**Project requirements:**

**Design:**

**1) Choose a UI to fix:** Identifying a UI that exists in the world that you think could benefit from a fundamental redesign.  It is going to be better if you find a UI that has a significant flaw, or for which you think a significant re-envisioning would be valuable and interesting to explore.  Don't pick a UI that is generally good, but that could be made better with a few small changes.

We are choosing the AltaFiber mobile UI.

* Submit this UI for a project checkpoint

Completed

* Present your initial idea for the redesign- considering whether you think the issue is a usability or design problem (A above) or you want to retarget the UI for a different goal (B above).

Big usability problem, it’s hard to find out details concerning the service, contacting people when help is needed. Convoluted settings and overall design.

We will redesign to make the UI easy and fast to use, and cover all the need features for a internet provider.

**2) Redesign research, data, planning:** Follow a process for planning your redesign, and gather research+data.

A good redesign process will proceed from a solid motivation, draw from research or data, and be specific in diagnosing the problems.  Since everyone's project is different, you need to consider what research you need to do or data you need to gather to justify and ground your redesign.

Do you think your UI has usability issues?  What precisely are those issues and how serious are they?  Do you think your UI could benefit from targeting different goals?  How?

Select from these methods to advance your understanding and justification for the redesign:

The methods our groups intend to use for the project are methods:

* 1. Capture UI/UX metrics, with a focus on self-reported metrics

G) System Usability Scale

H) Survey

We decided on these by voting and then discussing them, shown below:

* Joe F – I would like to use method A, C, and G for our design research.
* Eli P – I would like to use A, B, and G
* Joanne O – A, C, D, G
* Chimaroke U – I would like to use G and H.

a. Capture UI/UX metrics:

* Self-reported metrics
  + Users report their experiences, satisfaction, emotions, impressions…
* Performance metrics
  + Measuring user behaviors
  + Task success, time on task, errors, efficiency, ease of learning

g. System Usability Scale : [System Usability Scale](https://uc.instructure.com/courses/1712026/pages/system-usability-scale)

h. Survey: [Survey](https://uc.instructure.com/courses/1712026/pages/survey)

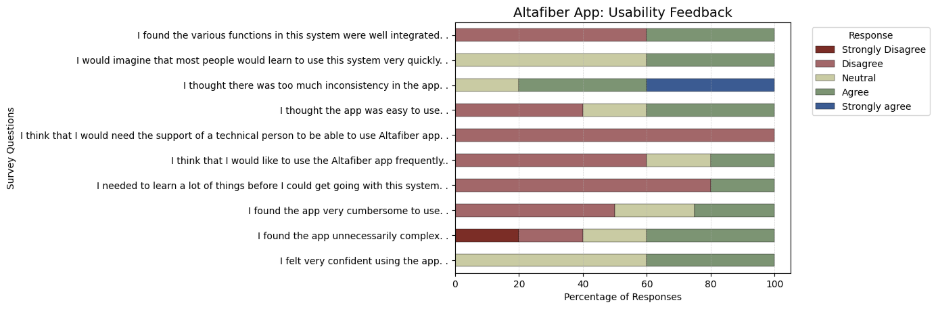
Present a clear and detailed description of your findings from the methods you select.  What precisely needs to be addressed.  This means instead of writing "Users make too many errors", you write: "Users make the following kinds of errors in the following tasks - list them".  Instead of writing "Design choices are poor", present the outcome from your heuristic evaluation with specific design problems.  Instead of saying "UI hard to learn", observe a user trying to learn the interface and note their points of difficulty.  Instead of saying "This UI hurts mental health", describe arguments and data from articles which explain how this UI or kind of UI may contribute to mental health problems.

Our Findings:

A)

G)

H) We used a short Microsoft Forms survey with 5 respondents to gather user feedback on the Altafiber mobile app, based on the **System Usability Scale (SUS)**. The results revealed specific patterns:



* **Perceived complexity:** Responses on unnecessary complexity were evenly split between agreement and disagreement. Respondents found the app cumbersome.
* **Learnability and support:** Responses to needing to learn a lot were mixed, with many selecting Neutral. However, respondents disagreed that they would need technical support, with only a minority agreeing.
* **Confidence and frequency of use:** Confidence levels clustered around Neutral and Agree. However, 60% disagreed they would use the app frequently, and only 20% expressed interest in frequent use.
* **Consistency and integration:** Most respondents agreed there was too much inconsistency in the app. And most disagreed functions were well integrated.
* **Ease of use and learning speed:** Ease-of-use responses were balanced across disagreement and agreement. Views on whether most people could learn the app leaned towards neutral and “agree”.

**Conclusion:** Key issues indicated by the chart are inconsistency, low desire for continued use, and uneven user experiences with complexity and ease of use. These findings justify a redesign focused on simplifying interactions and creating a more consistent, coherent interface.

Use this data to create a focused list of design plans.

