PROJECT 2

vocabulary flashcard app prototype

OBJECTIVES

RUBRICS

REQUIRED TERMS

STEPS

- RESEARCH + LO-FI WIREFRAME
- DESIGN SYSTEM + MED-FI WIREFRAME
- NAVIGATION MAP + FINAL WIREFRAME(S)
- APP PROTOTYPE BUILD
- MOCK-UPS

BEHANCE

GOOGLE CALENDAR

introduction

"Double-sided" flashcards are always a great way to learn vocabulary terms.

DIY cards:

- Buy a stack of 3x5-inch index cards. Write a term on the front of a card. Flip it over and write the definition on the back of the card. Then, you put one card down and reach for the next. Repeat. :)

Professionally printed cards:

- Same as above, but they are professionally printed on a nice, undestructible card stock. Depending on the audience (and the printing budget), the design system can either be simple grayscale on white paper or much more extravagent –even to the point of being collectible, beautiful pieces. The the cards also come in a package, oftentimes a "tuck box" that has a label repeating the design system on each card.
- See more here.

Free/paid apps online:

- There are plenty of free online, many already stocked with crowd-sourced information about different subjects. Quizlet, Brainscape, and Cram are great examples. There are not many free apps available that allow the user to create their own flashcards and also have at least some sense of design. This Google app is one of the few examples.
- Other apps, usually not free, are pre-stocked with vocabulary terms.

WHAT YOU WILL BE CREATING >>

Professional Flashcard App (Prototype):

- Reflecting the care that professionally printed cards would display while still keeping in mind the siimplicity and consistency of a design system, you will create a vocabulary flashcard app prototype of basic html/css terms. Due to the subject matter, no images are expected - only perfect typography. Color and pattern can also be explored, taking care to not override the learning experience with distracting elements.
- See this link for my Yoga Sanskrit Flashcards.
- See this link for my (very simple) start to html/css flashcards.
  Yours will require more/different terms and more advanced design choices.

MENU

objectives

This project reflects a tool that is traditionally found in printed form to the digital interaction of an app prototype for a phone. It also covers the following concepts whose graphic design foundations genesis has evolved to UI/UX interactive non-linear platforms.

- **Design Process** First Steps >> Research. Research more. Brainstorm. Mindmap. Sketch. Repeat.
- **Color**, **Contrast**, and **Content** – The 3 C's of Interface Design – should be considered when designing the interface. **Continuity** is the 4th C, which is easily controlled by creating a Design System that is enforced throughout the entire piece.
- **Typography** is about space as much as shape; plus, screens affect reading in a different way than print because the former is pixels of light while the latter is dots of ink on paper. Though type always has a voice, it is often calm and easy to read (and code) on a screen; therefore, students should be careful with fonts that are overly detailed or difficult to read.
- **User Experience** will be targeted to the constraints of basic XD capabilities as students grow increasingly comfortable with the software. (Remember that even if you end up using Figma, the two work in a similar fashion.)
- **XD building elements** will reflect those students are familiar with in InDesign, specifically **Character Styles** and (the less familiar) Object Styles. Once those are set and assigned, they can be adjusted across the entire build quickly. Character Styles are called the same in XD; however, Object Styles would parallel "**Components**" in XD.
- Like building out any design series, once one flashcard is built in XD, duplicate the Artboard

As always, please remember:

- **Every skill takes practice.** If it were easy, you wouldn't need 4+ years to get a degree in it. After that, you will continue to learn – and relearn – throughout your career.
  Never stop learning.
- Likewise, **the only way to learn software is to use it again and again and again.** There isn't any magic trick. It just takes time.
- If a student doesn't understand how to do something, **professional communication** with the professor is required. Fellow students with more experience may also have suggestions.
- You must understand how to **organize your time** and **prioritize what is important**.

MENU

rubrics

As explained in the syllabus, grade sheets will be returned with rubrics and brief comments. By each rubric point, CR, 1/2 CR, or No CR will be listed. The numerical equivalent would be:
CR = **100%** | 1/2 CR = **50%** | No CR = **0%**

OVERALL:

- **Directions were followed accurately.**
- **Participation** in all class and/or one-on-one discussions.
- Time management. **All work turned in on time or before due date(s).**
- Ambition. The overall evolution of work shows commitment and creative problem-solving.
- No spelling mistakes.

