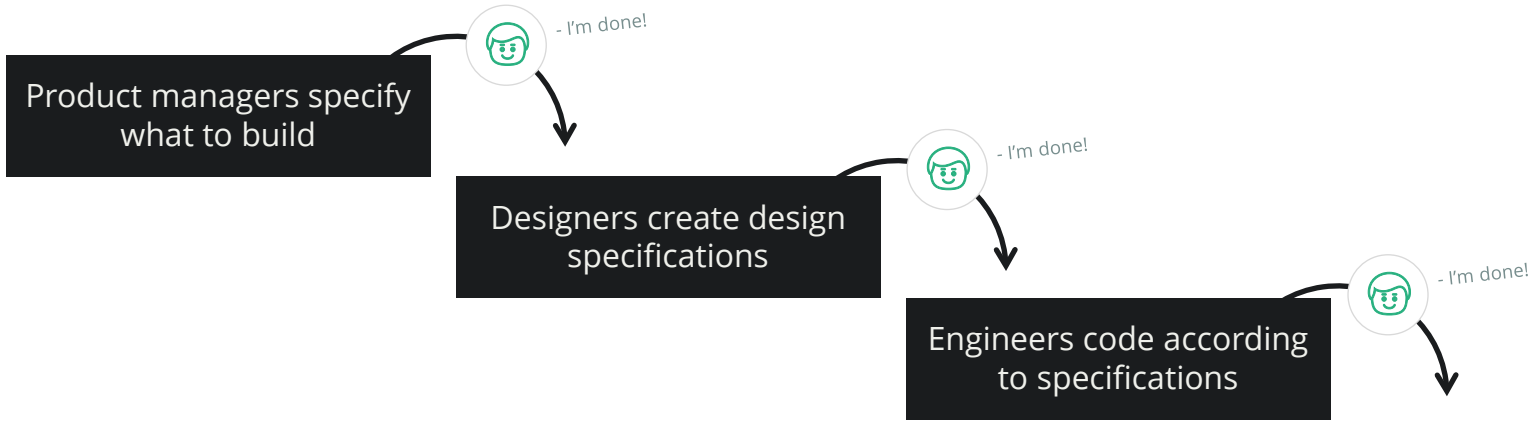
Common approaches to design and development coordination

Document and deliver (“Waterfall”)



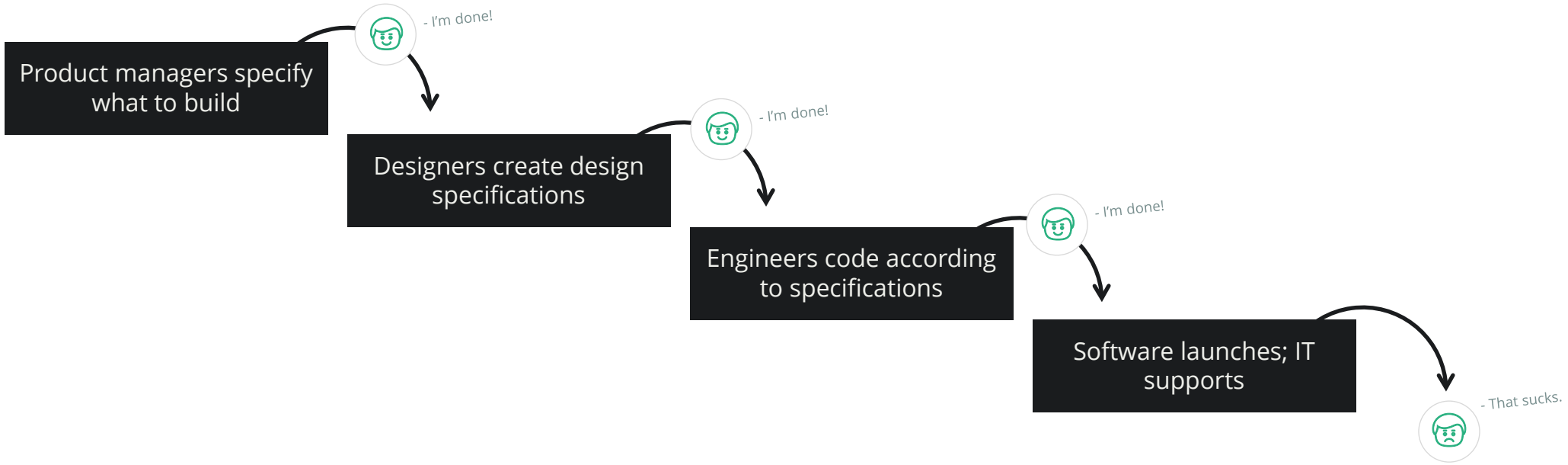
Common approaches to design and development coordination

Document and deliver (“Waterfall”)



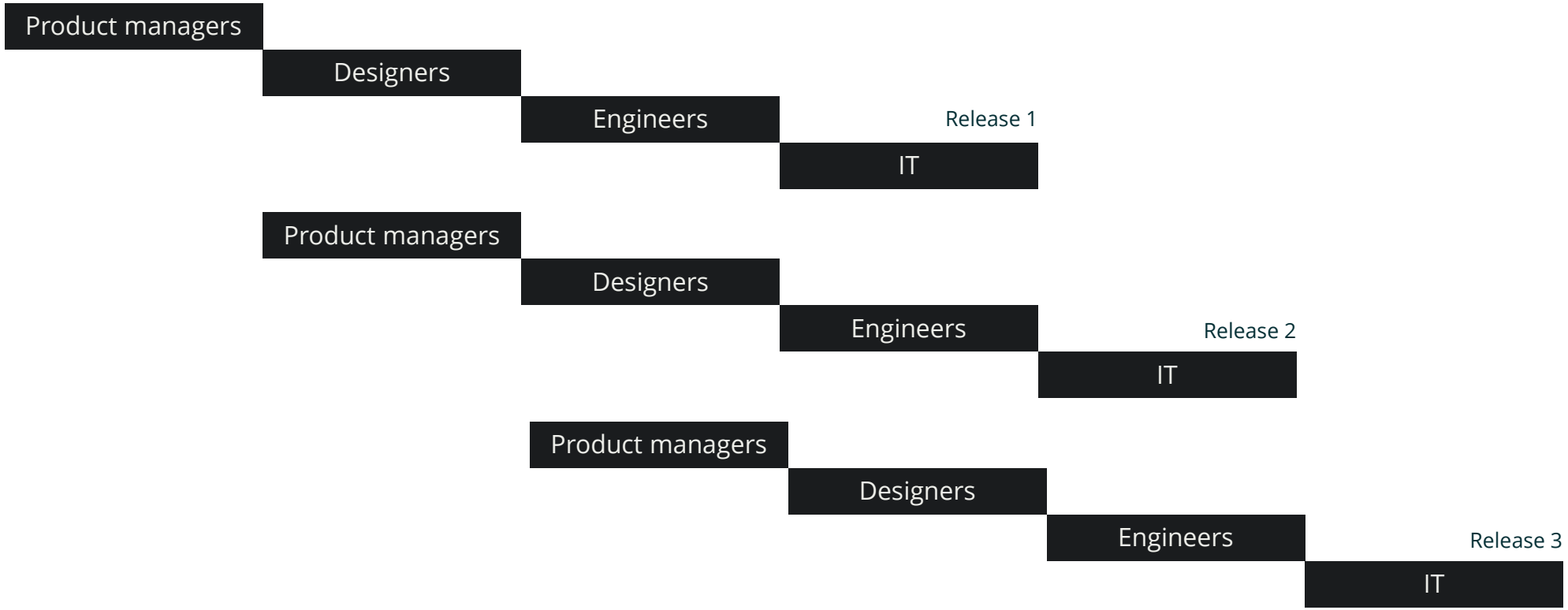
Common approaches to design and development coordination

Document and deliver (“Waterfall”)



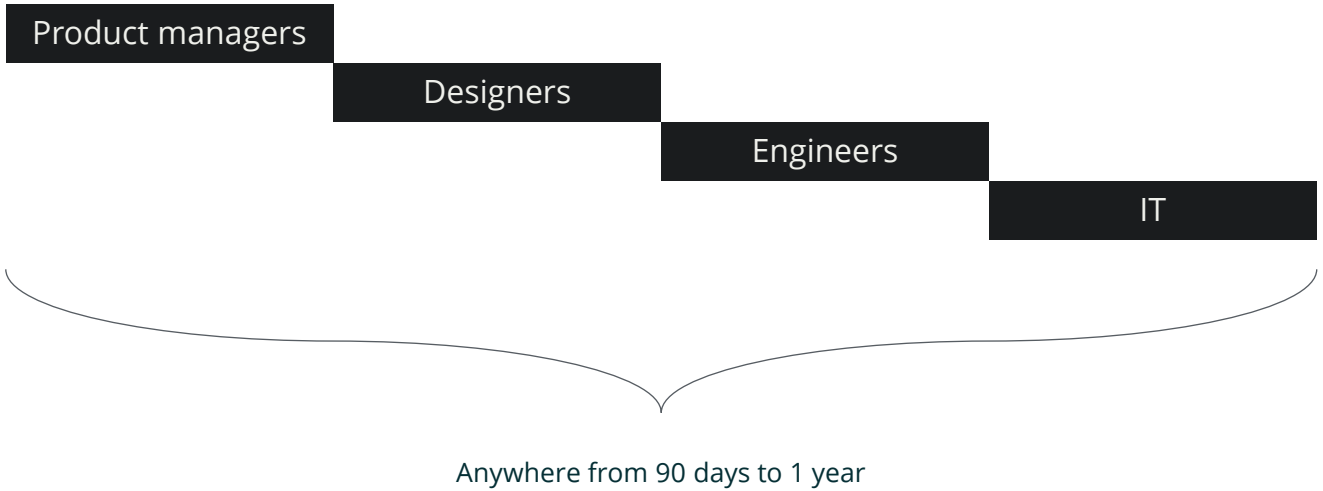
Common approaches to design and development coordination

Document and deliver (“Waterfall”)



Common approaches to design and development coordination

Document and deliver (“Waterfall”)



Common approaches to design and development coordination

Document and deliver (“Waterfall”)

Good for…

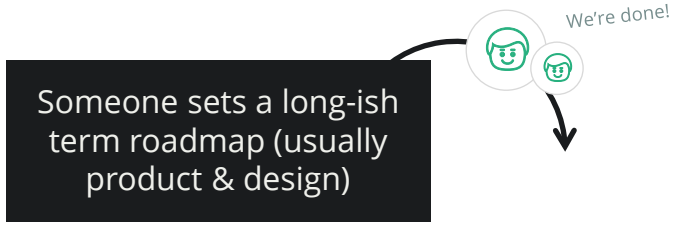
- Well defined problems—”we know exactly what to build”
- Commodity work, where there is a clear precedent and convention
- Offshore teams with a material disparity in skillset
- Working with a completely defined design toolkit
- When UI quality doesn’t matter (??!)
- Very old, conservative companies and thinking

Not good for…

- Rapidly changing problems
- Innovation and creativity
- A focus on users and customers
- Iteration and progressive feature definition
- Reacting to the market
- Modern development frameworks
- Attracting and retaining top-tier talent

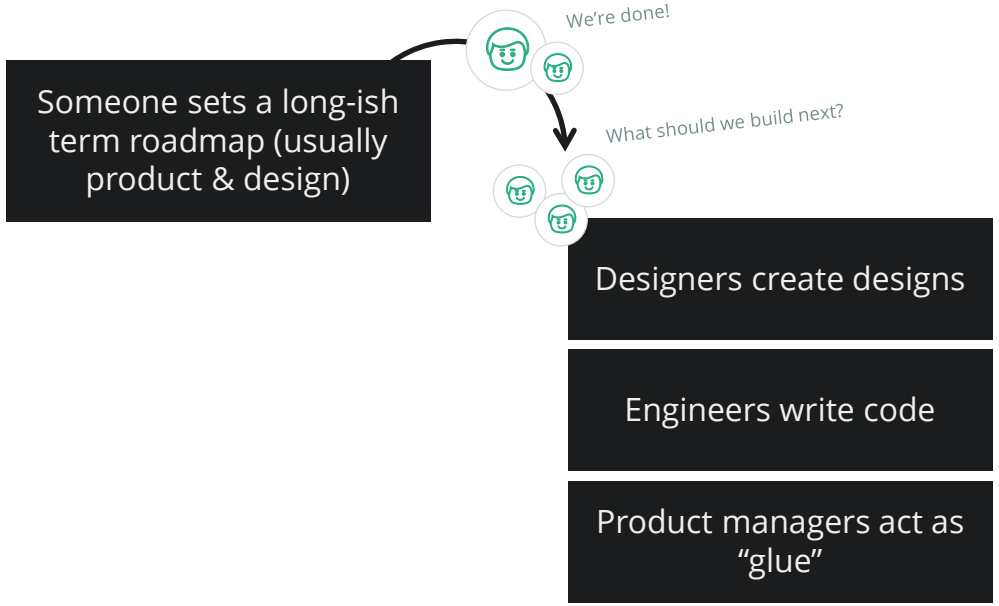
Common approaches to design and development coordination

Work together in a sprint or cycle



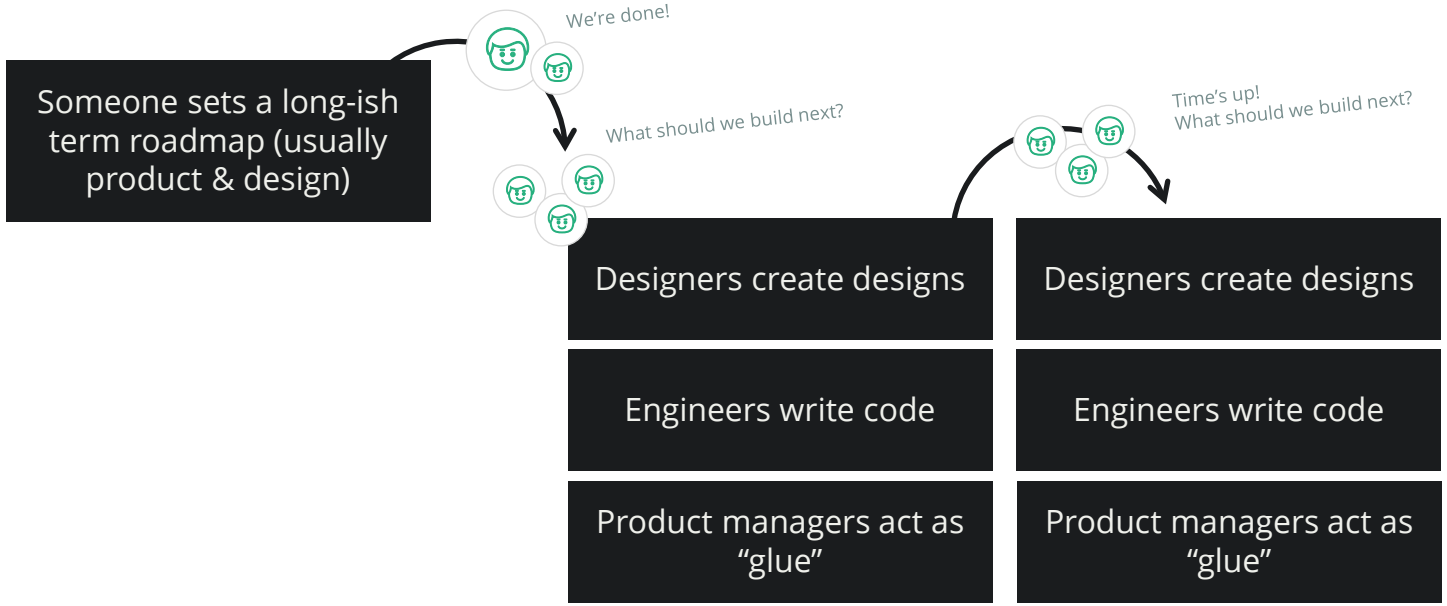
Common approaches to design and development coordination

