CS4131
Mobile Application Development

Name: _____ ( ) Date: _____

Chapter 5: Elements of UX in App Design

5.1 What is UX

UX is the short form of user experience, and it focuses on the creation of products that provide a meaningful and relevant experience to users. UX involves the design of the entire process of acquiring and integrating the product, including aspects of branding, design and functionality.

While the notes will not cover the entire UX process, it contains the key UX design components to be able to build an application which is **user-friendly**, **user-centered**, and is **accessible** to a diversity of members, such as the visually / audio impaired, in your intended user audience.

5.2 UX Design Process

The UX design process typically follows a framework like the design thinking approach, which consists of five phases:

- Empathise with the users (learning about your app's audience)
- Define the problem (identifying the users' needs)
- Ideate (generate ideas for the design)
- Prototype (turning ideas into concrete examples)
- Test (evaluating the design)

5.3 User Research (Empathise and Define the Problem)

The key idea in the UX design approach is to think from the perspective of the user, what the user needs and requires out of the app you are creating. You will certainly not want to build an app which does not meet the needs of the user in the first place.

As you start designing your app, you need to determine **what is to be produced** and the **context for its existence**. Every app has a target audience it needs to cater to. Hence, the app designer typically tries to find out more about the required features and aspects of the app, through,

- **Stakeholder Surveys and Interviews**: Interviewing your key target audiences for your app or the industry members or audiences of those you are intending to target your app.
- **User and Market Research:** It will be good to find out more about your key target audience to determine any issues or needs which need to be addressed.
- **Competitive Research:** Your proposed app may not even be a completely new thing in the first place, and hence, this step is to identify aspects of your app which will attract the intended target audience.

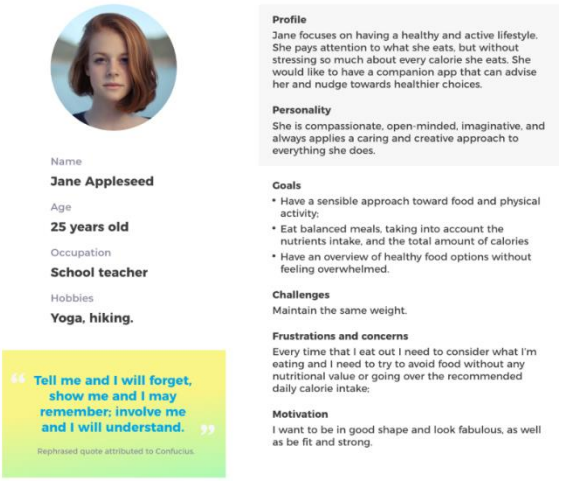
In a more professional setting, the stakeholder interviews also hold another purpose, which is to set goals, expectations, timeline, as well as success metrics for the app's development. In a professional setting, it is also imperative to manage client expectations to make the entire development process easier.

Empathising with the user through personas and user stories

After determining what your app is supposed to do and its context for existence, some members of the intended user audience will need to be interviewed and surveyed, to do the following:

- Understand more about the individual users (name, age, hobbies, career, hobbies)
- Understand some of the challenges and frustrations users face, and record how they feel
- Understand their goals and motivations to give context in what success criteria the user may want to see in your product.

Through **interviews and surveys** of **real people**, by observing behaviour patterns, or abilities / disabilities, designers usually develop **up to three "imaginary" users**, known as **personas**, which gives us both a **summary of ALL the findings**, and to **guide as to what different typical users**, under the broad classifications into the personas, **may require the app to do to solve a certain problem in his / her life**.



In essence, personas help the designer answer the question: **"Who am I designing for?".** An example of a persona can be seen in the right.

Note that the different individual personas generally can be classified under categories like age, level of ability with technology, physical ability or disability, and the current careers, goals and frustrations and the personas usually distinctly carry one of more of the mentioned attributes.

The crafting of the **user stories** will come in after, based on the personas created. The user stories answer the questions: **"Why am I creating this app?"** or **"What is the purpose of creating the app?"** The user story is a concise statement in the form of the following template:

User Story Template:

<Name of user from persona> **is a / an** *<type of user based on characteristics in persona>*. **He / She wants to** *<tasks / frustrations / challenges to be addressed>* **so that** *<the user's own success criteria for the app based on tasks / frustrations / challenges>* **Scenario A:** *<Contextual requirements the app should fulfill>* Given that *<some context that app is used>***, when** *<some action is done on the app>***, then** *<expected outcomes app must meet>* **Notes:** *<Any notes or findings you may want to add from research or persona characteriestics>* **Priority:** *<High / Medium / Low / Unprioritized>*

The user stories allow you to understand, in detail, what features your app needs to have and even the expected flow of the app so that you, as the designer, have a very clear idea of what the app needs.