RESEARCH + LO-FI WIREFRAME:

- Flashcard design research provided on Slack as directed.
- Low-fidelity wireframe and other sketches show significant critical thinking.
- Ideas presented are creative in relation to research and project goals.
- Minimum of 10 card "front" and its "back" created.

DESIGN SYSTEM + MED-FI WIREFRAME:

- Design system created in Illustrator.
- Minimum of 1 card "front" and its "back created in either Illustrator or XD.
- Design system font choice(s) make sense aesthetically per UI/UX needs.
- Design system color choices make sense aesthetically per UI/UX needs.
- Design system UI elements make sense aesthetically per UI/UX needs.
- Medium-fidelity wireframe displays design system.

FINAL DESIGN + NAVIGATION MAP:

- Navigation Map displays awareness of its purpose.
- Navigation Map is legible and turned in on time as asked.
- All elements in final design show awareness of the 4 C's of Interface Design:
  Color | Contrast | Content | Consistency
- Composition and Layout take into account shape vs. space within the artboard.
- Character Styles assigned in XD.

- Components created and assigned in XD.
- At least one button is created with multiple states.

APP PROTOTYPE BUILD

- **Directions were followed accurately.**
  This includes understanding how one publishes once, then continues to update the link.
  In other words, only one URL should be needed.
- All designs use Character Styles and Components correctly.
- Prototype wires used correctly to create proper interaction and flow.
- Prototype is shared correctly.
- Local XD file turned into Google Drive PROJECT 2 matches XD URL updated link.

MOCK-UPS

PHOTOSHOP

- Directions were followed accurately.
- Mock-Up displays work correctly.

QUICKTIME

- Directions were followed accurately.
- Recording displays prototype interation correctly.

BEHANCE:

- **Directions were followed accurately.**
- No spelling or grammar mistakes.
- Work displays significant, unrushed efforts of critical thinking and self-reflection.

MENU

required terms

There are **35 required terms**, so 35 required flashcards, each with its own front and back. This is easy to accomplish once the design is built and then copied on every artboard, so do not let this overwhelm you. To truly understand these, see the examples provided at the following sources. One may use these samples (re-create them in your style) if wished, as it would be an easy addition and make these definitions much more clear!

- https://www.codecademy.com/article/glossary-html
- https://www.codecademy.com/article/glossary-css
- https://physics.weber.edu/schroeder/html5/vocabulary.html

The following is available in a Google Doc at this link. All of these are required for flashcards. The term is the "front side" and the definition is the "back side."

HTML TERMS

HTML - HTML stands for Hyper Text Markup Language. It is the language used to define the logical structure of all web pages, breaking them down into various elements using tags.

attribute - A property or characteristic of an HTML element, such as id or src or style or onclick, that is specified inside its opening tag using the syntax attributename="value". Attribute values should be enclosed in quotes (though browsers don't always enforce this rule), and multiple attribute assignments are separated by spaces. The order in which multiple attributes are specified doesn't matter.

basic formatting - You can easily format text to be bold, italic, or underlined using simple formatting tags.

body - The body is the container for all of a page's content. Comes after the <head> tag, within the overall <html> tag. Almost all content belongs inside the body tag. The main exceptions are script and style tags, as well as the page title tag.

class - HTML elements can have one or more classes, separated by spaces. You can style elements using CSS by selecting them with their classes.

children - An element that is an immediate descendent of another element or nested within another element is called a child. These become useful when using CSS child selectors and psuedo-elements.

comments - HTML comments are sometimes used in code to explain parts of the markup. They are similar to comments in other languages. Users do not see comments in their browser.

content - Everything between the opening and closing tags of an HTML element. The content can consist of text and/or nested HTML elements.

closing tag - A unit of HTML code that marks the end of an element, such as </h2>, </p>, and . Each closing tag is the same as the corresponding opening tag, except for an additional slash after the first angle bracket and the omission of any attributes.

div - A block level container (or 'division' of the web page) for content with no semantic meaning.

head - Tag that surrounds important content that is invisible to the user, but is important to the browser. Elements within this tag contain metadata about the page and links to stylesheets, scripts, etc.

headings - Heading elements like <h1>, <h2>, <h3>, etc. allow you to use six levels of document headings, ranging from largest to smallest, breaking up the document into logical sections. For example, the word 'Headings' above is wrapped in a <h2> tag.

horizontal rules - This tag creates a black line one pixel thick that runs the all the way across its container. It can be styled to look differently with CSS.

href - Links tell the browser where to go using an href attribute, which stores a URL.

