Mobile Web Application Development

Session 1

Mobile Web Applications

Mobile Apps

- Mobile applications fall into two broad categories:
  - Native applications
    - Platform dependant (e.g. iOS or Android, etc.).
    - Installed on user machine.
    - Written using a programming language (e.g Java, Swift, etc.).
    - Full access to device features (camera, microphone, accelerometer, etc.).
    - Designed specifically for small screen devices.
  - Mobile Web Applications
    - Platform independent.
    - Hosted on a server, accessed via the Internet.
    - Requires a browser to run.
    - Written using HTML 5 (HTML, CSS and JavaScript).
    - Limited access to device features.
    - Designed to work on small and large screen devices.

Mobile Web Apps vs Mobile Websites

- Web applications are *functionality* focused.
- Websites are *content* focused.
- The dividing line between the two is not always clear:
  - Some applications provide extensive access to content.
  - Some websites provide access to app like functionality (e.g. TFL Journey Planner).

Mobile Web Apps – Examples

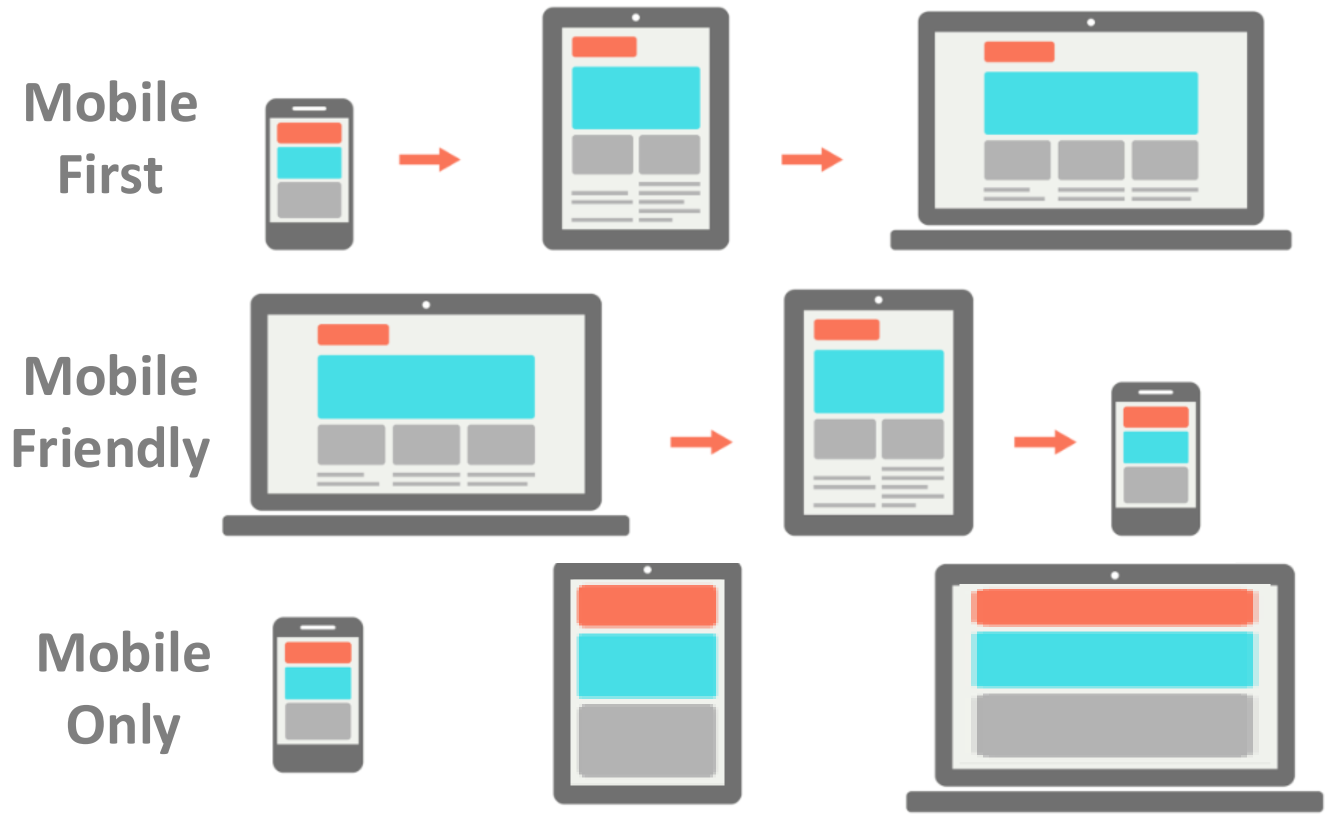
- Zoopla
- EasyJet
- Vehicle Tax
- Uber

- Note that the applications work equally well in both small and large screen devices.
- They use *responsive* implementation techniques to scale for different display sizes.

