The PLAN PLAN PLAN

1. Introductions - 10ish seconds each + team name. (60 secs) (Everyone)
   a. Slide with our SoC pixel characters
2. OUR TASK, leading into PAIN POINTS (45 seconds) (Deen)
3. Personas/User stories to focus on the pain points and potential solutions to solve these (90 secs) (George)
   a. USER IS AT THE CENTER - we know their pain
   b. → PROBLEM STATEMENT
4. Team organisation once the idea was somewhat solid - Trello, documentation of what we've done, Figma Jam, the idea to stick together for 90% of it so everyone did everything (60 secs) (Darren)
5. Our plan - project ideas BASED ON these pain points (120 secs) (Ben and Deen)
   a. Project Ideas → Ideation (add user flow at the end since we had iterations) (Deen)
   b. Anki clone because it ticks most boxes + versatile (Deen)
   c. → MVP (which in itself went through many iterations = AGILE) (Ben)
   d. Low res → High res → transition to the (Ben)
6. DEMO (THE APP) (120 secs) → don't show code? (Darren)
7. Stretch goals + RETRO + make it clear we kept agile THROUGHOUT (90 secs) (George)

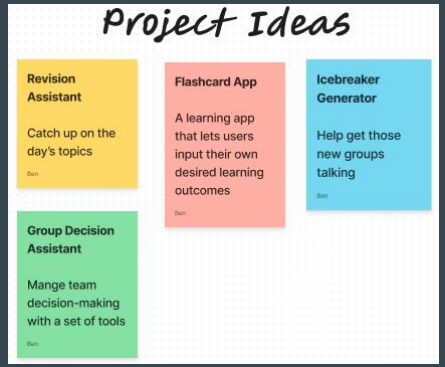
Total TIME: 585s / 9 mins 45s

→ 520s = 130s each

Questions they will likely ask, AT THE END:

1. What did you struggle with? → other team struggled with useEffect for example
2. A teamwork-based question*
    a. WE'RE A DEMOCRACY :)

*Could refer to the manifesto



No Nerds Left Behind

•••

Ben, Darren, Deen and George

Introducing...

Darren
Ensuring a comrade isn't alone in a difficult spot, I value collaboration and want to support my teammates as best as I can.

Ben
From Sound design to sound Design!

Deen
The team over everything.

George
I love computers and the fundamental logic that defines them.

Our Task and the relevant ‘Pain Points’

- To make a front end app that aids our fellow bootcampers, on their journey...

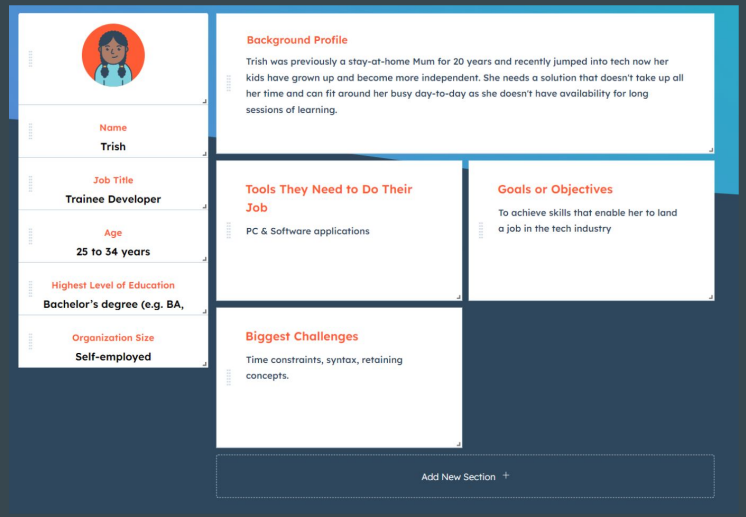
1. Not enough time to get to grips with difficult concepts
2. Time constraints in their busy lives