5.4 Ideation

Ideation is the process of conceptualizing the app and determining the flow of the app based on the findings obtained from the user research step. The key outcome from the ideation step is to **determine the user features of the app**, in the context of **realizing the intended user success criteria and scenario outcomes**.

In a professional setting, app designers usually work as a team, and brainstorming or mind-mapping sessions are done to talk through all the user findings and to develop ideas. However, in a more individual setting, we skip the brainstorming, going directly into storyboarding.

A storyboard's purpose is to, based on the user research and user stories, categorise features, then subsequently draw the flow or sequence of the **user interaction of your app**. You can do the storyboarding featuring the actual user interaction with the app, OR highlight the app flow just through the screens of the app.

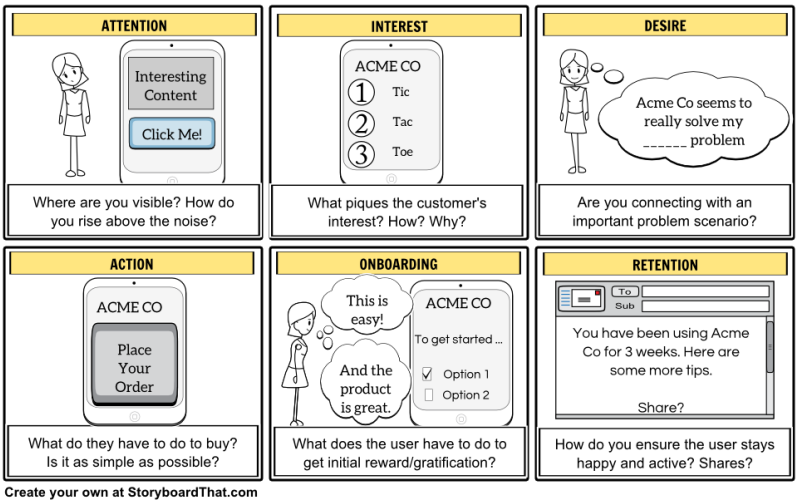


Figure 1: Storyboarding example featuring actual user interaction

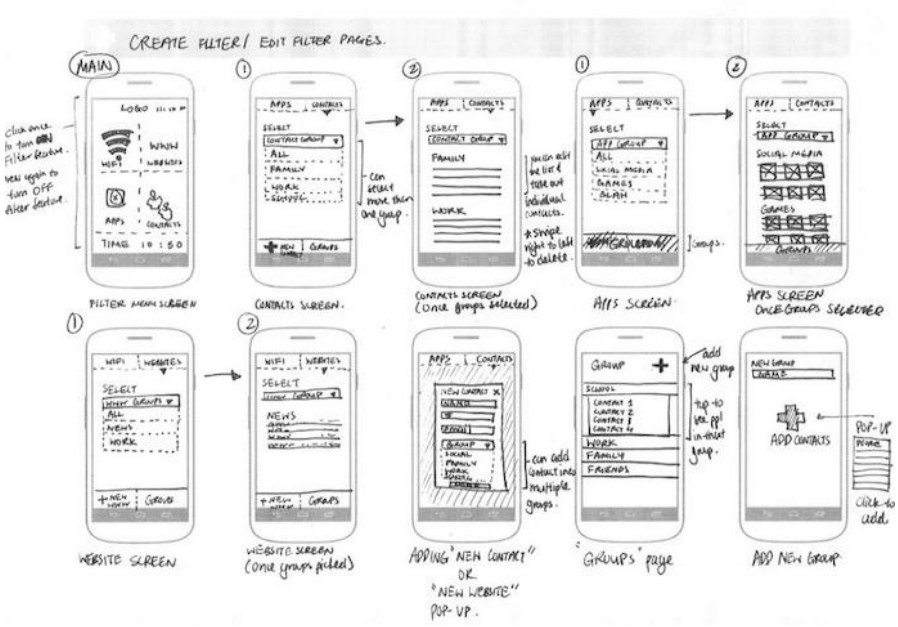


Figure 2: Storyboarding example featuring flow of draft screens of the app

5.5 Prototyping and Elements of Design

After the ideation phase and determining the app's features and user interaction details, we will now have to do prototyping of the app. The prototype is a primitive representation of an end-product that the front-end development team (the app designers) create during the design process. Prototypes have different "levels", otherwise known as the fidelity, which is how close the prototype is to the actual final product.