**3) Sketch and get feedback:**Use sketching and digital prototyping tools (e.g., Figma) to plan this redesign, and get feedback.

10 by 10 Sketches

Joe F-

A paper with writing on it

AI-generated content may be incorrect.

A drawing of a form

AI-generated content may be incorrect.

Eli P-

A notebook with writing on it

AI-generated content may be incorrect.

A paper with a drawing on it

AI-generated content may be incorrect.

Chimaroke U-

A page of a notebook with writing

AI-generated content may be incorrect.

A piece of paper with writing on it

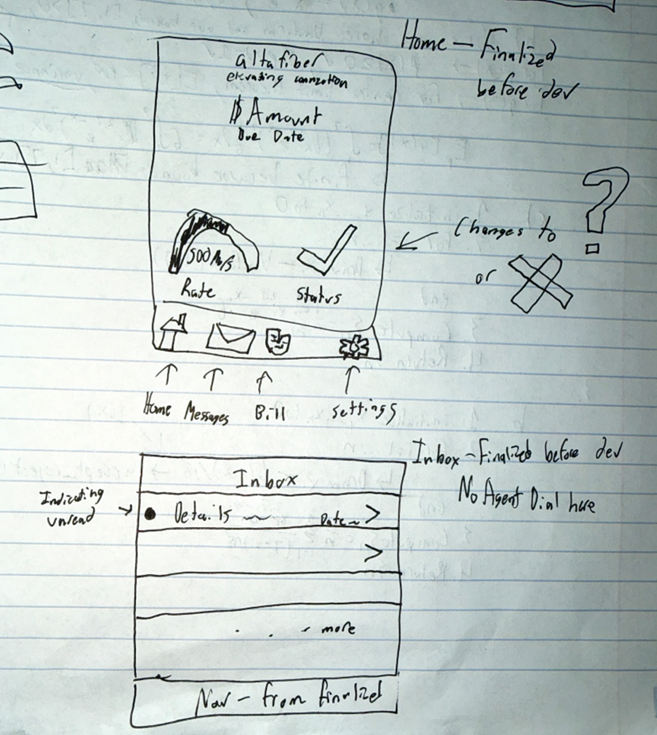
AI-generated content may be incorrect.

Joanne O-

A white board with a few sketches

AI-generated content may be incorrect.

Finalized Vanilla Sketch Based on 10 x 10.



A close-up of a paper

AI-generated content may be incorrect.

Figma sketches:

**4) Feedback:**Get and present feedback on these design plans. Your goal will be to gauge whether your redesign will address the identified issues before implementation.  Choose a method to get this feedback.

Feedback:

**Implementation**

Implementsome meaningful portion of this envisioned redesign in a working prototype, written using Svelte or SvelteKit and Javascript.  This prototype should be hosted publicly, so we can test it.

As with other projects- think of this as a prototype to illustrate the new design.  Select the most interesting aspects of your redesign- no need to spend time on login pages or a routine settings page if this is not illustrating core goals in your redesign.  By "a meaningful portion" I could envision a main page and a subpage or two.  This depends on the UI you select and the major issues you want to tackle.

As with other projects- there is no need for a backend or a database.  I tend to think this takes time away from front-end work and introduces un-needed complexity to projects in this class.  This time, however, I won't penalize you if you want to do a backend with a database.  Just make sure you can publicly host your code for us to test.   Do this at your own risk, because it does take time away from the front end work.

**Documentation:**

As before, document your work so you can showcase this project online.  This can then be used as a case-study for UI/UX positions.

For documentation: assume that someone is encountering your project for the first time. This documentation must be publicly available through one group members portfolio page.  I strongly recommend making a personal copy on your portfolio page, and copying the repo to your git account, in case your team mates someday take their portfolio page down.

* Describe the project
* Present your design work
  + \*\*\* For this project it is particularly important to explain and justify your redesign decisions.  Be sure to present your research, methods, and findings clearly and comprehensively. \*\*\*\*
* Describe your interface in detail:
  + Explain the features and controls
  + Include plenty of screenshots to illustrate your interface and different actions users can perform within it
* Explain how you implemented this application (libraries, code structure....)
* Optional- Use of AI-  If you used AI, describe how.  Did you identify strengths/limitations of using AI for your application.
* Future work- No project is ever fully done. What would you do next?  This is also a place to discuss the work you attempted but could not fully complete before the project deadline- include screenshots to illustrate and document your progress.
* Include a 2-3 minute demo video, showing your interface in action.
  + The easiest way to record this is with a screen capture tool, which also captures audio- such as Quicktime.  Use a voiceover to explain your application.  Include the name of the project, your name, the project components, and how your application works.  You can present it on your webpage or on youtube, but it must be linked on your webpage.
* Include a link to your source code on github and a link to the publicly hosted application.

**Presentation:**

In class: Give a 5-6 minute talk on your UI, with 1-2 minutes for questions.  Everyone in the group should participate in the talk.  You may use slides, videos or live demos to showcase your project.