CS-639 Building User Interfaces, Fall 2020, Professor Mutlu

Javascript β (2 Points)

Empathy-Driven Redesign Using Component Libraries

GitHub Classroom Starter Code for JavaScript β

In this assignment, you will practice design thinking and visual design methods to improve upon Badger Bank. You will complete this assignment in three parts.

In the **first** part, you will practice using the "empathy" method (the think-aloud protocol) that we have discussed in class as a method that will give you the most bang for the buck in terms of understanding user needs, preferences, and behavior. Specifically, in the think aloud, you will (1) identify users who represent your target group of users, (2) identify/develop tasks that represent the functioning of the target system, (3) observe users as they perform the tasks, and (4) analyze your data to develop design insight.

In the **second** part, you will practice the ideation and visual design principles you have learned in class to turn your design insight in the first part into a redesign of the Badger Bank website you have created, expressed in a layout sketch.

Finally, in the **third** part of the assignment, you will implement the design you created in the second part using the Bootstrap component library. This three part assignment will take you through user research to ideation and from sketching to implementation of user-facing elements. Follow the instructions below to complete the assignment.

Part 1: Think-aloud

(0.1 Points) **Step 1. Identify users.** Who uses online banking? Describe below the characteristics of this user group, identify one person (a roommate, a friend, or a family member) who might be willing to take part in your user research, and ask the person for their interest/availability. (If the person you identified is in this class, it is acceptable to swap roles.)

The user would be my friend. She is a 20 years old Chinese girl.

(0.1 Points) **Step 2. Develop tasks.** Study the new Badger Bank application, Javascript **β**, to develop a set of tasks that users might perform with the system. Rank your list in terms of importance (simultaneously considering impact, frequency, prevalence) and identify the top three tasks. Describe each task in 1–2 sentences in a way that your users can understand.

Note: As banking involves private and sensitive data, be sure that your tasks do not require your user to reveal private information (e.g., login credentials, bank balance) to you or anything else that may make them uncomfortable. Your tasks can focus on the general visual and click-through navigation of the site and interact with personal information in a very limited sense.

(Include more than three here but will only ask the top three of them)

1. Find the account numbers and balances of your checking account(s) and saving account(s)
2. Find the latest 5 transactions of your checking account(s) and saving account(s) and the details (date, description, amounts, balances after transactions)
3. Log out of your account
4. Download the tax for last year and account statements for last month.
5. Find the contact info of the bank
6. Find the beneficiaries and test to add a new one (not actually add one)
7. Check whether you enroll in paperless statements.

(0.2 Points) **Step 3. Perform think-aloud.** Hold a think-aloud session with your representative user (e.g., over Zoom using screen sharing) *on a real banking website*. First describe to your user how the think-aloud will work (refer to the reading and class notes), describe the tasks one at a time (answer any questions you might have), and ask them to perform each task while they say out loud what they are thinking.

> **Pro tip:** If you see your user performing the task but not speaking, probe them by asking what they are thinking or reminding them that they should be describing.

As you observe your user performing the tasks, take notes (using the other sheet) of important actions, problems they encounter, confusions they might voice, and so on. For anything that stands out, after each

task, ask your user why they did that or said that. Your observations and notes will form your data. Include your data below.

Note: If your tasks involve entering or reviewing personal information, you can ask your user to stop screen sharing while they perform these actions in order to preserve their privacy.

All of the tasks go well as the user is familiar with the system.
Something that I notice that can be improved from:

- Scrolling up and down would be burdensome.
- It would be slightly better to know the balance after each transaction was made to the account
- User do not want trivial information (such as some welcome text) to take large part of the web page
- User like clear separations between sections (e.g. between each account info) and want to see some contrast in color to show what is important. She also like light colors
- User does not like to have all the details about accounts directly shown on the webpage. For example, she would like to have the transactions and profile info hidden and click some buttons to show, and only display the overall balance of the account at first
- User want the text about overall balance of account be clearer to view
- Logging out in the user's account takes two steps (go to the user center and then log out), would be better if user could log out in the home page. User get used to have the log off button together with user profile.
- User gets used to have the contact info at the button
- User does not like info that are irrelevant to her account info such as welcoming texts