<html> tag - All HTML files live within an over-arching html tag. This is the basic tag that defines an html document.

hyperlinks - Hyperlinks (or just links) take the user to another webpage when they click on it. The most common attribute used with links is href, which tells the browser where the link goes.

id - An HTML element can have an id attribute to identify it. id elements should always be unique to that single element, and each element should never have more than one id.

images - The img tag embeds an image into your HTML. Always found with the 'src' attribute, which tells the browser where to find the image. Note that the <img/> tag is self-closing.

line breaks - This tag is used in a block of text to force a line break. This is to be used for things which are a single paragraph, but where this formatting is necessary such as poems or addresses. To separate paragraphs, separate each paragraph into a separate element instead.

links - Link elements are used to connect your document to a related resource (very different from hyperlinks, which take you to another webpage when you click on them). Links appear only in the

head section of a document so they do not alter the content, but only the presentation. Links are most commonly used to connect to a stylesheet, script, favicon, or alternate format of the page such as an RSS feed or PDF.

lists - HTML supports two kinds of lists: ordered lists and unordered lists. Within lists each individual list item has its own tag. Ordered lists' items are denoted with numbers. Unordered lists are just lists whose items are denoted with bullet points.

opening tag - A unit of HTML code, enclosed in angle brackets, that marks the beginning of an element. Examples include <h2>, <p>, and <a>. Some opening tags, such as <img> and <br>, are self-contained elements that have no content or closing tag.

paragraphs - <p> One of the most common tags in HTML - it denotes a paragraph of text. It often has other elements nested inside of it, such as <img/>, <a>, and .

property - Any characteristic of the appearance or placement of an HTML element that can be specified using CSS, such as color or font-family or margin. The syntax for specifying a property is property-name:value; (note the colon before the value, the semicolon after it, and the absence of quotes). The order in which multiple properties are specified doesn't matter.

semantic formatting - These tags are similar to the previously mentioned formatting tags which have fallen out of favor. The difference is that these tags have semantic value (meaning). is used for something that you wish to emphasize and is used for something that is important. With both of these elements, you can convey the level of emphasis or importance with nesting. The more times that you nest the element within itself, the higher the magnitude of the text it contains.

tables - An element for displaying information in rows and columns. Supports headers and footers for labeling columns. Divides information into rows (denoted by the tr tag) which contain cells (denoted by the td tag).

tags & elements - TTags tend not to do much more than mark the beginning and end of an element. Elements are the bits that make up web pages. You would say, for example, that everything that is in between (and includes) the <body> and </body> TAGs is the body ELEMENT. As another example, whereas "<title>" and "</title>" are TAGS, "<title>Rumple Stiltskin</title>" is a title ELEMENT.

title - This tag tells the browser what to display as the page title at the top and tells search engines what the title of your site is. It goes inside <head> tags. Try and make your page titles descriptive, but not overly verbose.

CSS TERMS

CSS - Cascading Style Sheets, the language used to specify the appearance (e.g., fonts, colors, borders, and placement) of HTML elements. CSS code can be placed in an external .css file or inside the special HTML <style> element, but for simple web pages it is easiest to assign it using the style attribute.

comments - Comments in CSS are signified by a forward-slash and asterisk.

properties - Properties are defined within selectors by defining a property and a value. They are separated with a colon and delineated with a semi-colon. Each CSS rule can have as many properties as you like. Each of them applies to the elements that the selector applies to.

padding - The padding is the spacing between the content and the border (edge of the element.). We can adjust this value with CSS to move the border closer to or farther from the content. Here, the div with id 'box' will get 10px of padding all around it.

margin - The margin is the space around the element. The larger the margin, the more space between our element and the elements around it. We can adjust the margin to move our HTML elements closer to or farther from each other. Here, the div with id 'box' will get 10px of margin above and below it, and 5px of margin to the left and right.

font-family - The font-family property sets the font of an HTML element's text.

selectors - Selectors are used in CSS to select the parts of the HTML that are being styled. You can use several different methods for selecting an element. These include: Class Name Selectors, Element Selectors, ID Selectors, Attribute Selectors, Child Selectors, Universal Selector, and Pseudo Class Selectors.

MENU

STEP 1 research +lo-fi wireframe

Evidence of all of the following may be required on the final Behance page, so **do not lose it**. Another suggestion is to take a clean, clear picture of work with your phone to easily access it later.