A low-fidelity (lo-fi) prototype, is typically a pen-and-paper sketch of the screens of the intended app. While it may just be a pen-and-paper sketch, the pen-and-paper sketches are expected to be very detailed (something like Figure 2 on the previous page), neat and accurate. Lo-fi prototypes are expected to contain:

- **near-to-scale UI elements relative to the size of your intended actual app,**
- **phone frames containing your app pages are all expected to have the exact same dimensions**,
- **navigation arrows going from the source page to the intended next page.**

An example of a lo-fi prototype can be seen below:

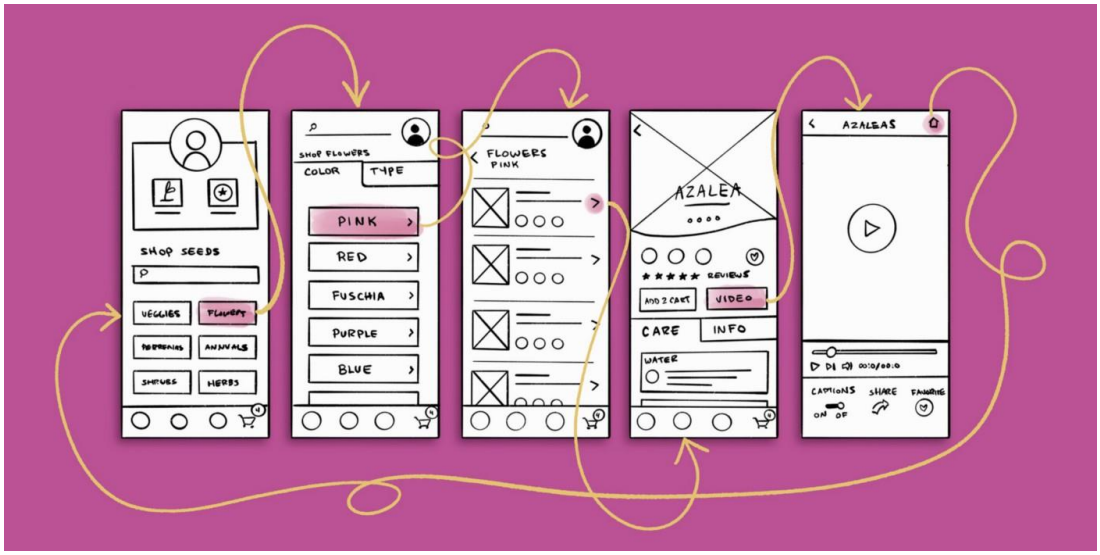


Figure 3: Example of Lo-Fi Prototype

A high-fidelity (hi-fi) prototype is essentially a digital representation of your app pages, which is usually **near or is the exact design of the app itself, with all interaction processes present**. This requires the designer to have an established understanding of the intended app's functionalities and navigation. As it is a very realistic representation of the intended app, hi-fi prototypes are normally used for **usability testing to solicit feedback** and for app designers to improve on their app design before it is even published (or beta-tested). The benefit of creating hi-fi prototypes is that it gives a very good idea of how the final app will look and feel. However, creating hi-fi is very time consuming and requires an established understanding of functionalities.

In the professional setting, software such as Figma or Adobe XD is used to create hi-fi prototypes which are clickable and will navigate to the exact intended pages when the respective UI elements are clicked on. They are usually also powerful enough to even include scrolling features in the prototypes.

