

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)

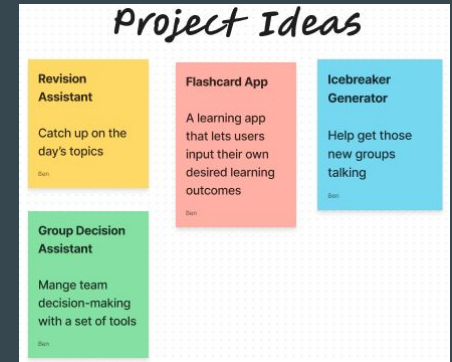
Total TIME: 585s / 9 mins 45s

→ 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)


**Getting the
SYNTAX RIGHT**

Deen Qureshi

**Forgetting
concepts used in
previous weeks**

George

User Personas



Name
Trish

Job Title
Trainee Developer

Age
25 to 34 years

Highest Level of Education
Bachelor's degree (e.g. BA,

Organization Size
Self-employed

Background Profile

Trish was previously a stay-at-home Mum for 20 years and recently jumped into tech now her kids have grown up and become more independent. She needs a solution that doesn't take up all her time and can fit around her busy day-to-day as she doesn't have availability for long sessions of learning.

Tools They Need to Do Their Job

PC & Software applications


Goals or Objectives

To achieve skills that enable her to land a job in the tech industry

Biggest Challenges

Time constraints, syntax, retaining concepts.

Add New Section +



Name
Gorky

Job Title
Trainee Developer

Age
35 to 44 years

Highest Level of Education
Some college, no degree

Background Profile

Gorky has worked for many years in unfulfilling roles and has decided to re-skill and start a new career in the tech world as a developer. He has no relevant experience or training and has always found learning new concepts challenging. He needs a user friendly learning aid that will help him keep up with the demands of his training schedule.

Tools They Need to Do Their Job

- Project Management
- Cloud-Based Storage & File Sharing Applications


Goals or Objectives

To learn new skills and make a new career in tech

Biggest Challenges

- Change Management
- Project Management & Disorganization
- Problem Solving & Decision Making
- Professional Development

Add New Section +



Name
Little Stevie

Job Title
Junior Developer

Age
35 to 44 years

Highest Level of Education
Bachelor's degree (e.g. BA,

Organization Size
Self-employed

Background Profile

Stevie has had a successful previous career but is now experiencing more difficult times. He wants to move into web design but doesn't have the confidence to make the change. He has been out of education for a long time and is unfamiliar with the intense nature of a training course.

Tools They Need to Do Their Job

- Coding software & Project Management Tools

Goals or Objectives

To learn the fundamentals of coding and working in teams

Biggest Challenges

- Struggles to learn effectively
- Doesn't have much free time for additional study

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

Day 1

Bootcamper Pain Points

1. Define your research to identify the main pain points of bootcampers



Needs



User Stories



Project Ideas



User Personas



Problem Statement

2. Define a single problem and provide a solution statement based on your research



The goal of this research is to develop a software tool that addresses the challenges of knowledge retention and finding an effective way of learning programming in a structured and engaging way.

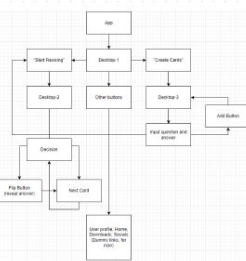
The goal of this research is to develop a software tool that addresses the challenges of knowledge retention and finding an effective way of learning programming in a structured and engaging way.

Ideation

3. Come up with 10-15 ideas in relation to the problem and map out your minimum viable product (MVP)



User Flow Diagram



MVP Feature Set - Primary

Body - Header - Menu

Logo

- Login - Not Functional
- Flashcard Library - Not Functional
- Explore Flashcards - Not Functional
- Settings - Not Functional

Body - Main

Page - Create Flashcard

- Input Text - Input question
- Input Text - Input answer
- Button - Add to Flashcard Library

Page - Start Testing

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

Footer

- Single - Simple links to social settings
- Copyright - use Bootstrap to set responsive copyright

MVP Feature Set - Secondary

Body - Header - Menu

Logo

Burger Menu

- Login - Add
- Flashcard Library - Called from API
- Explore Flashcards - Called from API
- Settings - Simple settings for changing them. Part 2.