Mobile Web Apps – Development

- Mobile web applications are designed and built using one of the following approaches:
    - Mobile first
        - Starts with small screen experience, including only essential functionality and features. Then scales up to larger screens using progressive enhancement approach.
    - Mobile friendly
        - Starts with large screen experience, which includes full functionality and features. Then scales down to smaller screens using graceful degradation approach.
    - Mobile only
        - Designed and built for the small screen experience only. No explicit effort is put in to catering for the large screen experience.

Mobile Web Apps – Development



Mobile Web Apps – Development

- Use **mobile first** when it is clear the majority of users will access your app via a mobile device.
- Use **mobile friendly** only when you are sure that the vast majority of your users will be using desktop devices.
- Use **mobile only** when you are sure that the vast majority of your users will be using mobile devices, and you are prepared to sacrifice the additional features and functionality that a desktop version might provide.

Mobile First – Google Indexing and Ranking

- A mobile first approach is crucial for **business success**.
- Google has previously used the page content of desktop versions of sites and apps for indexing and ranking.
- In July 2019, Google switched to mobile first indexing.
- Pages are now indexed and ranked based on mobile content.

Mobile First – Design and Implementation

- **Mobile first design** starts with the design for the smallest screen experience. The design is then progressively adapted for tablets, notepads and desktops.
- **Mobile first implementation** starts with writing the code for the smallest mobile version, then proceeds to add the code for larger screen versions.
- Mobile first implementation is a *responsive technique* and is based on the use of media queries.

Mobile Web Apps – Design

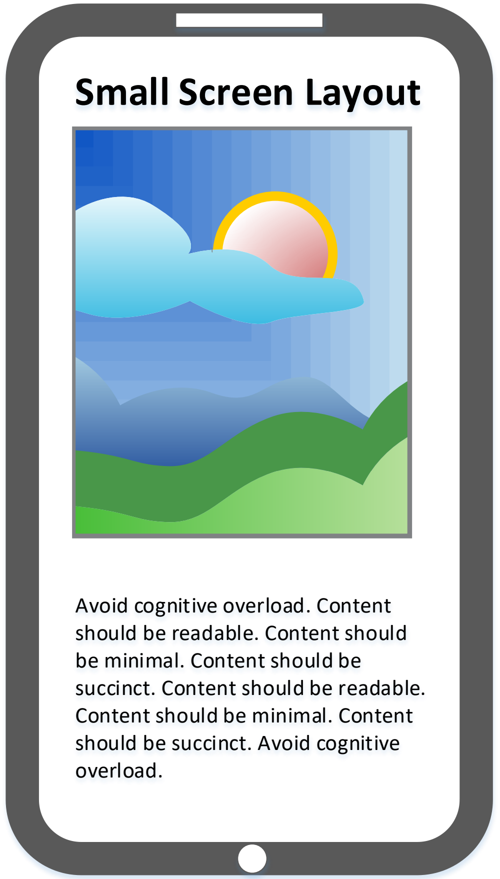
- **Design for mobile** is dictated by several key aspects of small screen devices, notably:
  - Limited screen real estate.
  - Touch interaction.
  - On screen keyboard.
  - Dual orientation (portrait and landscape).
  - Portability.
  - Mobile specific use contexts (e.g. on the move, whilst distracted, etc.).
- We can only create successful mobile apps if we factor these aspects into our designs.

Mobile Web Apps – Design – Guidelines

- A useful set of **guidelines** for mobile first design:
  - Reduce cognitive load.
  - Maximise content to chrome ratio.
  - Take a minimalist approach to content and aesthetics.
  - Use mobile specific layout.
  - Use mobile specific typography.
  - Use mobile specific navigation.
  - Design for touch.
  - Minimise the need for typing.
  - Integrate with device features where possible (e.g. phone, email, messaging).