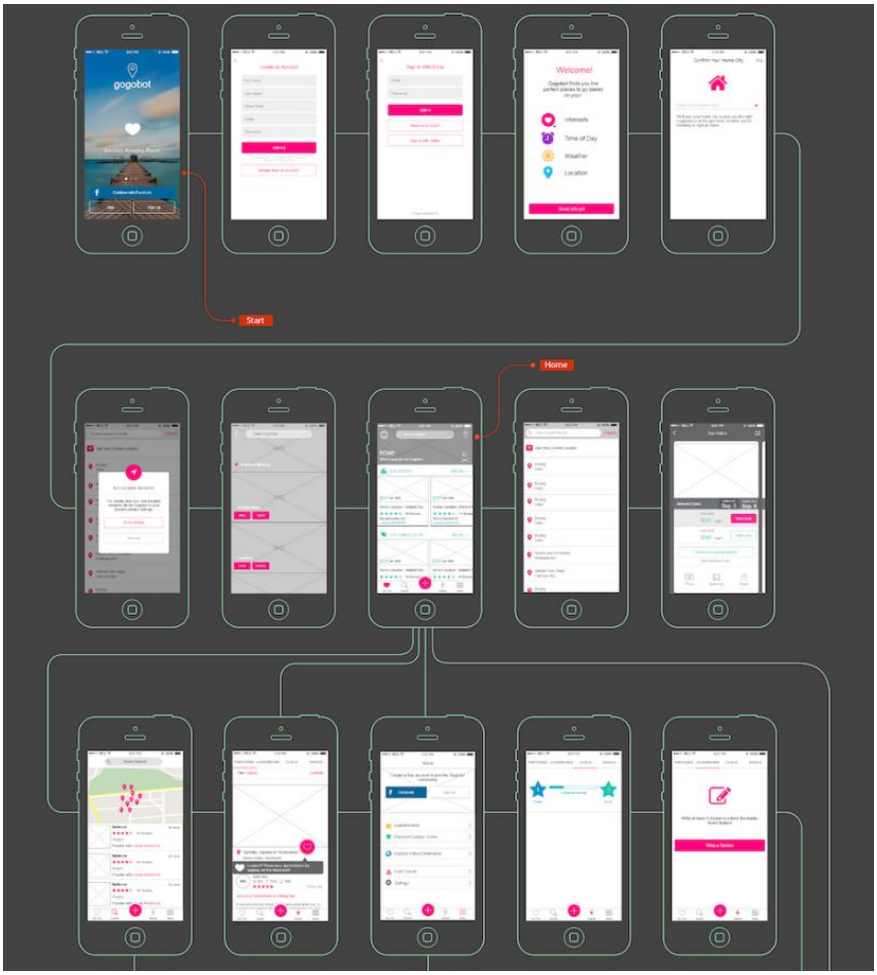


Figure 4: Example of Hi-Fi Prototype

While creating the prototype, many of the app's design features will need to be established and ironed out. Thankfully, there are guidelines present, specifically for Android apps, present to guide the design process for apps. We will look at Material Design in the next section.

5.6 Material Design for Android Apps

You may recall in Chapter 4 that UI components such as the FAB and the action icons / overflow menu item placement in the action bar. Material Design is a design system created by Google to help app designers build high-quality digital experiences in both apps and in websites. It covers guidelines on UI components, themes, colour, typography and even the use of shape styles.

Note that as there are many components of Material Design, this section will primarily take reference to and summarise the contents of: https://material.io/develop/android . It is strongly recommended that you find out more and read about it.

Elements of Colour

Colours are important in app development as it will impact any first impressions of your app and even impacts the usability of your app. Research has shown that most people preferred simple colour combinations that relied on only two or at most three colours, hence the notion of "less is

more". A combination of colours is known as a colour scheme, and the colour wheel can typically help in choosing these schemes. Colour schemes come in different categories:

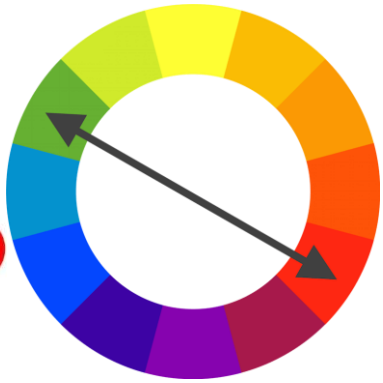
- Monochromatic schemes are simply different shades of the exact same colour and are usually very easy on the eyes, especially with **blue or green hues**

- Analogous colour schemes are created from related colours that do not stand out from one another; one colour is used as a dominant colour while others are used to **enrich** the scheme. Broadly, analogous schemes can be categorized as warm or cool schemes, each of which can bring about a different effect to your app. On the colour wheel, analogous colours are typically colours which are adjacent to each other.

  Warm colours signify importance and the sense of urgency. The warmer (or redder) the colour, the more important. They are usually also used in workout apps to signify sense of urgency or intensity.

  Cool colours bring about calm and are usually used for apps which are primarily for leisure or for a relaxing purpose like meditation or even socializing.

- Complementary colours on the colour wheel are colours which are opposite each other. They contrast strongly and can be used to attract the user's attention. When choosing complementary colours, it is important to choose a dominant colour and use its complementary colour for accents. For example, in an app with just many hues of green as its main colour scheme, red, its complementary colour, will stand out.



- Custom colour schemes are not that difficult to create. Adding a bright colour to a neutral colour scheme works and it is present in many apps today:

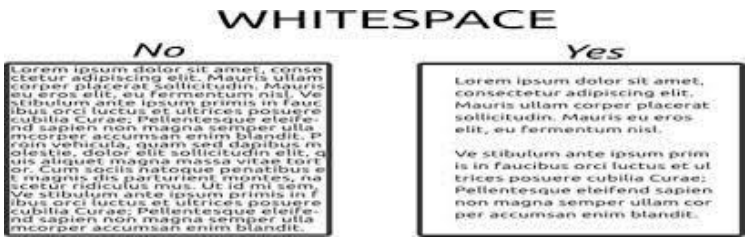
Contrast is also important as you will want your users to be able to see any text in your app, in particular users who may be hard of sight or are dyslexic. The general rule of thumb is to:

- Not have text which are nearly similar in colour or shade to the background
- Not have a colour combination such that you have bright text on a bright background or a dark text on a dark background, regardless of colour.

