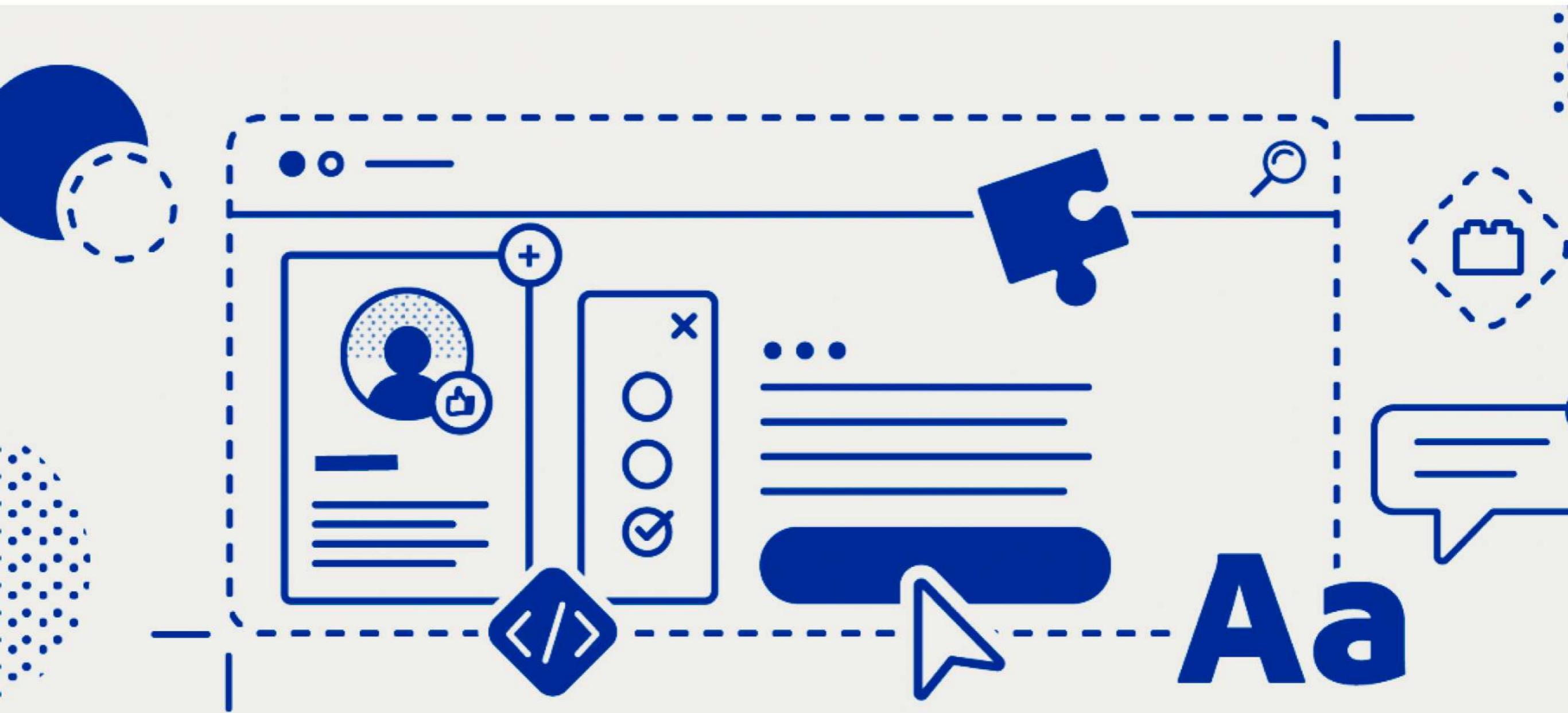


CASE STUDY

SPARK DESIGN SYSTEM



Aa

INDEX

1. Project overview	3	12. Design Tokens	18
2. Problem statement, Goals and Responsibilities	4	13. Components library	19
3. Project Kick start	5	13a. Components structure	20
4. Name and Logo	6	14. Versioning	21
5. Research and Analysis	7	15. Accessibility	22
6. Principles	8	16. Managing breaking changes	23
7. Structure	9	17. Governance and Roll out	24
8. Initial library	10	18. Dev build (Storybook)	25
9. Project setup	11	19. Community contribution	26
10. Libraries setup	12	20. Next steps	27
11. Core library	13		
11a. Colour	14		
11b. Typography	15		
11c. Iconography	16		
11d. Spacing	17		

Client



Duration

6 Months

Platform

Web platform and Native Mobile App

Project Overview

British Gas has a new product to launch called **Evolve** which needed a new design language that is unique to this new product but still complements the British Gas brand and stays true to its core brand values.



Problem Statement

The business at British Gas identified the need for a new design system for their new product (**Evolve**) to ensure consistency and efficiency across various features and interfaces. They recognised that without a design system, inconsistencies in design and user experience could lead to a confusing and frustrating experience for users and that could hurt brand's reputation.

Goals

- Provide designers and devs with an initial basic design system for them to kick start on their design work.
- Work through the basic design system and make it mature and robust.
- Create a state of the art modern design system which compliments the core brand as well as defines Evolve's design language.

Responsibilities

- Lead on all things design system.
- Audit existing practises with other design system that was being used for the core brand.
- Carry out research and competitor analysis and identify best in class practises for a brand new design system.
- Setup progress board to track progress of the state of design system evolution.
- Involve whole community of in house designers and developers to make it community driven and provide the relevant training for designers to be able to contribute.

Team

- 2 Designers
- 1 Dev lead
- 1 UX Researcher
- Design Community

Project Kick start

To start with the project, I organised a few **workshops** with designers, devs and involved stakeholders to find out what works with the core design system (**Nucleus**), what are the pain points and what are the potentials.

The workshops helped us to find out what an ideal design system should include, how it should behave and how it should be structured.



Name and Logo

In a parallel exercise, I wanted community involvement in suggesting the name for our new design system just so it gets its own identity and personality. There was a competition for the name suggestions and some great names were shortlisted which were in keeping with the product brand etiquette i.e. Gaia, Flare, Lumin, space but the one that stood out the most and won after a few sessions of voting was

Spark

There was a general consensus that the meaning and behaviour of the name suits the nature of the product and sits well with the brand values.

Next step was to give a visual to the Spark and similar to the name finding exercise, we asked the community to enter a logo design competition for Spark. Within a few days we received some very exciting suggestions on the logo.



After a few rounds of discussion and voting, we settled on a logo which was simple and yet modern representing the intent of the design language we were discussing to implement.



Research and Analysis

I began the research on various design systems which were regarded as industry standards serving the purpose within their design ecosystems, that included British Gas core design system called Nucleus. The aim was to understand how respected design systems have evolved and to study innovative and effective approaches.

Some of the design systems I found very useful in my study were;



Carbon Design System (IBM)



Government Digital Service



Atlassian Design System



Fluent

Fluent Design System (Microsoft)

I went through each design system in detail and made notes on areas of strength for each design system. That helped me take inspiration from these design systems and the way these are doing certain things so I could set those as benchmarks in preparation for our new design system.

- Goldman Sachs has good **DS Guidance**
<https://www.figma.com/community/file/1000112838495090231>
- Twilio design system has a good **Basic component setup**
<https://www.figma.com/community/file/1012844215105752327>
- Ant Design great for **Components in Grid format**
- Like the **Colour** page
[https://www.figma.com/file/pWjARIUdvY6BOBI3RWgbVe/Ant-Design-System-for-Figma-2.2-\(Light-Theme-Preview\)?node-id=371%3A8955](https://www.figma.com/file/pWjARIUdvY6BOBI3RWgbVe/Ant-Design-System-for-Figma-2.2-(Light-Theme-Preview)?node-id=371%3A8955)
- Airtable has good **Spacing** description
<https://www.figma.com/community/file/862805330899066752?preview=fullscreen>
- Coder house is good for **Spacing** specs
<https://www.figma.com/community/file/895664261764229458>
- Bold has excellent **Contribution** screen
- Bold has good **Icon usage** screen
<https://www.figma.com/file/TE9FUDtIgVQ4FWIAPtTagxQU/Bold-Design-System?node-id=0%3A1>

Principles

We carried out a UI audit and inventory to see how the designers and developers are using the current design system and what could be improved to help them overcome pain points. Based on the research, we streamlined the following objectives for Spark design system.

- Scalable

Develop a system that can accommodate future growth and new features.

- Modular

Based on atomic design principles, the structure of a design system should be modular, e.g. separate out foundations from core components, build components that are composed of smaller moving parts etc.

- Flexible

Develop a system which is flexible to logical changes and let the designers and devs modify where applicable.

