University of the West of England
BRISTOL

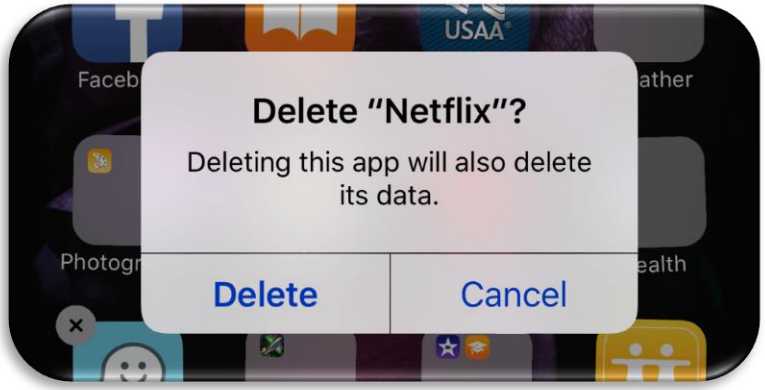
Week 6: Requirement, Wireframes and Composites

UFCF7H-15-3 Mobile Applications

Dr Kun Wei

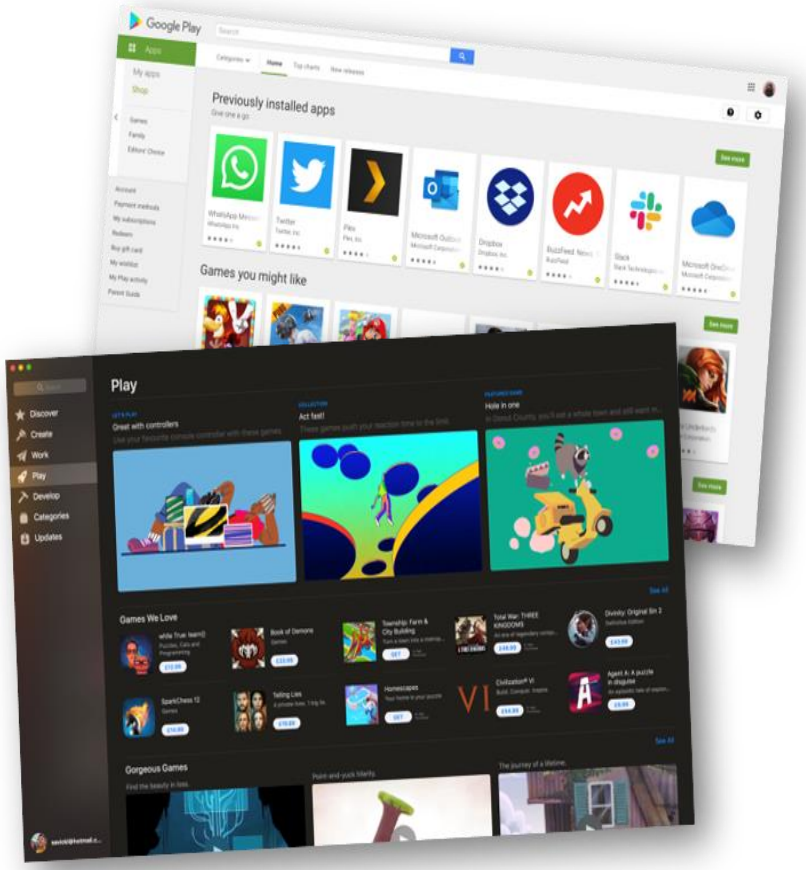
Why we do the design

- You need to make people love your app.
- It only takes **three taps** to remove your app from someone's phone.
- For every 100 apps downloaded, 50 of them are deleted after the **first time** are opened.
- You need to engage the user straight away through **good UI & UX**.



Mobile app requirement

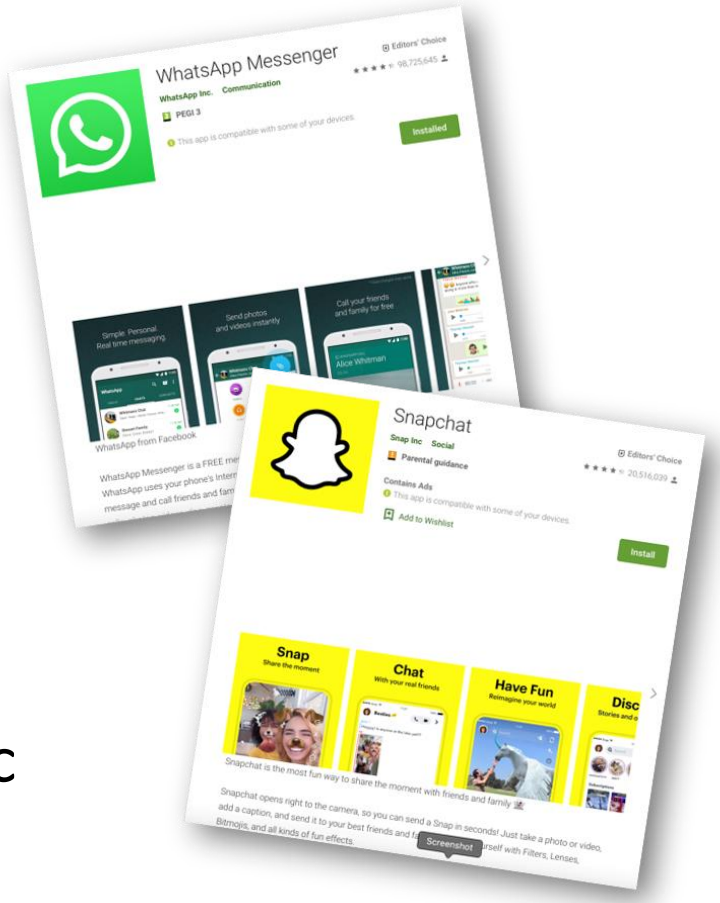
- Explore the app stores and more.
- Consider the target audience.
- Are there similar apps already?
- What makes your design different or better?