You may want to test the contrast ratios of your colours through the W3C success criterion, with some analysers available in the following link:
https://www.w3.org/TR/UNDERSTANDING-WCAG20/visual-audio-contrast-contrast.html

Figure 5: Good vs Bad Contrast of Colour

Within your app, what will make certain parts of your app stand out and even make the entire app more attractive will be the presence of **whitespace**. Whitespace may not necessarily be white, hence, more accurately, it is the negative space between layouts, UI elements or even lines of text.



Another point to note, since we are on the topic of user-experience, is to understand that there are users present who may have colour blindness, and may have difficulty or even inability perceiving certain colour combinations, like red-green, blue-yellow, or even monochromatic. **Hence, a rule will be to never solely rely on a colour to indicate certain functionality or system status in an app.** Instead use strokes, indicators, patterns, texture or text to describe actions or content.

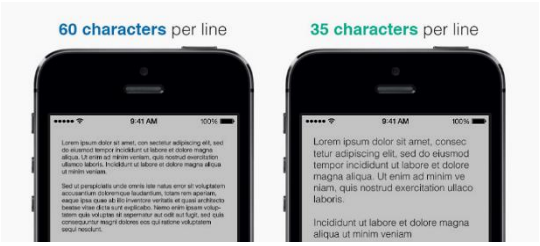
Thankfully, with the Material Design guidelines, we have means to easily pick and create colour combinations that suit our app needs. You can either go into the following Material Design Palette to pick and choose primary and secondary colours (and a preview will be given too) in the following website: https://www.materialpalette.com/ OR you can refer to Material Design's own online tool to choose your colour palette: https://material.io/design/color/the-color-system.html#tools-for-picking-colors

Elements of Typography

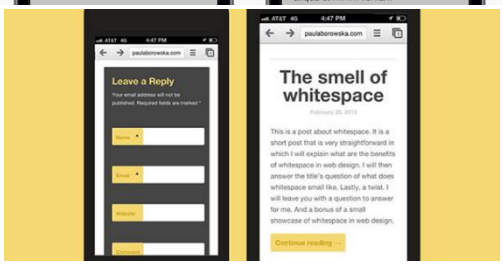
Typography is the art and technique of arranging typed text to make written language legible, readable and appealing when displayed. Typography is important in app design as text can convey things other elements cannot do, and good typography can enhance the user experience in apps by easing the user's eye and does make interaction more productive as a result.

Some general rules of thumb apply to good typography design:

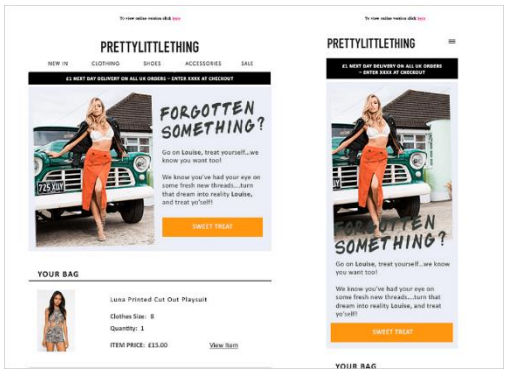
- It is better to keep fonts simple and legible with reasonable spacing in between letters. As we are building apps for the user experience, some fonts are better for those who are dyslexic. To find out what those fonts are, go back to Chapter 2.
- Fonts which are very stylistic and decorative may not be the most legible to read. While we want the fonts to stand out from other apps, we also want to make sure they are legible
- In the entire app, keep it simple and clean with only ONE primary font and one other secondary font to contrast the primary one.
- Font size should have a right balance and cannot be too big nor too small, especially since we are building for mobile devices that have smaller screens but fonts which are too small will harm the user experience. 14sp for main text for Android is typically preferred (means it should be bigger for headings)
- Keep number of characters per line to within 30 to 40 so that text is easy to read on a mobile screen. Note that font size and line length does matter as well, with respect to the number of characters per line



- Proper whitespace management will help users better interact with text. Aim for 10% to 20% of the app screen to balance between having some whitespace for the user experience and maximizing the small mobile screen space for content.