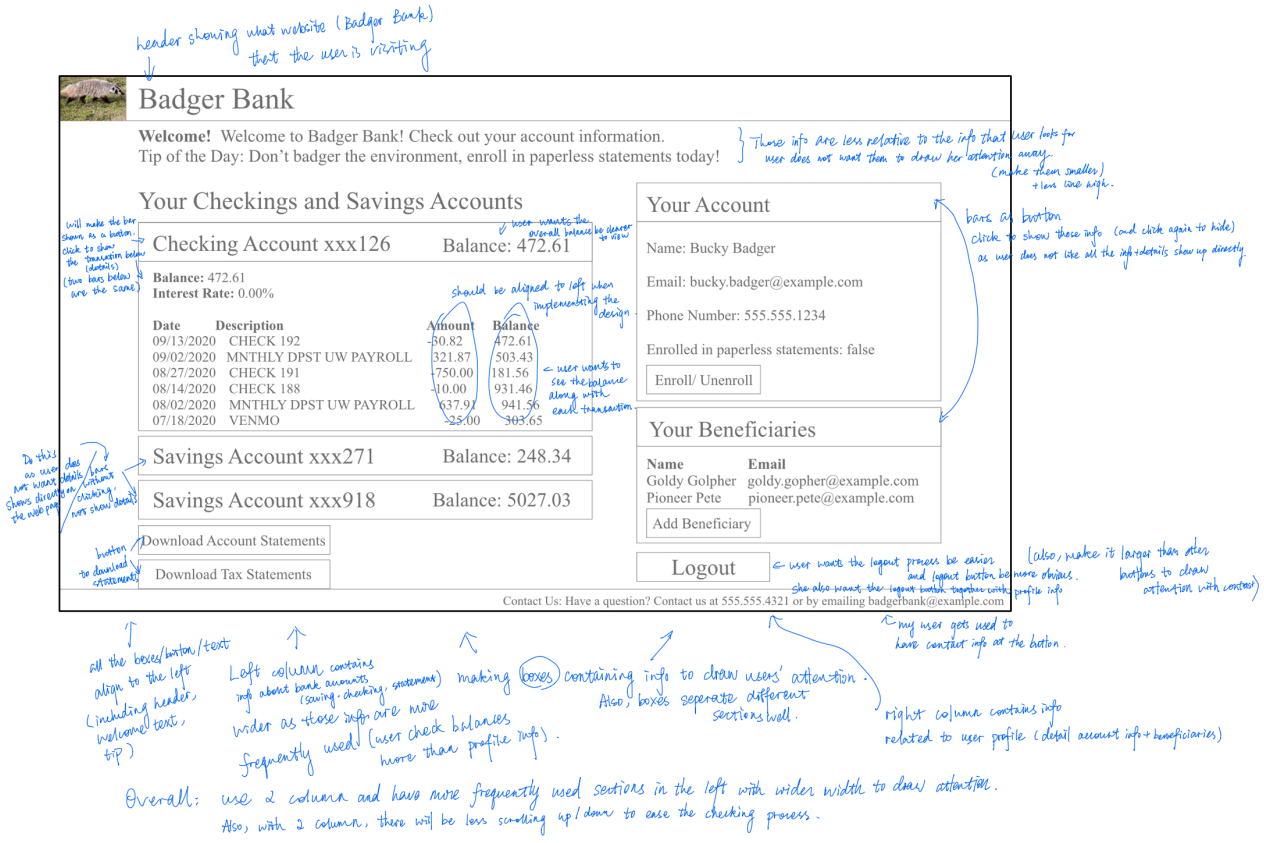
(0.2 Points) **Step 4. Create insight.** In your data (e.g., notes), highlight where you saw significant breakdowns in functioning, need for better functioning, or user preferences that would require an alternative design. Make a list of your findings as design recommendations.