- Accessible & Inclusive

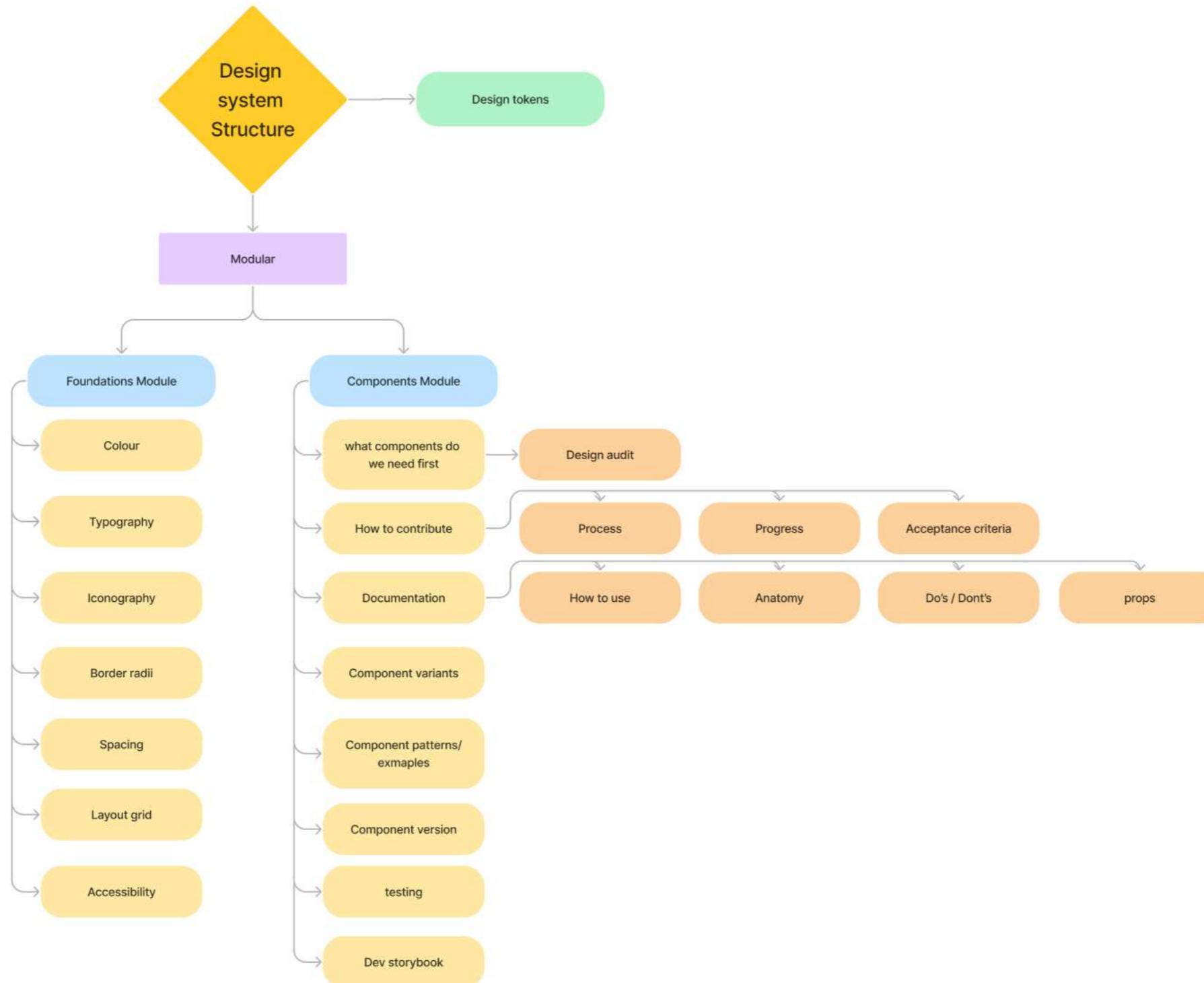
Integrate accessibility and inclusion into our component libraries, both from a design and code repository perspective.

- Well Documented

Create detailed documentation that would support every single aspect of our design system, and also be organised, consistent, and easy to use

Structure

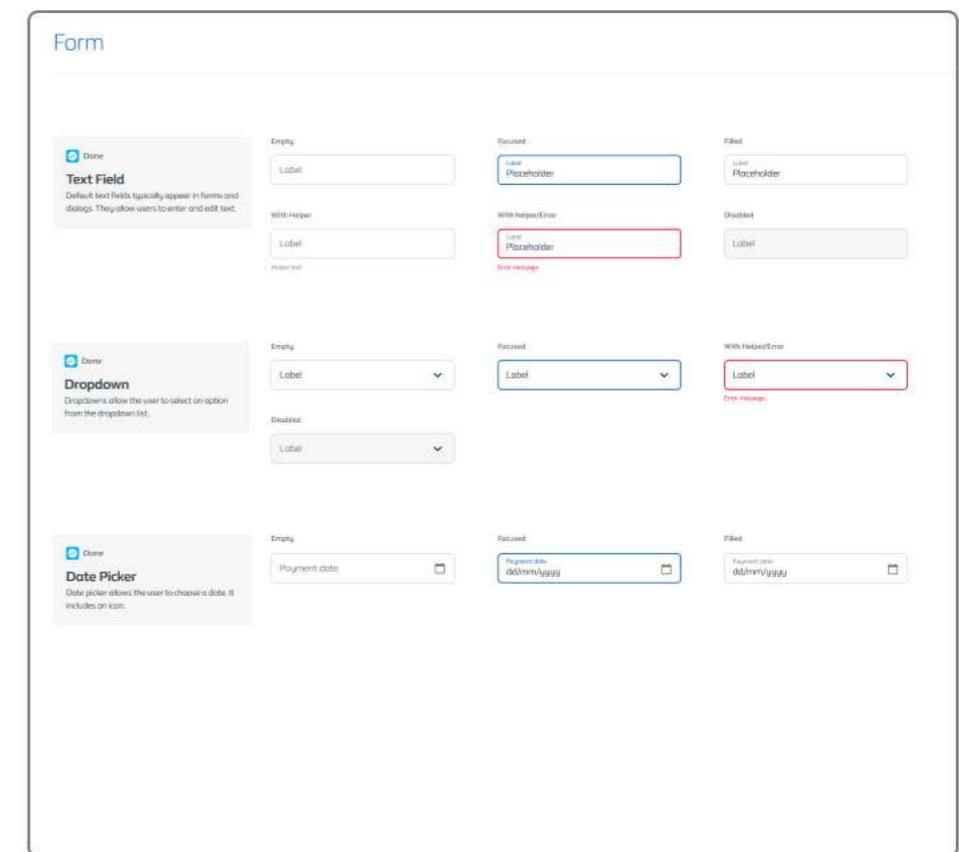
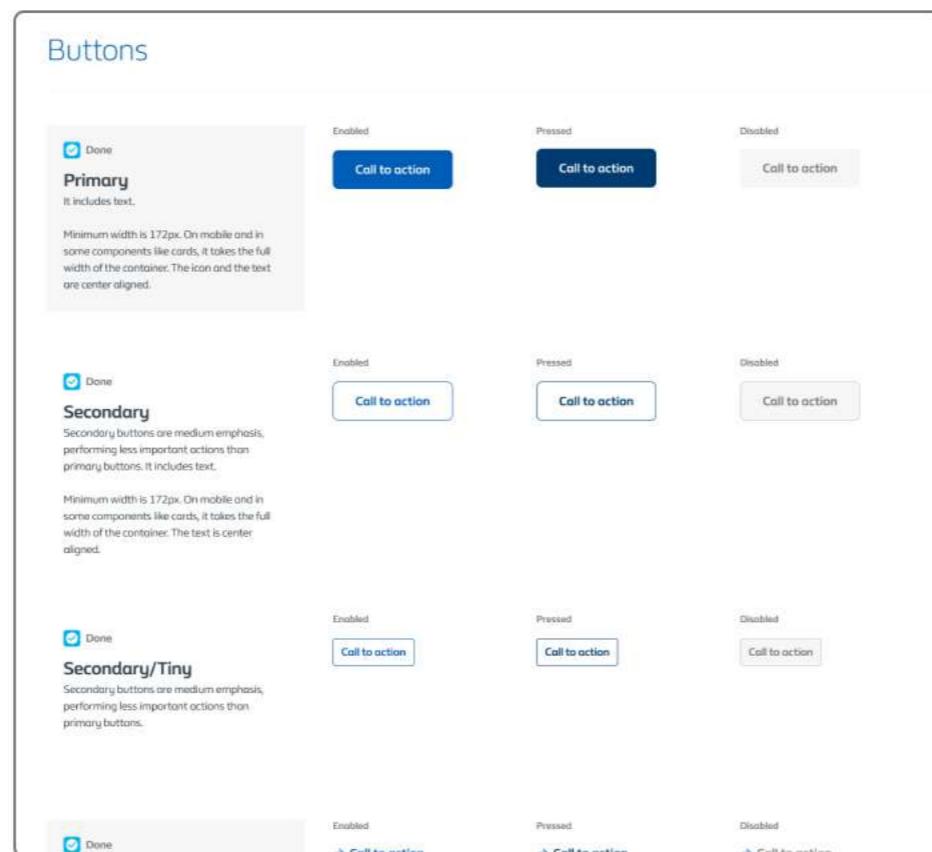
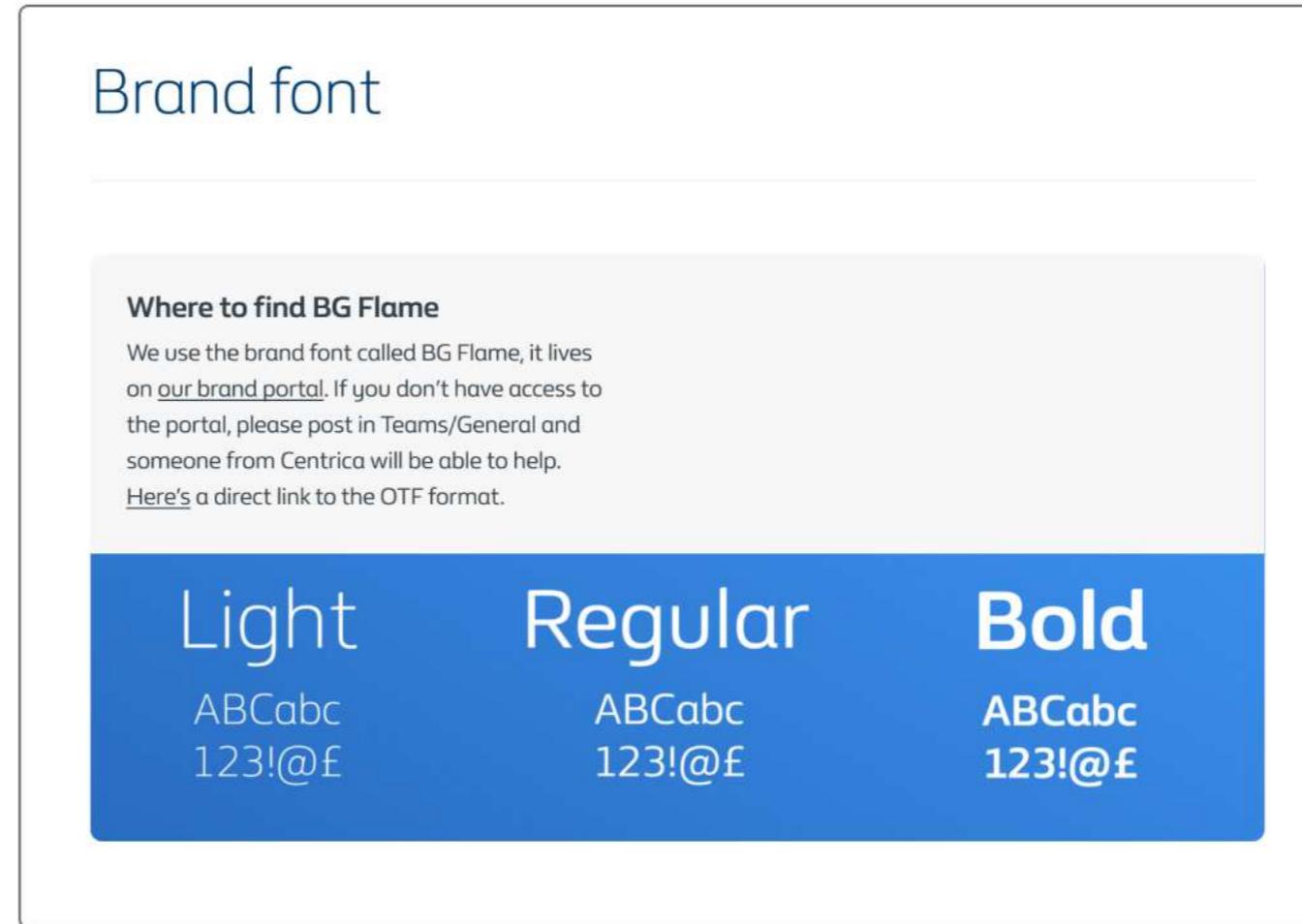
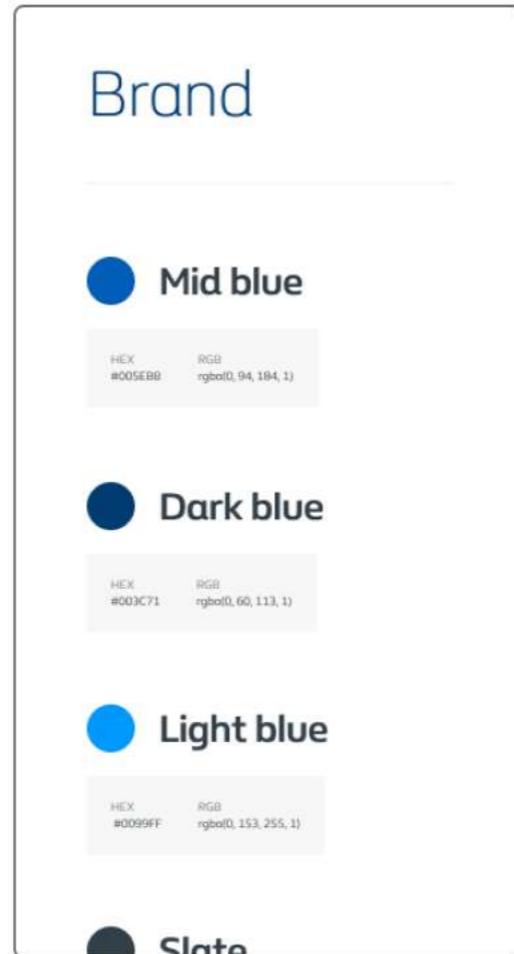
The below flowchart gives an overview of the structure of how Spark was going to be set up. We decided that at its core, Spark is going to be a design-led design system, i.e. the source of truth will live in the design file and code will follow the design specs from these files.