- Left alignment of text is usually the best alignment. Only use center alignment for titles or headings, only when necessary
- Have some form of hierarchy for text to create contrast. Mobile app designers usually apply 2 levels, a headline, and a body, both of which have different fonts, font sizes, and even bolded for headings to differentiate them. Headlines are to immediately grab attention and hence, the design should be big, have bigger weight and concise, whereas the body's focus should just be readability.



- Text should be kept functional if they are tagged to an action. Functional text should stand out among other elements and clickable elements should be big enough for tapping.

Material design offers a type scale generator for creating type scales which are automatically optimized based on Material Design guidelines: https://material.io/design/typography/the-type-system.html#type-scale

Elements of Iconography

Iconography is the design of meaningful icons which communicates some of the app's usual functions (like messaging, email, call, etc) in a very simple and concise way. It also helps with the branding of the app and enhances the user experience, as it allows the app designer to maximise functionality while decluttering the content available in a single small screen.

You would have seen some typical icons in many apps, such as the below icons representing "Email", "Home" and "Edit". Icons function in a way that while they look simple, they are descriptive enough to concisely communicate, usually accurately, the meaning and functionality of the icon. What makes good iconography fascinating is that once users get used to the icon-functionality associations, they become automatic and effortless in a way that even text cannot achieve.

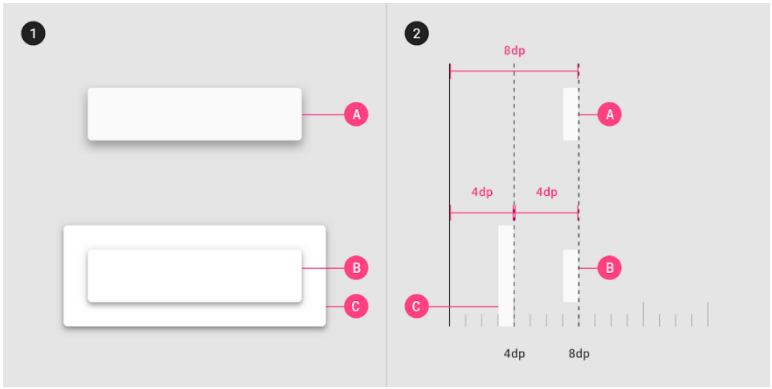
Material Design offers icons available for your app's use. Some examples of icons Material Design offers is the following, which also follows Material Design iconography guidelines. Note that while icons are useful, how the icons are utilized is equally as important. Note that icons are meant to draw user attention to important functionalities and should not compete with other visuals in the UI. Icons, at their best:

- Allow fast reading of functionalities in your app
- Saves space in terms of the UI design
- Draws user attention to important functionalities or content in your app
- Are language and culture neutral.

A good resource to get icons for Material Design will be the following:

- https://google.github.io/material-design-icons/#icons-for-android
- https://materialdesignicons.com/

Elements of Elevation and Surfaces

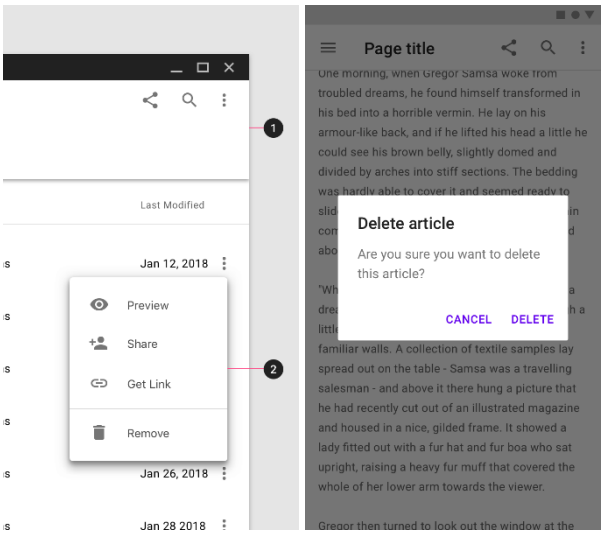


Elevation in UI components refer to the relative distance between two surfaces along the z-axis. From here, you may wonder how the z-axis is even introduced on a 2-dimensional screen. The are introduced through using **shadows** with units of measure defined as dp (same as the dp introduced in chapter 2). In the diagram:

1. Both surfaces A and B are at the same 8dp elevation. They cast different shadows because Surface B is in front of another surface (C) that already has elevation.
2. Elevation differences between surfaces (A), (B), and (C), as viewed from the side.

