Interactive Prototyping

Team 12: Afzal Miah, Ryan Hutton, Yo Han Jun, Naqash Nadeem, Andrew Hart

Atomic Design

The team used atomic design methodology for the generation of the interactive prototype. The team used the University of Dundee pattern library[1] as the inspiration for the design patterns used in the prototypes and mainly used the same patterns with little variation.

Atoms in the atomic design principle are the basic building blocks of the design. The team used atoms in the design not only for the interfaces such as inputs or buttons but also in the more abstract sense for colour palettes, fonts, and animations.

Global Atoms

The team broke up the design of the prototype into sections and then further applied the atomic design principles in development. Some things however were global across the entire design, so these atoms were to be set first. These were atoms made using the pattern library as referenced earlier, so mainly things pertaining to the overall design patterns to be used for the prototype. Starting with the colours, the colours used shades of greys, blues, white and red as atoms (figure 1).

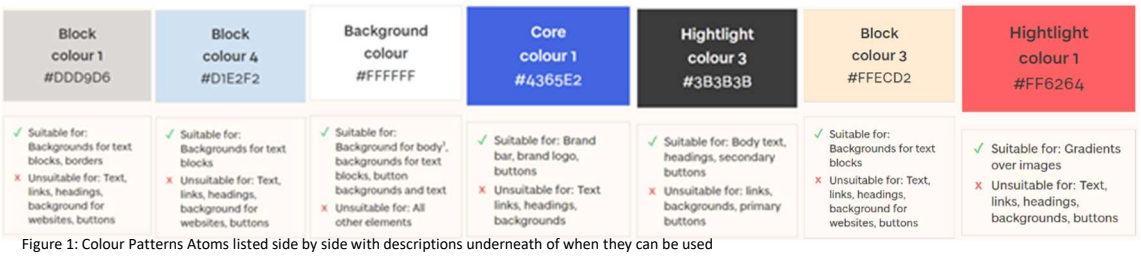


Figure 1: Colour Patterns Atoms listed side by side with descriptions underneath of when they can be used

For the font atom the team used the Dundee University font stack with 3 variations for various usages (figure 2). The team opted to use the Arial variations for the prototype due to it already being installed in the XD font library.

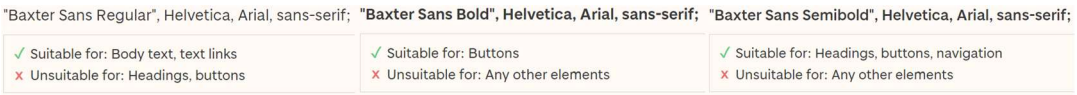


Figure 2: Font Stack Patterns Atoms side by side with descriptions of when they can be used

Homebar(Organism)

The homebar is an example of one of the components which were made using the atomic design principles. The homebar is an organism used across the prototype which used the atomic design principles. Starting with its atoms, the homebar was made using 2 molecules with a total of 5 atoms (figure 3). With the Logo and background as the atoms in one molecule, and the 3 tabs as the atoms in the other molecule, these were combined in order to create the homebar organism. In XD the interactions can also be set for organisms so the "links" for the interactions for the 3 tabs and the logo were also setup in the organism(on click interactions where on-click logo takes to homepage, buildings to buildings page and tours to tour page) before being saved as an organism using XDs save as component feature which allows for atoms/molecules/organisms/templates to be saved as

[1] [1] Dundee.ac.uk. 2020. *Colour Palette | Brand | University Of Dundee*. [online] Available at: <https://www.dundee.ac.uk/brand/design-guidelines/colour/> [Accessed 11 December 2020].

components allowing them to be instanced anywhere in the prototype as a design with also the interactions included.

Figure 3: Homebar Organism in its entirety. The logo and background are one molecule with the logo and background being atoms. The 3 tabs Buildings, Schools and Tours are another molecule with the 3 tabs being atoms.

Buttons(Molecule)

Figure 4: Content bar buttons example. Button on the top represents the default state, when hovered over by a user the button would switch states and its appearance to the button on the bottom.

The button molecule was made using 2 atoms, the text, and the box. The text was to be placed on the box to create the button (figure 4). The buttons however had 2 states, default and when hovered over. Using XDs state feature which allows you to give complexity to a molecule, the button organism was made to have 2 states saved, when hovered and default which in the prototype the molecule would change to accordingly. With the buttons being saved as a molecule using XDs component feature, the buttons could be instanced wherever need also allowing for any slight changes for each instance when needed however they would keep the same design. With this being the default molecule design to work as the template for all buttons in the prototype, further buttons made in the prototype followed and used the same design choices and the same concept of states.

Contents Bar (Organism)

The contents bar organism was made using buttons molecules and combining them together with a new contents frame molecule to create the organism. Buttons could be added/removed to the organism as required for any instance and the text changed and centred. For each type of contents bar, the bar is saved with also its interactions being linked allowing for the organisms to be copy-pasted where needed without needing to re-do the interaction for each instance.

Figure 5: On the left a fully populated contents bar example, on the right a less populated example

Tabs (Organisms)

The tabs organism was made using instances of the button molecules but editing the instances and their states to fit the function of working as tabs. Another state was added to the buttons for the tab's organism where now a selected tab will keep the same colours/design as it would when hovered over to indicate to the user what tab is currently selected instead of reverting back to the default state when selected. The tabs organism therefore is very modular so for any instance allows for adaptability, allowing for there to be changes in the prototype within instance for example tabs can be added or removed simply by adding or deleting a button molecule to the organism.