Work together in a sprint or cycle



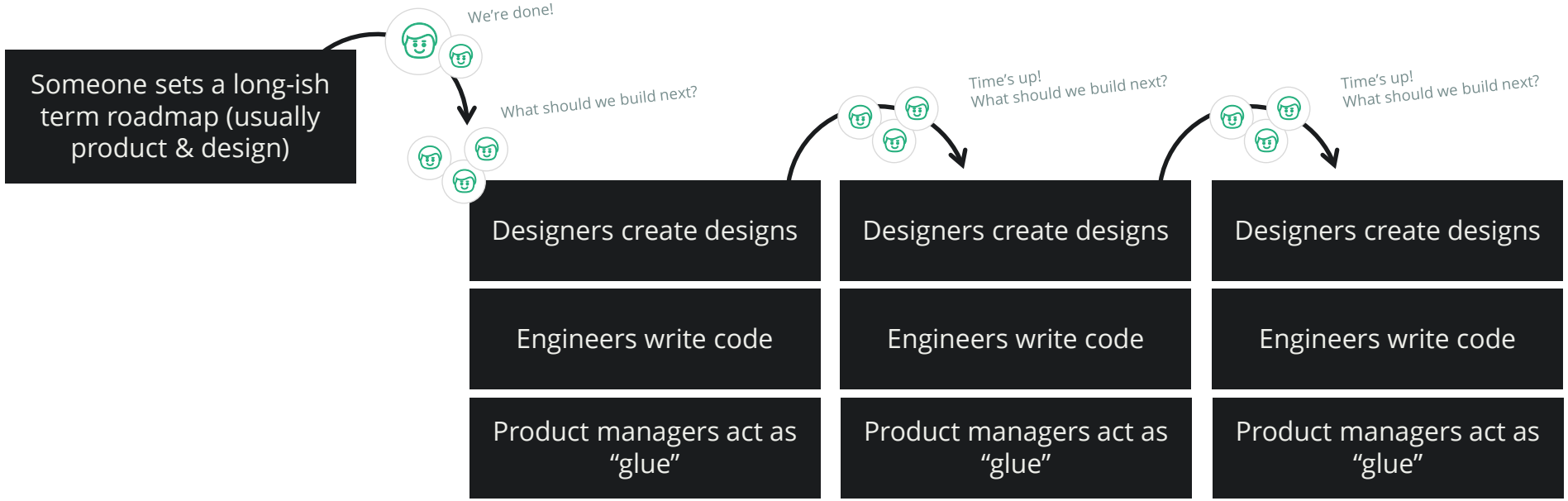
Common approaches to design and development coordination

Work together in a sprint or cycle



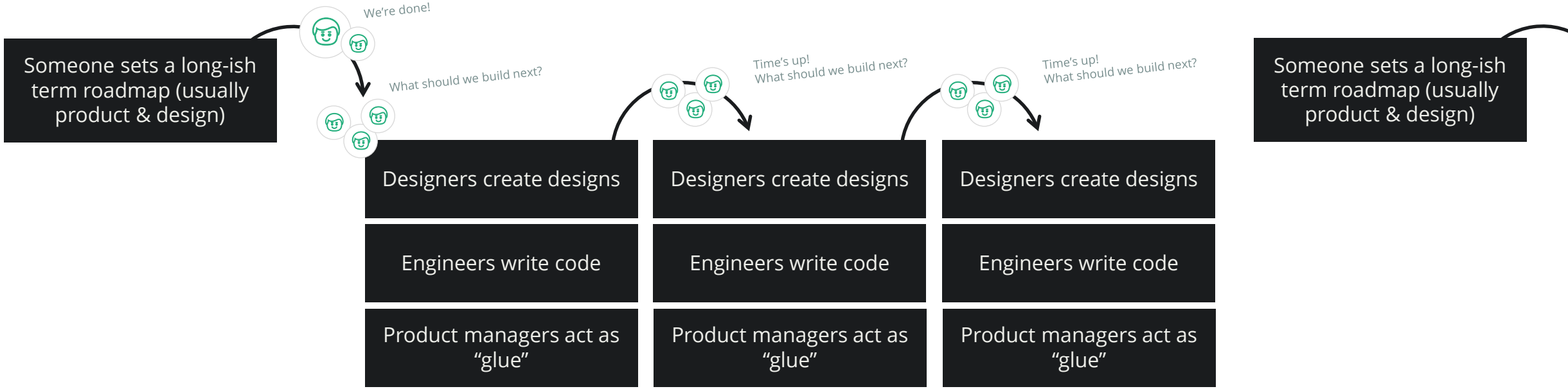
Common approaches to design and development coordination

Work together in a sprint or cycle



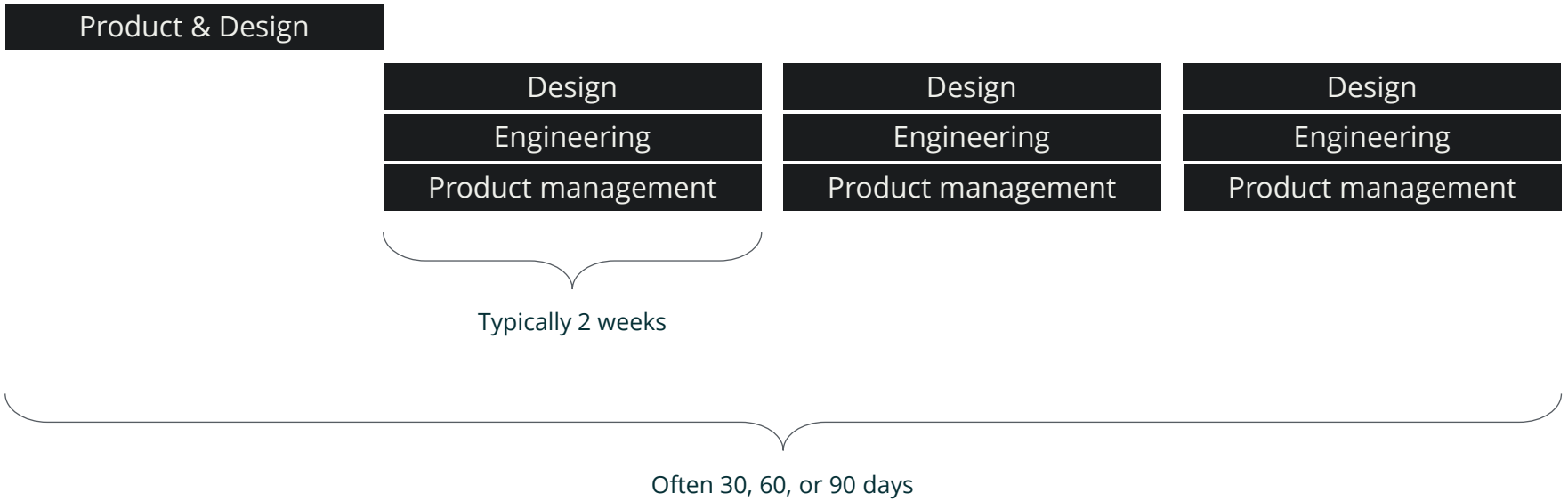
Common approaches to design and development coordination

Work together in a sprint or cycle



Common approaches to design and development coordination

Work together in a sprint or cycle



Common approaches to design and development coordination

Work together in a sprint or cycle

Good for...

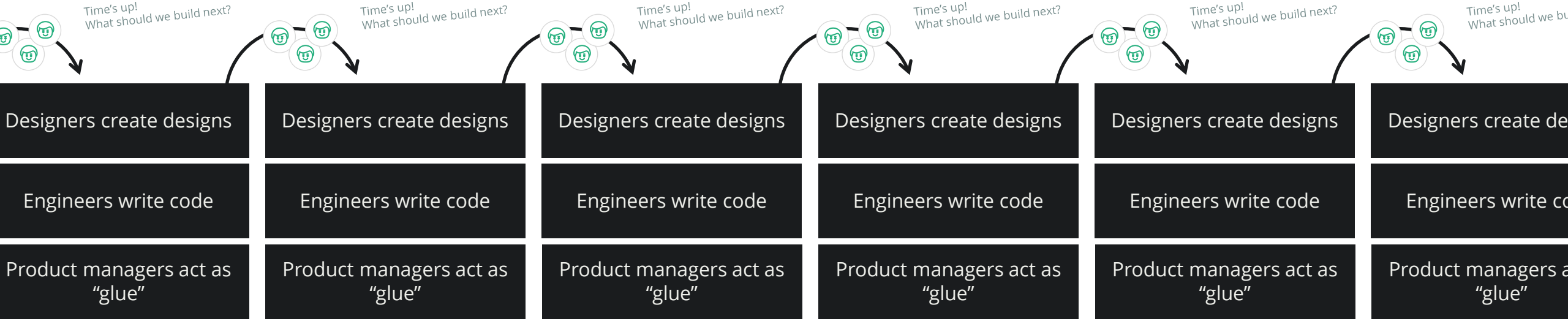
- Making and showing quick progress
- Feeling unified as a team
- Adjusting to changing business and market needs
- Launching and learning
- Startups
- Startup culture in a larger organization

Not good for...

- Feeling “fully organized”, as things are always in a state of change
- “Just tell me what to do” culture
- Junior talent
- Massive enterprise software
- Poorly organized codebases
- Poorly defined design systems

Common approaches to design and development coordination

“True agile” (whatever that means...)



Common approaches to design and development coordination

“True agile” (whatever that means...)

Good for...

- Making and showing quick progress
- Adjusting to changing business and market needs
- Launching and learning
- Startups
- Tiny teams

Not good for...

- Most designers, who usually feel like nothing is finished
- Users, who often encounter partially-built features
- Depth of thinking
- Vision and futuring
- Providing the business with a sense of what’s coming
- Communicating with the market

Common approaches to design and development coordination

The common problem of “we’ll fix it later”...

There is rarely a “later”

While product and development may fully intend to improve their work during later phases of a project, new features almost always push “later” to “never.”

Over time, a product accrues debt

- Technical debt comes from purposeful decisions to do things quickly instead of correctly
- Design debt comes from purposefully making compromises to move quicker
- Debt needs to be **repaid** (but frequently isn’t)...

What makes development difficult and time consuming?

Development concerns

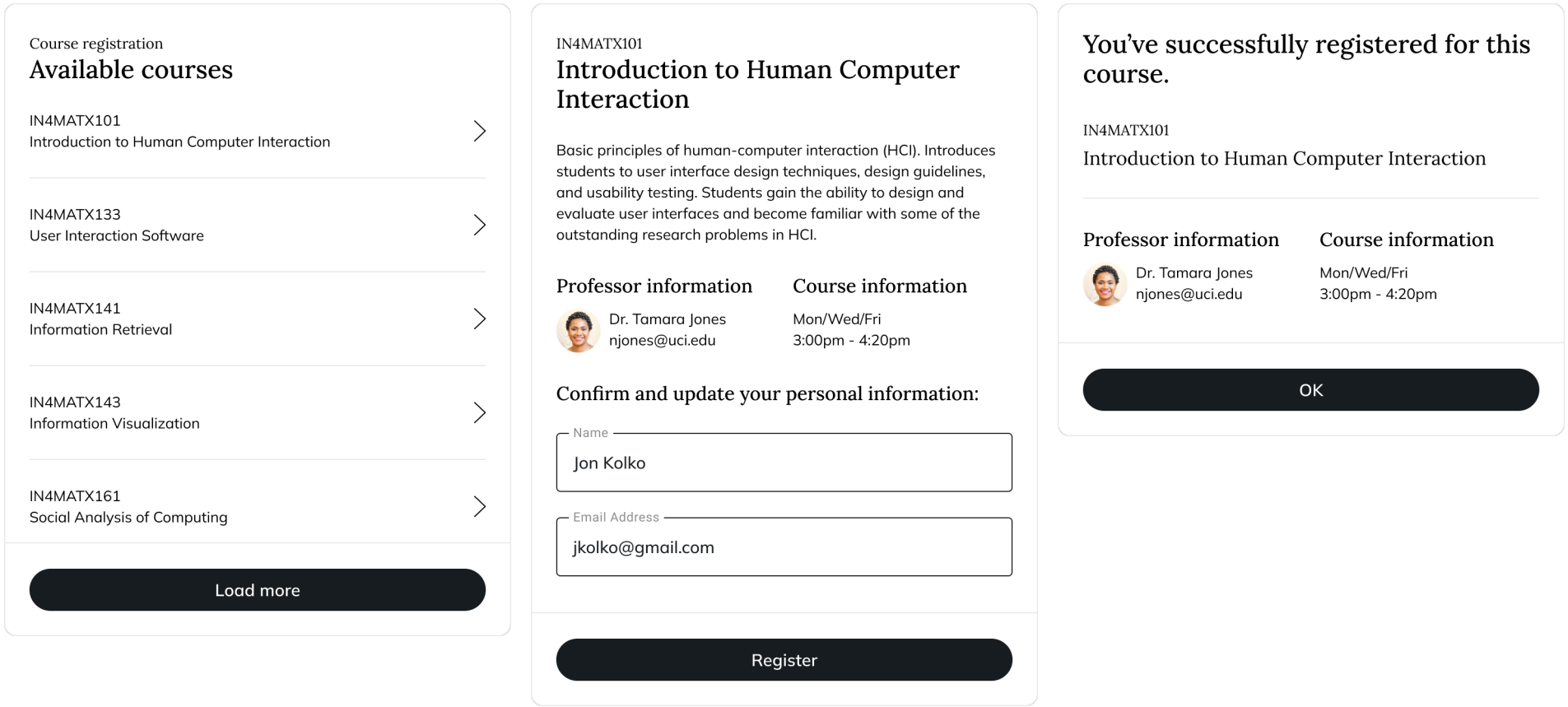
Developers have to care about a lot more things than just your design.

Their job includes more than you see, and your design has downstream impact you may not be aware of.

- Development strategy
- Feature dependency
- Data management
- Device support
- Accessibility
- Error handling
- Security
- Documentation
- Testing

For example...

You've designed a simple course registration flow:

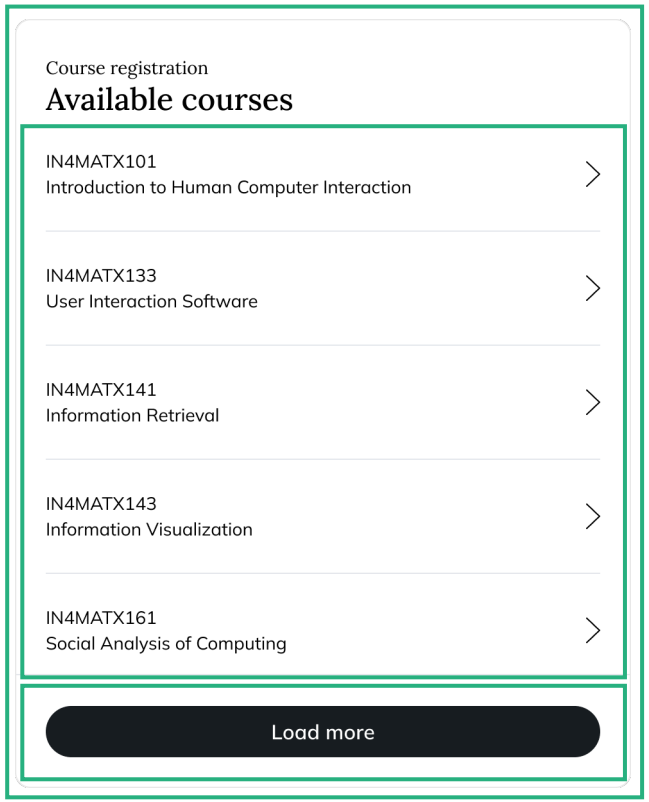


A developer will think about an implementation strategy.