Initial library

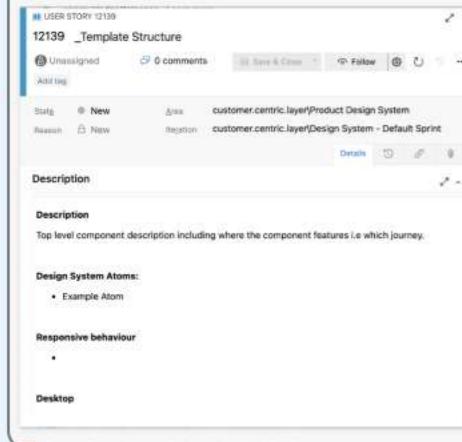
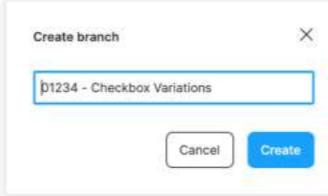
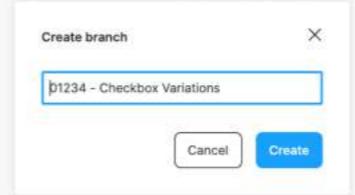
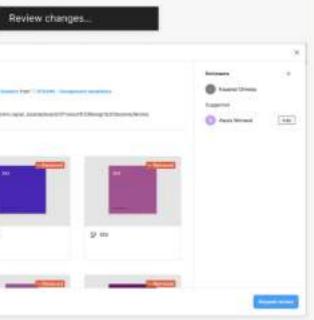
We needed to provide the most common components being used by the designers in their early prototypes of the product features. I decided to lay down a very basic UI kit for the designers to start mocking up designs using the basic components keeping in mind that these components will evolve to accommodate a proper built incorporating all the defined principles but with minimal breaking changes.

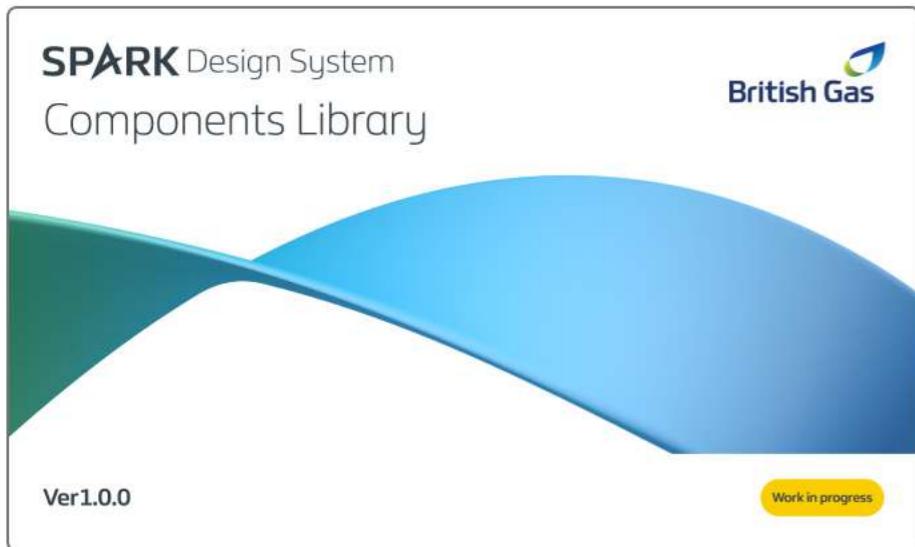
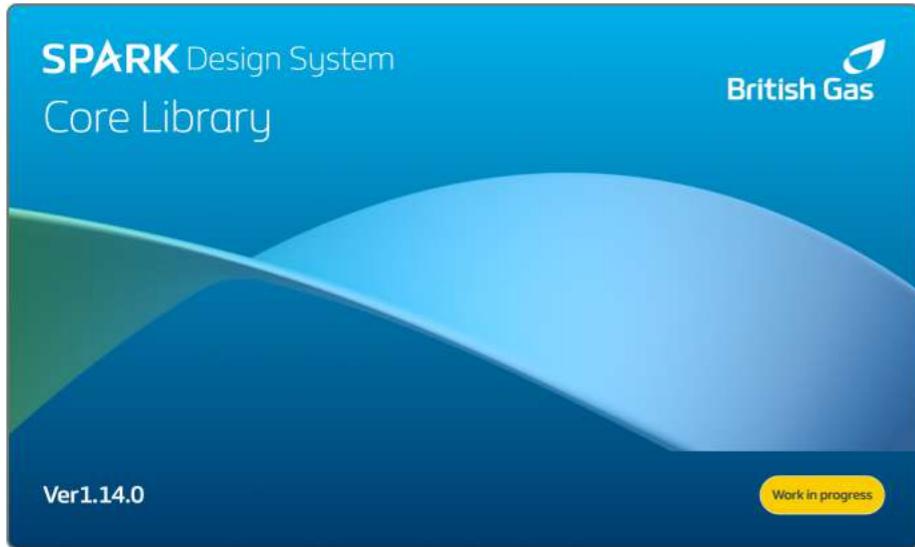
The atomic elements like colour, typography and iconography were quickly created using brand guidelines to start with but with the intention that these would be modified after due research and testing giving us solid design decisions to implement.



Process setup

Creation of design system needs a workflow process and a backlog of tasks to prioritise and focus on. I worked with the devs to setup a Kanban board and created documentation for the team to help them create the tickets and progress those through the process.