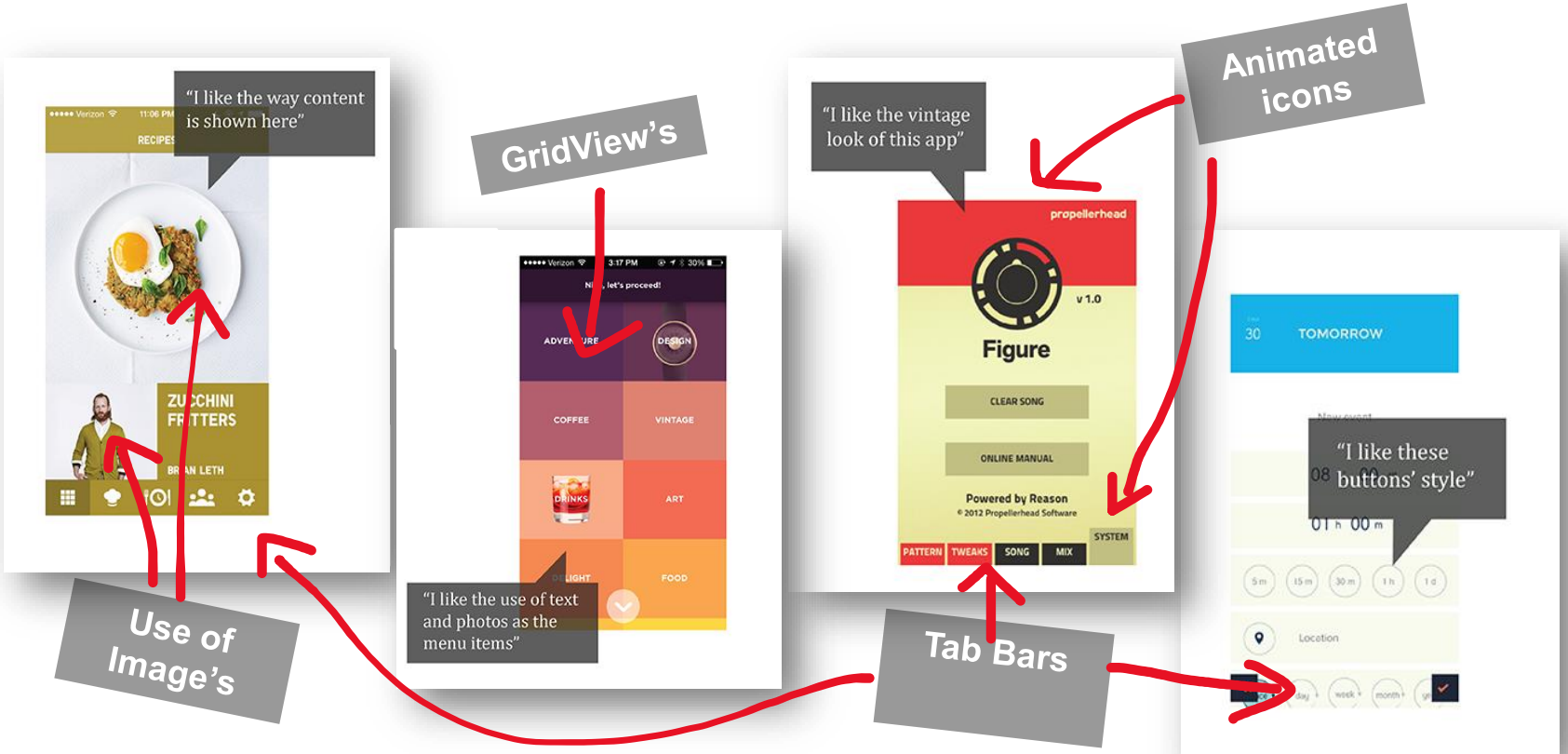
University of the West of England
BRISTOL

Digging deeper

- Compare multiple designs
- How your concept builds upon an existing design
- How your design will borrow from existing designs
- Are there UI / UX features that make these unique (can you mimic these)?



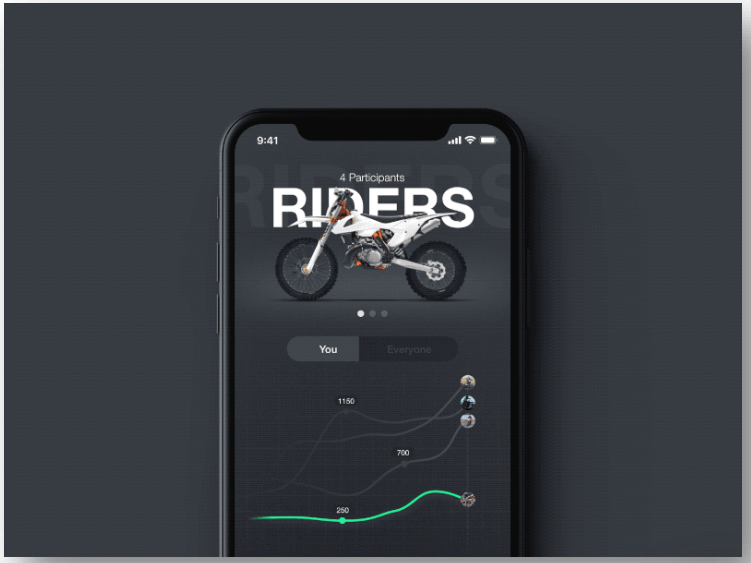
Extracting app features

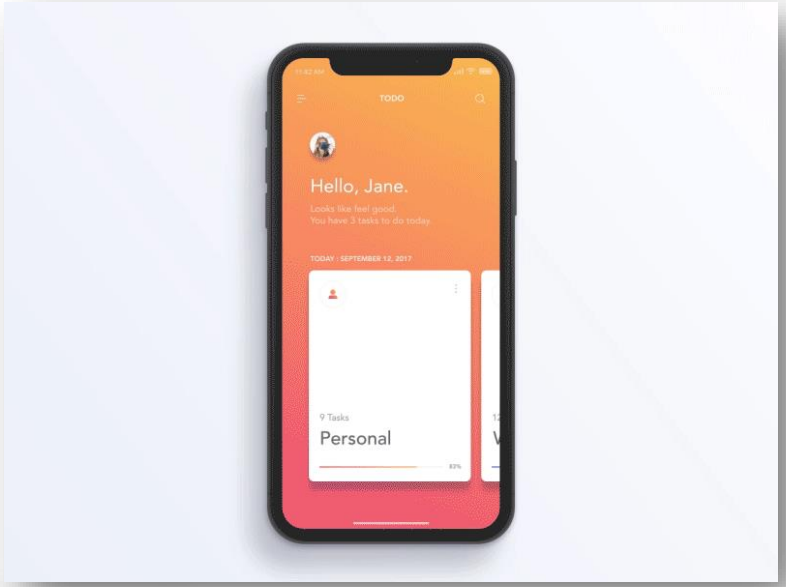


"I like something" is not really useable. Instead extract the actual UI elements to feed into your designs

University of the West of England
BRISTOL

Animations





Provide the type of animation and where the idea was sourced.

Sometimes animations can make a simple app more complex and appealing.