UI components are given a **resting elevation** by default. Note that all components will have resting elevations that are the same for each type. For example, all cards in CardView have the same resting elevation as each other. Due to user or system events, components move to preset **dynamic elevation offsets**, which will give the illusion of cards or buttons "sinking" into the layout when pressed. Due to this, there may be a case of **elevation interference**, where a change in elevation may "collide" with other components (which may make UI components "disappear" from view). Hence, as such, design measures must be implemented to avoid such collisions from happening, like:

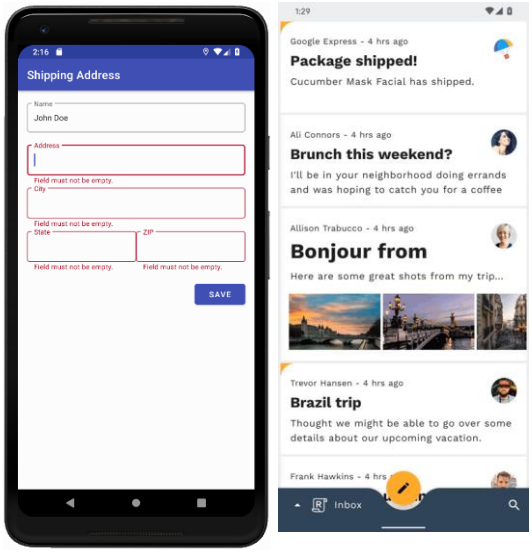
- NOT placing the FAB beside any cards in the CardView.
- UI components of different surfaces MUST have good contrast either with shadows or contrasting colours. Overlapping surfaces needs to have differing levels of elevation, either through shadows or colours. Material Design rules specify shadows need to have a purpose and not used for styling purposes only.
- When dialog boxes appear, the backgrounds must appear scrimmed (significantly darkened, indicating low elevation)
- Different animations can help to emphasise elevation to bring up its importance
- Elevation hierarchy should be reflective of the relative importance of the content in the UI. Components displayed "in front" should either have important content, or is meant to focus attention like dialogs, or is meant as a central control for all other components behind, like the action bar.
- If content are of equal importance (like cards in a CardView), they should all have the same relative importance or elevation.



You may read more about the default elevations of the different UI components in the link:
https://material.io/design/environment/elevation.html#default-elevations

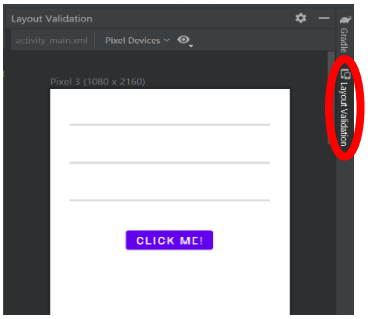
Adding Material Design Components into Your App Code

You can add material design components such as prompts under text fields and animated container transforms to add attractive transitions with follow Material Design guidelines, which ultimately enhances the user experience from the codes in the links below. Note that they involve just adding lines of code into the XML files directly.
https://codelabs.developers.google.com/codelabs/mdc-111-kotlin
https://codelabs.developers.google.com/codelabs/material-motion-android/



5.7 Designing for Different Android Devices; Different Screen Sizes

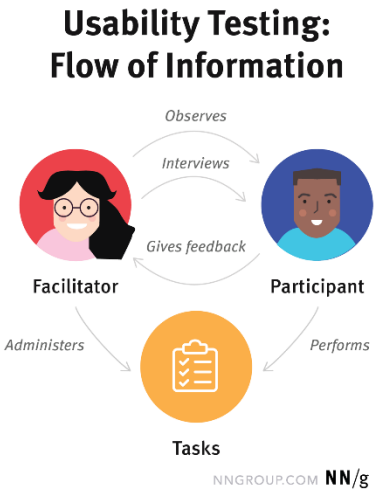
As users use a wide range of devices today with different screen-sizes, ranging from smart watches to smart phones, to tablets and even televisions (Android TV), it is important that your app works as intended for the many different devices which exists in the market today.



Thankfully, Android Studio provides a means to validate if your application views correctly on a wide range of devices. After selecting the appropriate XML file containing a layout of a fragment or activity, on the right edge of Android Studio, you will see the Layout Validation option. Opening that option will allow you to see how that layout will look like on different screen sizes, and you will be able to see if any adjustments are needed. If the ConstraintLayout is used, problems will occur if the Views are of a fixed height / width and not relative to the layout.

Now that you understand the use of fragments from Chapter 4 and how fragments can modularize the functionalities of your app, you may opt to programmatically perform a **responsive design** which will layout your fragments based on the screen size (or the applicationContext in terms of the code). You can also make design elements like the action buttons, and typography (the text size, etc) to be responsive to the screen size as well.

5.8 Testing and Evaluation of the App



Once the prototype or even the actual app is out, it is important to reach out to your users to receive feedback on what the app can do better in terms of the day-to-day use. Even seemingly trivial feedback like the colours and text must be taken seriously as this feedback may imply certain accessibility flaws or bugs in the app which could have been missed out during the development phase.