How does the overall page layout work?

Where does the data come from?

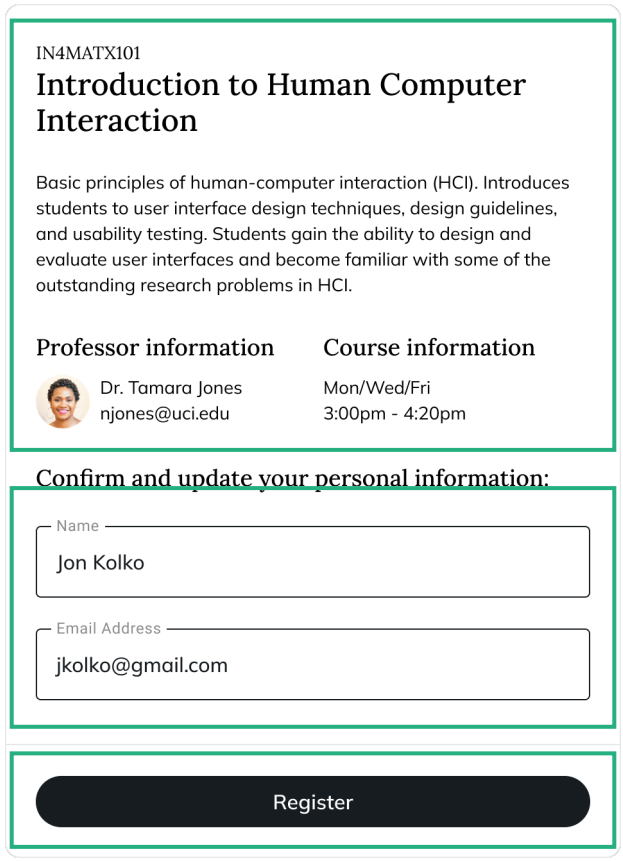
How does data loading and pagination work?



Where does the data come from?
What if it's not available?

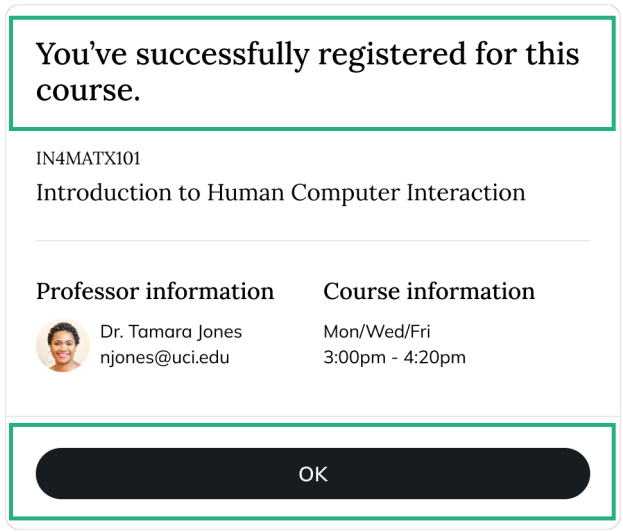
Where does this data come from?
Where does it go if it changes?

What does this button do?



What if this doesn't work?

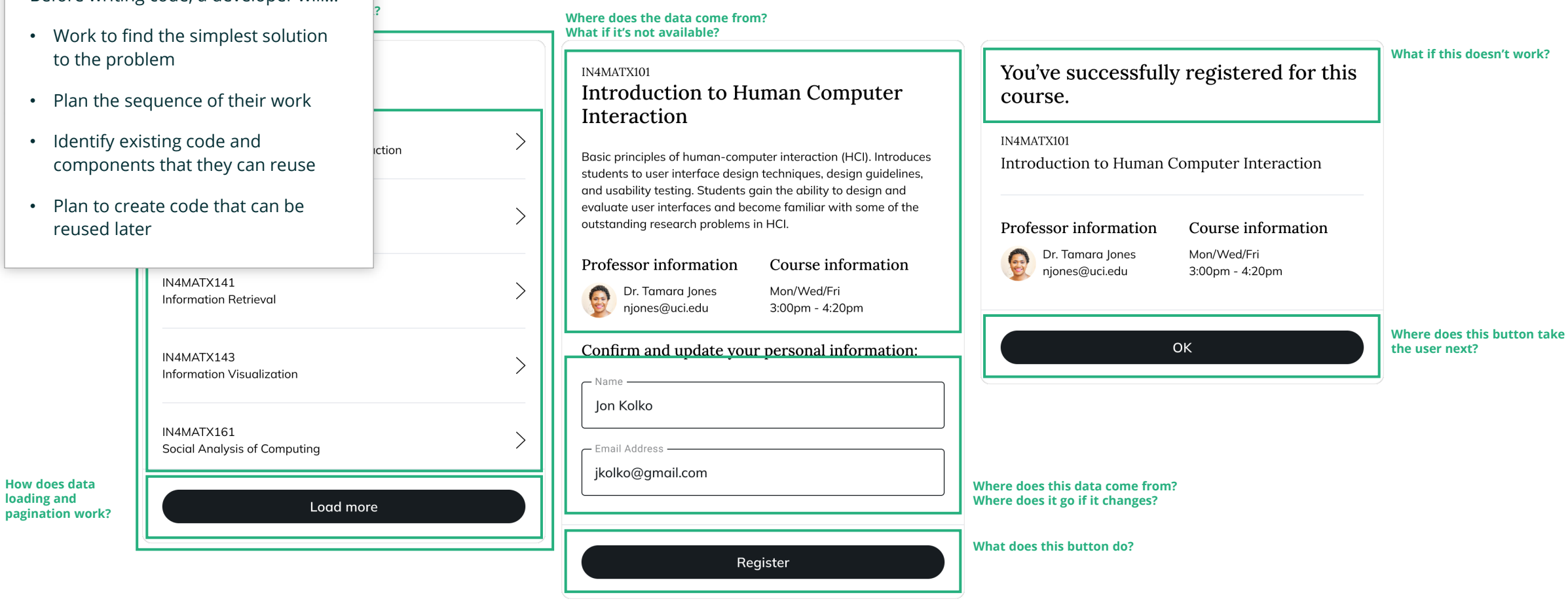
Where does this button take the user next?



A developer will think about an implementation strategy.

Before writing code, a developer will...

- Work to find the simplest solution to the problem
- Plan the sequence of their work
- Identify existing code and components that they can reuse
- Plan to create code that can be reused later



Some of their work may depend on other work-in-progress.

The courses haven't been loaded for the semester yet.

Course registration

Available courses

IN4MATX101
Introduction to Human Computer Interaction

IN4MATX133
User Interaction Software

IN4MATX141
Information Retrieval

IN4MATX143
Information Visualization

IN4MATX161
Social Analysis of Computing

Load more

The "avatar feature" is slated for the next release.

IN4MATX101

Introduction to Human Computer Interaction

Basic principles of human-computer interaction (HCI). Introduces students to user interface design techniques, design guidelines, and usability testing. Students gain the ability to design and evaluate user interfaces and become familiar with some of the outstanding research problems in HCI.

Professor information

Dr. Tamara Jones
njones@uci.edu

Course information

Mon/Wed/Fri
3:00pm - 4:20pm

Confirm and update your personal information:

Name: Jon Kolko

Email Address: jkolko@gmail.com

Register

You've successfully registered for this course.

IN4MATX101
Introduction to Human Computer Interaction

Professor information

Dr. Tamara Jones
njones@uci.edu

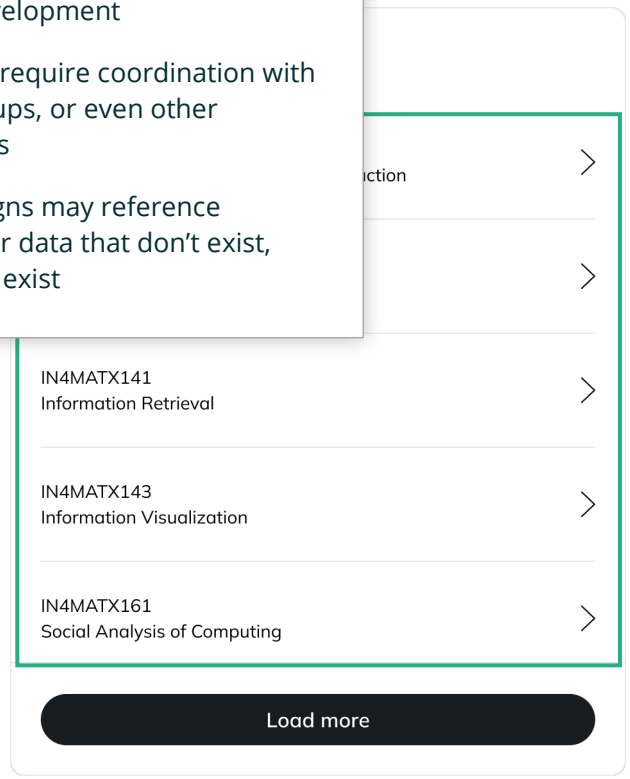
Course information

Mon/Wed/Fri
3:00pm - 4:20pm

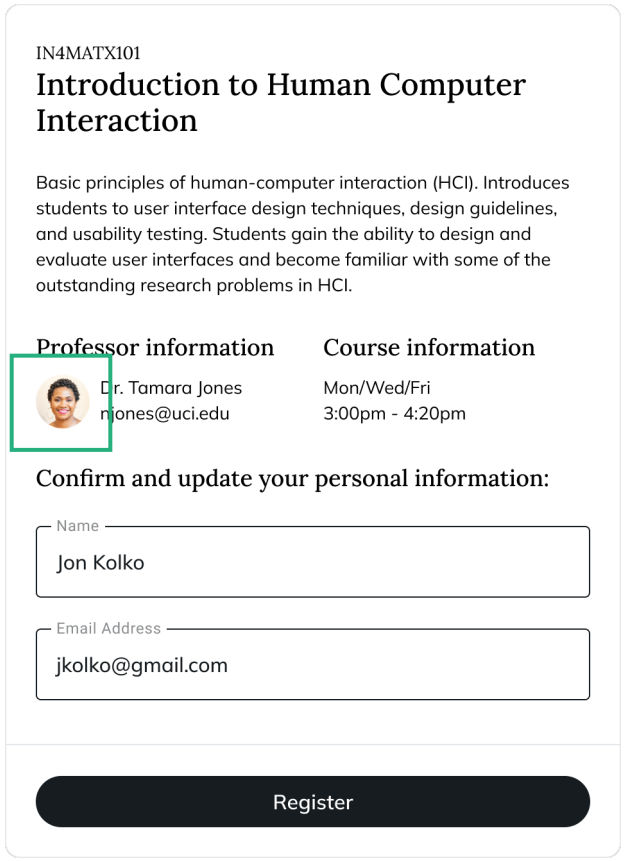
OK

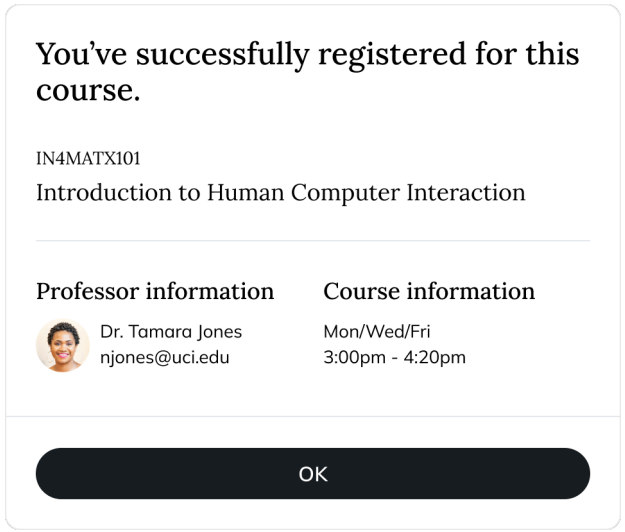
Some of their work may depend on other work-in-progress.

- Some features may depend on other features that are currently under development
- Data may require coordination with other groups, or even other companies
- Your designs may reference features or data that don't exist, and won't exist



The "avatar feature" is slated for the next release.





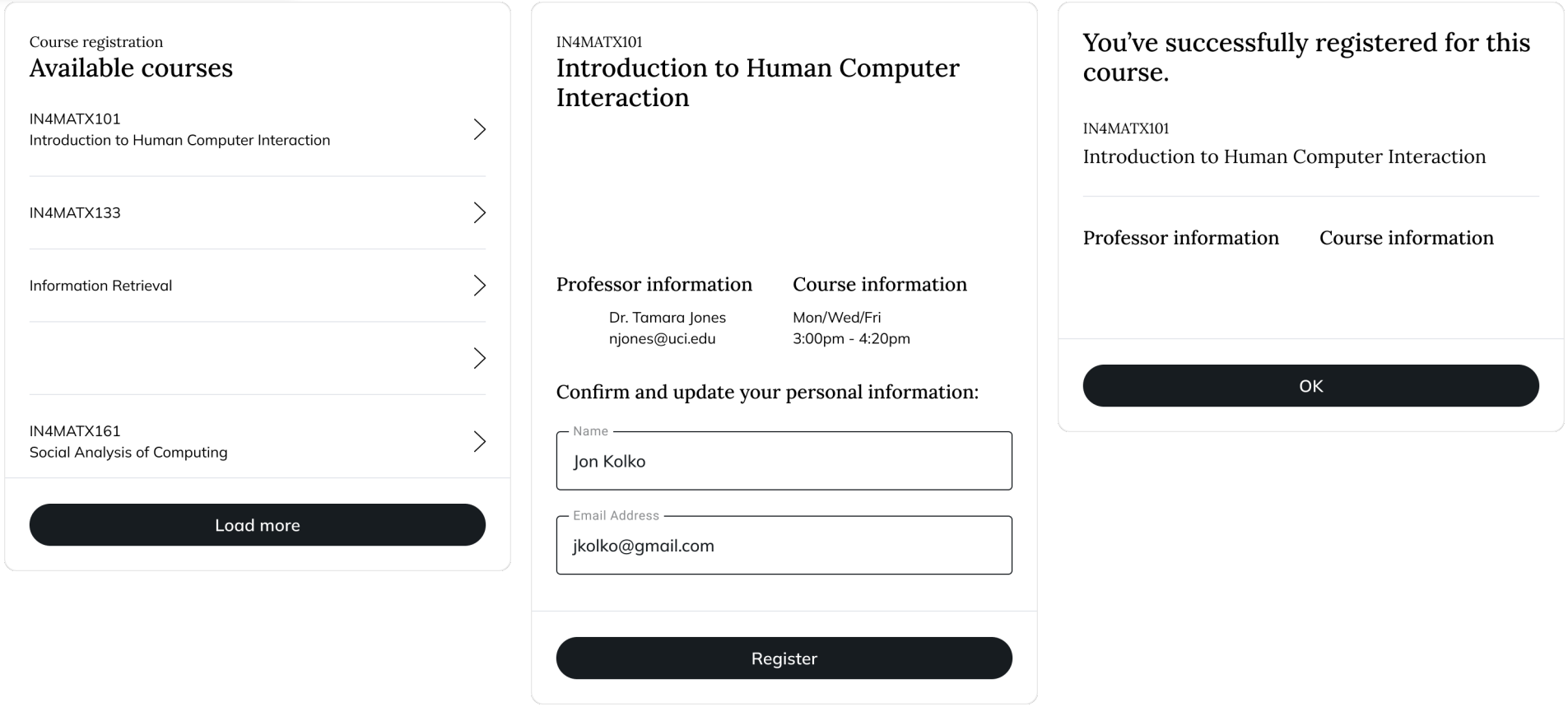
A *lot* of time is spent working with data.