Body - Main

Page - Create Flashcard

- Input Text - Input question
- Input Text - Input answer
- Button - Add to Flashcard Library
- Button - Keyboard to Flashcard Library
- Button - Delete card from library
- Button - Back to Home

Page - Start Testing

- Flashcard generated on load (useEffect)
- Flashcard Flip on tap to reveal answer - JS Animation
- Double tap to lock the Flashcard
- Back to Home

Footer

- Single - Simple links to social settings
- Copyright - use Bootstrap to set responsive copyright

Low-res Wireframe



Organisation & Documentation - Trello

A screenshot of a Trello board with three columns: "To do", "Doing", and "Done". Each column has a header bar and a list of cards. The "To do" column has a light blue header, followed by a light blue card "Morning", and then four white cards with orange borders: "Day 4 - Test > New Card renders a new card", "Day 4 - Test > Input takes text input (Not specified any special characters not valid)", "Day 4 - Test > Add deck adds the item to the array", and "Day 4 - Test > Add to deck clears the input fields". This is followed by a white card "Day 5 - Presentation Prep", a red header bar, a white card "Afternoon", and a white card "Day 4 - Presentation Practice". The "Doing" column has an orange header, followed by three orange cards: "Day 4 - Testing", "Day 4 - Test > Card Question / Answer Flip", and "Day 4 - Presentation work". The "Done" column has a green header, followed by two green cards: "Day 4 - Test > Page loads" and "Day 4 - Test > FlashCard renders". Each column ends with a "+ Add a card" button and a small icon.

To do

- Morning
- Day 4 - Test > New Card renders a new card
- Day 4 - Test > Input takes text input (Not specified any special characters not valid)
- Day 4 - Test > Add deck adds the item to the array
- Day 4 - Test > Add to deck clears the input fields
- Day 5 - Presentation Prep
- Afternoon
- Day 4 - Presentation Practice
- + Add a card

Doing

- Day 4 - Testing
- Day 4 - Test > Card Question / Answer Flip
- Day 4 - Presentation work
- + Add a card

Done

- Day 4 - Test > Page loads
- Day 4 - Test > FlashCard renders
- + Add a card

A screenshot of a Trello board with two columns: "Day 2 - Plan" and "Day 3 - Code". Each column has a header bar and a list of cards. The "Day 2 - Plan" column has a light blue header, followed by a light blue card "Morning 9:30am - 12:30pm", and then four white cards with orange borders: "Day 2 - Look back at the brief", "Day 2 - Standup with the team", "Day 2 - Nominate someone to standup with the team (Ben)", and "Day 2 - Name of the App - Code <Card>". This is followed by a white card "Day 2 - Logo (Ben)" with 2 attachments, a red header bar, a white card "Afternoon 1:30pm - 5pm", a white card "Day 2 - Color Scheme - Webaim" with 3 attachments, a white card "Day 2 - Hi-fi Wireframe" with 1 attachment, a white card "Day 2 - Component Tree (Ben)", and a white card "Day 2 - Start Coding - Details inside (Ben & George)". The "Day 3 - Code" column has a light blue header, followed by a light blue card "Morning 9:30am - 12:30pm", and then four white cards with orange borders: "Day 3 - Look back at the brief", "Day 3 - Standup with the team", "Day 3 - Nominate someone to standup with the team (Deen)", and "Day 3 - CSS Header (Ben)". This is followed by a white card "Day 3 - Implemented Bootstrap and Classname dependencies - To add to presentation", a white card "Day 3 - Flashcard component added (Deen)", a white card "Day 3 - Deck array created and imported (Ben & Deen)", a white card "Day 3 - handleClick functionality operational - (Team effort - Writer: Darren)", a red header bar, a white card "Afternoon 1:30pm - 5pm", a white card "Day 3 - Split off for pair programming (Logic - Darren/George) / (CSS - Ben/Deen)", a white card "Day 3 - Group Decision to scale back the MVP. Refer to AGILE Card.", a white card "Day 3 - CSS tidy up", and a white card "Day 3 - Stretch goal - MVP Done". Each column ends with a "+ Add a card" button and a small icon.