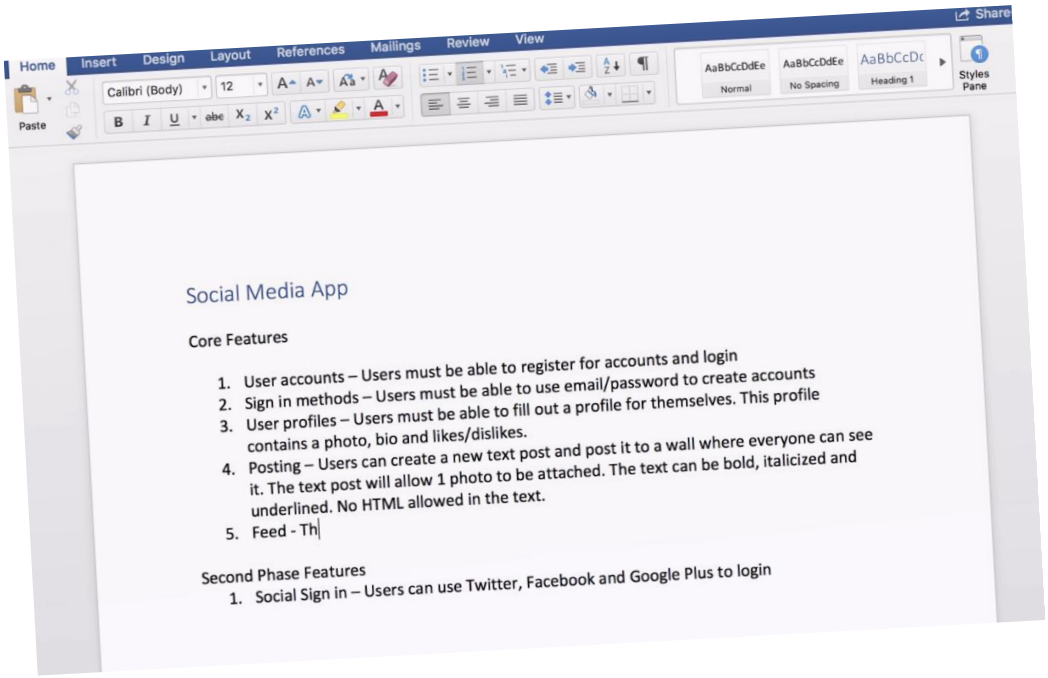
External requirements

- Are you using any:
  - API's?
  - Web tools?
  - Asset collections?
  - Database?
  - Fonts
  - Image sources

- It is highly recommended you use existing tools / libraries rather than building you own.
- Ask yourself – does the app really need a login page?

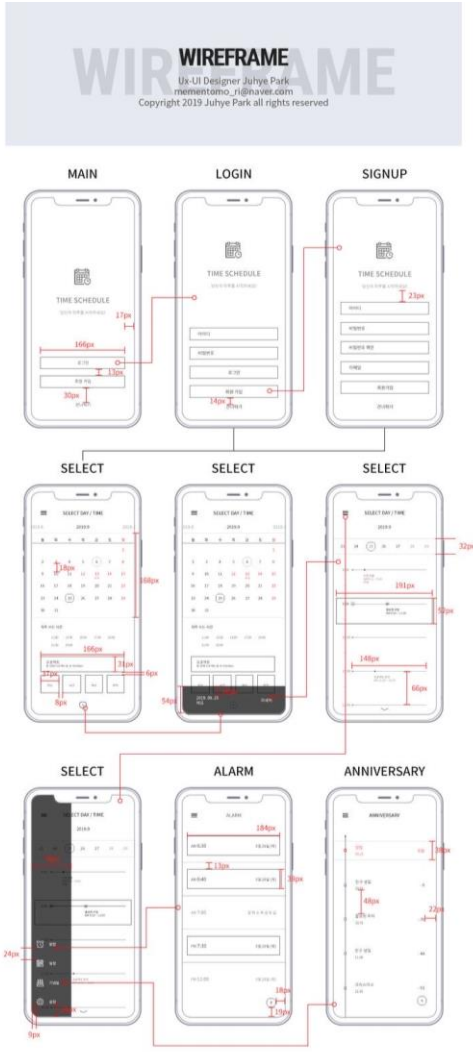
Listing your findings

- Core and desirable features.
- What UI elements are required?



(A full MoSCoW breakdown is somewhat of an overkill and not very “client facing”)

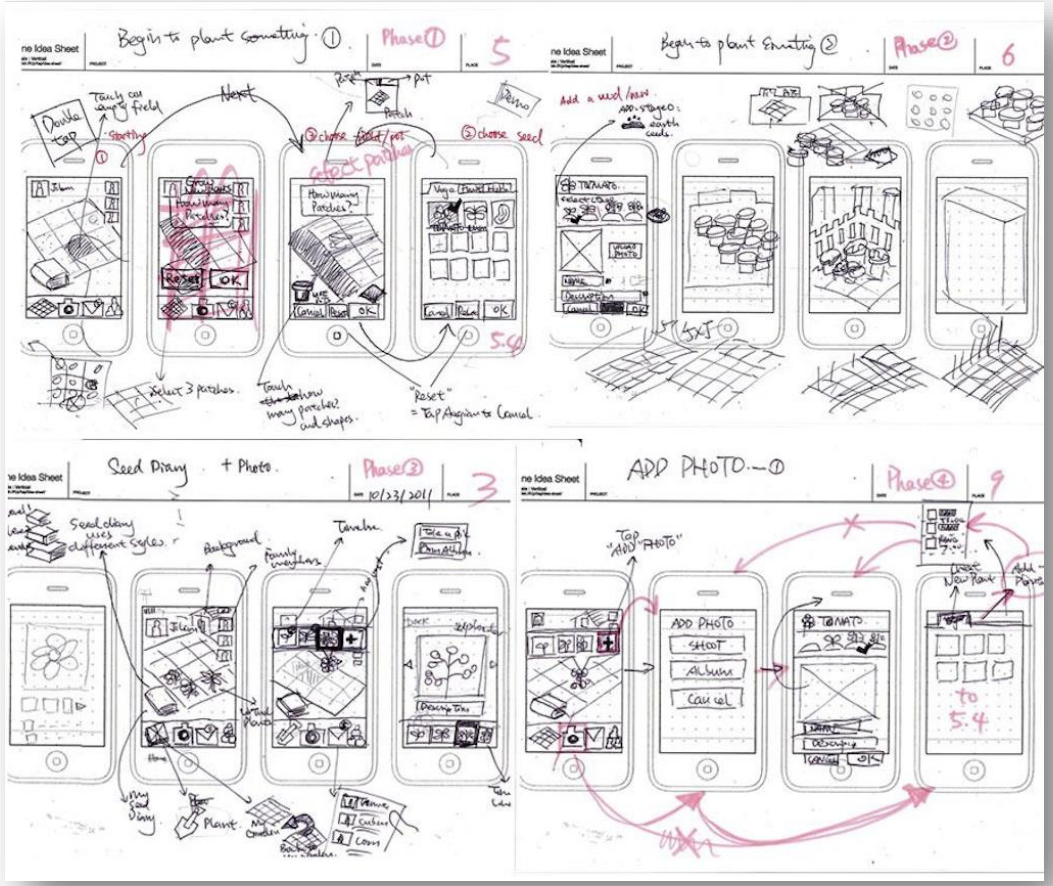
What are wireframes?



- They are low fidelity sketches of your design.
- Not concerned at this stage its aesthetic design or how final app will look.
- Wireframes should focus on layout, structure, flow and consider the usability and organisation of information.
- Put yourselves in the shoes of your target user.
- Use tried and tested structures which are provided by each OS (see links on spec and on Bb).
- Forces you to stop and think through things – have you met your requirements?
- Can be used for (very) early user feedback.