1. Define	2. Design	3. Review & Sign Off	4. Prod ready & Documentation	5. Approve & Merge
<p>Define your design need/design need. Use the ADO board to create a ticket but before this have a quick glance through the backlog in the "Define" column of the board and see if you can find a similar ticket just to avoid duplication of tickets and tasks.</p> <p>If you can't find a similar ticket then proceed to add a "New item" in the backlog and give it a meaningful title which captures the intent of the design work.</p> <p>Fill the ticket with ample description, add screenshots as references where applicable.</p> <p>Assign tags so your ticket can be categorised and searched by category easily. Please choose from the tags below:</p> <ul style="list-style-type: none">• Modification - For tasks in which the design is modified/enhanced/changed in terms of its visual or behaviour.• New - For tasks which propose a new concept/component altogether.• Bug - For visual or functional bugs in the existing design/components. <p>Note: If the bug ticket is only related to figma design and does not have a dev impact, please add a tag called 'Figma' to indicate its a figma only bug.</p> <p>Try to leave it in a state where there is enough detail enclosed for a designer to pick up and understand easily.</p> <p>NOTE: There is a template ticket in the board which you can use as an example on what details to include for the work item.</p> <p>https://dev.azure.com/mcas/ms/plab-platforms/customer.centric.layer/_boards/board/t/Product%20Design%20System/Stories/?workitem=12139</p> <p>Not everything in the description template is mandatory, please fill the details where applicable.</p> 	<p>Pick up the ticket from the "Define" column of ADO board and move it to the "Design" column. Take the ownership of the ticket by assigning the ticket to yourself.</p> <p>Check the Design System and see if there is an existing component there which is similar to what you need and if the existing component can be changed to meet your new requirements without compromising on the existing behavior?</p> <p>If yes then go inside Figma "Components Library" file for the Design System and create a new branch. Give the ticket number and ticket title as Branch's name and then proceed to propose changes to the existing component.</p> 	<p>Once the design proposal is ready it can be forwarded to the Review and Sign off phase.</p> <p>Move the ADO ticket to "Review & Sign off" column.</p> <p>Provide comments in the ticket and include figma link for the prototype</p> <p>Run the prototype through design review/critique and collect feedback.</p> <p>If you need to make changes to the design proposal after the feedback and critique, please move the ticket back to the "Design" phase and carry out the changes.</p> <p>Repeat with the review after each iteration until the design proposal is signed off.</p> <p>As the ticket is moved between the columns, please provide comments in the ticket clearly stating the reason of move.</p> 	<p>The signed off design can now be prepared to be merged in the main production file. Please move your ticket to "Prod ready & Documentation" column in ADO while it's still assigned to you.</p> <p>If your design lives in the "In progress" file, you will need to move this to a branch in the production file.</p> <p>Open Figma "Components Library" file for the Design System and create a new branch. Give the ticket number and ticket title as Branch's name and copy your signed off design from In progress file and paste it to this new branch.</p> 	<p>When the design is ready to be merged in, please move ADO ticket to "Approve & Merge" column and assign it to @Kaushal Chheda and provide the relevant comments in the ticket including the link to the Prod ready branch.</p> <p>In figma, put your branch in Review state by selecting "Review changes" and give some description. Add dev lead (@Kaushal Chheda) as a reviewer for a dev feasibility review.</p>  <p>Review changes...</p> <p>After the dev review, the ticket will then be assigned to the design system lead (@Awais Waheed) who will be responsible for design QA.</p> <p>Dev lead or design lead may leave some comments detailing issues/concerns with the design through figma comments tool. If this happens you will see your ticket moved back to "Prod ready and Documentation" column and the ticket assigned back to you. Please discuss with the dev through comments or direct communication and address the comments. You will need to resolve the comments before asking for another review.</p> <p>After design review if everything seems good with the design, it will be approved. Your design will then be merged in the Production file and changes will be published.</p>



Libraries setup

I aimed for a modular structure of the visual parts, and decided to split the whole UI Kit in 2 parts, foundational parts being in the core library while the core components being in the components library.

This has several advantages i.e.

- It makes a clear distinction on what are the foundational elements and what are tangible artefacts/components.
- Modularity allows mixing and matching of the libraries.
- It allows scalability, i.e. multi-brand theming.
- The core library can be easily swapped with another core library with different styles, designers can swap the library in their design files and this will reflect instantly.

Core Library

I focused on the foundational elements (atoms) first as these elements are base of almost all the components that will be constructed. These elements are Colour palette, Font & Typography scale, Iconography, Spacing, Layout and Elevation.

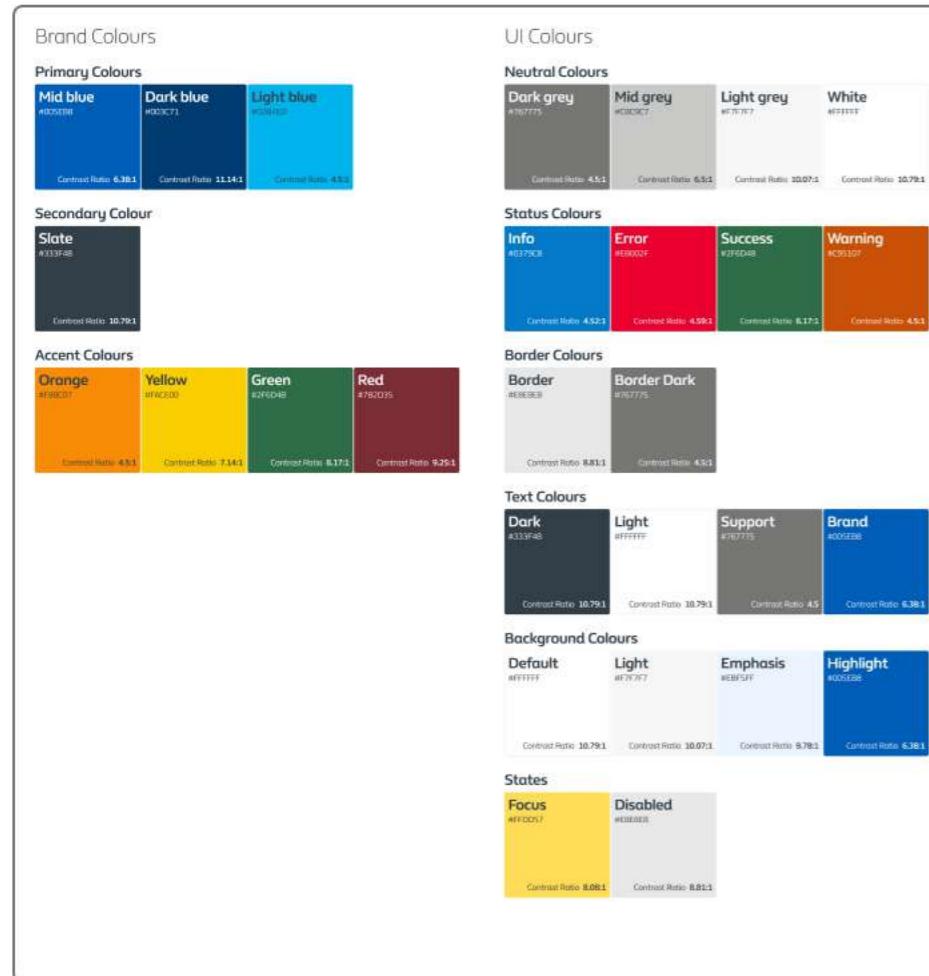
Brand Colours		
Primary Colours		
Mid blue #005EB8	Dark blue #003C71	Light blue #02B4ED
Contrast Ratio 6.38:1	Contrast Ratio 11.14:1	Contrast Ratio 4.5:1
Secondary Colour		
Slate #333F48		

Text	Weight	Size	Line height	Spacing
XL	Light	60px	70px	-1px
XL (Bold)	Bold	60px	70px	-1px

System Icons				
Arrow/Back	Arrow/Down	Arrow/Forward	Arrow/Up	Balance
Chevron/Left	Chat	Clear	Clock	Close

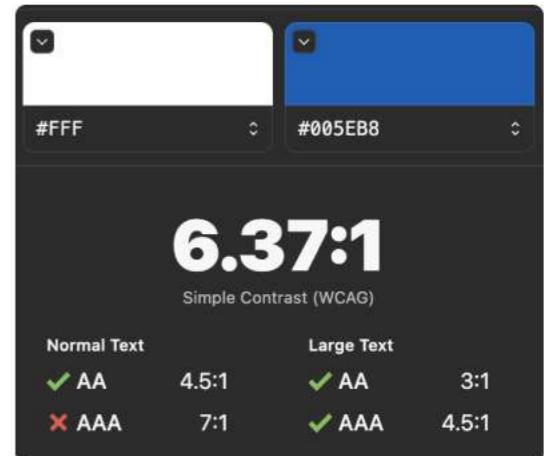
Colour palette

We wanted to define a basic yet scalable palette which had enough colours to fulfil UI needs but at the same time we didn't want to overload the palette. We worked with our primary brand colours, Cyan and Blue and a handful of supporting colours and then managed to create a palette in order which was minimal but had all the elements to express **hierarchy, state and brand identity**.



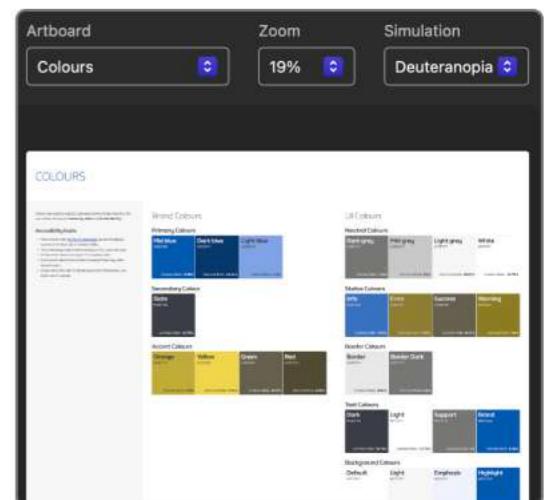
Colour contrast

The colour palette was based on the Brand colours but were modified slightly to be compliant with colour contrast standards. We strive to meet [WCAG AA standards](#) across this design system, including colour contrast ratios.



Colour blindness

We also ran our colour palette through colour blindness tests and made sure there's enough contrast for most common colour blinded conditions. Rules were set like colour alone should not convey a message, it should always be accompanied by another visual aid to help colour blind users perceive the message properly.



Colour roles and usage

Each colour's role was defined and baked into the style description so they are readily available for user reference in figma.

Name	Mid blue
Description	Default Primary Colour Primary interactive and accent colour, used in components like buttons and inputs

Color roles and usage

Primary Colours



Mid Blue

Primary interactive and accent colour, used in components like buttons and inputs



Dark Blue

Darker variation of Primary Colour. Used for Hover/Pressed states



Light Blue

Lighter Variation of Primary Colour, used only for elements that need to stand out

Secondary Colour



Slate

Used for elements that need a neutral dark treatment

Accent Colours

Font, font weight, base font size

We used the brand font called **BG Flame** which is a customised typeface to British Gas that has a clear, legible and elegant style.