In a professional setting, typically those who were originally interviewed for the initial user study and the stakeholders (who were involved in the stakeholder interview) will be invited back to perform beta testing on the fully functional app, with the developer running a step-by-step setup session and presentation, then some time given to the "beta-testers" with **tasks to perform** to test and use the app and provide the feedback to the developer.

User ID	Category	Task	Problem	Tag 1	Tag 2
1	Search	Find a red item	User unable to locate filter features	Filter	Confusion
1	Shopping Cart	Saving an item for later	"Save for Later" button did not work- simply removed item from cart	Broken element	
1	Checkout process	Entering payment details	Accidentally exited payment process by clicking on shipping options button	Icons	Confusion
2	Checkout Process	Entering payment details	User expressed disappointment that Paypal wasn't a payment option	Payment	Disappointment

As the testing is done, app designers typically keep track of a table of any difficulties or "pain points" users experience with respect to certain tasks, and also keeping track of their feelings and spoken words so that when this feedback document is taken back, the designing and development team has a very clear idea of what happened during the user testing phase, before drawing any conclusions for further action and adjustments to the app.

5.9 Summary of the UX Design Approach

The UX design process, in short consists of the following steps:

1) Determine the product to be produced and the context for its existence
    a. Stakeholder interviews to understand requirements and manage expectations
    b. Perform competitive, user and market research
2) Develop user personas and user stories to determine user needs
3) Ideate through brainstorming and creating a user interaction storyboard.
    a. Use opportunity to uncover required features of the app or determining any gaps which need to be addressed
4) Create a lo-fi or/and hi-fi prototype of your app:
    a. Showcase how the different app pages interact between each other
    b. Showcase the interaction experience (look and feel) of the app
    c. Usability study conducted if hi-fi prototyped is developed. Make improvements after.
5) Develop and test the app. Get feedback on user experience and make improvements.

Ultimately, the most important point is that your app needs to be **user-oriented** and treat the user as the focus in development of the app's design and functionalities.

[References]

[1] UX Design Process (Adobe XD): https://xd.adobe.com/ideas/guides/ux-design-process-steps/

[2] UX Design Process. An Actionable Complete Guide: https://careerfoundry.com/en/blog/ux-design/the-ux-design-process-an-actionable-guide-to-your-first-job-in-ux/

[3] Design Thinking Process - Ideate https://www.interaction-design.org/literature/article/stage-3-in-the-design-thinking-process-ideate

[4] Storyboarding an App: https://medium.com/@jjman505/how-to-storyboard-an-app-ede5ce249ea5 https://www.storyboardthat.com/storyboards/alexcowan/customer-acquisition-storyboard--aida-or--example2

[5] Prototyping User Experience: https://www.uxmatters.com/mt/archives/2019/01/prototyping-user-experience.php

[6] Designing wireframes and low-fi prototypes: https://bootcamp.uxdesign.cc/creating-wireframes-and-low-fidelity-prototypes-in-figma-for-the-google-ux-course-326a7e11fa60

[7] High-Fidelity and Low-Fidelity Prototyping. What is the difference? https://medium.com/@tristaljing/high-fidelity-low-fidelity-prototyping-what-when-and-how-4f2a7037be1f

[8] Material Design 2 Tutorials for Android: https://material.io/develop/android

[9] Colour Theory in Design: https://www.smashingmagazine.com/2017/01/underestimated-power-color-mobile-app-design/

[10] Guide for designing better mobile apps typography: https://uxdesign.cc/guide-for-designing-better-mobile-apps-typography-5796495ef86f

[11] Typography in Mobile Design: https://medium.muz.li/typography-in-mobile-design-15-best-practices-to-excellent-ui-5eaf18280ad

[12] Using iconography to enhance UX design: https://blogs.windows.com/windowsdeveloper/2016/06/02/using-iconography-to-enhance-ux-design/

[13] Foundations of Iconography: https://uxdesign.cc/foundations-of-iconography-f95d7233a3e6

[14] Material Design: Elevation: https://material.io/design/environment/elevation.html

[15] Elevation in the Indigo Design System: https://www.infragistics.com/products/indigo-design/help/style/elevation

[16] Ideal Screen Sizes for Responsive Design: https://www.browserstack.com/guide/ideal-screen-sizes-for-responsive-design

[17] Usability Testing 101: https://www.nngroup.com/articles/usability-testing-101/

[18] Usability Evaluation and Analysis: https://www.hotjar.com/usability-testing/evaluation-analysis/