RESEARCH

Professor will place samples of flashcard designs in a thread on Slack's #1413_proj-2 channel. These can be print or apps. The point is to observe modular elements with a consistent design system, plus consideration of helpful elements to a user of that application.

- Each student will post examples of their favorites in the same thread. Directions provided in class and on Slack.
- Students must also **review all of the required terms and their definitions**. Fully understand the comprehensive design of the system and how to best convey the information to the user.

DUE > WHAT • WHEN • WHERE

DUE for discussion in class **11:00am** on **Tues Feb 13**.

- Significant evidence of research. This includes required posts and explanations in thread on Slack #2803_proj-2.

With this research, student will proceed to next steps in class on Tues Feb 13. >>

BASIC IDEA(S) + LO-FIDELITY WIREFRAME

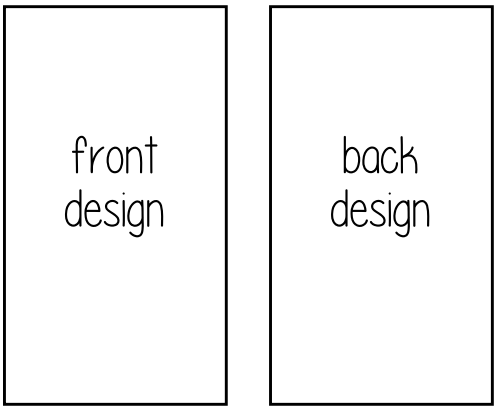
Just like sketching ideas for a layout for print, wireframing in UI/UX Design is one of the most crucial steps which involves visualizing the skeleton of digital applications. A wireframe is a layout of a product that demonstrates what interface elements will exist on key pages. It is a critical part of the interaction design process.

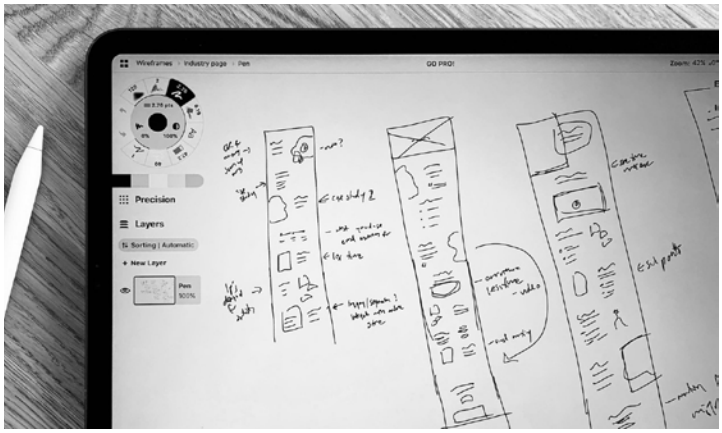
Unlike a website's need for resposonsive design, this app prototype is built to display only on your smartphone. The "back" of the flashcard can either be hidden/viewed via:

- Building an artboard longer than the phone screen, scrolling down on the phone to see the second side. (This is much easier in XD and suggested.)
- OR can be built in XD to flip on the phone.

While considering, start sketching up a rough wireframe like the one shown below, considering both the front and back of the card, (usually) regardless of how it will show on the screen. This can be done on paper, Procreate/Fresco, or Illustrator.

Include at least 10 sketches, concentrating on the design via shape vs. space, typography, color, and texture. Illustrations can be considered, but would only be picked if they significantly added to the user experience and objective of the flashcards.





DUE > WHAT • WHEN • WHERE

DUE for discussion in class **11:00am** on **Thurs Feb 15**.

MENU

STEP 2 design system + med-fi wireframe

DESIGN SYSTEM

Like all design that extends beyond just one "thing" (page, screen, etc.), student should decide on elements for a solid and consistent visual system and hierarchy to use throughout. It is suggested that each student create a new Artboard and resize it to a more traditional page size than a flashcard.

Design system should be obvious on Medium-Fidelity Wireframe. It is optional to actually create an artboard with the system, but it will make a more informative Behance post.