Starting your design

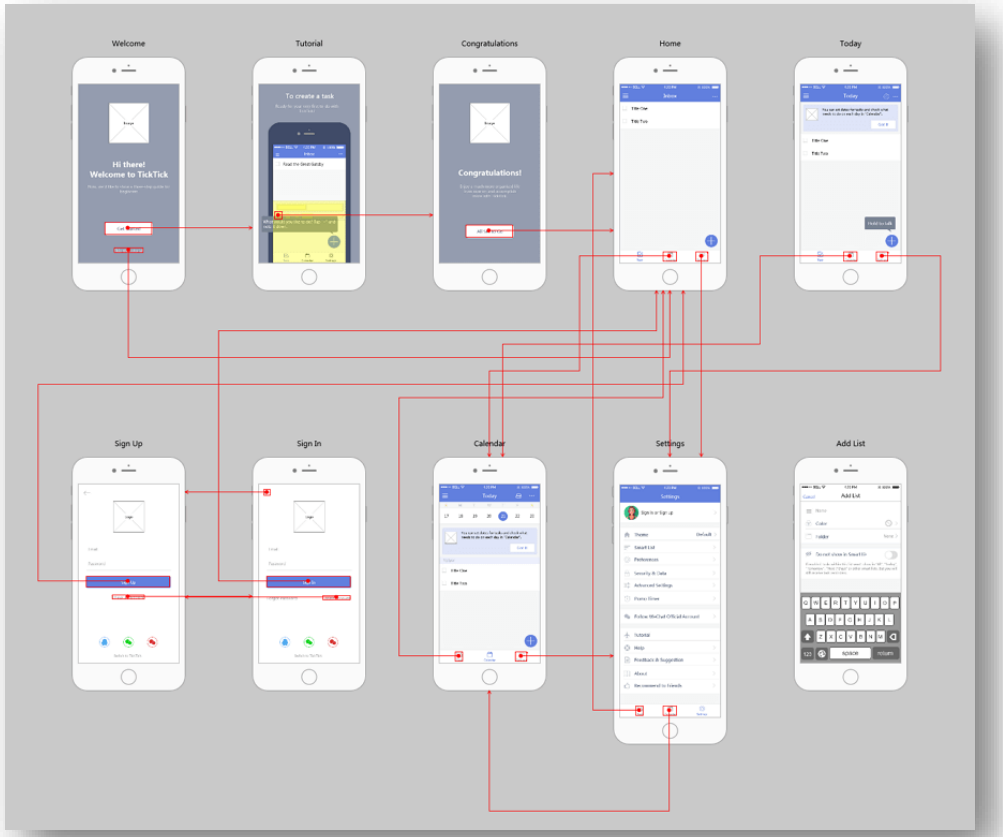
- You should all start here
- Paper prototype
- Quick / Cheap
- Easy to iterate
- NOT client facing



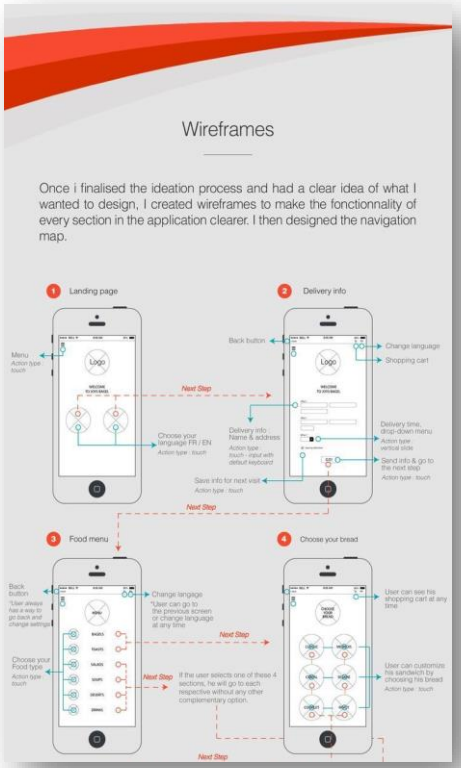
(you should be submitting digital wireframes – not hand drawn ones as shown here).

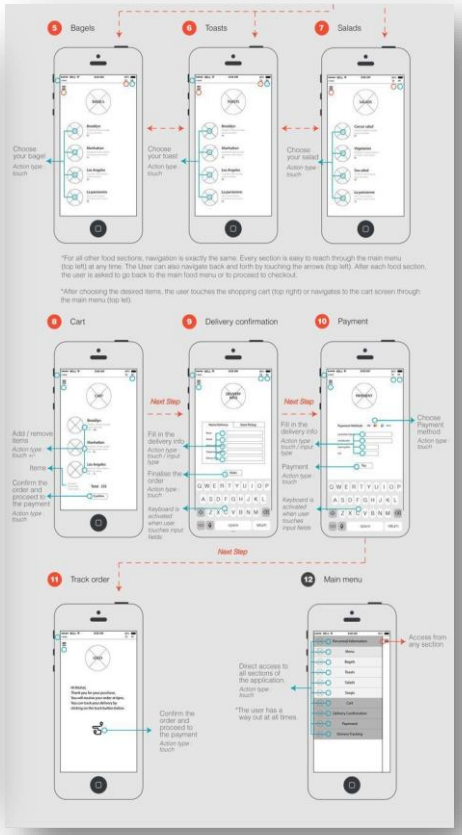
App flow

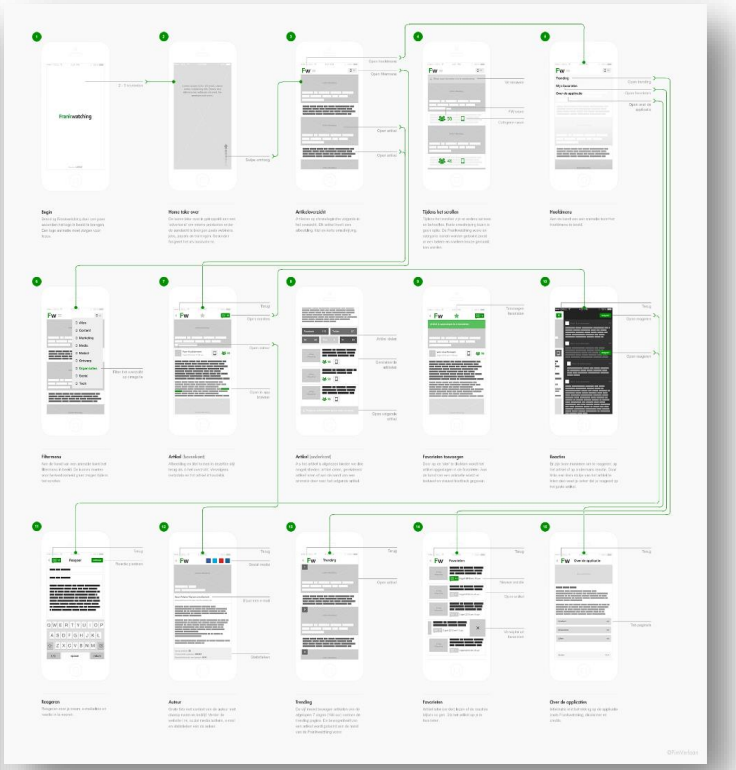
- Show the flow and navigation of your design.
- Clarity is key. Avoid ‘spaghetti’ diagrams
- Can be used by frontend developers to build the final application.