- For the Welcome text and tips of the day and info about badger bank, we can make the size smaller and less highlighted. Maybe we could put the contact us info at the button (which is what the user get used to). Or maybe have a small header bar for the bank name and welcome message.
- We can have the "Badger Bank" with the photo in the header with a smaller size to tell the user where she is and also meets her requirement to put less highlight on irrelevant info than accounts.
- For the user profile, we can make the details to be shown only by clicking some button and have some icon to show that clicking the button will show user's info. We can also have the log out button next to the user profile but do not included in the user info so that the user can log out in one step.
- For the account info, we can have two large bars showing the ending number, type, and balance of each account and let user click on those bars to see the detailed transactions for each account.

- For the overall webpage, it might be better to have all the element be one the same page (reduce scrolling up and down) by having two columns in the webpage, let left column be wider with bank accounts, and put profile info in the right.
- For the overall color, light color are preferred but need some contrast between important info and others and between different sections.

Part 2: Design

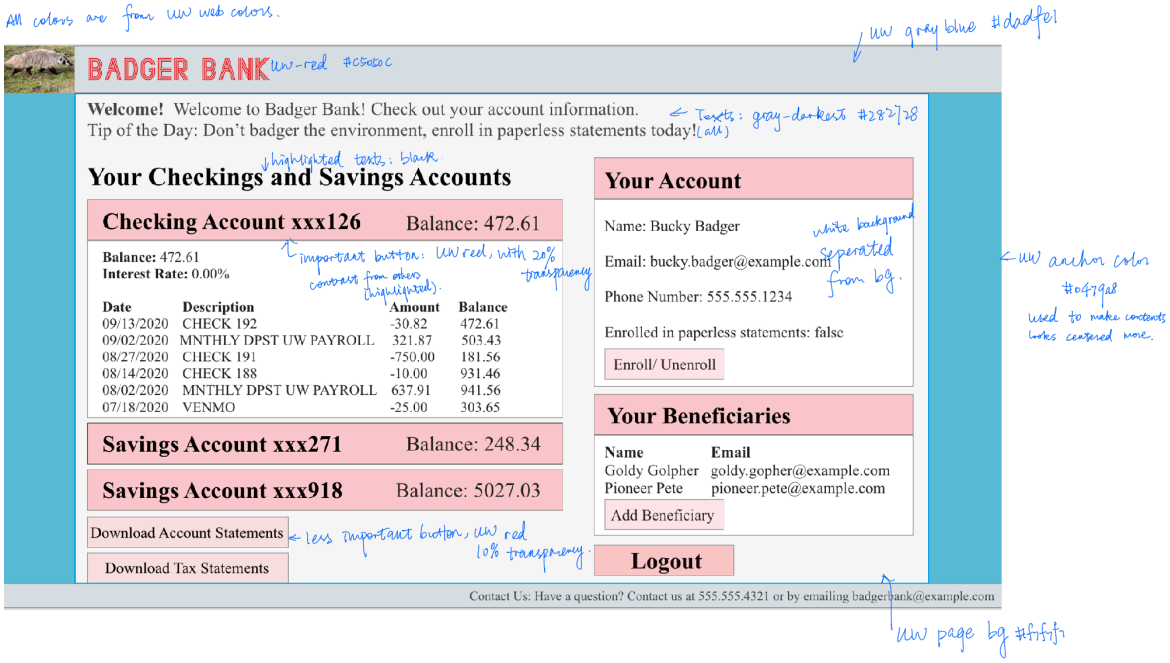
(0.4 Points) **Step 1. Visual Design.** Next, using knowledge of banking operations from the think-aloud, you will redesign JavaScript **β** with the goal of improving its visual design. First consider what elements you must place on the canvas. All elements should have a function. If the original design included elements with no function or use, you can exclude them in your redesign. Next, consider the design principles, and ask yourself, "how can I direct user attention appropriately," "do I need to create contrast between elements," "how do I achieve unity," and so on. You will need to go back and forth between the elements and principles. For example, you must determine what user attention should be directed to, such as a product photo, a button, or a paragraph of text, in order to place the focal point to it. Similarly, you should think about the composition of elements to create an appropriate level of balance on the page. Your redesign should be in the form of a digitally- or hand-drawn wireframe with annotations that justify the use of the elements and principles.