These include:

- **Font(s)** to use for headings, subheads, body text, etc. Instead of using many fonts, try to differentiate the same font with such parameters as size, weight and color. Google Fonts provide a great collection of choices, though not all will work well.
- **Color Palette**: There are many apps online to help you with this. Know the basics of color psychology and choose the color palette that will be suitable for a certain kind of a product/industry.
- **General Shape/Line**: Will you use straight edges or curves on everything? Be consistent.
- **Buttons/Arrows:** How will you guide the user through the app? These might be how you show the previous design system requirement.
- **More to Consider:** Do you have drop shadows on elements or definitely not? Will you use solid flat colors or will you attempt gradients? Do you want patterns on the edges or elsewhere that bring the piece together without impeding the read?

MEDIUM-FIDELITY WIREFRAME

After discussion with professor, student should proceed to refining their idea. Building out a medium-fidelity (tighter, but not final) wireframe should correspond to the development of the design system. Use **Illustrator or XD** to create (at least) one version "front" and "back" to be approved by professor before building the entire set, which obviously would express the design system created.

Figure out what size to build ideas to by first checking size possibilities on XD that most closely respond to your own phone.

If working in **XD**:

- See requirements for **design system** on resized artboard that will be ignored when doing the prototype OR use Illustrator to create your design syste.
- Save your file as **YOUR LAST NAME_2803-Proj2**.
- Character Styles and Components do not need to be assigned yet, but can be.

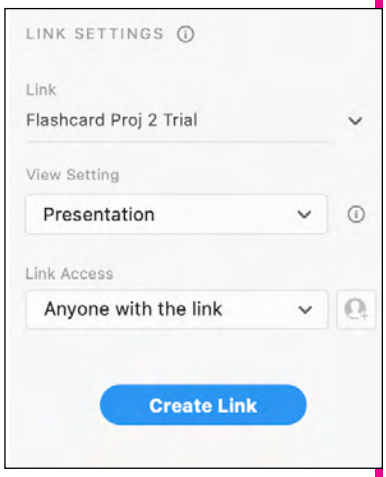
If working in **Illustrator**:

- Start a new file for Web that reflects the pixel width and height.
  - If creating a flashcard that scrolls down, resize your artboard to a longer document, but be aware of the original screen's view.
- See requirements for **design system** on resized artboard.
- Name the file **YOUR LAST NAME_2803-Proj2_Preliminary**.ai.

DUE > WHAT • WHEN • WHERE

DUE for discussion in class **11:00am** on **Tues Feb 20**.

- ***If on Illustrator:***
  YOUR LAST NAME_2803-Proj1_Preliminary.ai with all required elements in Google Drive PROJECT 2 sub-folder.
- ***If on XD:***
  Do not create a local file. Share as shown here. Then paste the **URL** into a document that will provide a live link, like an interactive PDF.



- ***If on Illustrator:***
  File: Export as a **PNG** and post to thread in Slack **#2803_proj-2** requesting it.
- **If on XD:**
  Share the URL on thread requesting it on Slack **#2803_proj-2**.