Data quality

IN4MATX101 Introduction to Human Computer Interaction	 Dr. Tamara Jones njones@uci.edu	Mon/Wed/Fri 3:00pm - 4:20pm
IN4MATX101 INTRODUCTION TO HUMAN COMPUTER INTERACTION	 Tamara JonesDr. njones@uci.edu	MONWEDFRI 3:00PM-4:20PM
IN4MATX101 INTRODUCTION TO HUMAN COMPUTER INTERACTION IN4MATX101 INTRODUCTION TO HUMAN COMPUTER INTERACTION	 JonesDr.Tamara njones@uci.edu	Mon/Wed/Fri 15:00 - 16:20
IN4MATX101 Introduction to Human Computer Interaction Basic principles of human-computer interaction (HCI). Introduces students to user interface design techniques, design guidelines, and usability testing. Students gain the ability to design and evaluate user interfaces and become familiar with some of the outstanding research problems in HCI.	 Dr. Tamara Jones njones@uci.edu	

A *lot* of time is spent working with data.

Missing data



A *lot* of time is spent working with data.

Long strings and truncation

IN4MATX101
Introduction to Human Computer Interaction

IN4MATX101
Introduction to Human Computer Interaction and History of Computational Theory As It Applies to People, Places, and Environments

IN4MATX101

Introduction to Human Computer Interaction

Basic principles of human-computer interaction (HCI). Introduces students to user interface design techniques, design guidelines, and usability testing. Students gain the ability to design and evaluate user interfaces and become familiar with some of the outstanding research problems in HCI. HCI surfaced in the 1980s with the advent of personal computing, just as machines such as the Apple Macintosh, IBM PC 5150 and Commodore 64 started turning up in homes and offices in society-changing numbers. For the first time, sophisticated electronic systems were available to general consumers for uses such as word processors, games units and accounting aids. Consequently, as computers were no longer room-sized, expensive tools exclusively built for experts in specialized environments, the need to create human-computer interaction that was also easy and efficient for less experienced users became increasingly vital. From its origins, HCI would expand to incorporate multiple disciplines, such as computer science, cognitive science and human-factors engineering. Today, HCI focuses on designing, implementing, and evaluating interactive interfaces that enhance user experience using computing devices. This includes user interface design, user centered design, and user experience design. Students will study the various elements of this history.

Professor information


Dr. Tamara Jones
njones@uci.edu

Course information

Mon/Wed/Fri
3:00pm - 4:20pm

Confirm and update your personal information:

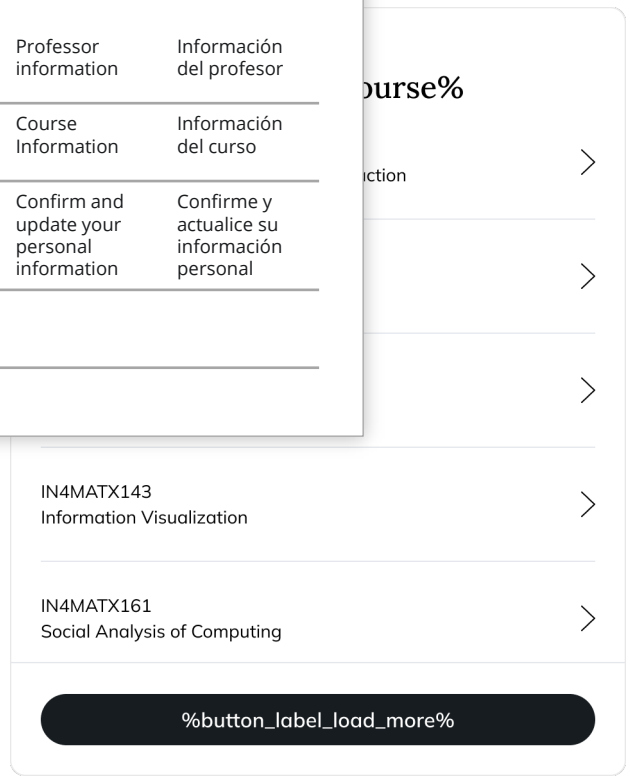

Dr. Tamara Jones
njones@uci.edu

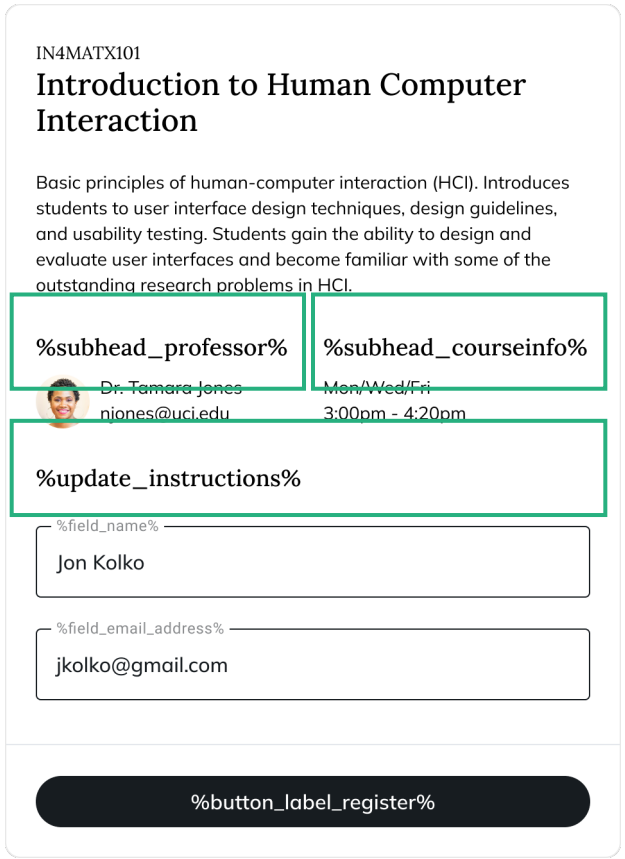

Dr. Tamara Weizenstein/Abromotiz
drtamaraweizenstein-abromotiz@uci.edu

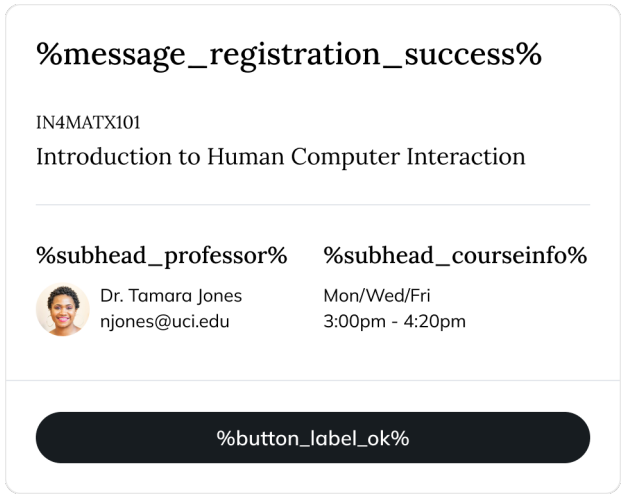
A *lot* of time is spent working with data.

String abstraction and localization

%subhead_professor%	Professor information	Información del profesor
%subhead_courseinfo%	Course Information	Información del curso
%update_instructions%	Confirm and update your personal information	Confirme y actualice su información personal
...		







A *lot* of time is spent working with data.

Form validation

IN4MATX101
Introduction to Human Computer Interaction

Basic principles of human-computer interaction (HCI). Introduces students to user interface design techniques, design guidelines, and usability testing. Students gain the ability to design and evaluate user interfaces and become familiar with some of the outstanding research problems in HCI.

Professor information
Dr. Tamara Jones
njones@uci.edu

Course information
Mon/Wed/Fri
3:00pm - 4:20pm

Confirm and update your personal information:

Name: Jon Kolko
Email Address: jkolko@gmail.com

Register

Name: Jon Kolko
Name: Jon Jones Kolko
First: Jon Jones — Last: Kolko
First: Fuck — Last: Off

Name: Jon
Name: Jonasdkfjakdsfjaodsidsajfaldskjfalsdjflkadsjflkasjdlkfaj
Name: Jon1 Kolko
Name: Jön Kölko
Name: יוֹנָתָן kolko

Development needs to accommodate platforms & breakpoints...

Obnoxious breakpoint sizes

IN4MATX101

Introduction to Human Computer Interaction

Basic principles of human-computer interaction (HCI). Introduces students to user interface design techniques, design guidelines, and usability testing. Students gain the ability to design and evaluate user interfaces and become familiar with some of the outstanding research problems in HCI.

Professor information

Dr. Tamara Jones
njones@uci.edu

Course information

Mon, Wed, Fri
3:00pm - 4:20pm

IN4MATX101

Introduction to Human Computer Interaction

Basic principles of human-computer interaction (HCI). Introduces students to user interface design techniques, design guidelines, and usability testing. Students gain the ability to design and evaluate user interfaces and become familiar with some of the outstanding research problems in HCI.

Professor information

Dr. Tamara Jones
njones@uci.edu

Course information

Mon/Wed/Fri
3:00pm - 4:20pm

Confirm and update your personal information:

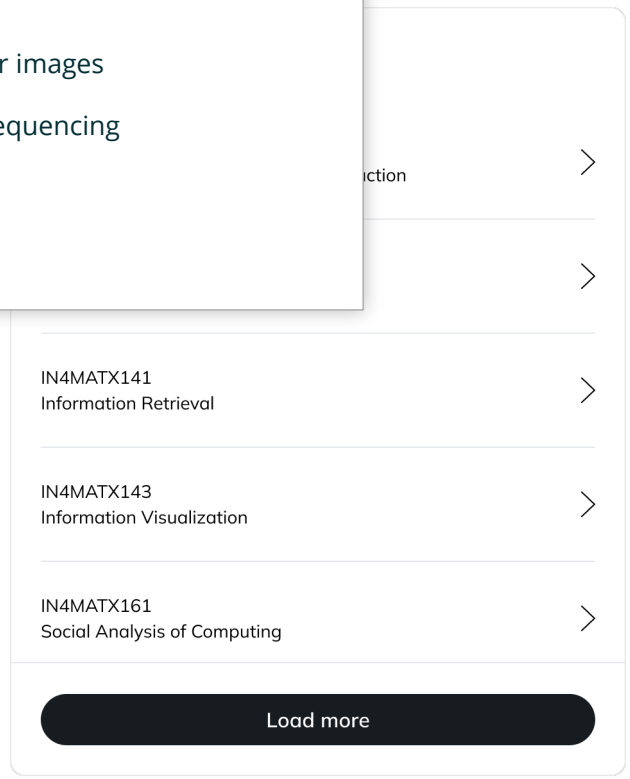
Name
Jon Kolko

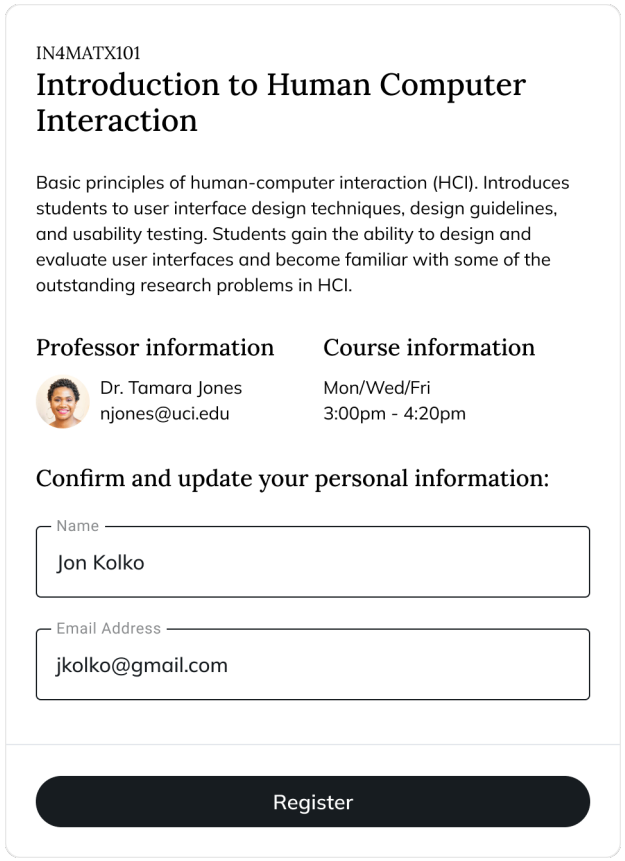
Email Address
jkolko@gmail.com

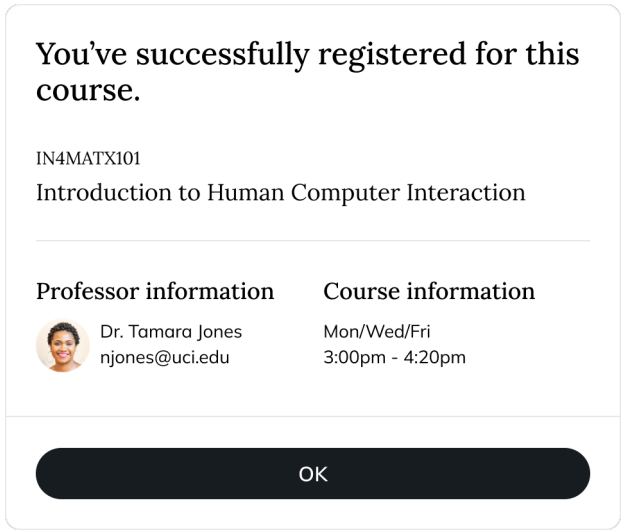
Register

Accessibility...

- Keyboard access
- Tab order
- Alt tags for images
- Markup sequencing
- ARIA
- ...