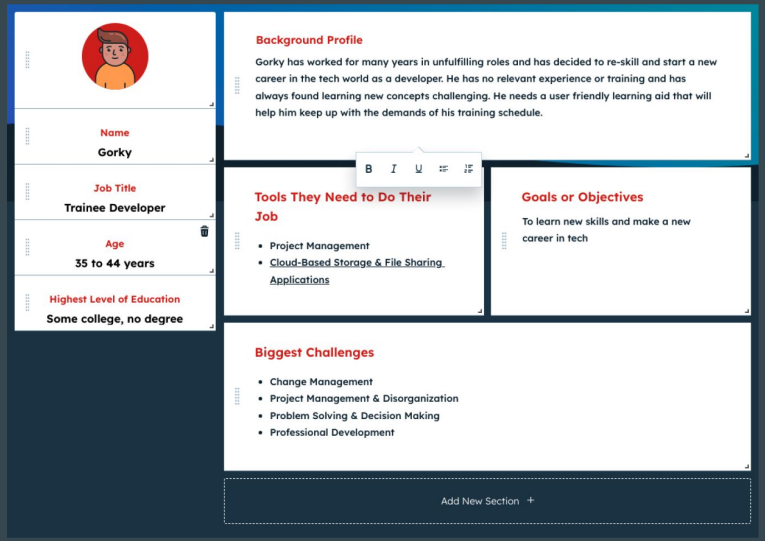
And more. All normal and very much human!

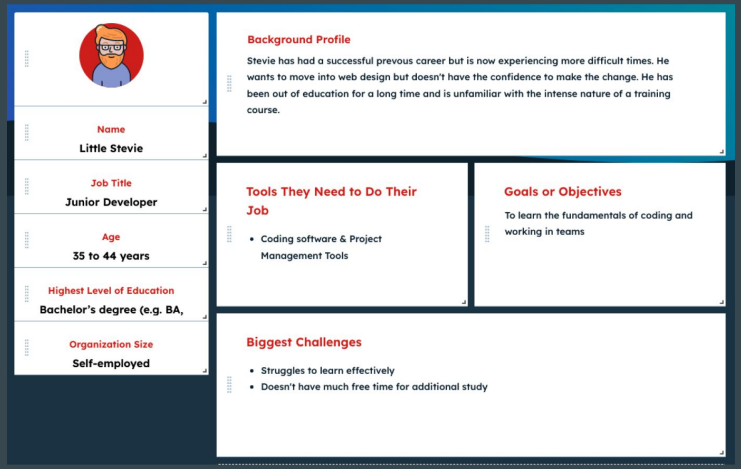
User-centric thinking - what ticks most of the boxes?

Of course, there’s only so much software can address (narrow down our ideas)