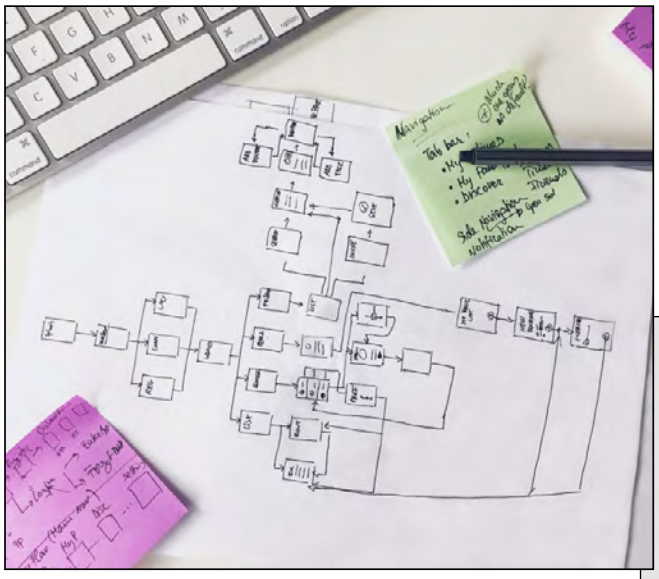
MENU STEP 3

navigation map + final wireframe

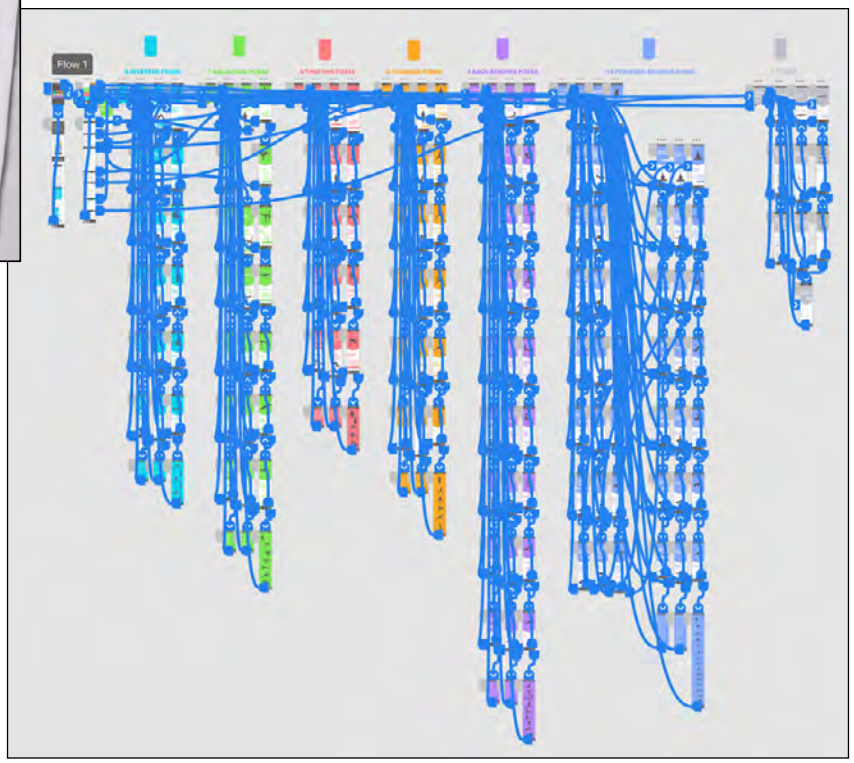
NAVIGATION MAP

In Illustrator or Procreate, student should create a basic Navigation Map. Save it whatever name you would like. It is due for discussion with Prof. Nikki and will be shown in the final Behance post.

- Consider a home page and an index too.
- You may organize the cards in a way that makes sense or you can just have them all in one “stack.”
- Consider how you can make it flow well, understanding that the prototype in XD will not randomize flashcard order. For example, once the main build is completed in XD, it would be very easy. to duplicate boards to reverse what is on the “front” and the “back”.



Below is shown the level of map needed, though a tighter version will show better on Behance. The image below shows wires in XD Prototype mode, which is essentially the realization of a navigation map.



FINAL WIREFRAME(S)

Student should revise anything needed on flashcard front and back to become final art.
It is also highly suggested, but not requred, that student give a try at creating a home page for the app that is similar to what would be on the package that would hold print flashcards.

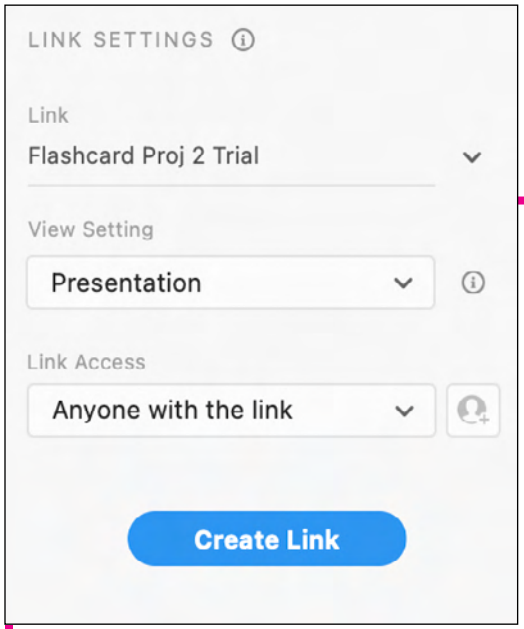
If working in **XD** Design mode

- Assign all Character Styles and Components in Document Assets panel.
- Build out any buttons that will change with rollover or toggle.

If working in **Illustrator**:

- Create clean artwork for your final pick. THEN >
  > Work in **XD** Design mode:
  - Paste or Import in XD. Save your file as **YOUR LAST NAME_2803-Proj2.**
  - All elements pasted into XD will be grouped. Keep clicking to select grouped elements.
  - Assign all Character Styles and Components in Document Assets panel.
  - Build out any buttons that will change with rollover or toggle.