We ran some comparison tests with different font weights applied to a few common components to determine which font weight/size works the best for desktop and mobile experiences.

Test: Round 1

The wireframe shows two mobile screens, A and B, comparing font weights for a tariff overview page. Both screens include a header, a sidebar with payment methods, and a main content area with a 'Continue' button. A central callout box at the bottom says 'No preference'.

Typography scale

We found that an optimised size for base font should be **18px** and the scale we opted for was **1.333 - Perfect Fourth**. which gave us the below typography styles.

Headline

We primarily use **bold** variation for headlines and subheadlines, while we use the regular variation for hero/display texts which bold applied to the key words in the sentence.

XL

Weight	Size	Line height	Spacing
Light	60px	70px	-1px

XL (Bold)

Weight	Size	Line height	Spacing
Bold	60px	70px	-1px

L

Weight	Size	Line height	Spacing
Light	44px	51px	-0.6px

L (Bold)

Weight	Size	Line height	Spacing
Bold	44px	51px	-0.6px

M

Weight	Size	Line height	Spacing
Bold	32px	37px	-0.5px

S

Weight	Size	Line height	Spacing
Bold	26px	32px	-0.4px

XS

Weight	Size	Line height	Spacing
Bold	22px	30px	-0.4px

XXS

Weight	Size	Line height	Spacing
Bold	20px	28px	-0.3px

Content

We primarily use **regular** variation for content styles while we use **bold** variation for links and the content(styled with M, S and XS) which needs an emphasis.

XL

Weight	Size	Line height	Spacing
Regular	32px	38px	-0.5px

XL (Link)

Weight	Size	Line height	Spacing
Bold	32px	38px	-0.5px

L

Weight	Size	Line height	Spacing
Regular	24px	32px	-0.4px

L (Link)

Weight	Size	Line height	Spacing
Bold	24px	32px	-0.4px

M

Weight	Size	Line height	Spacing
Regular	18px	27px	-0.3px

M (Bold)

Weight	Size	Line height	Spacing
Bold	18px	27px	-0.3px

M (Link)

Weight	Size	Line height	Spacing
Bold	18px	27px	-0.3px

S

Weight	Size	Line height	Spacing
Regular	16px	24px	-0.2px

Components

Call to action

M (Bold)

Weight	Size	Line height	Spacing
Bold	18px	24px	0px

Typography guides

Documentation on the usage of typography was compiled to help users understand how typography should be implemented in the designs across smaller as well as wider screens.

Typography styles

H1 Quick brown **fox** jumps over the lazy **dog**

H2 Quick brown **fox** jumps over the lazy **dog**

H3 Quick brown fox jumps over the lazy dog

H4 Quick brown fox jumps over the lazy dog

H5 Quick brown fox jumps over the lazy dog

H6 Quick brown fox jumps over the lazy dog

Note The H1-H6 structure is here for [guidance](#) only. Select the headline that works best in your context. The semantic markup might differ.

Body We'll keep an eye on your [consumption](#) and let you know if you are using more or less than what we quoted you for. This might mean we have to adjust your payments.

Small Body We'll keep an eye on your [consumption](#) and let you know if you are using more or less than what we quoted you for. This might mean we have to adjust your payments.

Sizing Guide

The base size we use is **18px** and the default text style is **Content/M**.

Below is the sizing usage, we use the same size for paragraph text across the devices while heading sizes change between mobile and tablet/desktop sizes.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

The rule for heading size is we use 1 style size lower on mobile view compared to desktop view, e.g. if we use **Headline/XL** for desktop, it will propagate to **Headline/M** for mobile.

Iconography

Icons symbolise ideas or actions and provide context. Icons should be used purposefully and sparingly. At best, they enhance usability, are easily remembered, and guide the users to where they need to go.

Since the brand design language is unique to the British Gas brand, we wanted to create custom icon style which would identify with the brand. We opted for **Outline** style with a default size of **24px**.

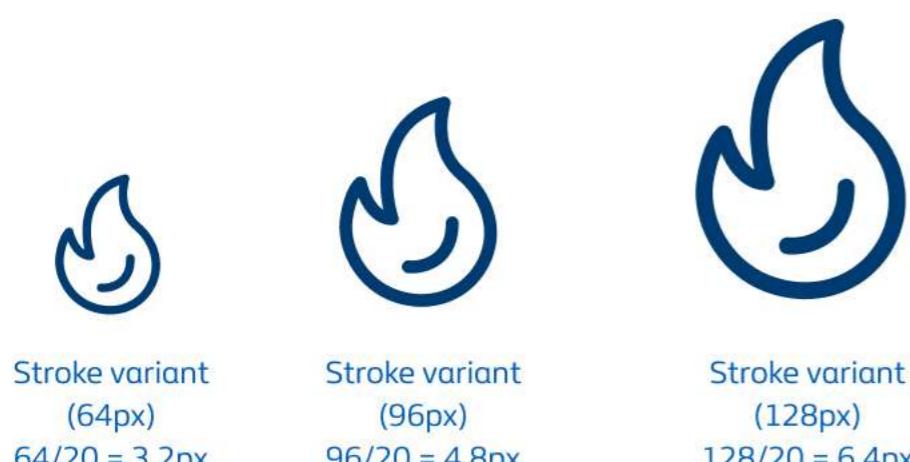
Default icon properties

Icon Style: Outline
Icon Size: 24px
Padding: 2px
Live area: 20px
Stroke width: 1.5px
Stroke alignment: Center
Stroke ends and joints: Round

System Icons							

Stroke width and Resizing

We realised that stroke width of **1.5px** worked well with default sized icon and resizes nicely for sizes of up to 40px but when the icons were resized to more than 40px width/height, the stroke would look too thick on them which effects the weight and visual balance. To tackle that, I provided a stroke variant for each icon on which the stroke width could be customised when the icon size gets too big.



A test was carried out to determine the ideal stroke width of the Icon but letting the designers choose the stroke width that they find balanced on large sized icons. I analysed the pattern and devised a formula which would give an ideal balanced stroke width.

For a balanced stroke weight of icons over 40px of size, we used the formula, **icon size/20**, and this gave us the stroke weight as per size of icon; e.g. 96px size of icon should have $96/20 = 4.8$ px stroke weight



Finally i documented a guide on the usage of the available icons and also on how to create new icons adhering the the set standards.

Usage

We suggest that you review your choice of icons against the icons used on Britishgas.co.uk (also referred to British Gas Classic), paying attention to their connotation and consistency of messaging.

Use icons to symbolize metaphors and actions. When using icons keep the following principles in mind:

- Icons are used to emphasize meaning, not as a replacement
- Adding an icon should be intentional
- Not everything needs an icon: consider a text label before adding an icon
- Keep icons at their aspect ratio

Icon Sizing

The default icon size is **24px** and they are scalable. We try to stick to the 8pt system on the size scale i.e. the icon should be of the size that fits within 8pt grid system.

There are 4 standard icon sizes (16px, 24px, 32px, 40px) which should be used in most cases, but if you are using icons outside of these sizes, scale the sizing based on 8pt grid.

Small (16px/16px)	Medium (24px/24px)	Large (32px/32px)	X-Large (40px/40px)
-------------------	--------------------	-------------------	---------------------

Sizing

✓ Do ✗ Don't

Always resize proportionally, i.e. icon should be the same width and height size Do not squash the icon

Icon Weight

Inside each icon, there are two variants, **Flat** and **Stroke**. The default variant is Flat while the Stroke variant is a hidden layer.

The weight of the icon is proportional to the size and it sits well with the standard sizing, but if you are using an icon that is above 40px in size then you may want to adjust the

How do I create an Icon?

Create custom icons based on the following guidelines to yield a consistent style. Each icon is designed to be simple, friendly and reduced to its minimal form, expressing essential characteristics.

Icon Frame and Bounds

All icons are built on a **24px/24px** frame. Icons sit inside this frame centrally aligned vertically and horizontally.

Note: There is a Layout grid style saved as "Icon" which is recommended to be applied on the frame to help better align the icon to the frame bounds.

Square Icon
Icons that have the same width and height usually have greater visual density and should be 18px/18px in dimensions

Circular Icon
Dimensions of circular icons should be 20px/20px

Horizontal Rectangular Icon
Horizontal rectangular icons should be 20px wide and scale to a proportional height

Vertical Rectangular Icon
Vertical rectangular icons should be 20px tall and scale to a proportional width

Spacing

Consistent spacing makes the UI look more cohesive and helps build relationships between different elements while bringing balance on the screen.

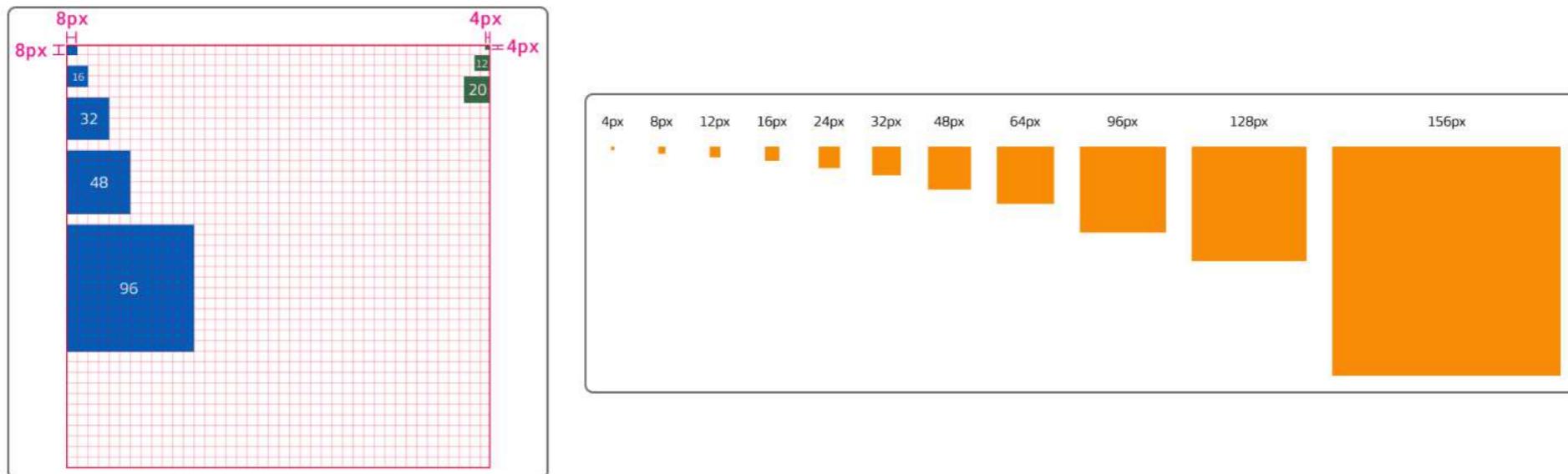
I analysed what a page layout is usually composed of and found out that all of the content can be classified as 3 elements which are;

Sections i.e. header, body, footer.