User Personas



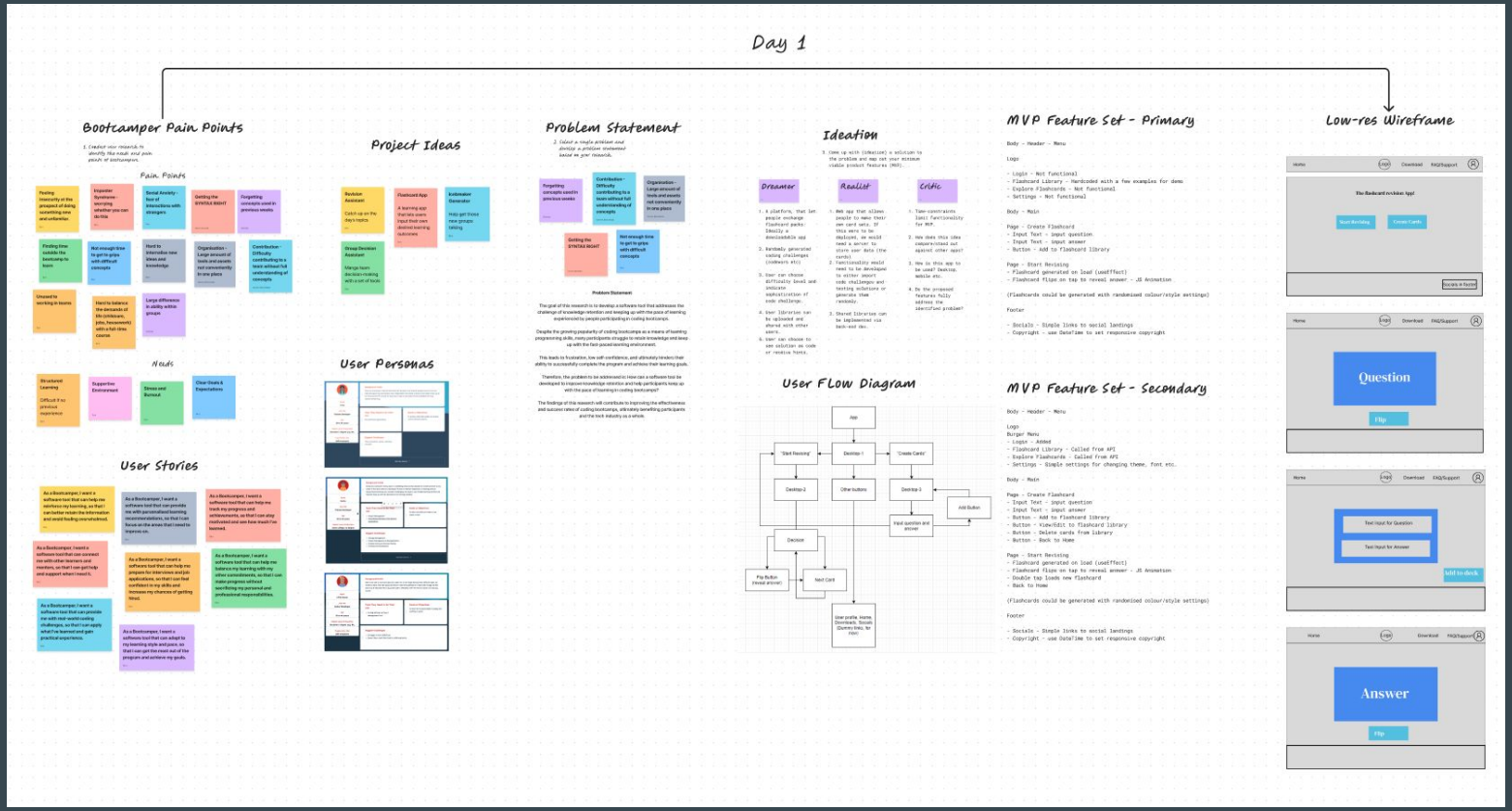




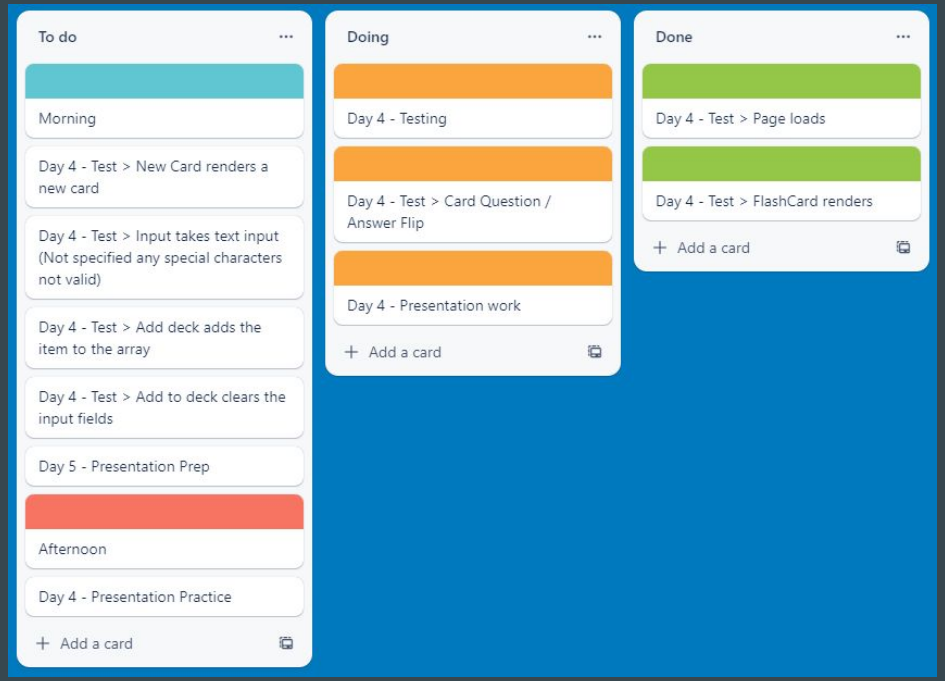
Problem Statement

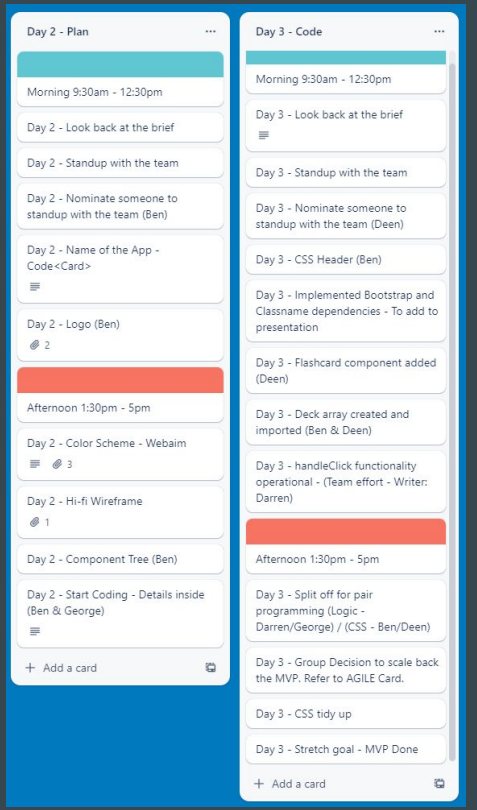
Research goal: develop a software tool that addresses the challenge of knowledge retention and keeping up with the pace of learning experienced by bootcampers.