Layout examples



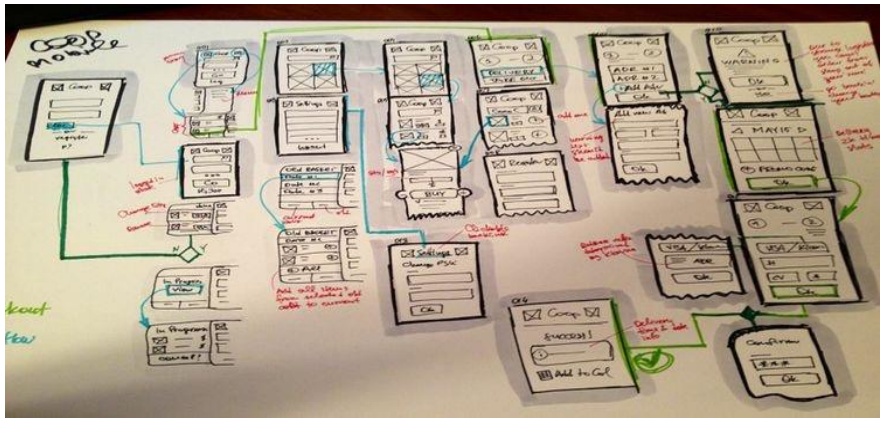


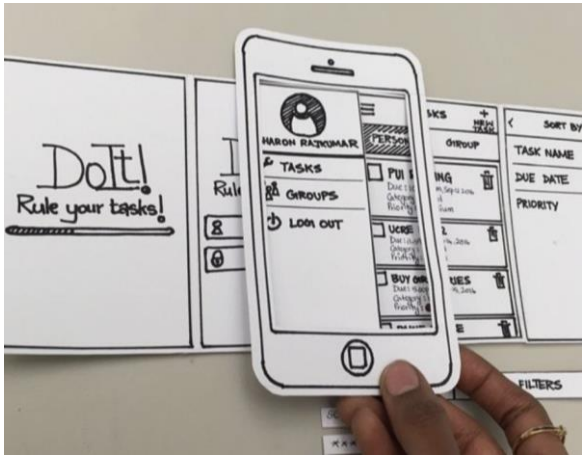


University of the West of England
BRISTOL

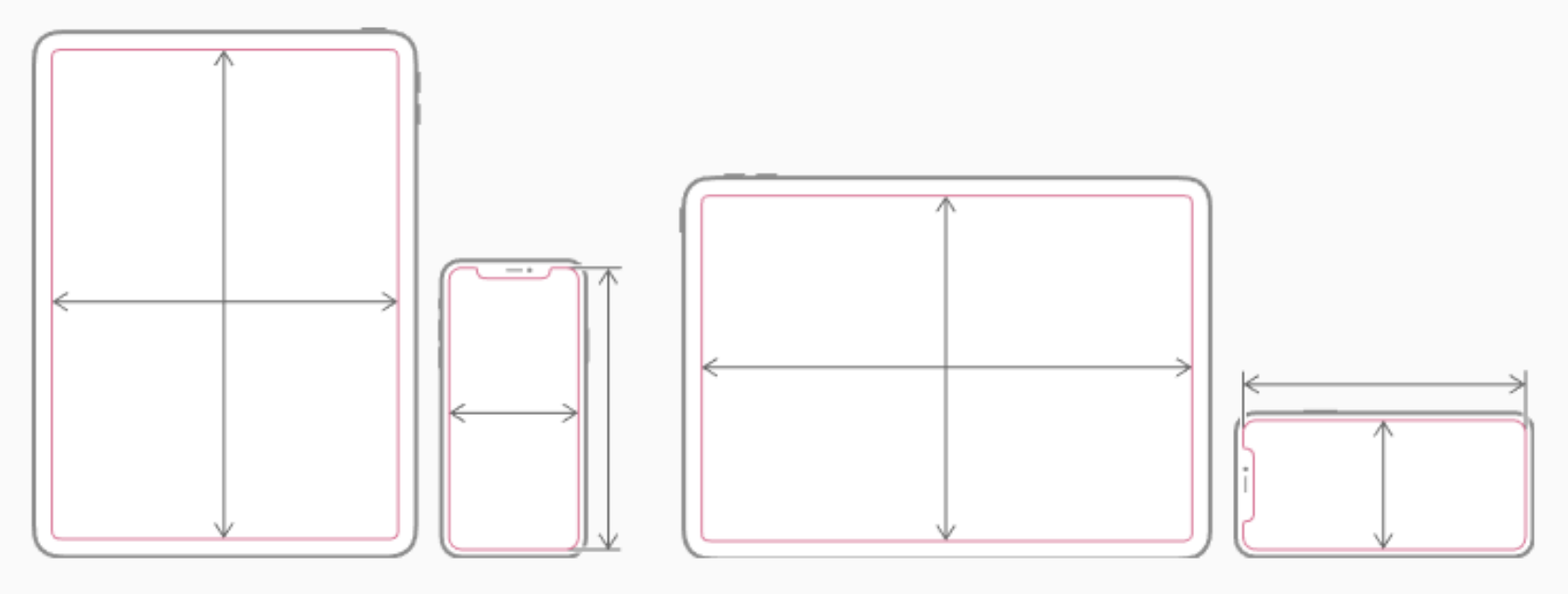
No single method

- We recommend starting with paper prototypes and there are many ways to approach this.
- Some methods are better for early user testing. Others simply help with initial ideas.
- Pick the best method for your app design / project.