- There would also be the “back to top” link at the buttom of the left column (left column would be really long if the user open all the bank accounts

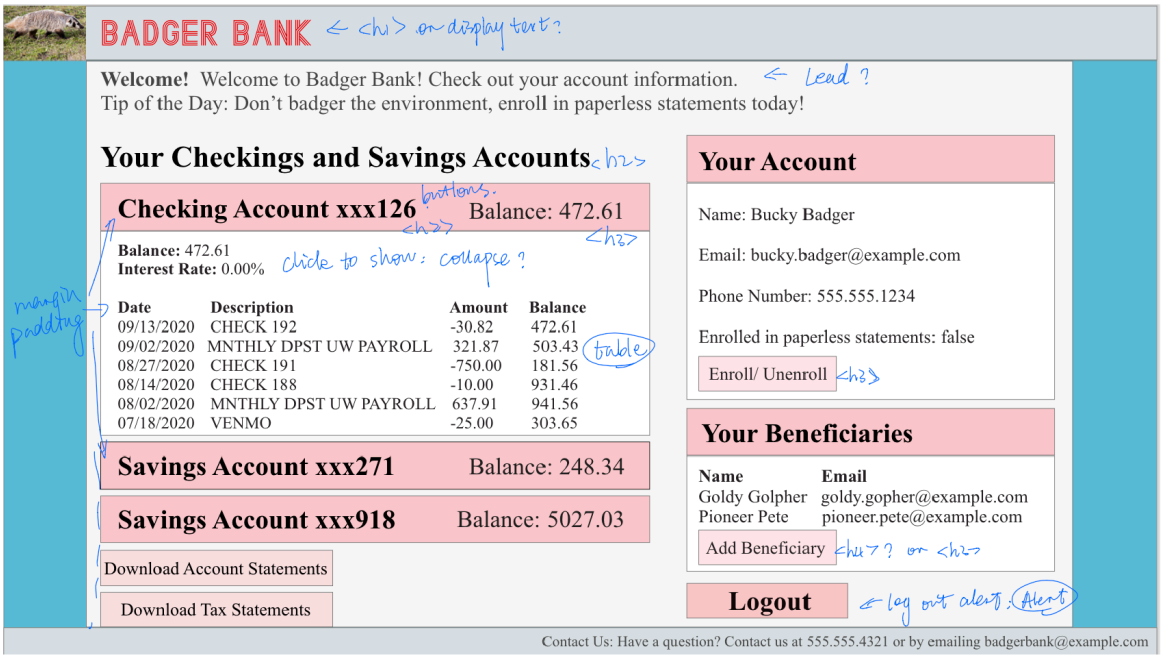
(0.2 Points) **Step 2. Specify color and type choices.** Finally, determine what color palette your redesign should follow. How many colors/shades will you use? Why will you use these colors? Keep design principles in mind when you are choosing colors. For example, using contrasting colors, you can create contrast and manage user attention. Additionally, determine what category of typeface and what font you will use, whether or not you will use multiple fonts on the page, and how you will parameterize each font. You may review the fonts in your computer’s fonts folder (on Windows, go to “My Computer > Control Panel > Fonts” and select “View > Detail. on the Mac use the Font Book app) or the Google Fonts collection to give you ideas and get familiar with available typefaces.

- Use “Times New Roman” for most of the font for unity. Also, as it is a bank system, using this font would be more formal. I also make some of the text in Bold and others in regular (I do not use italic as italic might make texts harder to read, which might not be a good representation in bank system.)
- I also use “Paralines” for the title “Badger Bank” to create a sense of logo.
- For color choices, as we are building “Badger Bank” I use the colors from UW-Madison color for web (https://brand.wisc.edu/web/colors/) As my user likes light color, I will use light grey and white for most of the pages
- I will also make the color of the primary buttons darker when the mouse is on the button to let user knows what she/he will click



Part 3: Implementation

(0.2 Points) **Step 1. Inspect component library elements.** In this step, you will inspect the Bootstrap component library to see how you can realize the layout, visual-element, and color/type choices you have made in the previous part using the component library. You are not expected to change the library components to exactly match your design choices, but to identify which component elements might best meet your design goals. Below, copy the design and the choices you generated in Part 2 and annotate them to describe which components from the library you will use to accomplish your design goals.