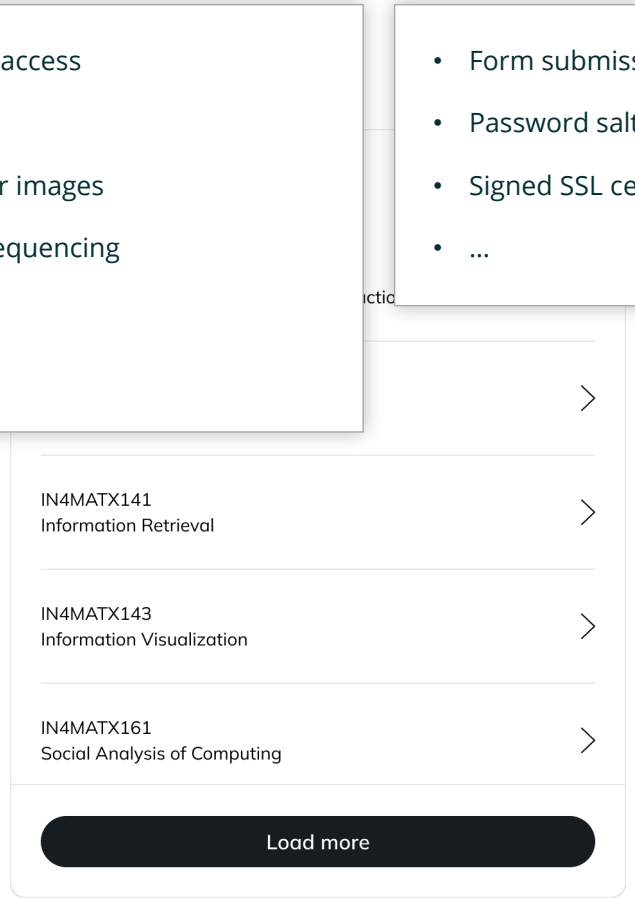


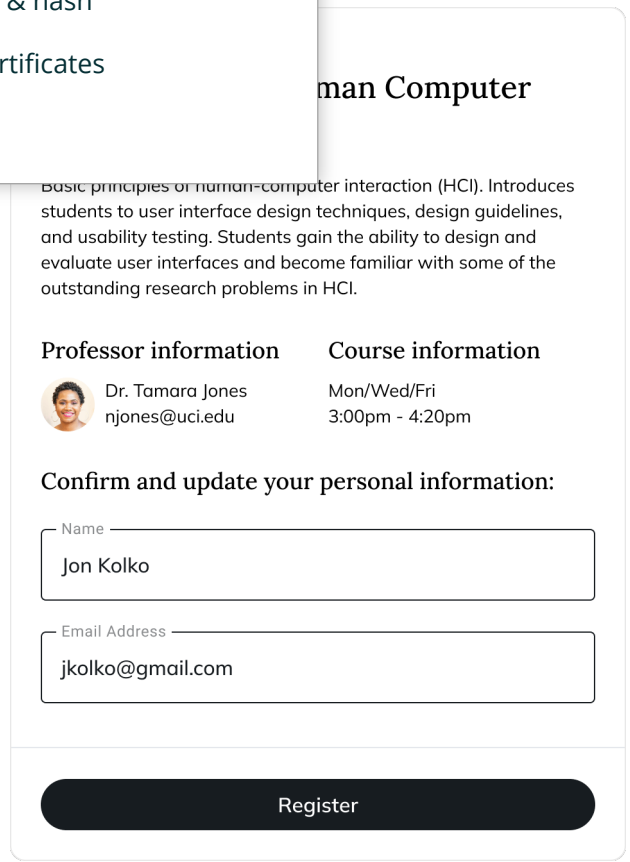


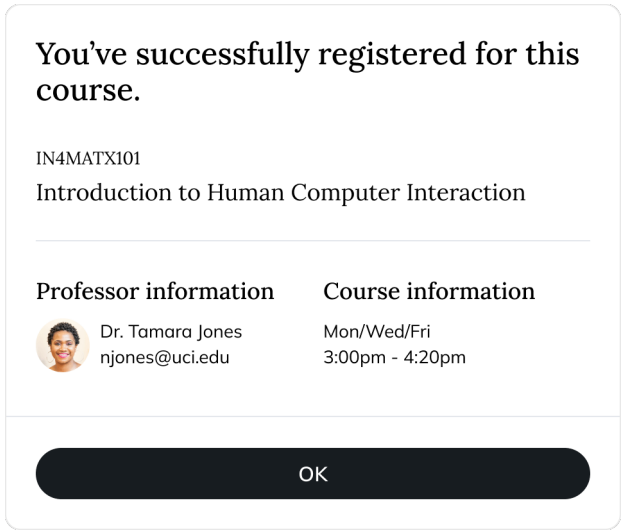
Security...

- Keyboard access
- Tab order
- Alt tags for images
- Markup sequencing
- ARIA
- ...

- Form submission
- Password salt & hash
- Signed SSL certificates
- ...





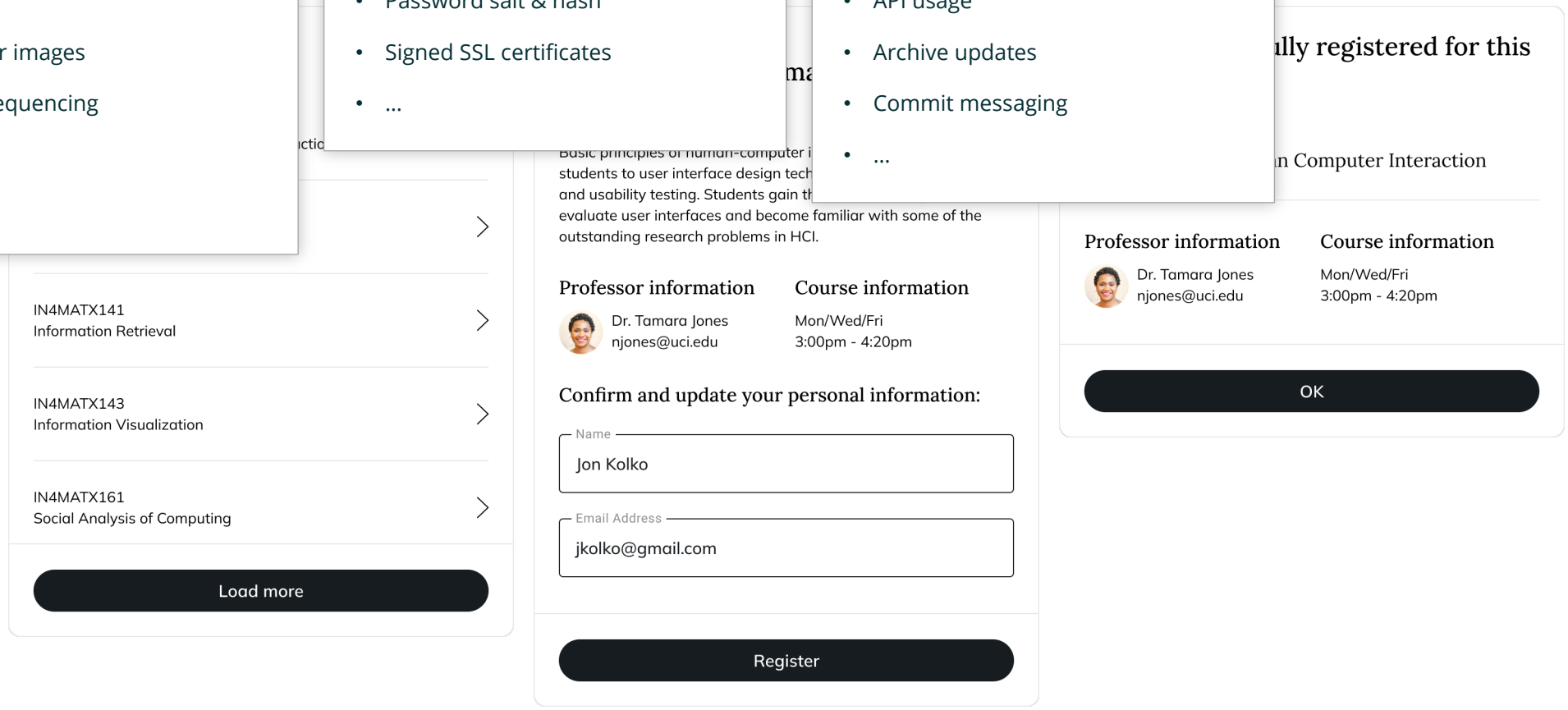


Documentation...

- Keyboard access
- Tab order
- Alt tags for images
- Markup sequencing
- ARIA
- ...

- Form submission
- Password salt & hash
- Signed SSL certificates
- ...

- Code commenting
- API usage
- Archive updates
- Commit messaging
- ...



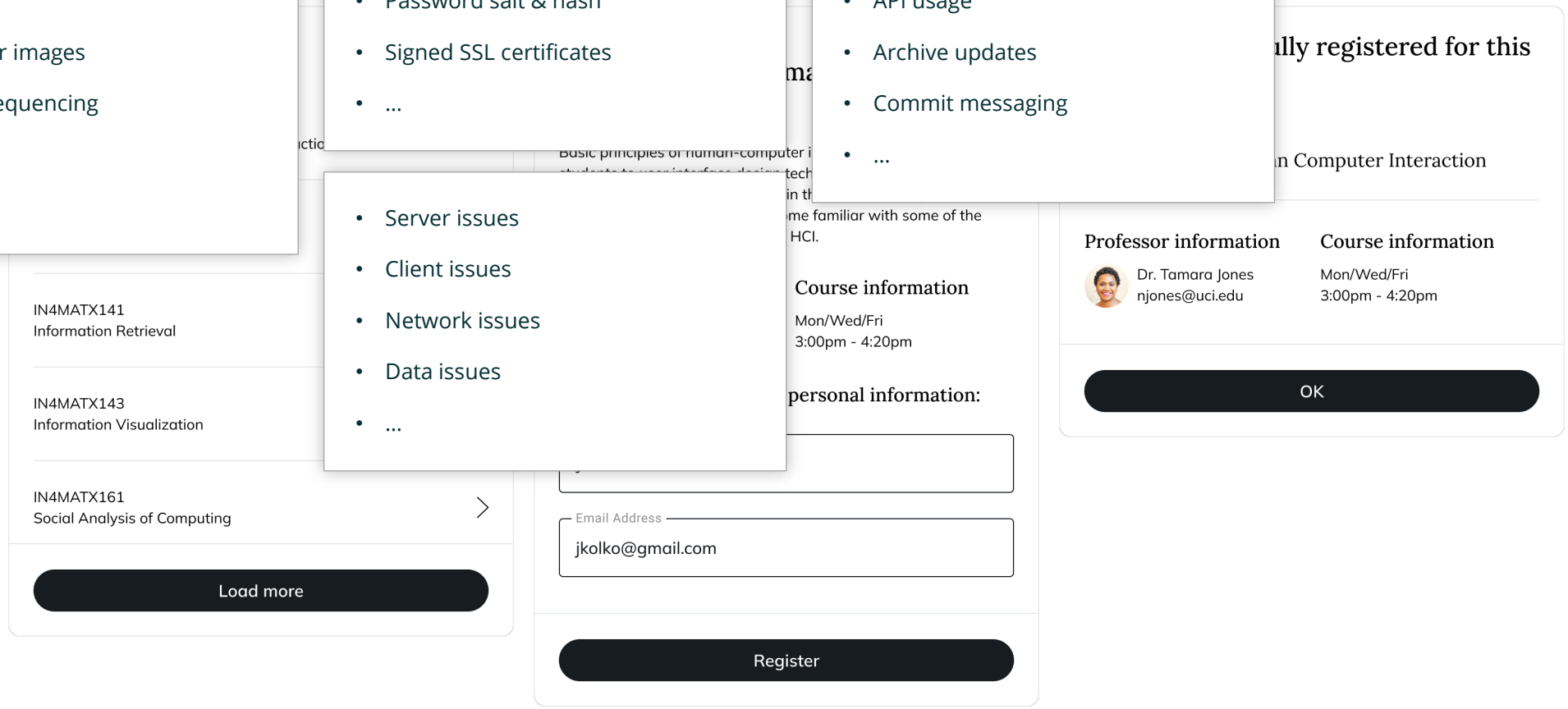
Error handling...

- Keyboard access
- Tab order
- Alt tags for images
- Markup sequencing
- ARIA
- ...

- Form submission
- Password salt & hash
- Signed SSL certificates
- ...

- Server issues
- Client issues
- Network issues
- Data issues
- ...

- Code commenting
- API usage
- Archive updates
- Commit messaging
- ...



Testing...

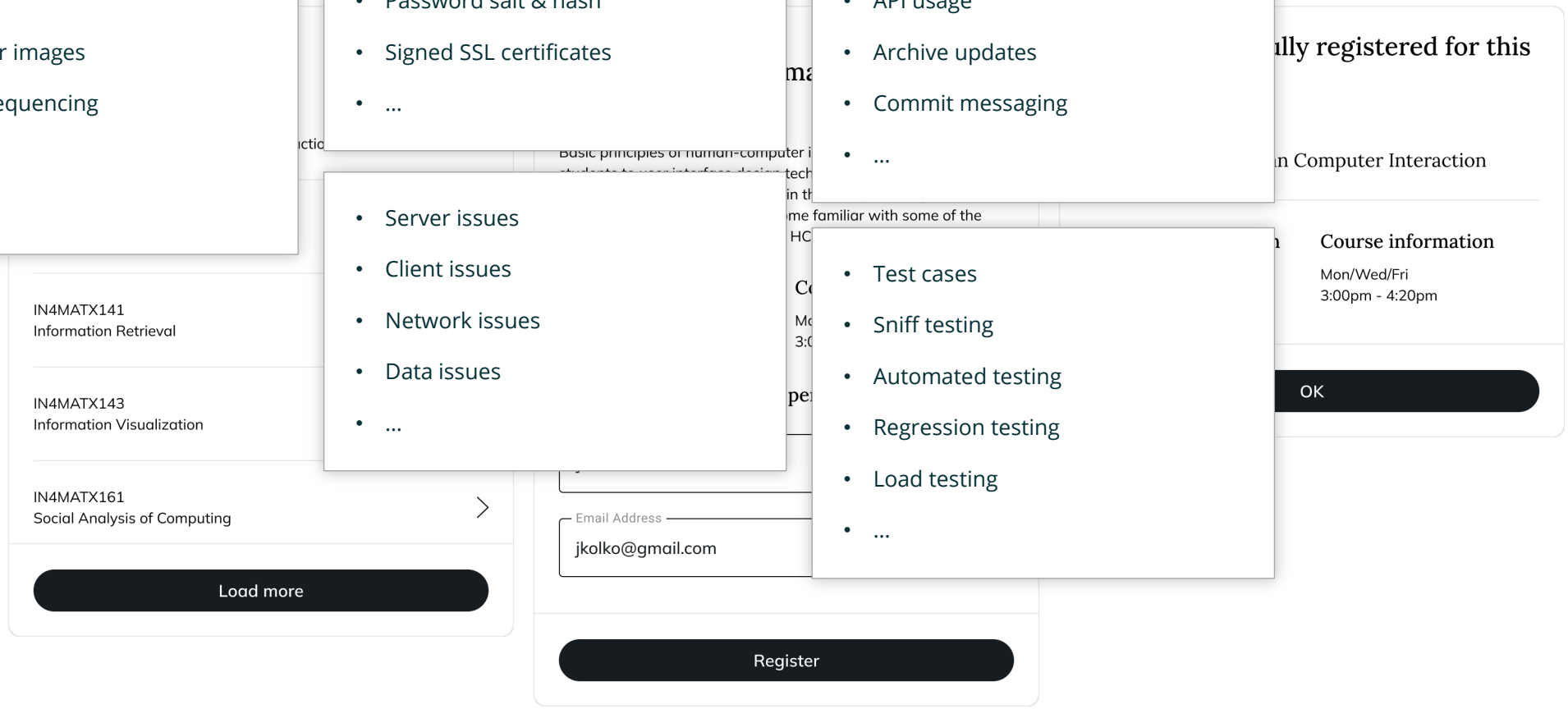
- Keyboard access
- Tab order
- Alt tags for images
- Markup sequencing
- ARIA
- ...

- Form submission
- Password salt & hash
- Signed SSL certificates
- ...

- Code commenting
- API usage
- Archive updates
- Commit messaging
- ...