Team Brainstorm - Figma

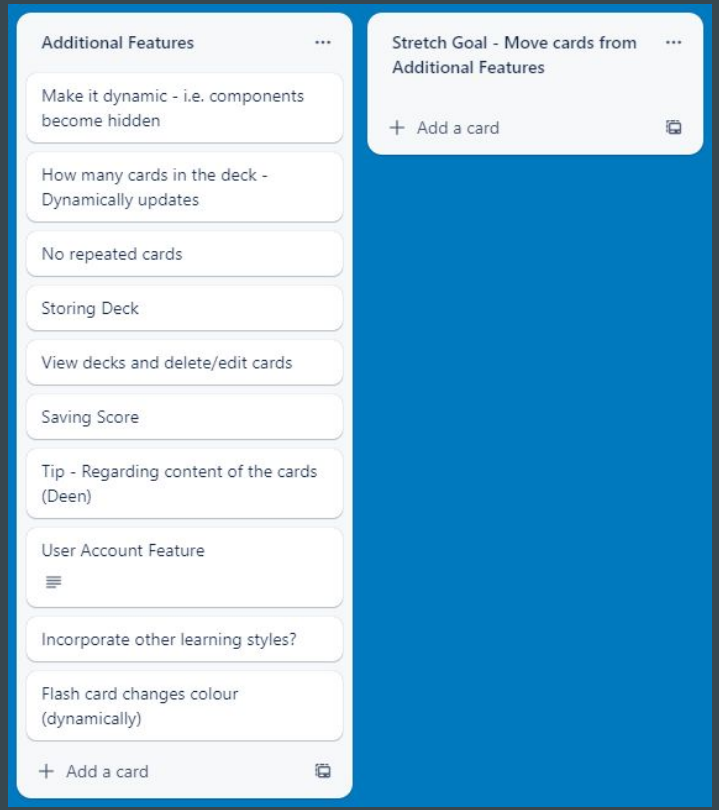


Organisation & Documentation - Trello

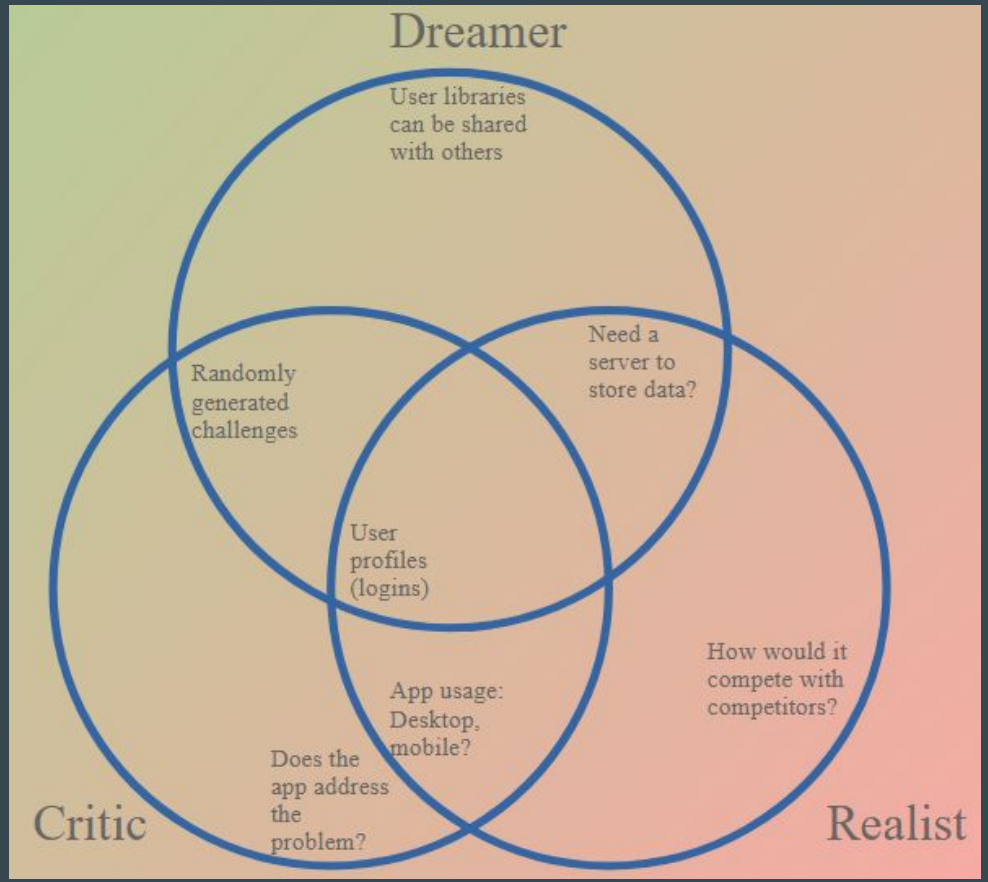




In the event we did deviate...



Slide 13 ideas



The Product

Code <Cards>

Inspired by Anki, but with code in mind