Content blocks, i.e., blocks of content containing components or visual items inside the sections.

Block elements i.e. elements/components inside the content blocks.

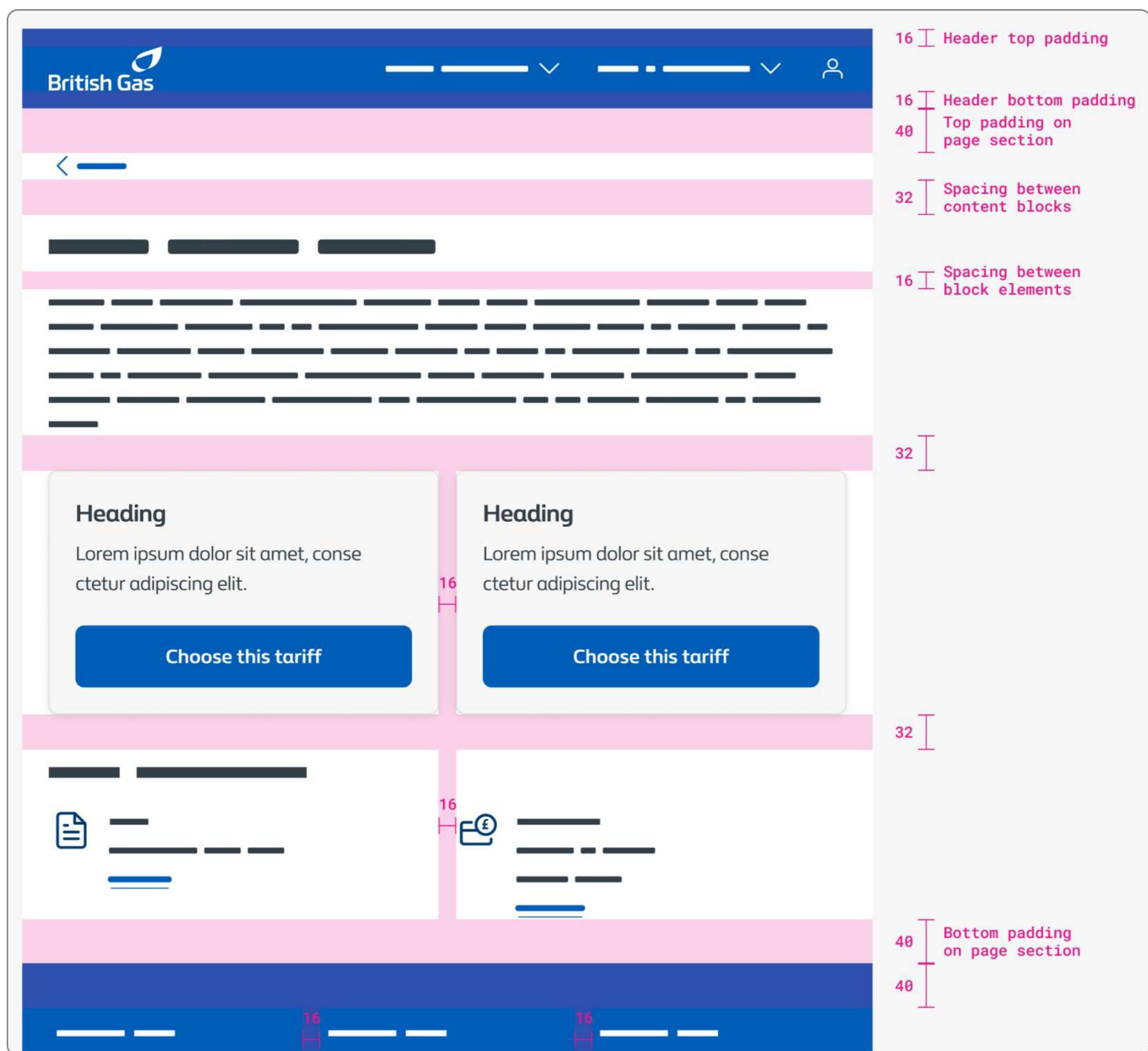
I then tried to standardise the spacing rules and using **8 point grid system**, defined spacing values that could be implemented among these 3 elements.



Section should maintain **40px** padding on top and bottom unless there's an exceptional use case.

Content blocks should maintain **32px** gap between them, unless there's an exceptional use case

Block elements can have a custom spacing based on the baseline grid.



Design Tokens

We wanted to place our design decision in a central repository and sync our design and dev tools based on those centralised design tokens. I used a figma plugin called `Tokens Studio` to generate the design tokens from our foundation styles and structured them in a semantic order. The idea was to make these core design decisions framework agnostic and with the power of design tokens, these design decisions could be parsed anywhere including Figma and storybook.

We used style dictionary to convert the json file generated by tokens studio and then converted that into css(sass) variables which was then being used in the storybook which is the dev counterpart of our design system libraries.

The image shows two screenshots of the Tokens Studio for Figma interface. The left screenshot displays a sidebar with categories like Primary, Secondary, Accents, and UI, each with color swatches. The right screenshot shows a detailed view of a specific token: Colour.Brand.Primary.Mid. It includes fields for Name (Colour.Brand.Primary.Mid), Color (#005eb8), Modify (PRO), and Description (Default Primary Colour, Primary interactive and accent colour, used in components like buttons and inputs). Below this is a JSON representation of the token:

```
{  
  "Colour": {  
    "Brand": {  
      "Primary": {  
        "Mid": {  
          "value": "#005eb8",  
          "type": "color",  
          "description": "Default Primary Colour\nPrimary interactive and accent colour, used in components like buttons and inputs"  
        },  
        "Dark": {  
          "value": "#003c71",  
          "type": "color",  
          "description": "Darker variation of Primary Colour\nUsed for Hover/Pressed states"  
        },  
        "Light": {  
          "value": "#02b4ed",  
          "type": "color",  
          "description": "Lighter Variation of Primary Colour\nUsed only for elements that need to stand out"  
        }  
      }  
    }  
}
```

Components Library

After putting the foundations together, we shifted our focus to the components library which was themed by the core library which we just finished.

I moved all the basic components from the initial library to the a new figma file where each component would live in its own page along with a detailed documentation on the component. I published the core library so the colours, typography, icons, layout, spacing and elevation styles could be consumed in this new components specific library.

The screenshot shows a Figma interface with a sidebar on the left and a main content area on the right.

Sidebar (Pages):

- Cover
- Start here
- Appointment picker
 - Banner
 - Button
 - Card
 - Content media
- Feature
- Form
- Divider
- Hero Banner
- Masters
 - Button
- Guidelines
- Usage
- Anatomy/Props
 - Header
 - Body
- Button

Main Content Area:

BUTTON

Buttons are used to initiate a particular action. Their labels describe what action will be performed when the user interacts with them and indicates what will be the next step.

Status: In Development

Version	Date	Ticket
P 1.2	29 Jun 22	276000 - BUG: Territory button alignment fix
P 1.1	14 Apr 22	271392 - Button Light variants
P 1.0	14 Apr 22	22185 - Button component: refactoring 22186 - Update loading state for buttons.

Primary button

- Use a primary button to indicate the most prominent action a customer would make on a screen. It should be a safe and, if possible, reversible action without much cost.
- Try to use only one primary button on a screen. Using multiple primary buttons might be distracting.
- Minimum width of primary button is 188px.
- On mobile and in some components like cards, it takes the full width of the container.
- The icon and the text are centered aligned.

Enabled	Hover/Pressed	Disabled	Focused	Loading
Label only	Call to action	Call to action	Call to action	Call to action
Icon & Label	→ Call to action			

Secondary button

- Secondary buttons are medium emphasis, performing less important actions than primary buttons.
- Use a secondary button to indicate an action that should be easy for a customer to make but isn't the most prominent on a screen.
- It should be a safe and, if possible, reversible action without much cost.
- Minimum width of primary button is 188px.
- On mobile and in some components like cards, it takes the full width of the container.
- The icon and the text are centered aligned.

Enabled	Hover/Pressed	Disabled	Focused	Loading
Label only	Call to action	Call to action	Call to action	Call to action
Icon & Label	→ Call to action			

Light button

- Light buttons are used only for the purpose of call to actions that are needed on a dark background.
- Minimum width of regular size light button is 188px.
- The icon and the text are centered aligned.

Enabled	Hover/Pressed	Disabled	Focused	Loading
Label only	Call to action	Call to action	Call to action	Call to action
Icon & Label	→ Call to action			
Small	Call to action	Call to action	Call to action	Call to action

Versioning

We maintained the version of the libraries through a version log. The evolution of the library could be tracked through that version log. The log contained the link to the ticket change where more details of the change could be found and also was linked to the relative branch in figma.

We decided to keep a global change log for the Core library, while in the components library, each component change log was maintained separately. This is because foundation library once developed would usually go through rare changes and can have a global change log rather than individual change log for each element, while components may undergo a more frequent change so a separate change log for each component makes more sense.