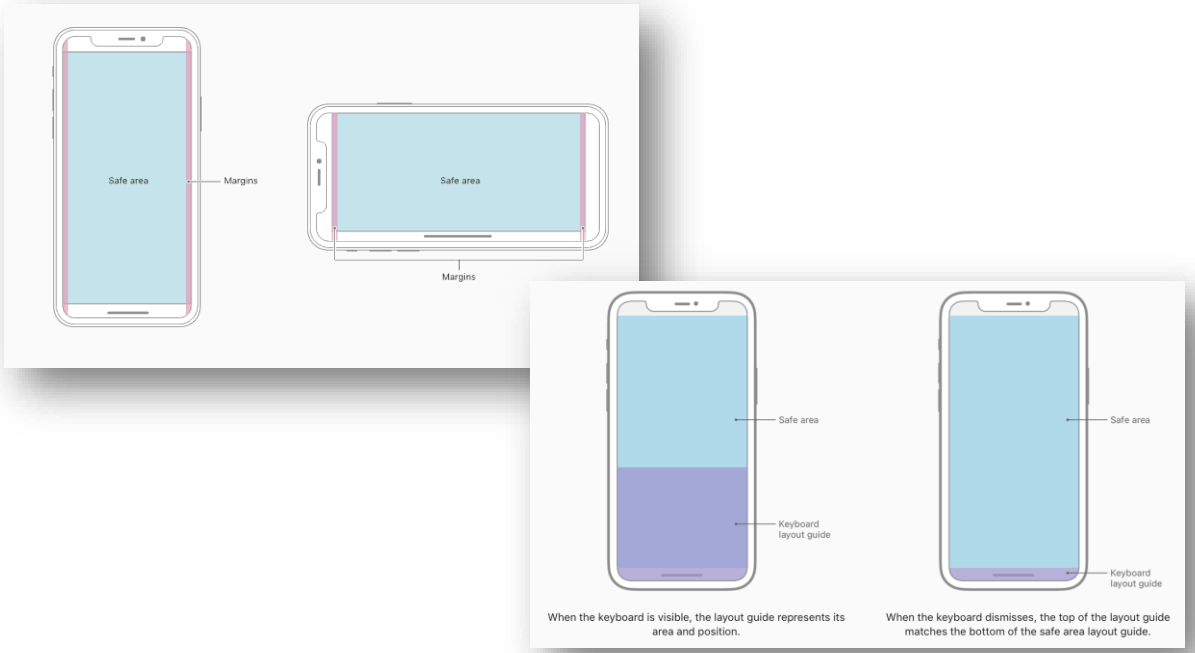


Orientations



Whilst is it likely the majority of you will choose a portrait layout, we will always assess whether this was done as a design decision or to make your life easier.

Consider safe areas



iOS – Recommend areas to avoid. Can affect UX and cause adverse responses to interaction (Available on Bb).

Android – More to consider... Recommend you focus on one device type.

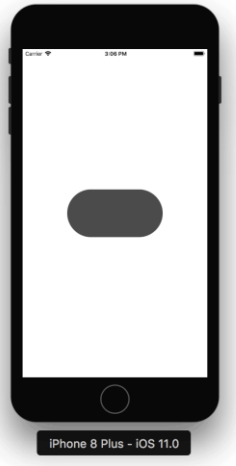
Fixed and Fluid layout

- Fixed layout:

  Does not change based on the screen size or orientation. Currently there are only a small amount of screen sizes that limit you (watches etc.) so generally it is better to design a layout that dynamically changes.

- Fluid layout:

  When the size of the elements inside the layout are defined by percentages, so the proportionally changes based on the view size. Users expect certain views to accommodate various devices / orientations.

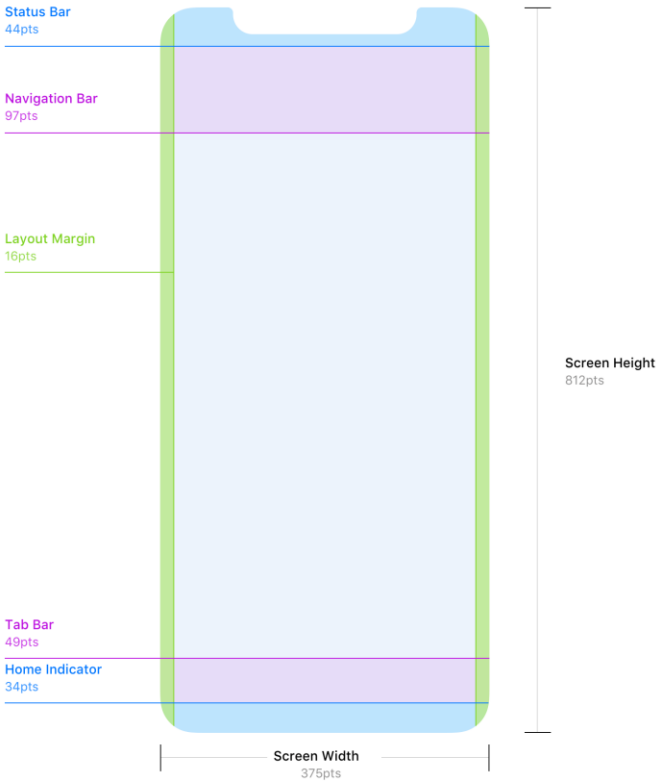


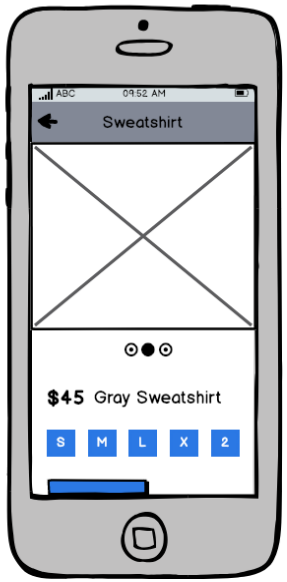
Scale and constraints

Things to consider:

- *Rotation / orientation*
- *What device it runs on (It should support all current devices)*
- *Screen position and percentage.*
- *Higher marks available if you documentation shows multiple devices/sizes/orientations*

Representing Scale





University of the West of England
BRISTOL

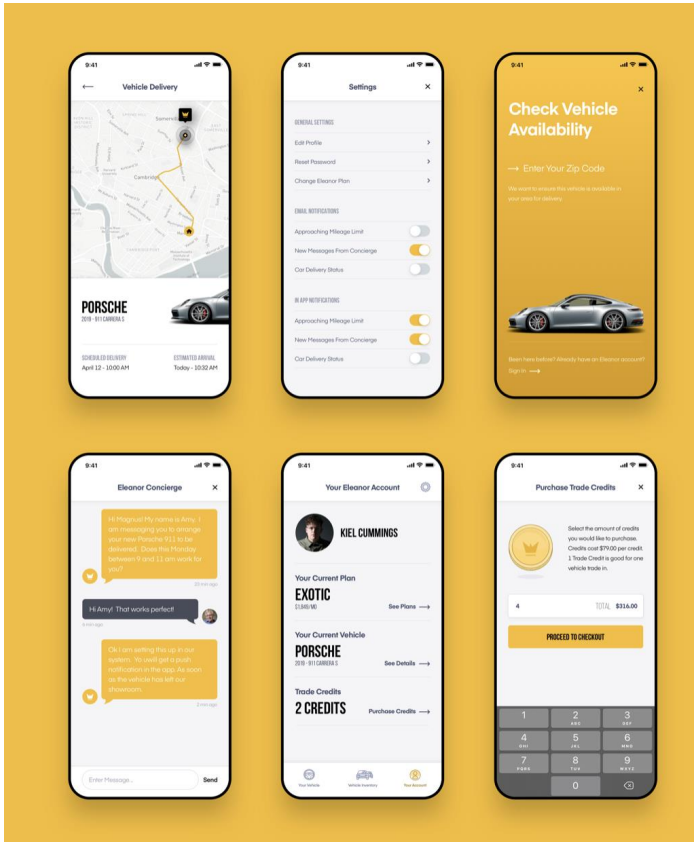
What are composites?

Essentially usable mock-ups with the focus is on the final visual design.

Tells anyone what the final app will look like (whilst not necessarily showing the flow).

Layout is drawn from the wireframes but now with the context for the images, colours and UI.

These will be a mixture of placeholder images (to show people what it will look like as a final app) and usable assets that can be used directly in the final design.