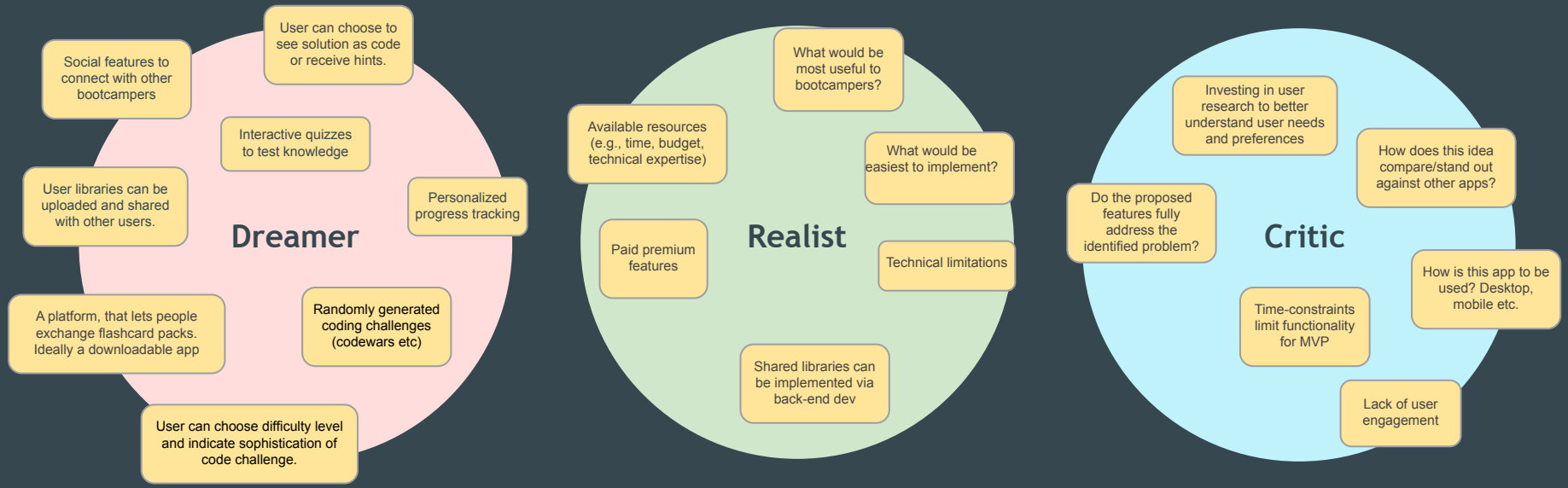
1. Ticks most of the boxes:

```
// Knowledge retention from the BARRAGE of learning ✅
// Help get the syntax right ✅
// Little time investment, for those with busy lives ✅
// Build confidence ✅
```