Version	Date	Updated component	Description
P 1.14.0	11 Aug' 22	Iconography	32001 : New icon - External link
P 1.13.0	25 May' 22	Start here page	Defined intro, vision and principles
P 1.12.2	13 May' 22	Colours	24895 : Status colours accessibility fix
P 1.12.1	12 May' 22	Elevation	24724 : Remove border styles from core library
P 1.12.0	19 Apr' 22	Iconography	23679 : Upcoming icon
P 1.11.0	19 Apr' 22	Iconography	23522 : New icons - info, warning
P 1.10.0	19 Apr' 22	Version log	22811 : Version log for core library
P 1.9.0	14 Apr' 22	Layout, Spacing	22742 : Layout and Spacing examples in core library

Core library change log and library version

Version	Date	Ticket
P 1.4	29 Jun 22	27599 : Spacing tweaks in quote card
P 1.3	29 Jun 22	27600 : Added pattern, Card with tertiary button
P 1.2	24 May' 22	25769 : Quote card header subtext removal
P 1.1	13 May' 22	24925 : Non-overriding Unlinked components update
P 1.0	04 May' 22	22195 : Card component refactoring

Card Component versioning/change log

Accessibility

Accessibility ensures that all persons can access and benefit from a website, system or application, regardless of their gender, age, culture, or ability. Our goal was to ensure Level AA conformance. Within Spark, this meant that any new component needs to meet related criteria.

The four accessibility principles (**POUR**) are derived straight from WCAG 2.1. We tested our components for each of these principles which are;

- **Perceivable** - Users must be able to perceive that information is available.
- **Operable** - Users must be able to operate the interface.
- **Understandable** - Users must be able to understand both the information and interface provided.
- **Robust** - Users relying on user agents, including assistive technologies, must be served with compatible content.

As some components were contributed by the community of in house designers, we made sure that enough guidance and education is documented to help them understand how to implement these accessibility principles in the proposed designs.

Accessibility guidance

Spark cares deeply about inclusivity. To help everyone, we aim to understand the impact each and every decision we make.

Accessibility is a key foundation of Spark. We've made sure our components integrate [WCAG 2.1 Level AA practices](#).

The following guidance has been provided by our accessibility partner Digital Accessibility Centre (DAC).

Colour

A high level of colour contrast assists people with sight loss, including colour blindness, as well as people with cognitive disabilities.

We follow two primary rules of accessibility related to colour:

- We never use colour as the only means of providing information or requesting an action.
- The combinations of text and their background colours do not fall below the [WCAG recommended threshold](#) ratio of 4.5:1 for standard or small text and 3:1 for larger text.

There are a number of tools that we've listed in our Accessibility Tools section which may help.

Forms

- All form fields are labelled or, where it is not possible to label a form, make sure the layer has an appropriate title.
- Where errors occur on a form, the user is informed as to which fields need attention and why.
- Provide informative error messages and a means of navigating away from an error message back to useful information.

Types of barriers

Inclusivity and accessibility are fundamental. No one should be left out. Ever.

Physical barriers

When designing experiences for users who have physical disabilities which impact their mobility, it is important to acknowledge the following:

- Provide shortcuts.
- Make large clickable actions.
- Give form fields space.
- Design with mobile and touch screen in mind.
- Design for keyboard or speech only use.

Cognitive development

When designing experiences for users who have cognitive disabilities, it is important to acknowledge the following:

- Reading age may be lower.
- Busy page layout can be confusing (Keep it simple).
- Ambiguous labels and icons can be confusing.
- Users may be easily distracted by moving animation.
- Short term memory may be affected.

Hearing loss

Accessibility tools & sources

Tools

Colour contrast

- [Colour Contrast Analyser](#) - The Colour Contrast Analyser (CCA) helps you determine the legibility of text and the contrast of visual elements, such as graphical controls and visual indicators.
- [Colorable](#) - A quick web based tool to check colour contrast.
- [Contrast](#) (Figma plugin) - Contrast makes it easy to check the contrast ratios of colors as you work.

Colour blindness

- [Sim Daltonism](#) - Sim Daltonism lets you visualise colours as they are perceived with various types of colour blindness. Use the camera on your iOS device, or use the Mac app to filter a region of the screen. Sim Daltonism is open source.
- [Stark](#) (Figma plugin) - Stark is a figma plugin which has a range of accessibility tools including vision simulator.

Using another tool?

If you use any other tools that you think we should include, please get in contact with the [Spark Design System Team](#).

Sources

Here are some interesting sources you can explore:

- [Inclusive Design Principles](#)
- [Microsoft Inclusive Manual](#)
- [2 thing every designer needs to know about accessibility](#)

Managing breaking changes

Our aim was always to update the components without any breaking changes. We try to update the component in a way that it would not lose the link to the design files and the overrides would stay intact. This process is usually straightforward, we make the new changes to the component, test the new changes, if there aren't any breaking changes in the component, we will simply document the change and then publish it. The users will then get a system notification indicating that there is a new update for the component that they are using in their design file. With the click of a button, the update will then apply to all the instances of the updated component in the file.

But sometimes the breaking changes are inevitable. This will happen when there is a major change in the component structure and layer names etc. To tackle that we devised a process which involved educating the users on breaking changes and how to handle those.

- **New component** - A new copy of the same component was put in place instead of updates applied to the old one as it would result in loss of overrides if the component was to be updated through Figma's automatic updates notification.
- **Deprecated component** - The old component was labelled as deprecated and deprecation indication was made clear through the component name. This update gets published along with the new component.
- **Manual replace** - The designers were asked to manually replace the old component with the new one where applicable and then do the overrides manually.

Component with breaking changes: Quote card

Deprecated Quote card

The screenshot shows a component named "The Green One" with a price of £91.42/month. It lists eight bullet points under "Bullets of key value for user". Below the list is a "Tariff detail" section with a "Choose this tariff" button. At the bottom is an optional promotional message section with a gift icon. On the right, a sidebar indicates the component is deprecated, showing a "Cards" list item with a "DEPRECATED: Quote Card" badge. A tooltip says "This component is now deprecated, ... Show more". A note below the sidebar says "This component is now deprecated, please use the new component (Quote card) from the library and manually override details."

New replacement for quote card

The screenshot shows a component named "The Green One" with a price of £86.94/month. It includes details like "Gas & Electricity", "Payment date: 20th", "Sort code: 20-20-20", and "Account number: 1234 5678". A note says "Your first payment will be 20/10/20". Below is a "Choose this tariff" button and an optional promotional message section with a gift icon. On the right, a sidebar shows a "Cards" list item with a "Quote Card" badge, a "Quote Row" badge, and a "Quote Card" tooltip. A note below the sidebar says "Used for marketing messages within the Direct Sales Journey." Another note at the bottom says "The promotional banner is displayed conditionally."

Governance and Roll out

I published the 2 libraries as soon as they were created with migrated elements from the initial library. This was done without breaking any links to the original styles and components so the designers still had all the library links intact and the initial library was marked as deprecated. The deprecated library was switched off from the admin settings in Figma's Organisation workspace and the new Core and Components libraries were switched on so the users will have these libraries available in their figma files by default.

Documentation of changes was placed in the library to help users understand what to expect and how to get on board with the changes.

The first proper release was the Core library and it was published in one go with all its elements updated, however the components library was published progressively whenever each component was refactored from its old state to the new version.

All of these changes were then announced in the MS Team's **Spark design system channel** and also vocalised in the Design team meetings update.

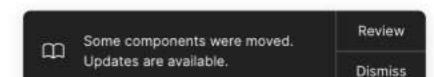
The design system is changing

There are number of changes planned for the design system to make it more robust and optimised. Some of these changes are under the hood changes to aid structural organisation of our design system. The original design system in production is replaced by 2 libraries which are;

- **Core Library** — This library contains all the details on the atomic level i.e. colour styles, typography styles, grid and spacing details etc.
- **Component Library** — This library contains all reusable components that can be consumed in the design files.

All files in the British Gas Organisation in Figma have been linked to the components and styles from these new libraries but you may have to do some changes in your "Drafts" files in order to relink to the latest components and styles.

1. You may get a prompt saying "Some components were moved". Review and then select "**Update all**". This will relink most of the components from the old library to the new one.