To turn in XD file, go to **SHARE** mode. Do as shown here.



After this link is created, remember that any work you do will require you to **Update Link**.

DUE > WHAT • WHEN • WHERE

DUE for discussion in class **11:00am** on **Thurs Feb 22**:

Navigation Map:

Print to discuss with professor.

XD File:

Do not create a local file from XD. Share as shown here. Then paste the **URL** into a document that will provide a live link, like an interactive PDF. Upload to the **PROJECT 2** sub-folder:

Share the URL on thread requesting it on Slack **#2803_proj-2**.

STEP 5 app prototype build

Contine working with the previously created **YOUR LAST NAME_2803-Proj2** XD file that is in the cloud. DO NOT work with a Local Document.

Guidance will be provided in class; however, student is expected to learn by doing and Googling. Professor will be around to help answer questions. **Remember to save early, save often**, underestanding that you aren't working with a Local File yet.

IF BUILDING FLIP CARD IN XD

If student chooses to do the flipping card, he/she/they must **design and prototype FIRST** before going any further. *If student just wants to TRY this on an artboard, credit can still be given even if the prototype ends up not using this artboard in the prototype.*

The following directions assume student is choosing the scroll method to see the "back" side of the flashcard:

DESIGN

Build out the rest of the flashcards. Rememer the easy way to quickly duplicate an existing artboard is to select the top left name of the artboard. Then click Option + Command + Shift and drag. Command + D will also quickly duplicate, but then the artboard may need to be repositioned.

Make your life as easy as possible by working with **Components** and **Character Styles**! Remember if building buttons with hover and toggle stages, you will use Design and Protoype mode to complete this. Then when the button itself becomes a multi-state Component, it is easy to use repeatedly.

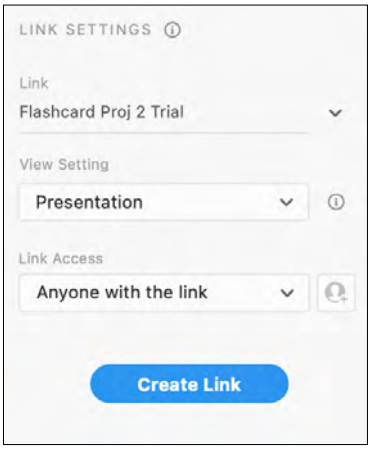
PROTOTYPE

This should be an easy construction with wires that imitate your navigation map. Remember that randomizing flashcard order is not expected. This should be a linear read like going through a deck of cards. Card groups relating to an index page will require additional wires, but these are also very easy to organize.

SHARE

When ready to share, go to **SHARE** mode.

- There should be an option to Update Link, assuming file was shared correctly before. If this is available, click this and then check the URL shared previously on Slack.



- If the only option available to Create Link, then do as shown here. Paste the new URL in where it was asked for on Slack previously.

DUE > WHAT • WHEN • WHERE

DUE for discussion in class any day before final due date. Check Slack for new requirements.
Share the URL on thread requesting it on Slack **#2803_proj-2**.

Final work **DUE 11:00am** on **Thurs Mar 7**.

XD File:

Upload both of the following to the PROJECT 2 sub-folder:

- Share as shown above.. Then paste the **URL** into a document that will provide a live link, like an interactive PDF.
- Save as a local XD document (named **YOUR LAST NAME_2803-Proj2**.xd). Upload t

Share the URL on thread requesting it on Slack **#2803_proj-2**.

MENU

STEP 6 mock-ups

This last step will be completed together in class on Thurs Mar 7.

PHOTOSHOP

- A layered template PSD file will be provided in class.
  File : Save As **YOUR LAST NAME_2803-Proj2_MockUp**.psd
- Student should build the mock-up to best represent their work. Keep the PSD layered file.
- **File : Export : Quick Export as PNG**. Turn this PNG in as explained below.

QUICKTIME

- In class, students will be shown how to use QuickTime, which is native on all Macs, to complete a Screen Recording of a cropped space of them using the XD prototype.
- Adobe Media Encoder may also be shared to explore how to resize massive video files to something more manageable.

DUE > WHAT • WHEN • WHERE

All final work **DUE 11:00am** on **Tues Mar 12**, including Behance post.

YOUR LAST NAME_2803-Proj2_MockUp.png in Google Drive PROJECT 1 sub-folder.