2. Versatile option → Simple but beautiful MVP, with scope to build

Project Development

Ideation



Project Development

Minimum Viable Product

MVP Primary

Header

- Logo - Icon only
- App Name - Text
- Profile - Icon only
- Settings - Icon only

Body - Main

Page - Create Flashcard
- Input Text - input question
- Input Text - input answer
- Button - Add to flashcard library

Page - Start Revising

- Flashcard generated on load (useEffect)
- Flip card to reveal answer - JS Animation

Footer

- Copyright

MVP Secondary

Header - Menu

- Burger Menu
- Login - Added
- Explore Flashcards - Called from API
- Settings - Simple settings for changing theme, font etc.

Body - Main

Page - Create Flashcard
- Button - View/Edit to flashcard library
- Button - Delete cards from library
- Button - Back to Home

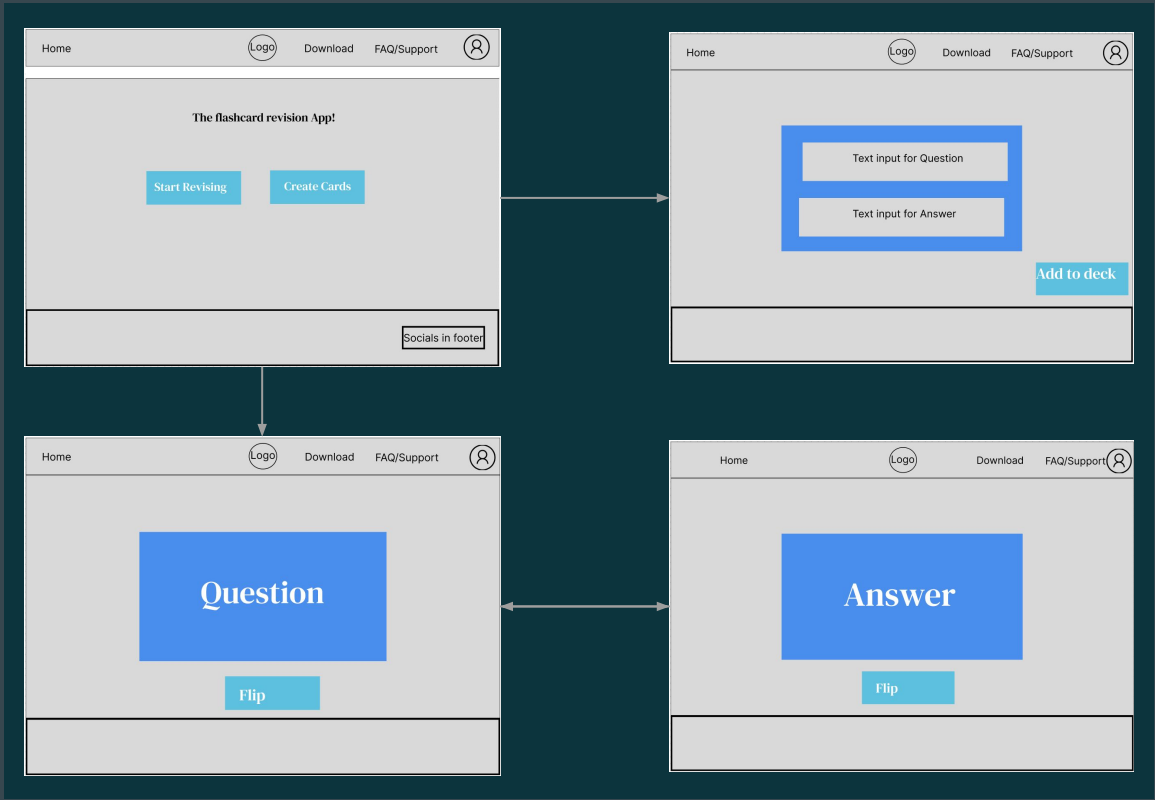
Page - Start Revising
- Double tap loads new flashcard
- Back to Home