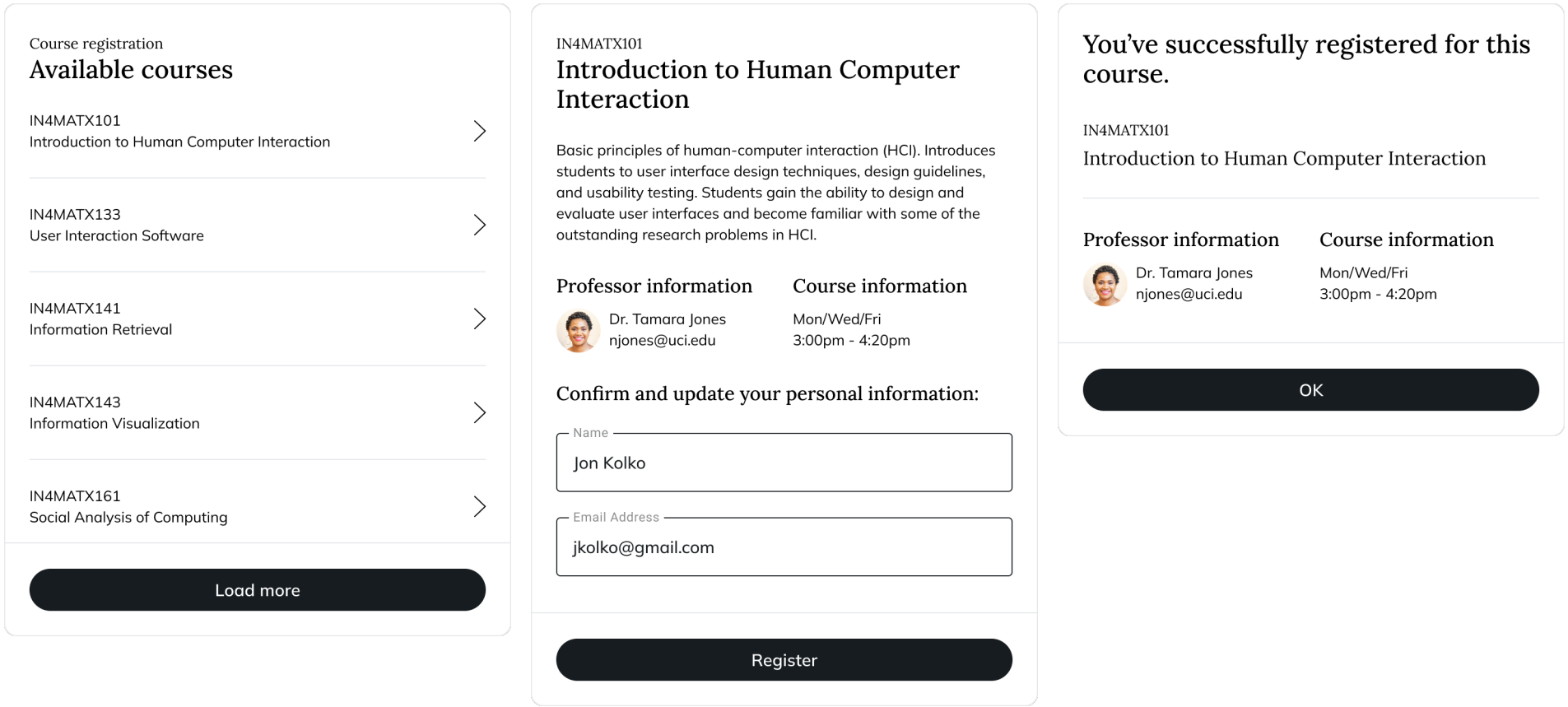
- Server issues
- Client issues
- Network issues
- Data issues
- ...

- Test cases
- Sniff testing
- Automated testing
- Regression testing
- Load testing
- ...



For example...

“That will take two months to build” starts to make sense...



Your job doesn't end at design

You can support development efforts with many of these tasks and items.

- Feature dependency—you often have a view of the whole system that others may not have
- Data management—you can specify data intent and ideal states, and define the rules for form validation
- Device support—you can design with responsiveness and agentness in your mind
- Accessibility—you can understand and push best practices
- Error handling—you can supply error messaging and rules

Your role before, during, and after development

Tickets and stories

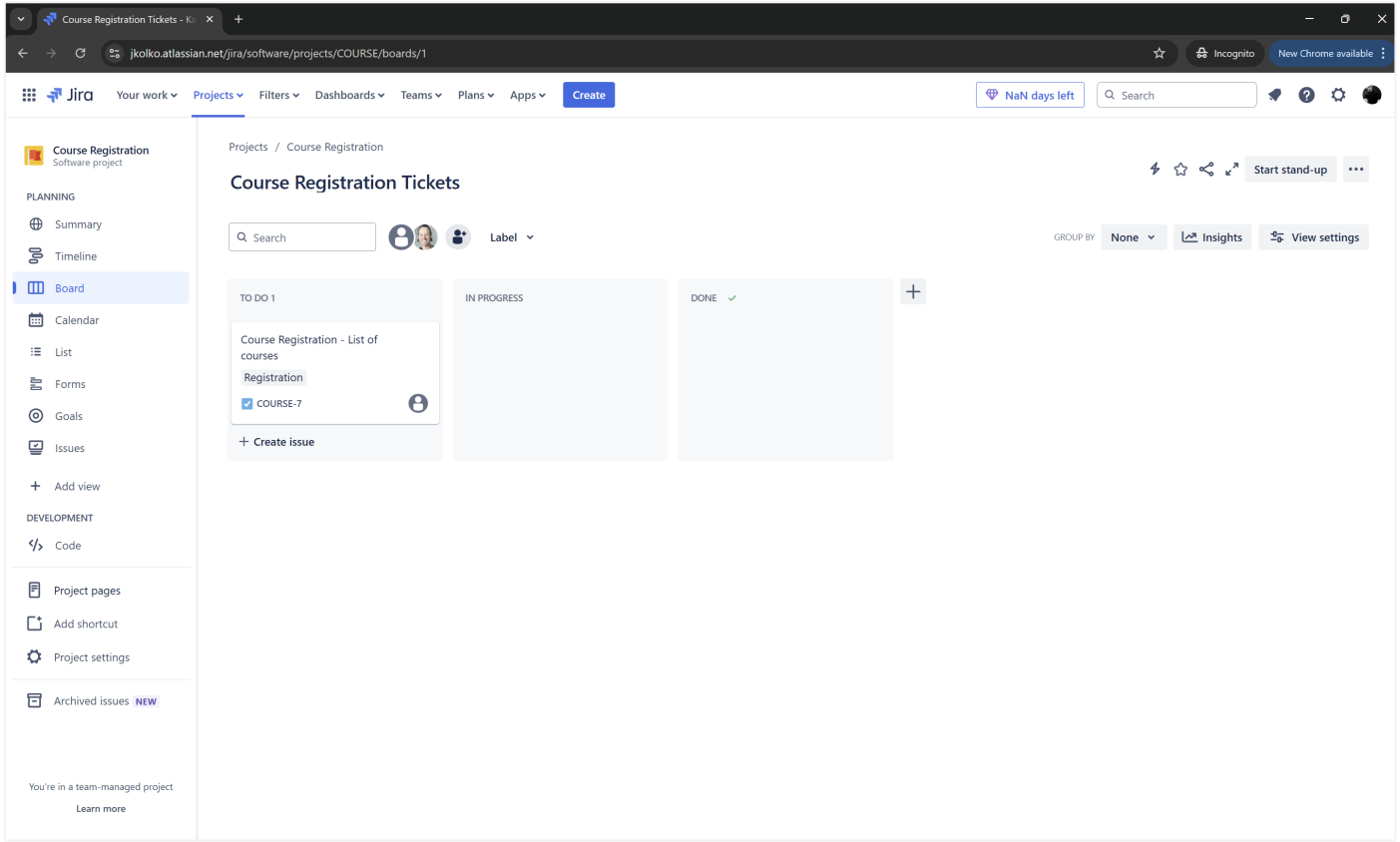
A “ticketing system” is used in nearly all software development practices.

Tickets act as a definition for what to build

- Act as a container for various details (that extend beyond design)
- Make it somewhat simple to track progress across larger capabilities
- Provide a centerpiece for conversation, to focus and align meetings
- Provide a central sources of truth for decisions that have been made

Tickets and stories

Jira is a common ticketing system.



Tickets and stories

You have multiple responsibilities during development, particularly if you want your design to see the light of day.

Before (or soon after) development starts...

- Establishing the “stubs” of the work you will be including
- Socializing your designs
- Documenting and attaching your design work
- Describing what a user will do with your designs
- Indicating how a developer will know they have completed the work successfully

As development continues...

- Viewing in-progress builds
- Answering questions
- Redesigning, based on new information and constraints
- Documenting defects

When development is completed

- Viewing live capabilities
- Documenting defects
- Being a squeaky wheel

Before (or soon after) development starts

Getting ready and collaborating with product management

Getting ready and collaborating with product management

Pre-game your design with everyone that might encounter it—there should be no surprises for anyone!

Make sure the developers *really* understand it

- Show your work in a casual way, not a formal presentation
- Discuss why you are making various decisions
- Ask if various data streams are available
- Learn about what will make various parts of the work difficult or overly time consuming
- Explain what you feel can be negotiable, and what is a bare minimum

Getting ready and collaborating with product management

Anticipate that you may need to split your design into smaller pieces; have those designs ready.

IN4MATX101
Introduction to Human Computer Interaction

Basic principles of human-computer interaction (HCI). Introduces students to user interface design techniques, design guidelines, and usability testing. Students gain the ability to design and evaluate user interfaces and become familiar with some of the outstanding research problems in HCI.

Course information

Mon/Wed/Fri
3:00pm - 4:20pm

Register

IN4MATX101
Introduction to Human Computer Interaction

Basic principles of human-computer interaction (HCI). Introduces students to user interface design techniques, design guidelines, and usability testing. Students gain the ability to design and evaluate user interfaces and become familiar with some of the outstanding research problems in HCI.

Professor information


Dr. Tamara Jones
njones@uci.edu

Course information

Mon/Wed/Fri
3:00pm - 4:20pm

Register

IN4MATX101
Introduction to Human Computer Interaction

Basic principles of human-computer interaction (HCI). Introduces students to user interface design techniques, design guidelines, and usability testing. Students gain the ability to design and evaluate user interfaces and become familiar with some of the outstanding research problems in HCI.

Professor information


Dr. Tamara Jones
njones@uci.edu

Course information

Mon/Wed/Fri
3:00pm - 4:20pm

Confirm and update your personal information:

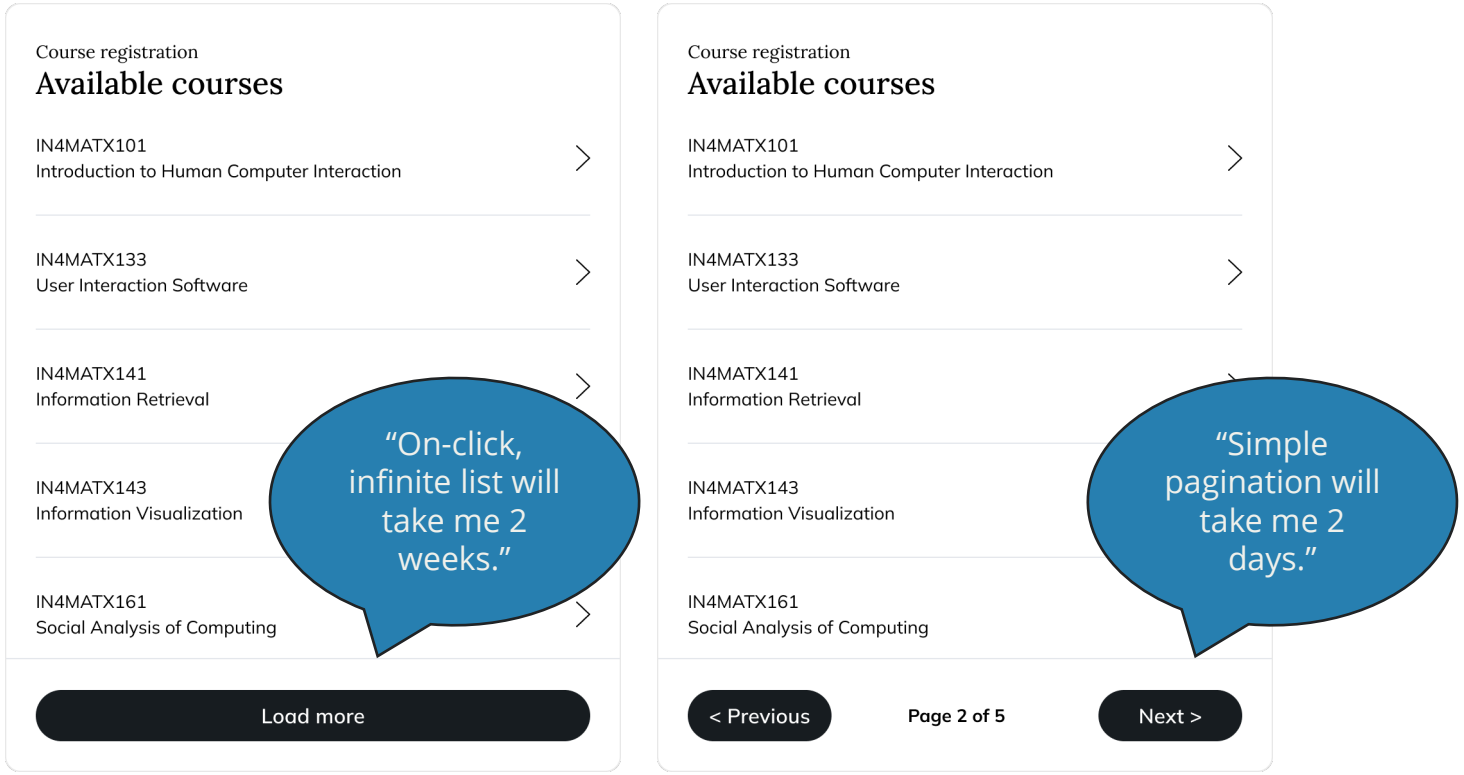
Name
Jon Kolko

Email Address
jkolko@gmail.com

Register

Getting ready and collaborating with product management

Anticipate features that are notoriously difficult, and have an alternative design ready.



Getting ready and collaborating with product management

Write a strong, detailed, comprehensive story for your work; include the story overview, design specifications, non-obvious details, and acceptance criteria.

- **The story overview** describes who will use your design, what they will use it for, and why they will use it.
- **Design specifications** are your design assets: flows, images, prototypes, written documents, and any other material that will help developers understand your intention.
- **Non-obvious details** are the rules for the page that aren't always evident in your design materials.
- **Acceptance criteria** are the details that indicate a developer has successfully completed their work, so that it can be launched.

Getting ready and collaborating with product management

For example...

Course registration

Available courses

IN4MATX101
Introduction to Human Computer Interaction

IN4MATX133
User Interaction Software

IN4MATX141
Information Retrieval

IN4MATX143
Information Visualization

IN4MATX161
Social Analysis of Computing

Load more

Getting ready and collaborating with product management

For example...