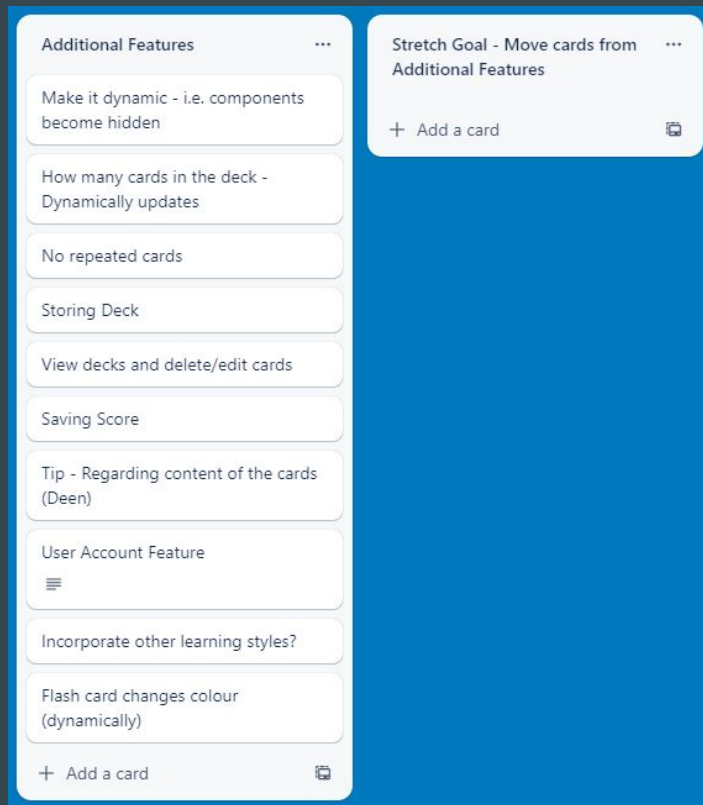
Day 2 - Plan

- Morning 9:30am - 12:30pm
- Day 2 - Look back at the brief
- Day 2 - Standup with the team
- Day 2 - Nominate someone to standup with the team (Ben)
- Day 2 - Name of the App - Code <Card>
- Day 2 - Logo (Ben)
- Afternoon 1:30pm - 5pm
- Day 2 - Color Scheme - Webaim
- Day 2 - Hi-fi Wireframe
- Day 2 - Component Tree (Ben)
- Day 2 - Start Coding - Details inside (Ben & George)
- + Add a card