Figure 6: Pictured on the left - tabs Organism with the selected state being "Campus" in and the remaining two in the default state. Pictured on the right – tabs organism making use of tabs organism by adapting to needed use case by changing contents and adding another button molecule

With the tab organism done, the organism was used together with a box to create how the tab would work for content in the prototype where the tabs would be connected to a screen which would show the needed content for that tab (figure 7). By default, the content box is empty. The tabs have been by default are the tabs for the tours pages but can be changed as needed as mentioned earlier.

Figure 7: Tab with content box organism.

Tours Template and Pages

Using the organisms created and stored using the components feature in XD, the layouts were made. This is the example for the template made for tours pages. By first creating a template for all the tours pages to follow, this made creating the pages themselves easy to prototype as all it required was changing placeholders/organisms where needed and adding organisms to fit the use case. Figure 8 pictures the transition from how one template transitioned into two examples of page's prototypes doing the aforementioned.

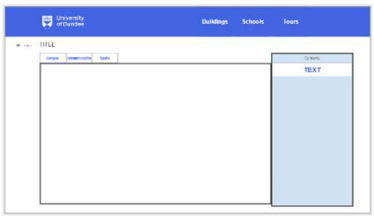


Figure 8: Pictured above is the template made to be used for the tour's pages.

Building hover states

The hover state of the buildings on the landing page is an organism made with the molecules being the drawn shape of the building, the rectangle used as a label and the text within the label to say what building it is. The molecules of the text labelling the building and the rectangle which holds the label work together to create an atom to display the name of the building to the user. However, one single organism displaying the hover state could not be reused because every building has a different shape. Also, there is a different name for each building so the molecules of each hover state had to be remade to suit the building which you can see in figure 9.

Figure 9: Pictured above is the result of what happens when you hover over a building on the interactive map on the landing page.

Adobe XD Elements and Features

Hover

One of the ways in which the hover feature was used was throughout the interactive map on the homepage. When you land on the homepage, you will find the interactive map which you use to select building to find information on them. When you hover over a building on campus it will light up red and display a text beside it labelling the building.

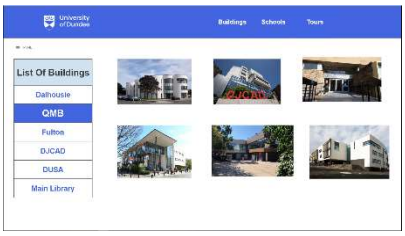
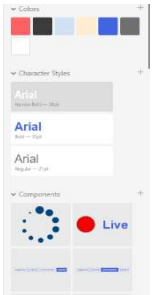


Figure 10: Highlighted option can be seen

Hover state was one the features of Adobe XD that was used throughout the design of the prototype. For example, on the buildings page the 'List of Buildings' table user can hoover over different options and it highlights and indicates the user which option they are on see figure 10. Asset Bar (Components and Colours)



The team utilised the components asset bar by storing the nav bar as a global component so that all the pages could use the same nav-bar with the same interactions which was more efficient. The team stored a colour set following the colour palette used for the University of Dundee website. These colours were consistently used across all our pages to avoid any

inconsistencies and to make life easier for the team, so they did not have to keep inserting the colour codes every time. Any other components/atoms/organisms/molecules made were added to the assets as required.

Transition

Transition was also one of the other features of XD that team made use of. Team used transition for the buildings tour pages, there is a 1.5s time delay from buffer video page to the video playing page. It was used to demonstrate a video playing as the team did not feel there was purpose in showing an actual video within our Adobe XD prototypes, but the use of transition feature it illustrates how it would like on an actual website.

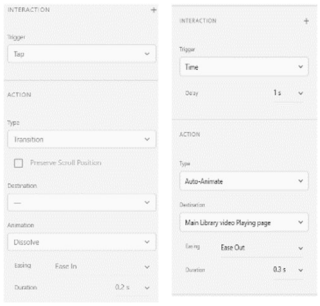


Figure 12: interaction setup menu in XD

States

The team utilised the states feature of Adobe XD on every interactable component across all the pages. When a component is created, a default state is also created for it. This default state allows for each component to have a consistent base design that it will revert to if no action is being performed on it. For several components, a hover state was also implemented, as mentioned above. This hover state was also kept consistent across every component that was selectable by the user. Having these two separate states allowed the team to have different design iterations for a component based on the users input, while ensuring that they did not have to create a separate element. It also ensures that the component reverts to its' original design when unselected.

Figure 13: State feature in XD

Overlay

The Team created named master navigation bar which is connected to every menu button. Menu button is located into left top side of slides. Overlays can be used for modal windows, drop-down menus, pop-up notifications, all without leaving current screen. Team utilized navigation bar as applying dissolved animation to pop up for entire slides that leads to connect all slides globally. we would like an option to go back to current slide once closing it and keeping the artboards current location.

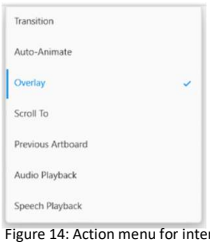


Figure 14: Action menu for interaction feature in XD with overlay selected

Drawing and Shapes

Throughout the landing page, shapes had to be drawn around the border of the buildings so that they can be used for the hover state and would highlight when hovered over. Beside these shapes of the drawn buildings, a label would also appear which is made up of text inside a rectangle.

Conclusion

Overall, the team utilized the atomic design principles, and the Adobe XD features. Atomic design really helped in making prototype robust and consistent whereas with the help of adobe XD features, it was easy and efficient to implement those principles. The main feature of adobe that was a big help was having a components section where the team made master components and then copied instances of those components across to the prototype which meant there were no inconsistencies in the design. The other feature of adobe XD was real time collaboration across the team which helped keep everything in accordance with the design rules the team had set. These features sped up the design process as the team did not had to keep recreating these components. Other main features like hover and ability to link components with different components and then having ability to control those transitions with the help of option of tap, action, destination, and animation.