Course registration

Available courses

IN4MATX101
Introduction to Human Computer Interaction

IN4MATX133
User Interaction Software

IN4MATX141
Information Retrieval

IN4MATX143
Information Visualization

IN4MATX161
Social Analysis of Computing

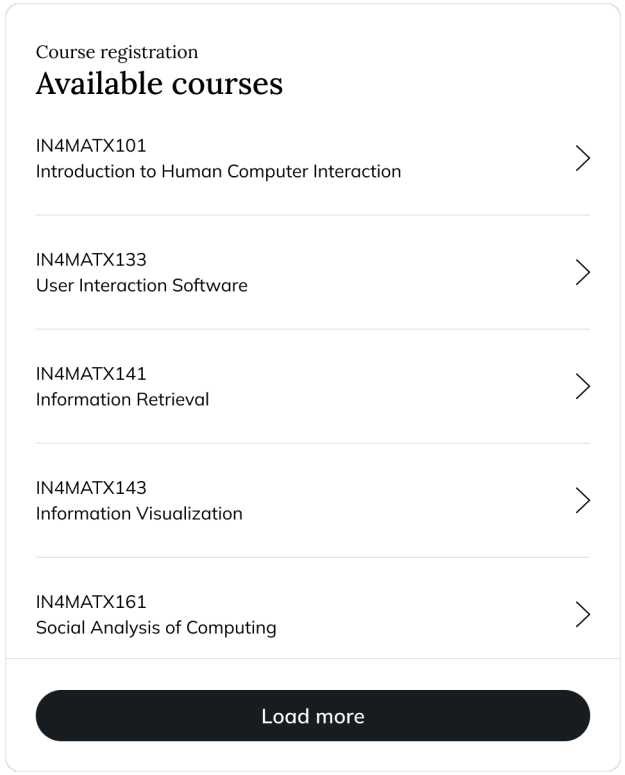
Load more

Overview

As a student, I want to see the courses that are available, so that I can view information about them and register for class.

Getting ready and collaborating with product management

For example...



Overview

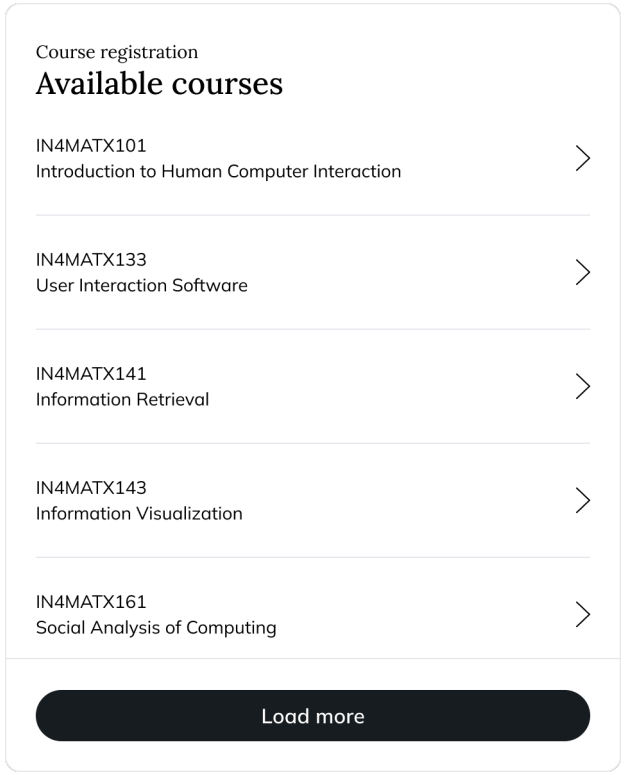
As a student, I want to see the courses that are available, so that I can view information about them and register for class.

Non-obvious details

- Five courses are displayed on page load
- Courses are sorted by course identifier, from lowest to highest
- Course identifiers are always in capital letters
- Course names are always in Title Case, with these Minor Words in lower case: a, an, the, of, in, on, to
- The “Load more” button appends another five courses at the end of the list, and automatically scrolls the screen to display the first new course at the top
- When there are no additional courses to display, the “Load more” button disappears from the screen
- Clicking on a course name takes the user to the respective Course Details page

Getting ready and collaborating with product management

For example...



Overview

As a student, I want to see the courses that are available, so that I can view information about them and register for class.

Non-obvious details

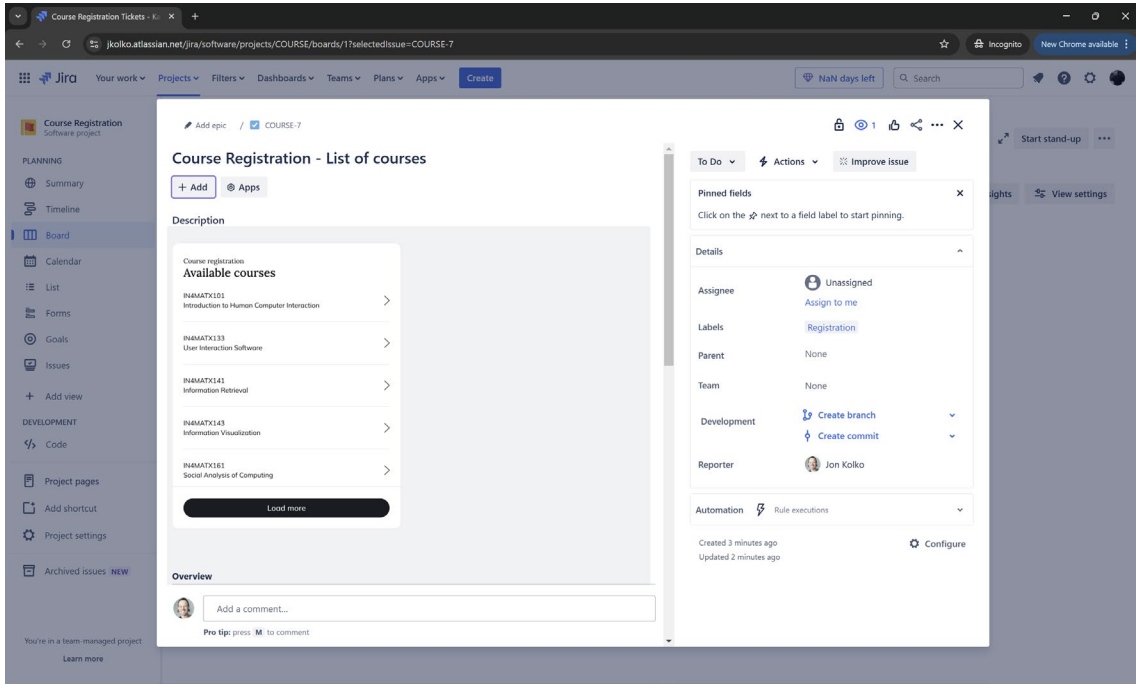
- Five courses are displayed on page load
- Courses are sorted by course identifier, from lowest to highest
- Course identifiers are always in capital letters
- Course names are always in Title Case, with these Minor Words in lower case: a, an, the, of, in, on, to
- The “Load more” button appends another five courses at the end of the list, and automatically scrolls the screen to display the first new course at the top
- When there are no additional courses to display, the “Load more” button disappears from the screen
- Clicking on a course name takes the user to the respective Course Details page

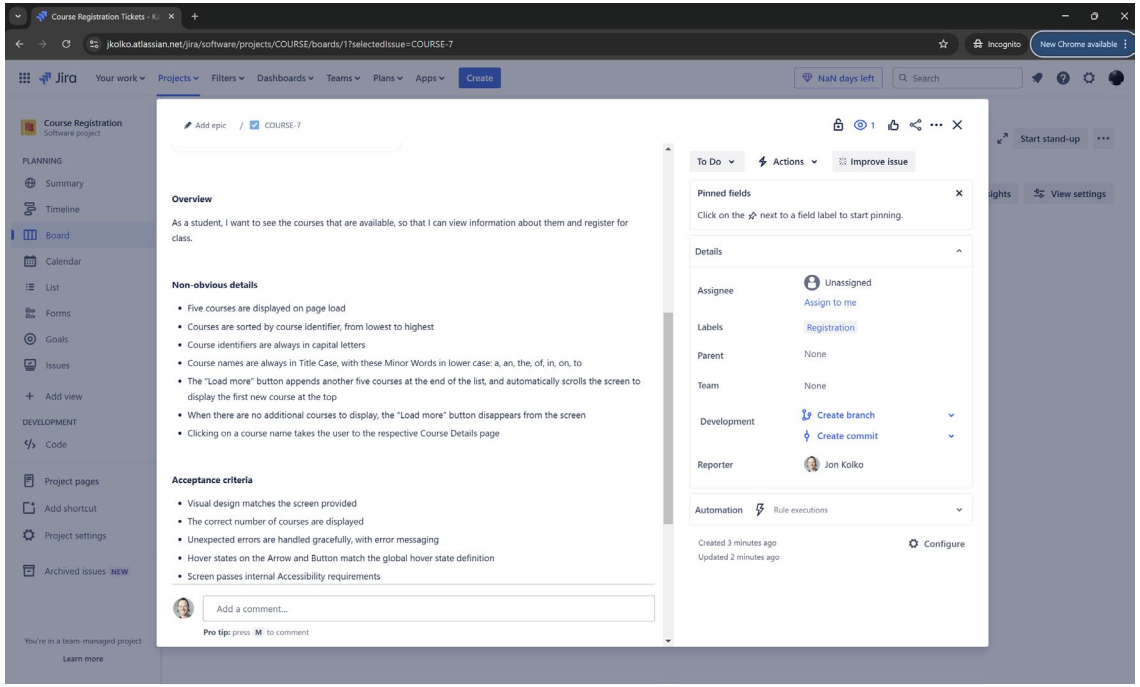
Acceptance criteria

- Visual design matches the screen provided
- The correct number of courses are displayed
- Unexpected errors are handled gracefully, with error messaging
- Hover states on the Arrow and Button match the global hover state definition
- Screen passes internal Accessibility requirements
- All strings translates successfully to Spanish when Spanish language is set by the user in their profile
- Unexpected long strings break to the next line, and the layout adjusts as defined in the provided screens

Getting ready and collaborating with product management

Add your material to your ticket.





As development continues…

Supporting your design during development

Supporting your design during development

Your primary role during development is to support your developers.

Support through conversation

- Don’t use the ticket as a way to hide—have conversations first
- Ask questions; when you’re asked to do something, understand why before you do it
- Move quickly—don’t hold up progress—but hold on to craft and method

Support through tickets

- Document everything in the ticket as a central source of truth
- Be specific and detailed
- Attach assets, even if you’ve already provided them

Supporting your design during development

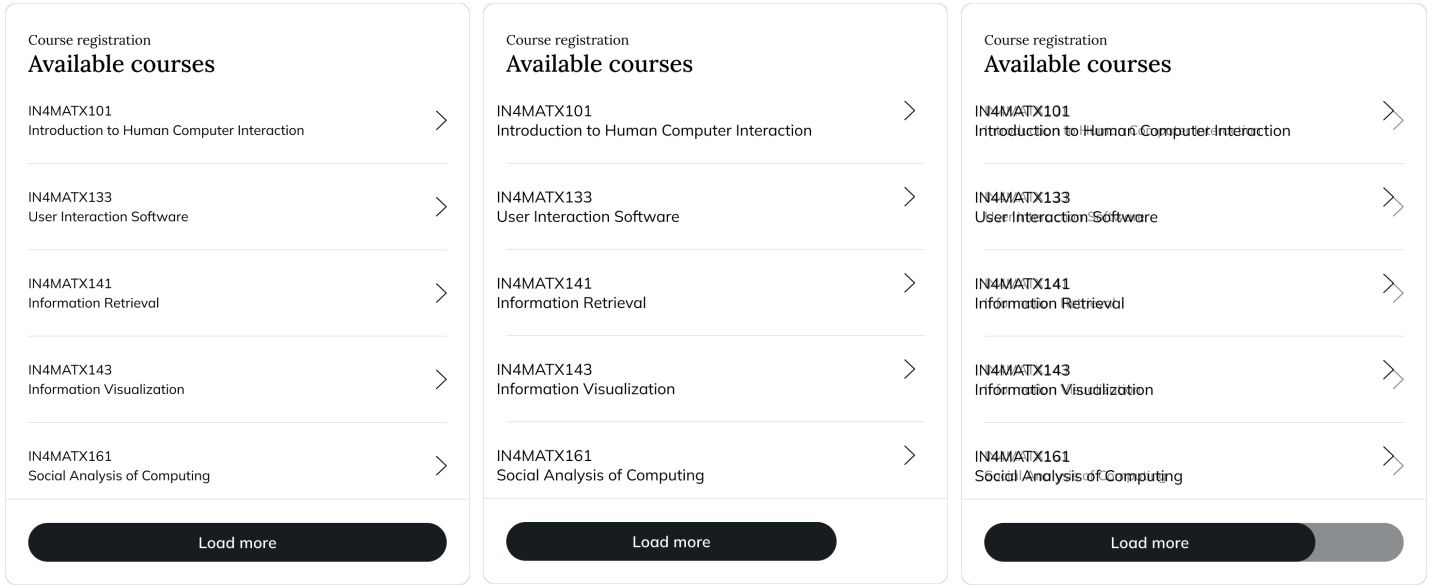
Insert yourself into the formal QA process (or if there isn't one, create one!)

You'll look for, and notice, different things than a developer or QA engineer

- Your focus on the user means you'll naturally identify defects related to experience
- A product can pass *functional tests* but still be incorrect

It's really hard for many developers to *see* small differences in design

Overlay what you made, and what they made, to help them notice differences:

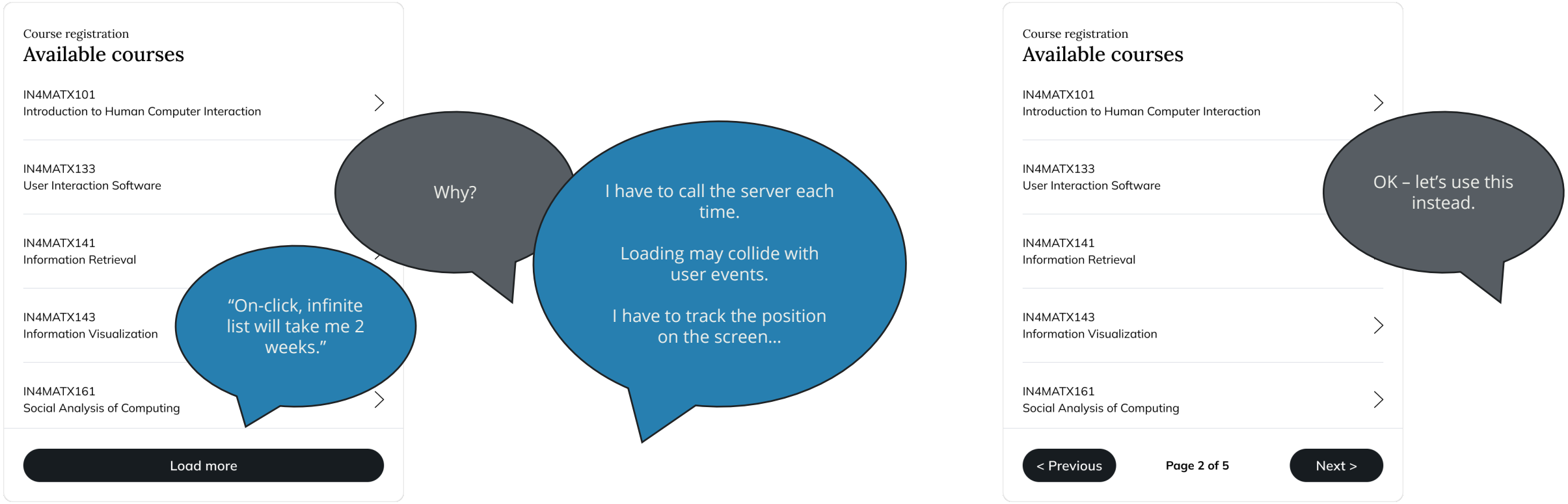


Supporting your design during development

Ask, learn, and when necessary, compromise (selectively).