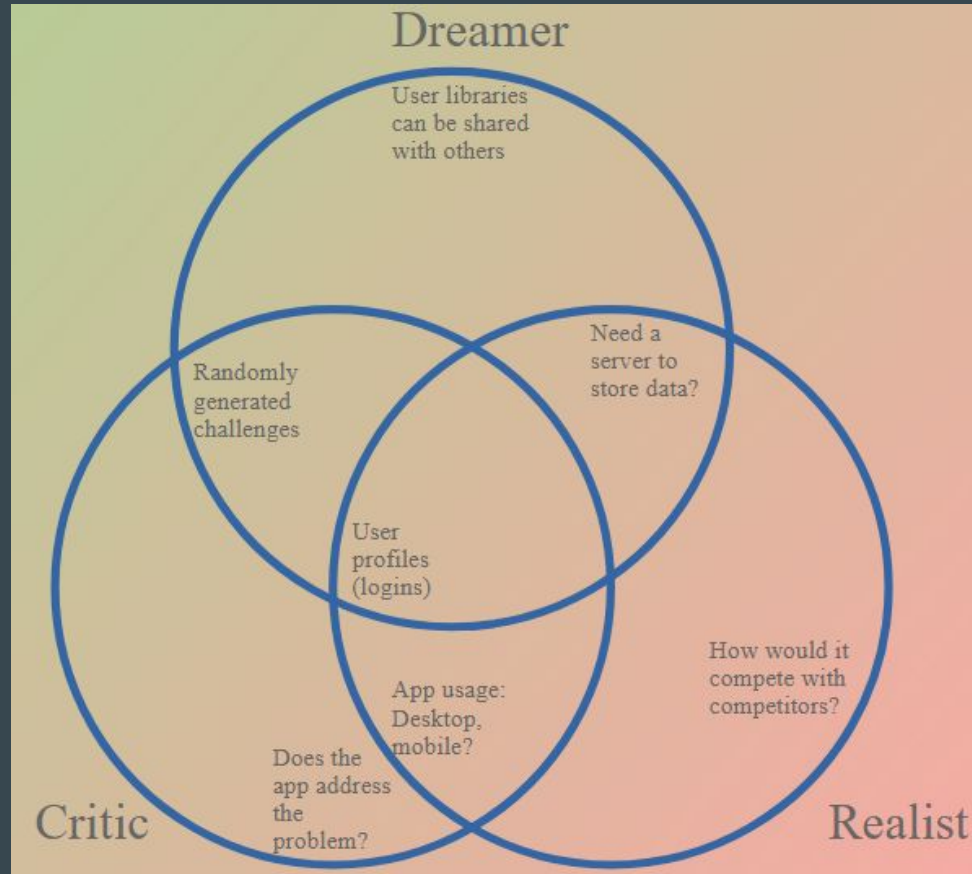
Day 3 - Code

- Morning 9:30am - 12:30pm
- Day 3 - Look back at the brief
- Day 3 - Standup with the team
- Day 3 - Nominate someone to standup with the team (Deen)
- Day 3 - CSS Header (Ben)
- Day 3 - Implemented Bootstrap and Classname dependencies - To add to presentation
- Day 3 - Flashcard component added (Deen)
- Day 3 - Deck array created and imported (Ben & Deen)
- Day 3 - handleClick functionality operational - (Team effort - Writer: Darren)
- Afternoon 1:30pm - 5pm
- Day 3 - Split off for pair programming (Logic - Darren/George) / (CSS - Ben/Deen)
- Day 3 - Group Decision to scale back the MVP. Refer to AGILE Card.
- Day 3 - CSS tidy up
- Day 3 - Stretch goal - MVP Done
- + Add a card

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

Dreamer

User can choose to see solution as code or receive hints.

Interactive quizzes to test knowledge

Personalized progress tracking

Randomly generated coding challenges (codewars etc)

User can choose difficulty level and indicate sophistication of code challenge.

User libraries can be uploaded and shared with other users.

Social features to connect with other bootcampers

A platform, that lets people exchange flashcard packs. Ideally a downloadable app

Realist

What would be most useful to bootcampers?

What would be easiest to implement?

Technical limitations

Shared libraries can be implemented via back-end dev

Available resources (e.g., time, budget, technical expertise)

Paid premium features

Critic

Investing in user research to better understand user needs and preferences

How does this idea compare/stand out against other apps?

How is this app to be used? Desktop, mobile etc.

Lack of user engagement

Time-constraints limit functionality for MVP

Do the proposed features fully address the identified problem?