YOUR LAST NAME_2803-Proj2_MockUp.png posted to thread in Slack **#2803_proj-2** requesting it.

QuickTime movie should be shown in Behance. TBD if additional way to turn in this file.

MENU

behance

Save all steps of this project, and be ready to write a brief explanation of your design choices and self-evaluation. Your Behance page will be updated with each project.

» **It is highly suggested you write all of this out in a document that can check your grammar and spelling first** (Word, Pages, Google Docs, etc.). Then copy : paste your text and insert images in one sitting on Behance. This way you also avoid the internet suddenly crashing midway and losing information that wasn't yet cached.

- Remember that you are **telling a story to the public**, not just the professor who knows what you did and why you did it.

» When creating the post, remember this should also display comprehension of design. **You must use visual hierarchy** by making sure sections, like **SUMMARY**, are clearly defined via type weight (and/or other techniques) so the information's "chunks" are clearly defined.

Below are specific directions for this project's Behance artist statement page:

- **SUMMARY:** Minimum of one paragraph summarizing the project and its objectives. Do not copy the project sheet. Instead, explain it like you would explain to a friend.
- **GOALS:** What did you want to learn with this project AFTER you read the preliminary (or full) project sheet. Aside from "my goal was to get an A," what did you hope to LEARN for your future use as a designer?
- **FLASHCARD DESIGN:** In the following story of your design process, you are explaining how you came up with the overall design vs. the entire deck; therefore, concentrate on only one "front" and "back" side of a card.
  - » **LO-FI WIREFRAME:**
    - » Briefly **EXPLAIN** in a minimum 1 paragraph your starting point. You might reference earlier research here and any notes you made while considering your design choices. Show your thinking!
    - » *Optional to **SHOW** image(s)/screenshot(s) of research, especially if it inspired your lo-fi wireframe.*
    - » **SHOW** image(s)/screenshot(s) of your low-fi wireframe.
  - » **MED-FI WIREFRAME AND DESIGN SYSTEM:**
    Briefly **EXPLAIN** in a minimum 1 paragraph how you refined your design to this level.

    - » **SHOW** image(s)/screenshot(s) of your med-fi wireframe.
    - » **SHOW** image(s)/screenshot(s) of your design system.
  - » **FINAL DESIGN:**

    Briefly **EXPLAIN** in a minimum 1 paragraph how you got to the final overall design of the flashcard's "front" and back."
    - » **SHOW** image(s)/screenshot(s)
- **NAVIGATION MAP:** Minimum of one paragraph explaining your choices for how the user will move through the flashcards in the prototype. If you have greater plans for the interaction experience than what will be able to be shown in XD, explain it here. Show your thinking! :)
- **APP PROTOTYPE BUILD:** Minimum of one paragraph each bullet point below, explaining your steps and design choices. **Explain your progress**. Use screenshots. Tell a story. :)
  - **DESIGN:**

    Briefly **EXPLAIN** your process of building out the app using your assign **Character Styles** and **Components.**
    - » **SHOW** screenshot(s) of as much as you can, showing how much work you put into it by taking these shots THROUGHOUT the process. Also, remember the more you show, the less you have to verbally explain.
  - **PROTOTYPE:**

    Briefly **EXPLAIN** your process of building the app's interactions by pulling the wires and assinging transitions, etc.
    - » **SHOW** screenshot(s) of as much as you can, showing how much work you put into it by taking these shots THROUGHOUT the process. Also, remember the more you show, the less you have to verbally explain.
    - » When finished, go **Command + A to display all wires**. Take screenshot of this.
- **FINAL WORK:** Show what you can of the following, depending on the abilities of Behance.
  - » Mock-up made in Photoshop, obviously resaved to a JPG or PNG.
  - » Video of you using the app prototype.
  - » URL link from XD (under **SHARE**).
- **KNOWLEDGE GAINED:** Minimum of one paragraph explaining what you learned. Share the technical skills of the software used, design choices, UI and/or UX considerations, but also discuss any changes to your mindset or expectations of graphic design.

see next page>>>

Make sure you check your grammar and spelling using Grammarly, or similar. **Improper grammar and misspellings greatly reduce your project grade!**

DUE > WHAT • WHEN • WHERE

Project 2 Behance post **DUE by 11:00am** on **Tues Mar 12**.

Make sure your post is **published** on Behance. Then check the URL for your page on Slack channel **#2803_behance**. Your new project should be up! :)