Supporting your design during development

Ask, learn, and when necessary, compromise (selectively).



When development is completed

Continually monitoring your work in production

Continually monitoring your work in production

Look at the shipping product regularly, in order to find and track *regressions*.

A regression is when a shipping feature that used to work no longer works.

- A developer isn't aware of the relationship between their feature and another feature
- "Knowing the whole" is difficult or impossible
- Testing was poor or incomplete

Design is often in the best place to see regressions, because they almost always impact the user experience.

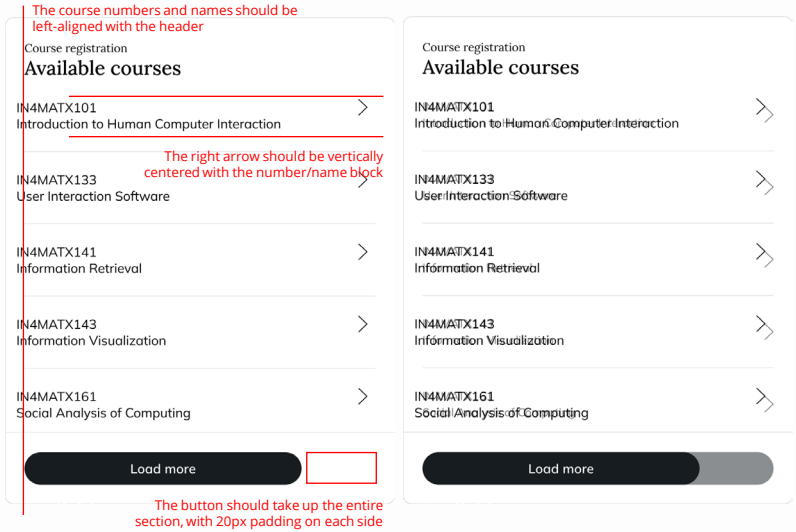
Continually monitoring your work in production

Log defects as you find them, and communicate your findings to developers and product managers. For example...

Description

On the Available Courses page, the alignment of text, selector arrows, and action button are incorrect

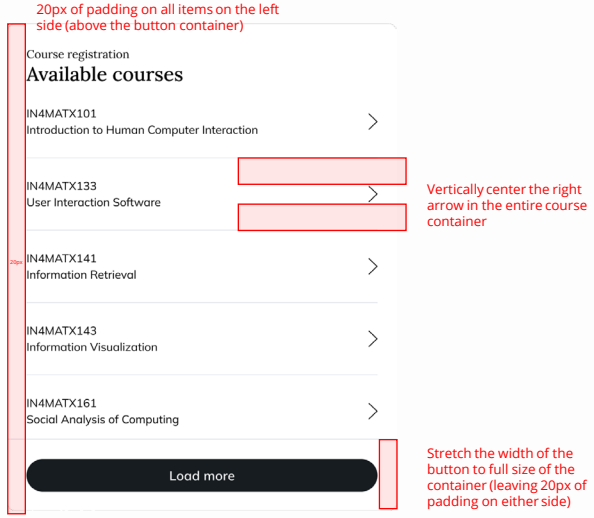
Visual example



How to reproduce the problem

1. Log in as a student user
2. Visit the Course Registration page
3. Click on “Select Available Courses”
4. Observe that:
   - The course labels and names are incorrectly position on the left
   - The right arrow is incorrectly positioned
   - The load more button is incorrectly positioned, and the incorrect size

Details for fixing the problem



Continually monitoring your work in production

Be an advocate for your backlog items.

New features are often prioritized over regressions and design flaws

- New features are exciting
- New features feel more timely and time-sensitive
- Backlog items often live in the backlog forever
- It’s easy to forget about things that were incomplete or never fully realized

A squeaky wheel…

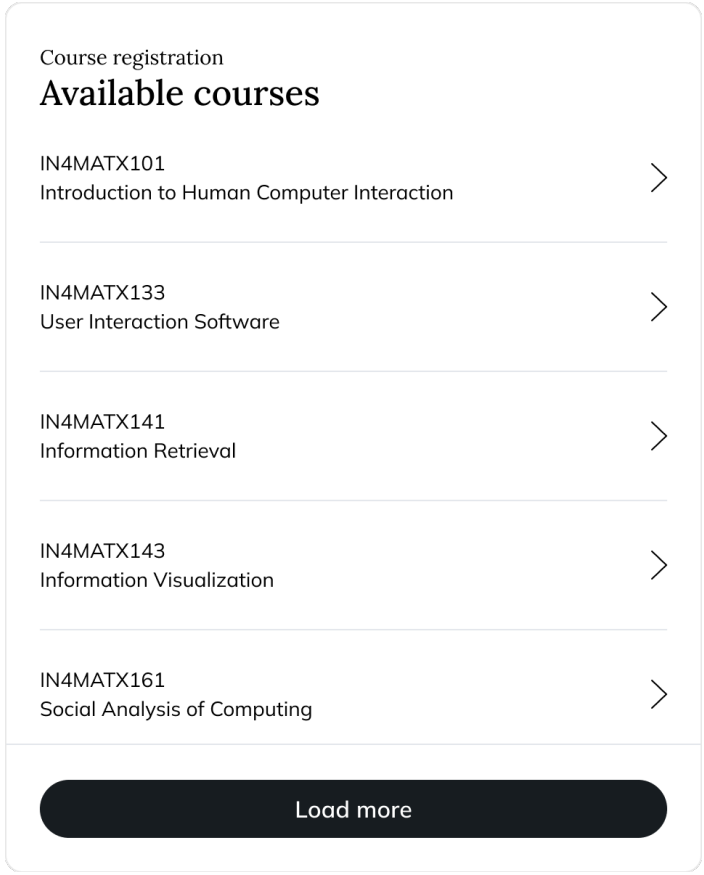
- Remind the team of the importance of the backlog
- Voice your concerns (but try not to be annoying)

Changing your culture to understand and respect design

The biggest design complaint...

"Why didn't they use what I designed?"

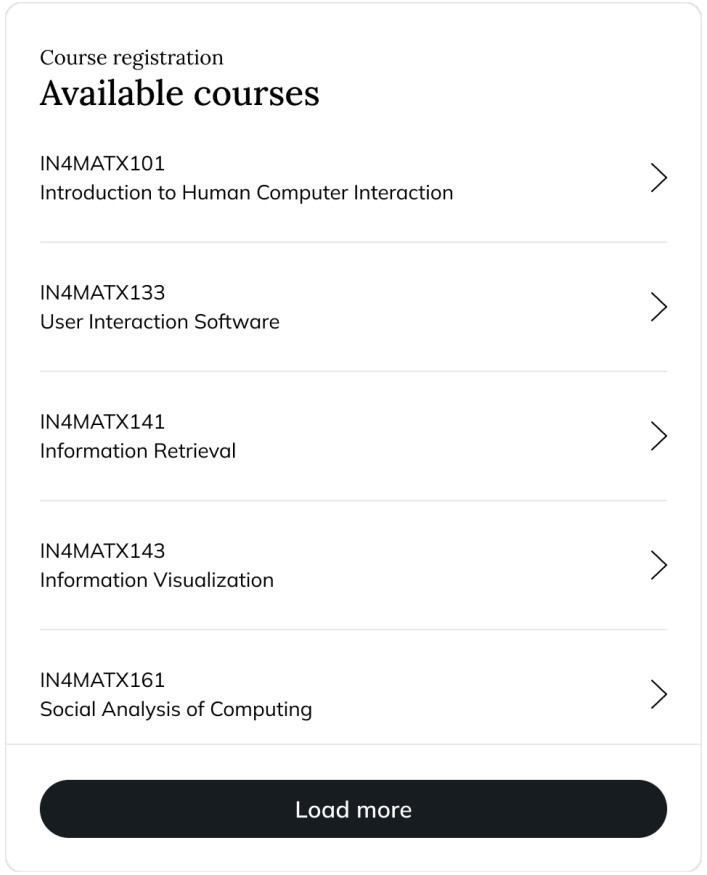
I made this



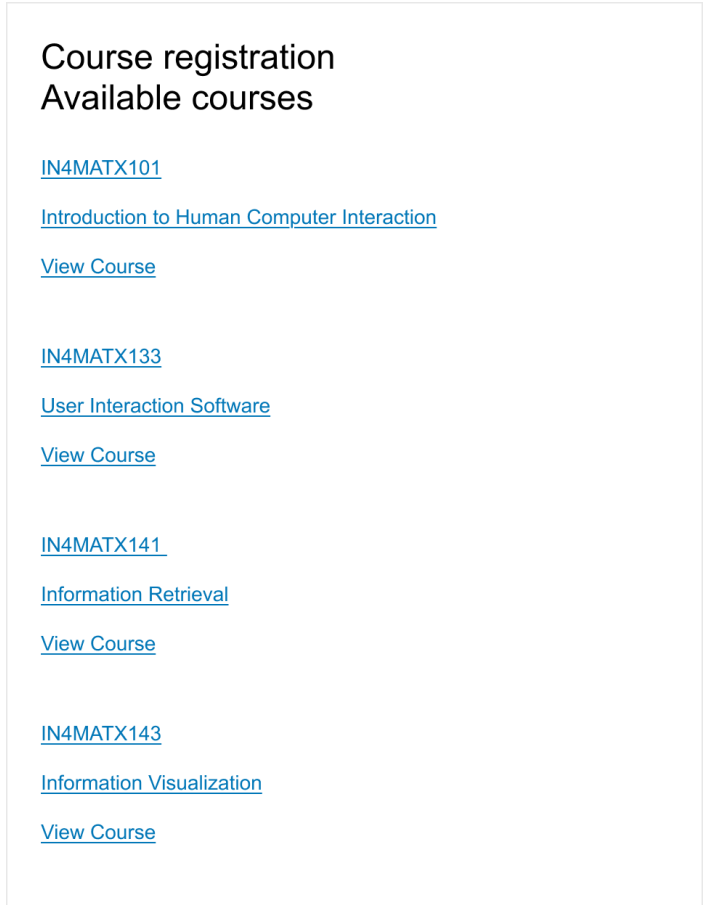
The biggest design complaint...

"Why didn't they use what I designed?"

I made this



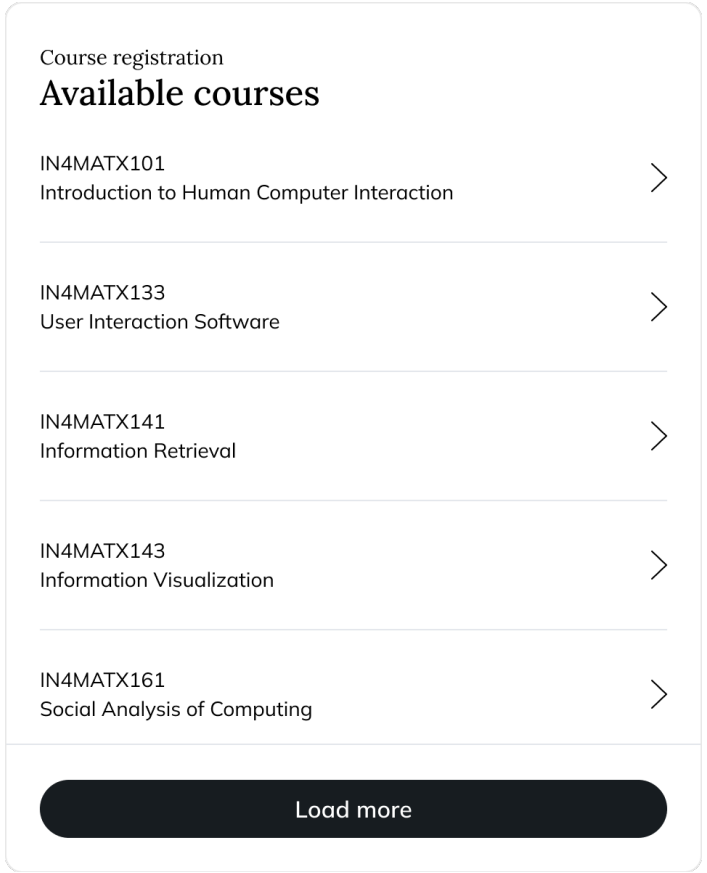
They made that



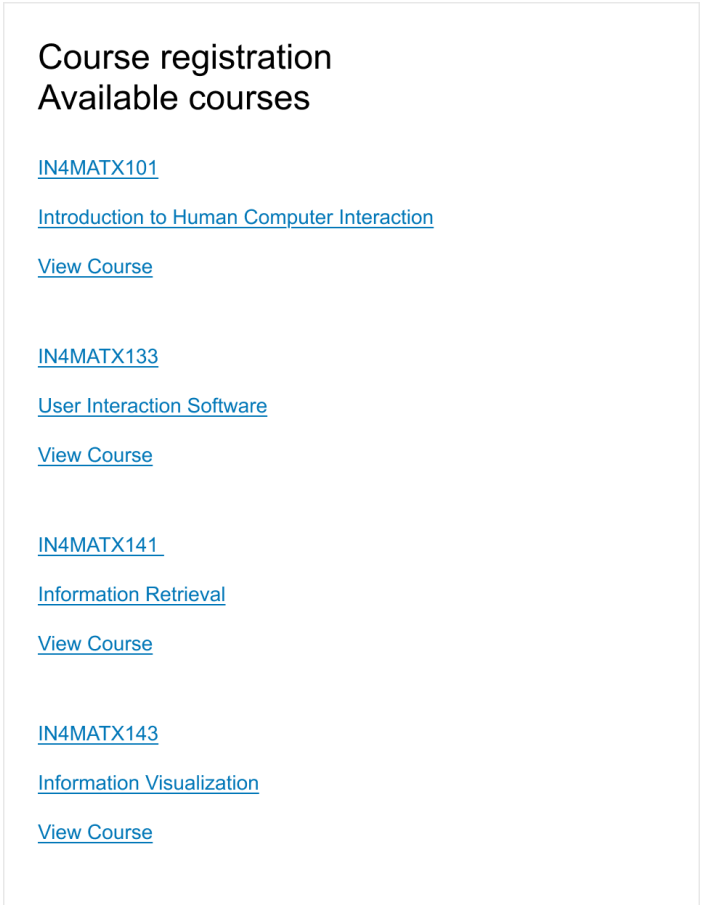
The biggest design complaint...

Most likely: you didn't advocate for the design.

I made this



They made that



What happened?

- You are thinking in terms of "I' and "They" instead of "We"
- You didn't explain the design
- You didn't explain the rationale for the design
- You didn't ask questions about technology constraints
- The design was too hard to build
- The design would take too long to build
- You didn't document your work
- Your documentation was too hard to understand
- You thought you were done
- You waited too long to see what was being made
- You aren't responsive
- You won't compromise
- You aren't respected

Changing your culture to understand and respect design

Do they respect you, your work, and your leadership?

Do you respect them, their work, and their leadership?

Changing your culture to understand and respect design

Push, relentlessly, for a culture of interaction design, visual design, accessibility, localization, content strategy...

Be charismatic, kind, thoughtful

The best way to gain respect is by making friends.

- Build up cultural capital
- Explain why design benefits everyone
- Be consistent in your positivity
- Be consistent in your drumbeat

Be aggressive, forceful, direct

Occasionally, you'll need to break glass.

- You only get to do this a few times
- You'll have to already have built up respect
- You'll need an existing reputation of success

Thank you!

jkolko@wonderfulnarrative.com