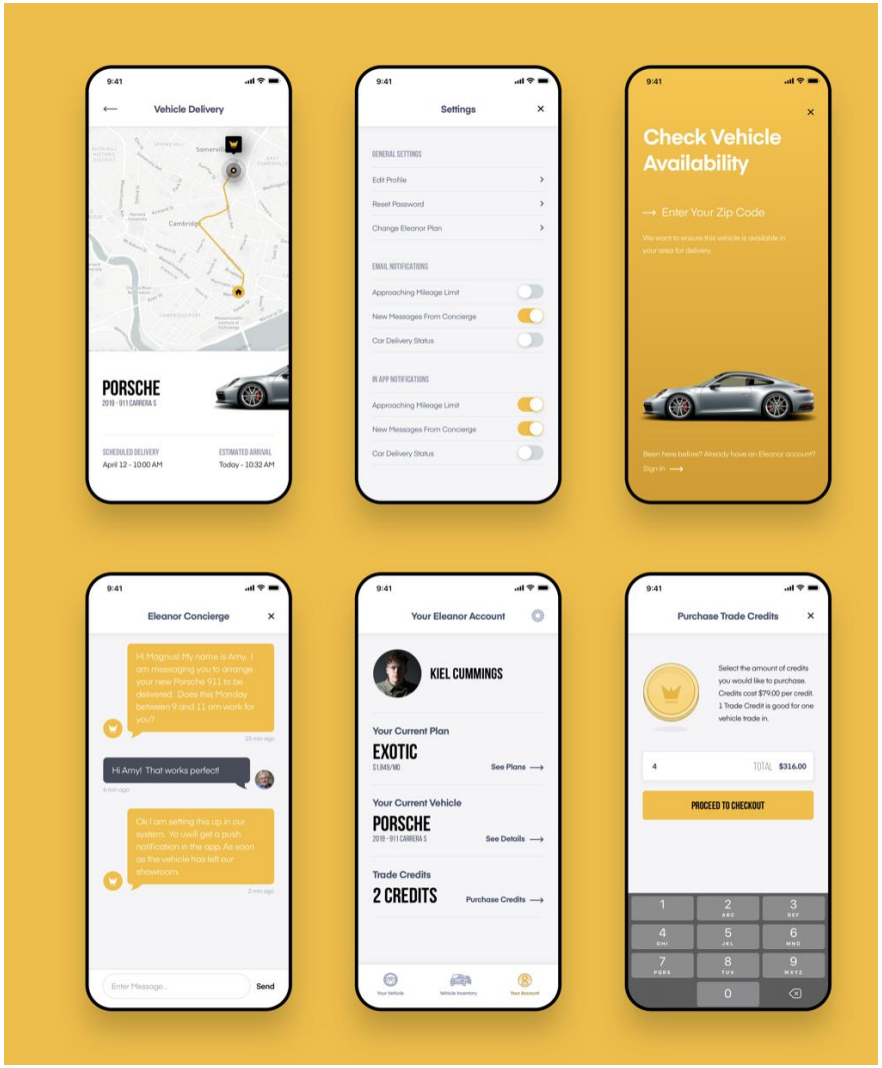
University of the West of England
BRISTOL

A frontend developer

- Remember your role in this module.
  - UI Designer prototypes the screen (s),
  - Frontend developer takes those designs and uses Kotlin, etc. to make them “work”.
- You can achieve high marks in this module using free assets, but we ask that you avoid “programmer art”.

Mock-ups

- Show a near completed aesthetic app design.
- Things can still change during development and implementation.
- Colour, layout style are all decided and agreed upon.



University of the West of England
BRISTOL

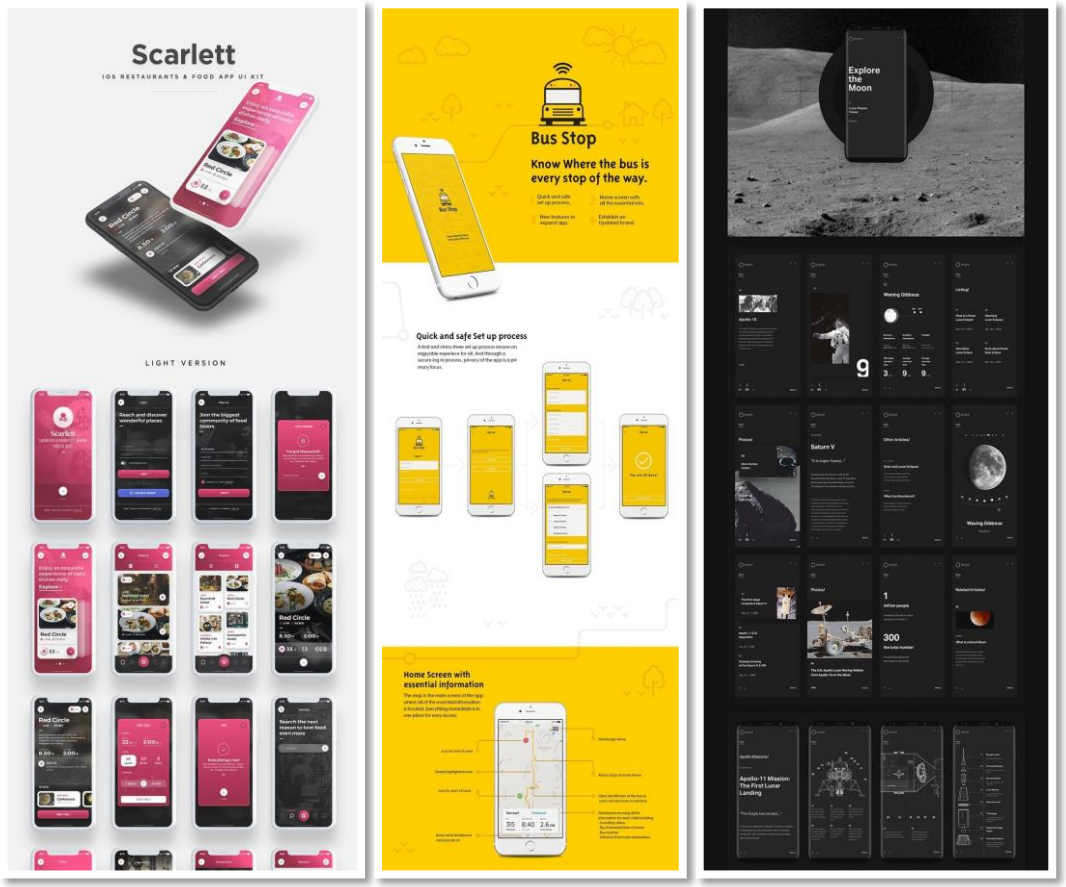
Composites

A chance to show your app as you intend it to look.

You may not achieve exactly what you are presenting at this stage, but remember this is a separately assessed piece of work.