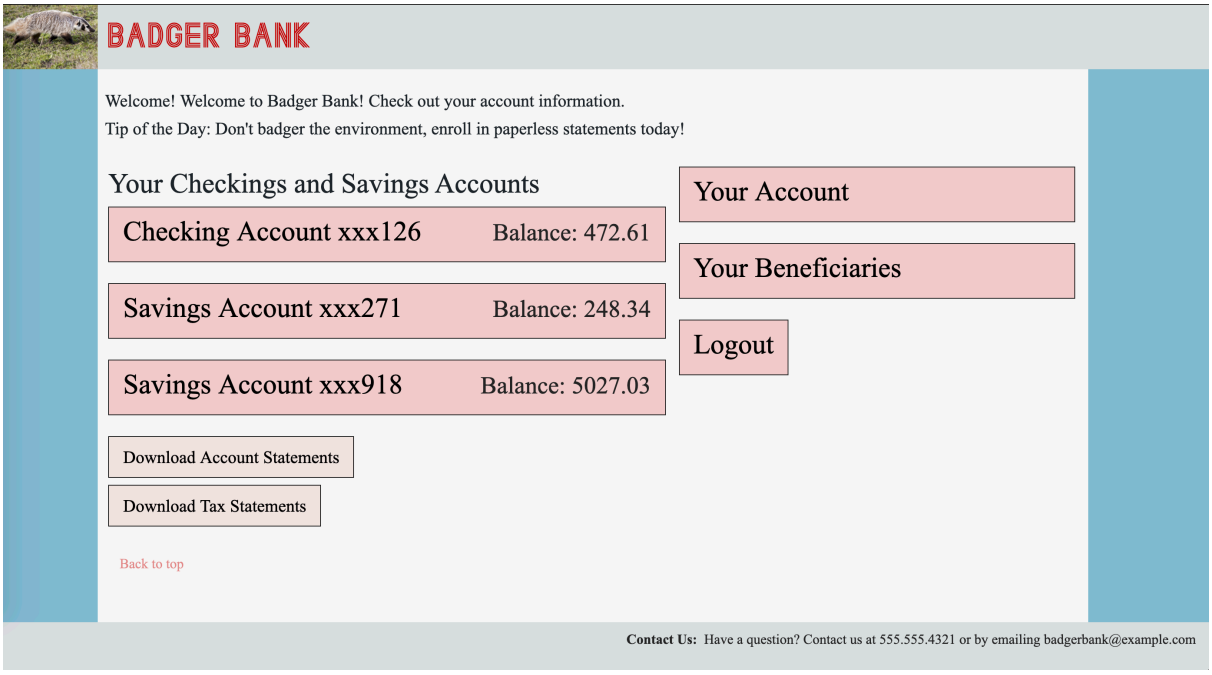


- There are also other components in bootstrap that can be applied to layout, such as for margin and padding (for which we can add class instead of specified in styles)
- Can use the container for the whole body except header and footer and use navbar for header and footer.
- There are also floating classes, such as float-left and float-right
- For table, can use small table and table-hover. (works better in collapse div and can make it clearer about what data user is hover on)

(0.6 Points) **Step 2. Implement your redesign.** The last step of this part will involve implementing the layout and components you had identified in the previous step. You will include Bootstrap in your project and use it to implement your design. You do not have to implement new *functionality*; focus on implementing your *design*.

Your deliverable will be a completed version of this document, attached to the canvas assignment as a PDF, and the GitHub Classroom repository name and latest commit hash.

Screenshot before expending nothing



Screenshot after expending something on both sides