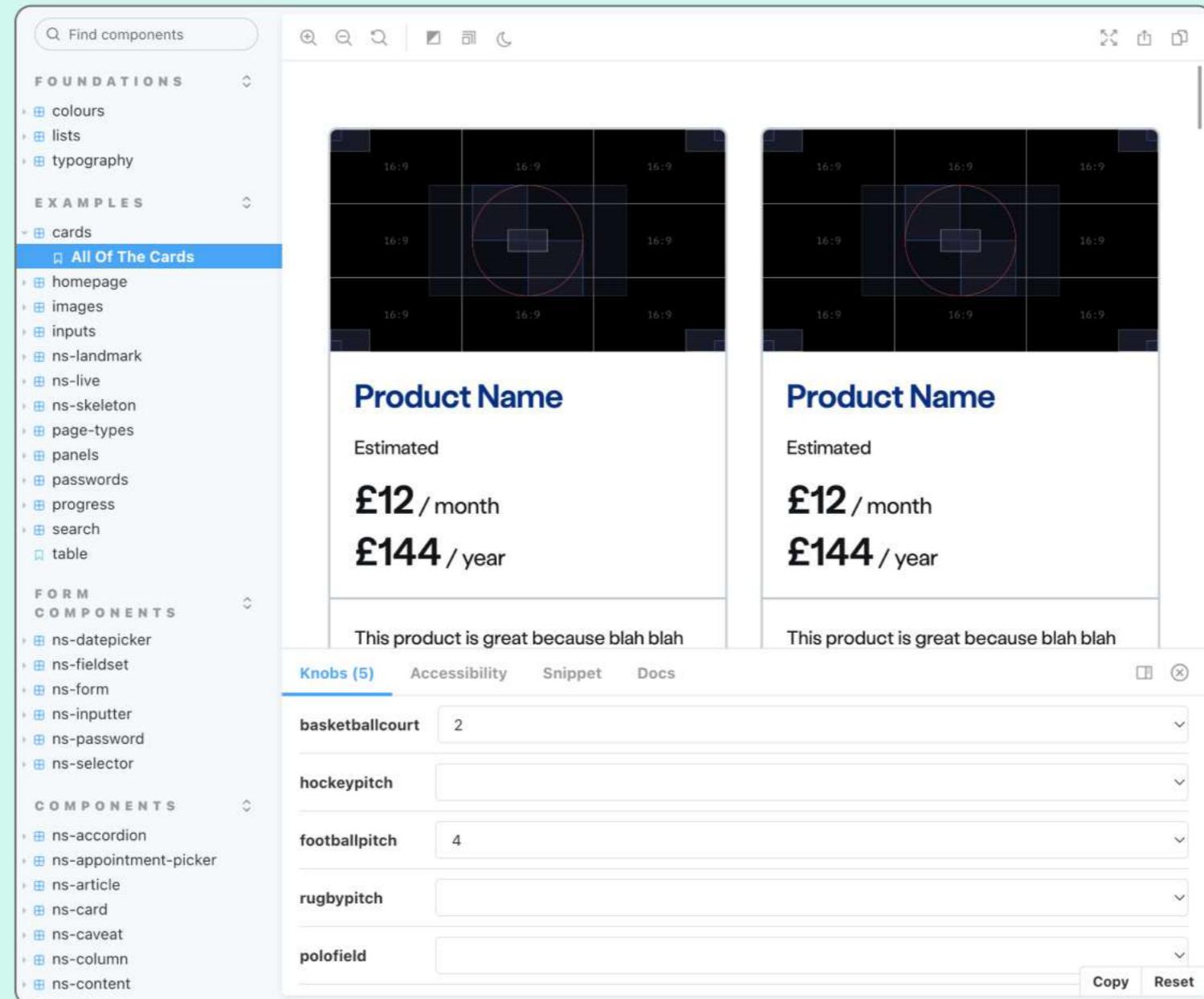
2. You may want to toggle the "**Core Library**" and "**Components Library**" **ON** and toggle the "**Design System**" library **OFF** just so you source your styles and components from the new libraries rather than the old one. Press **option/alt+3** to bring up the libraries panel and do the toggles on there.

Note: The Core Library and Components Library will be switched on by default in your new files.

With each roll out, we wanted to onboard all design teams consuming our design system, so I put a weekly drop in meeting in place where we would demonstrate the changes and run the audience through the usage, configuration and related documentation on the update. The recorded session was then placed in the training archive for Spark design system.

Dev build (Storybook)

It was important to sync the state of design system with the dev build. We chose **Storybook.js** as a tool for coded components repository and tried to follow the same principles and structure as we employed in Figma in terms of asset management and configuration.

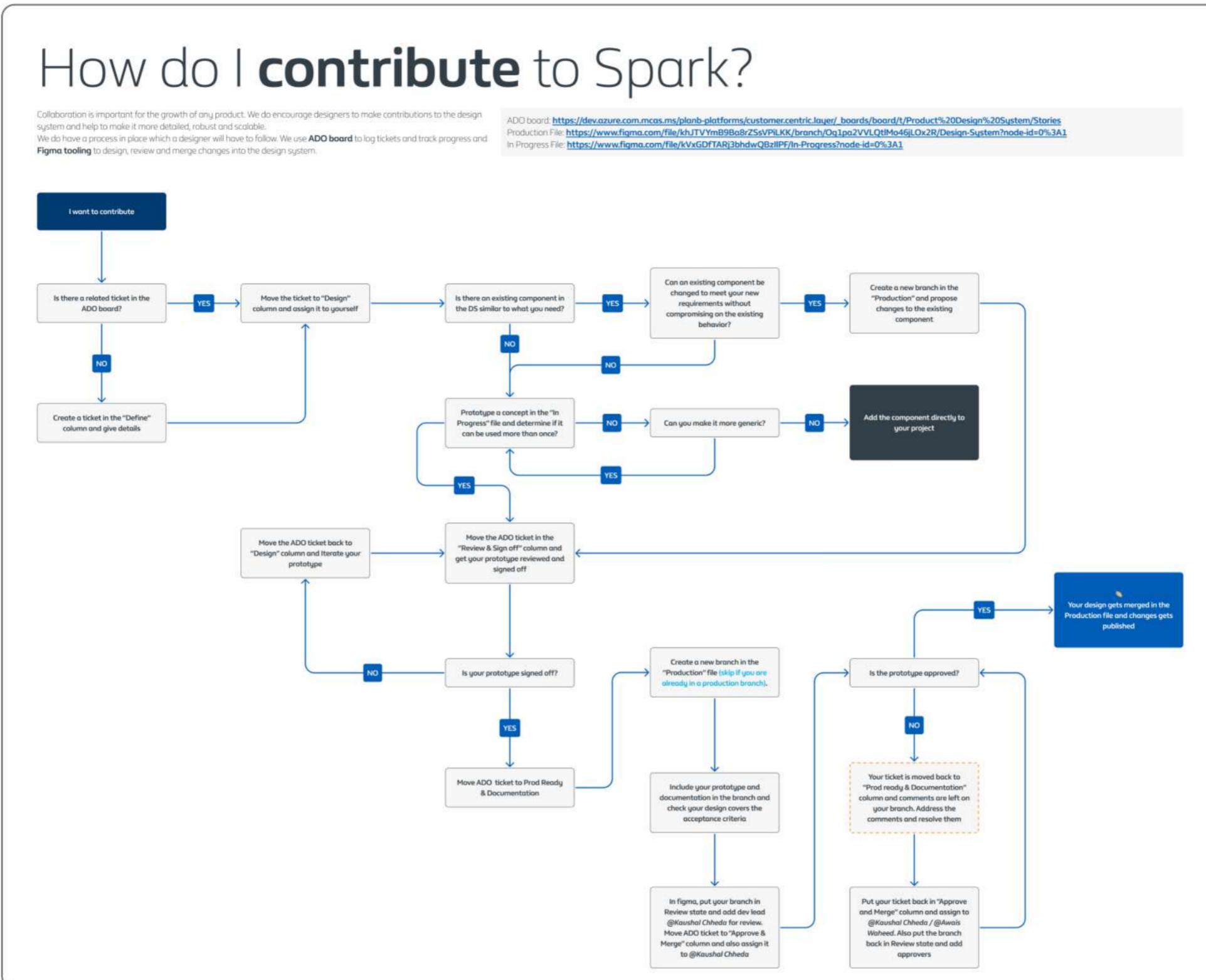


I worked closely with the dev lead to ensure that the coded components are in sync with the designed components. As this was a design led product, all the information on design tokens, behaviour of the components and guidelines were handed off to the devs to implement the same experience on storybook components.

Community contribution

British Gas has a huge volume of in house talented designers and developers whose contributions to design system could be valuable as well as important to speed up the process of updates and at the same time get them familiarised with the current state of Spark.

To bring the designers on board with how to contribute to the design system, we put a process in place which enabled the designers to bring their explorations to the table and use the Design system space to be vocal about their design needs and then participate to make those valuable addition to the design system.



Tutorials and workshops were put in place to train the resources for contributing ideas, bug fixes and changes to the design system. We managed the design system proposals in branches and would then review and sign off the branches if the proposed designs were in keeping with the acceptance criteria.

We held 2 times a week component review/sign-off sessions to enable the contributors share their needs for a component or change in a component and have an open discussion on how we can solve the problem either with the existing set of components or exploring a new component behaviour as a solution.

We were really proud of the contributions made by the community that really helped a small design system team to achieve a mature design system state.

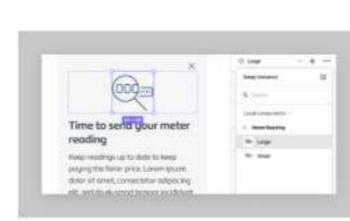
Making your components **prod ready** for inclusion in the Design System

Once your component has been all signed off, you need to prepare it for production. The steps below cover our **acceptance criteria**.



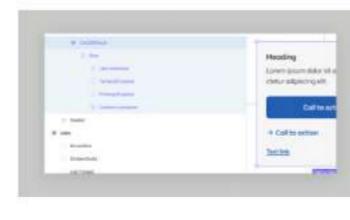
#1: use auto layout

We use auto layout in our components where possible. This allows us to change content and the layout will adapt accordingly. Less resizing for us! Use auto layout when your layout of the component consists of elements that are stacked in a row or a column, but for the elements which overlap each other, just use a normal frame/group.



#2: all linked up

Make sure the components and design foundations in your components are all linked to the design system. Such as, the icons, typography and colour.



#3: name the layers

Name your frames and sub components. Typically, we call the frames that contain content "containers". We use auto layout to apply spacing to them and their sub components.



#4: name your components

Next steps

The journey of Spark to maturity has been whirl wind with many challenging situations and insights. We did appreciate all the things we learned off it and are extremely proud to bring the design system to a state where the processes were running in a automated way i.e. the design teams had access to all the stuff on where to find the needful things in the design system and what to do in order to use or contribute to Spark.

Spark by no means is a finished product, it's an ever evolving product and there are a few next steps which we were keen to start on to bring more standardisation, simplicity and effectiveness.

Some of our team's next steps include:

- **Usage audit**

Carry out an audit on the user behaviour, how users are using the design system elements, are they comfortable with configuring the components easily? Are they detaching the components. These insights will help improve the quality of life.

- **Surveys**

Carry out user surveys, gather user inputs and synthesise results. This will give us quantitative results on how the design system is being received by the users.

- **Retros**

Engage stakeholders and teams in retrospective exercises to find out what is working and what can be improved in the system and processes.

- **Advanced Design tokens**

Refine tokens and apply them throughout the components. Work with dev on creating component level tokens.

- **Simplify processes**

Refine involved processes i.e. usage and contribution. Revisit the touch points which are causing delays and simplify them.