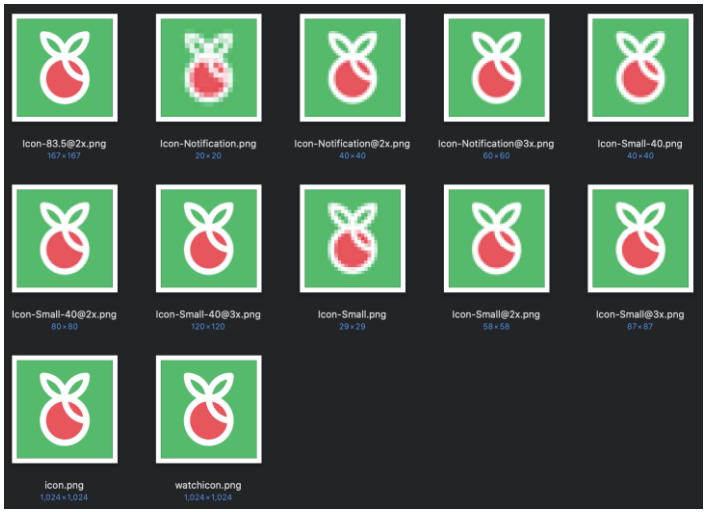
Presentation is important. This is where you really tell people about your app design.

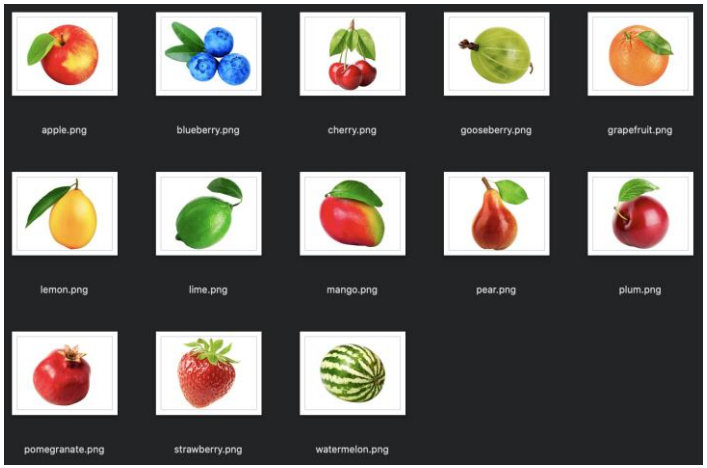
The majority of assets should be exportable to the final app.



Useable Assets

- Much of what you produce can likely be created programmatically.
- Where you are using images, consider naming conventions and scale.
- These should be final and for this assessment you are welcome to use sourced images if required



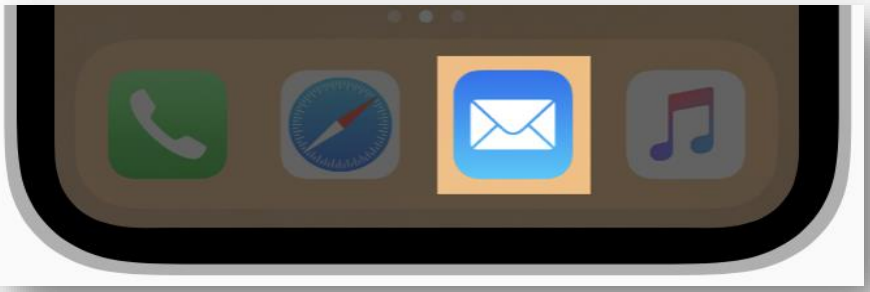


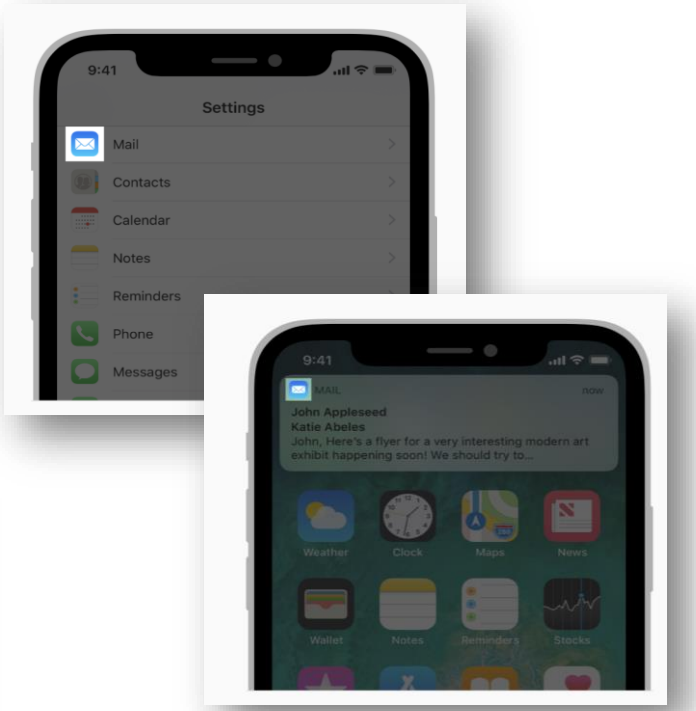
App Icons

App Icon Attributes

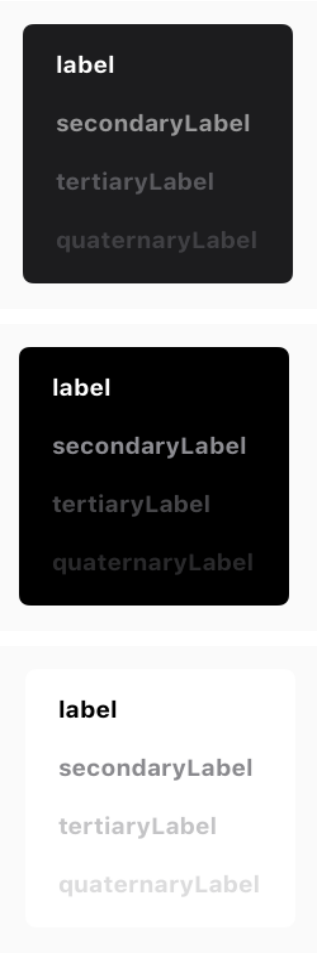
All app icons should adhere to the following specifications.

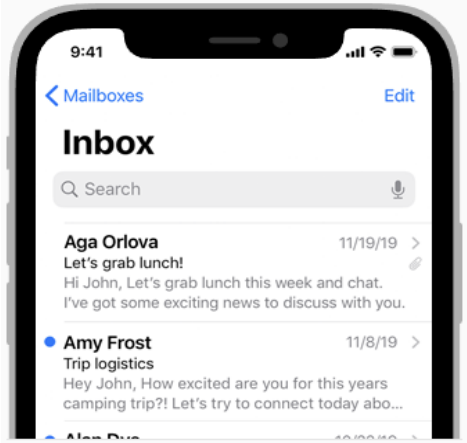
Attribute	Value
Format	PNG
Color space	Display P3 (wide-gamut color), sRGB (color), or Gray Gamma 2.2 (grayscale). See Color Management.
Layers	Flattened with no transparency
Resolution	Varies. See Image Size and Resolution.
Shape	Square with no rounded corners





Dark Mode





Users make these choices at a systemwide level, they generally expect all apps to respect their preferences.