Footer

- Socials - Simple links to social landings

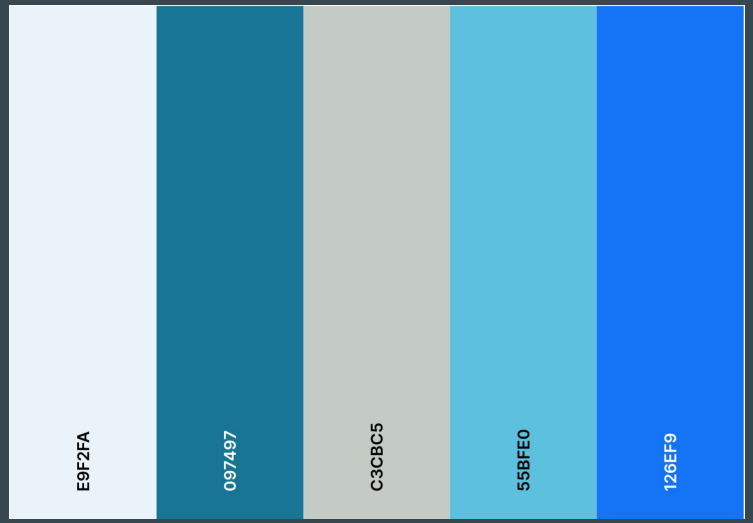
Project Development

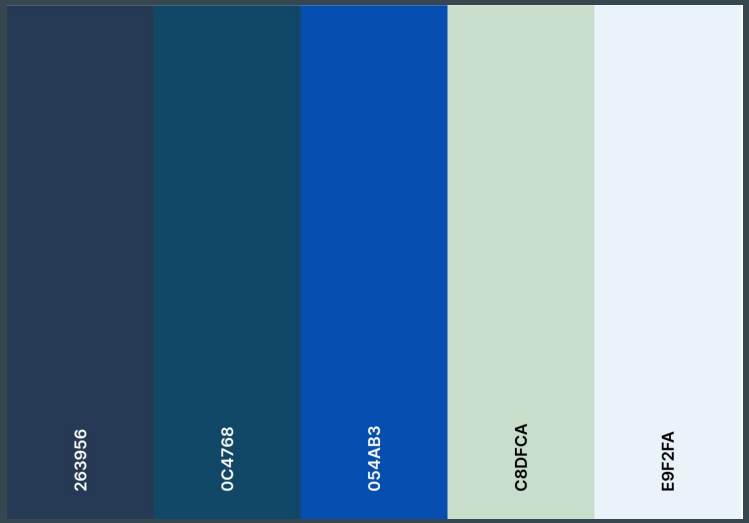
Low Fidelity Wireframing



Project Development

Colour Palette

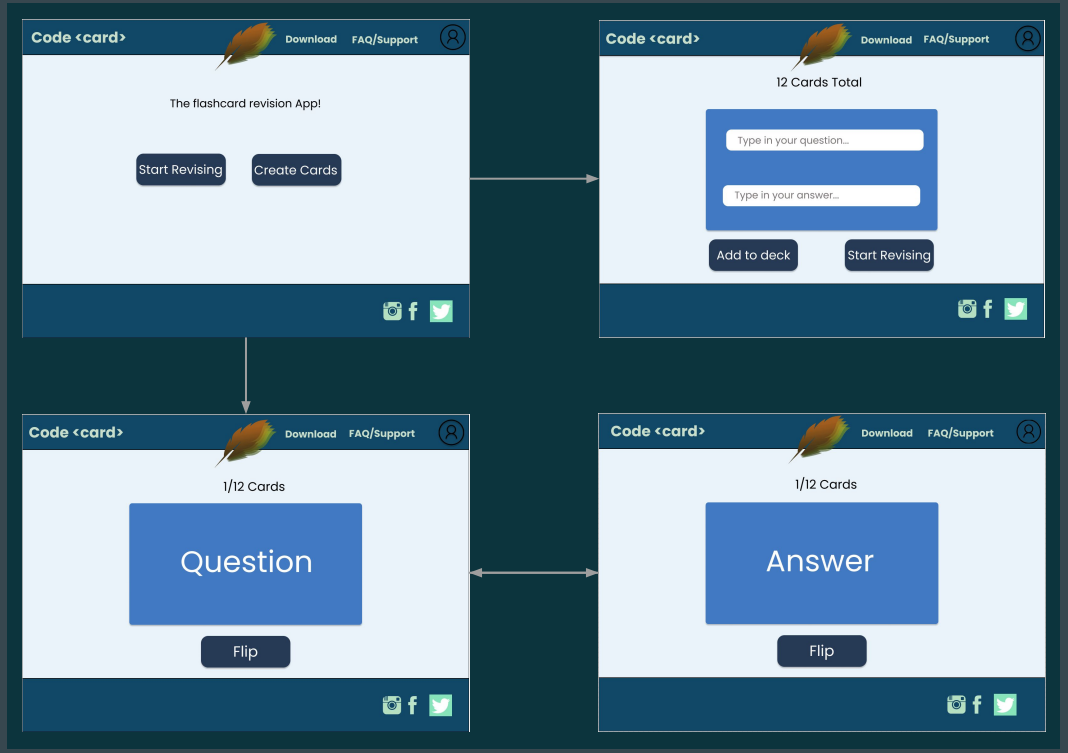




(Adheres to AAA standard)

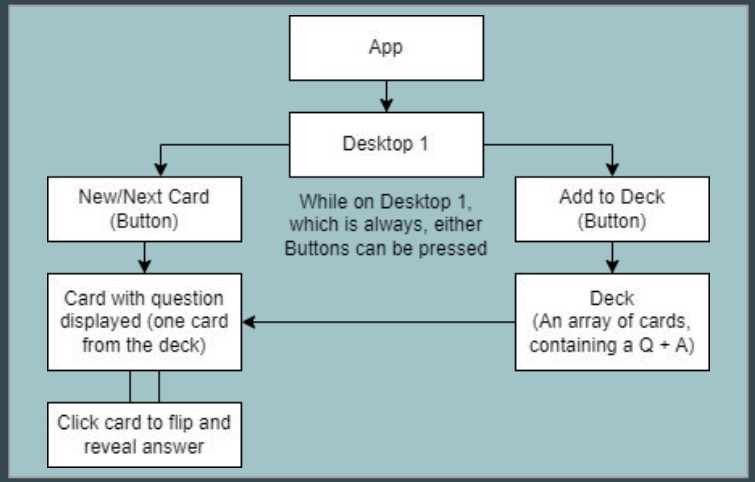
Project Development

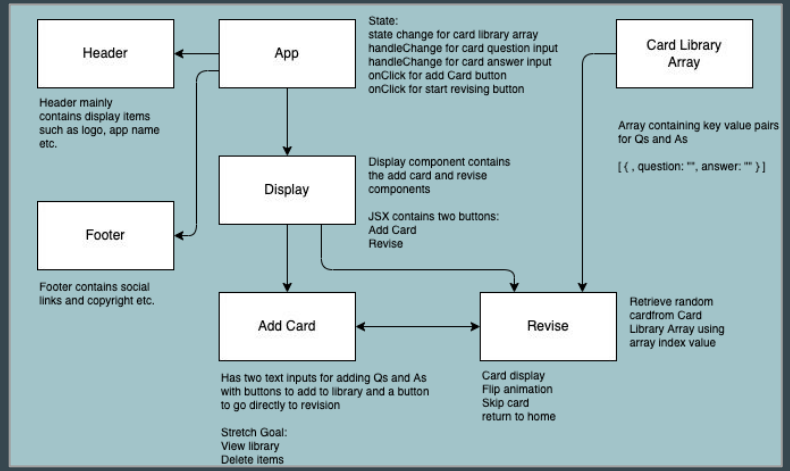
High Fidelity Wireframing



Project Development

User Flow & Component Tree





Following Agile methodology

Plan, Design, Develop, Test and Evaluate

Incorporating Agile into our project allowed to us to keep our MVP focussed on solving the problem statement.

Stretch Goals

- A multi-page app (having relevant components on different pages)
- Functionality for user profiles and then allowing them to save their own data
- Importing other libraries of cards via APIs
- Other functionality such as typing your answer in and scoring when correct

Retrospectives

What worked well for us?

- Planning and ideation
- Minimised tangents to keep productive

What did not work well for us?

- Testing
- State management

What actions can we take to improve our process going forward?

- Explore different react libraries

Thanks for listening!

On break, we are still here!

...

Ben, Darren, Deen and George