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

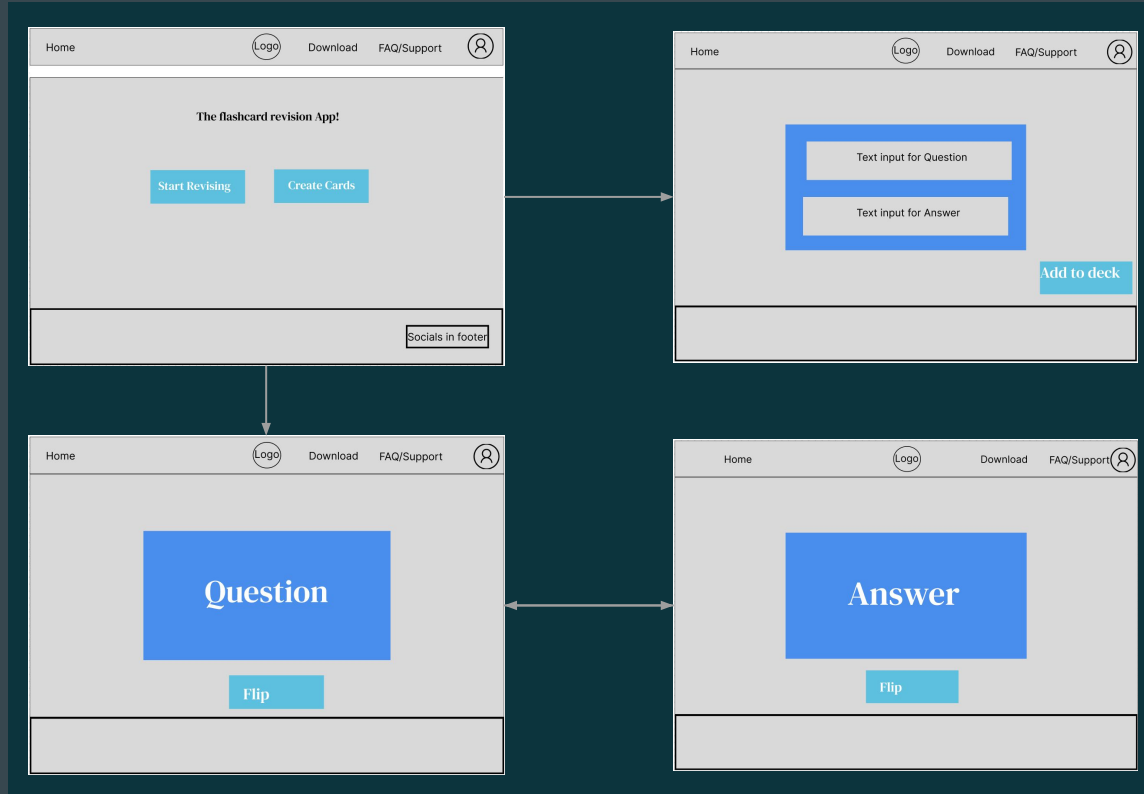
- Double tap loads new flashcard
- Back to Home

Footer

- Socials - Simple links to social landings

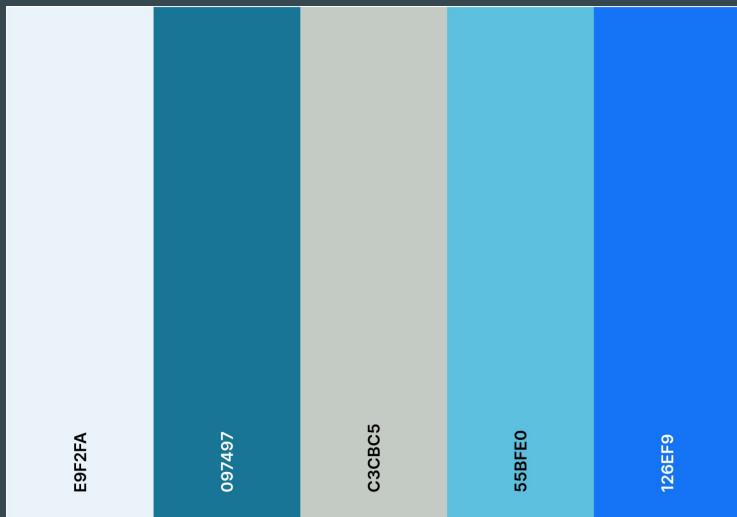
Project Development

Low Fidelity Wireframing



Project Development

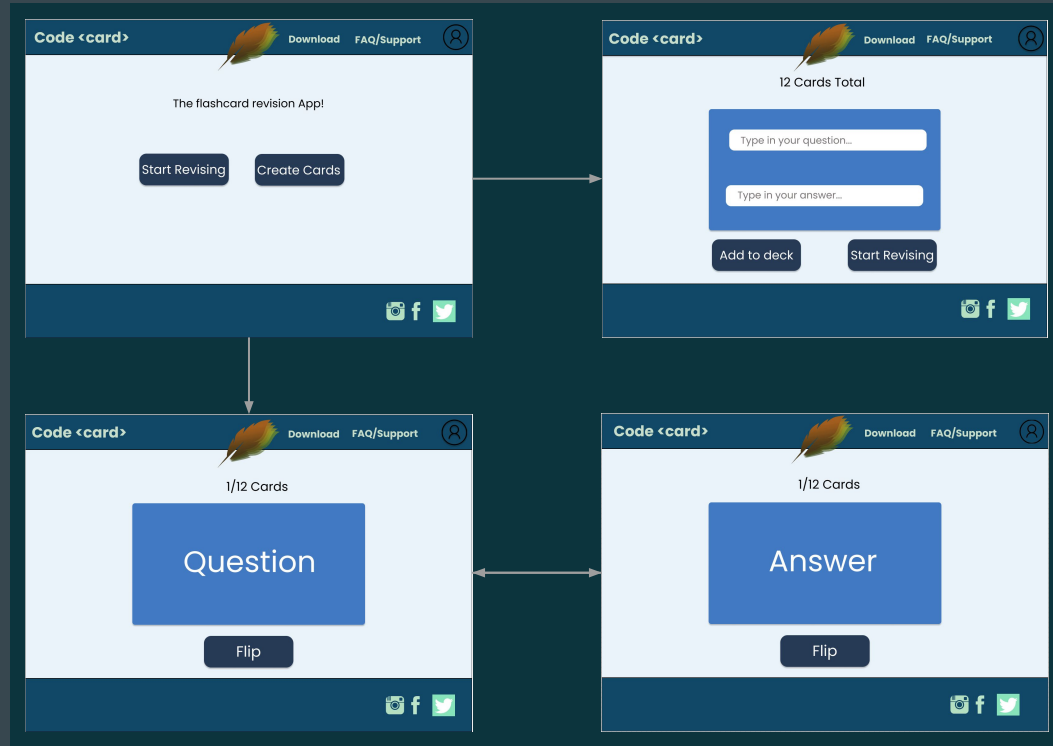
Colour Palette



(Adheres to AAA standard)

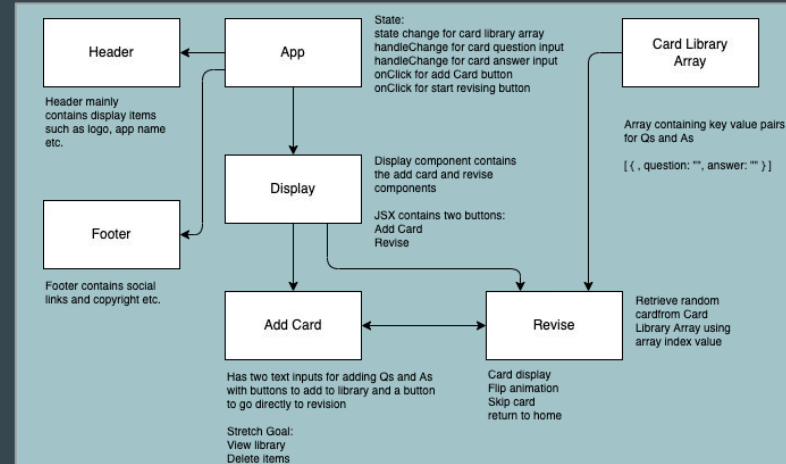
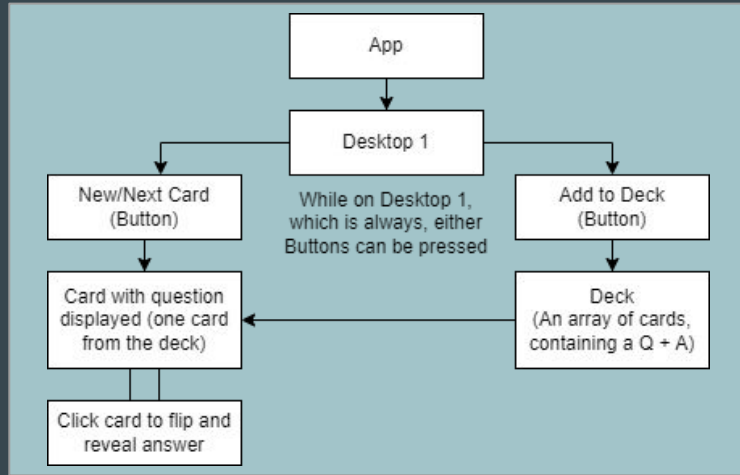
Project Development

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