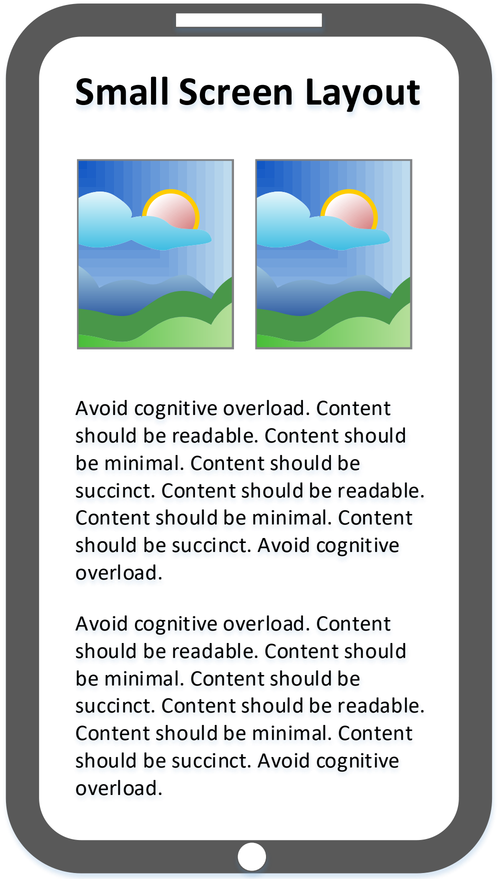
Mobile Web Apps – Design – Layout

- **Layout** for small and large smartphones, and small tablets should consist of a single column with elements stacked vertically.
- For larger tablets, two columns can be used, but this is not necessary, as one column does the job just fine.
- Single column layout is easy to implement and renders consistently across browsers and browser versions.


Small Screen Layout
Avoid cognitive overload. Content should be readable. Content should be minimal. Content should be succinct. Content should be readable. Content should be minimal. Content should be succinct. Avoid cognitive overload.

Mobile Web Apps – Design – Layout

- Two columns can be used in certain instances. But should be the exception rather than the rule.
- For textual content, using two columns will limit line length to around 15 to 20 characters per line in extra-small displays, which is too short.
- Column numbers can be increased for large tablets, notepads and desktop.

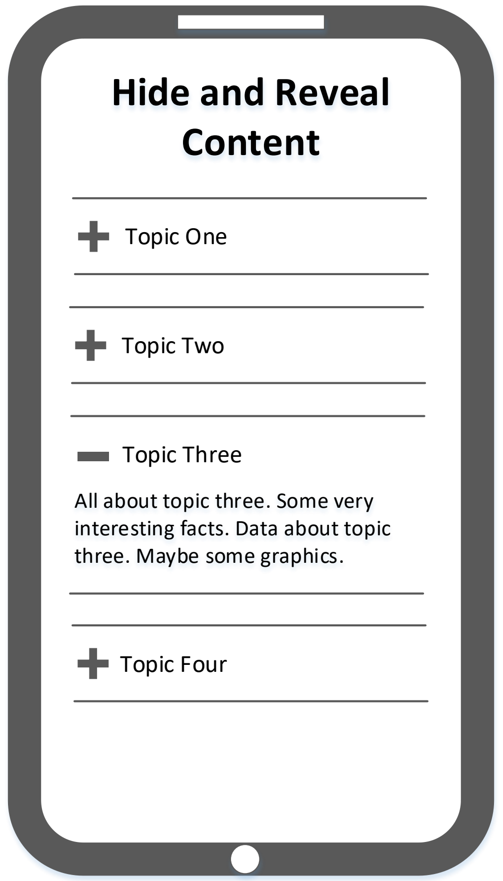


Avoid cognitive overload. Content should be readable. Content should be minimal. Content should be succinct. Content should be readable. Content should be minimal. Content should be succinct. Avoid cognitive overload.

Avoid cognitive overload. Content should be readable. Content should be minimal. Content should be succinct. Content should be readable. Content should be minimal. Content should be succinct. Avoid cognitive overload.

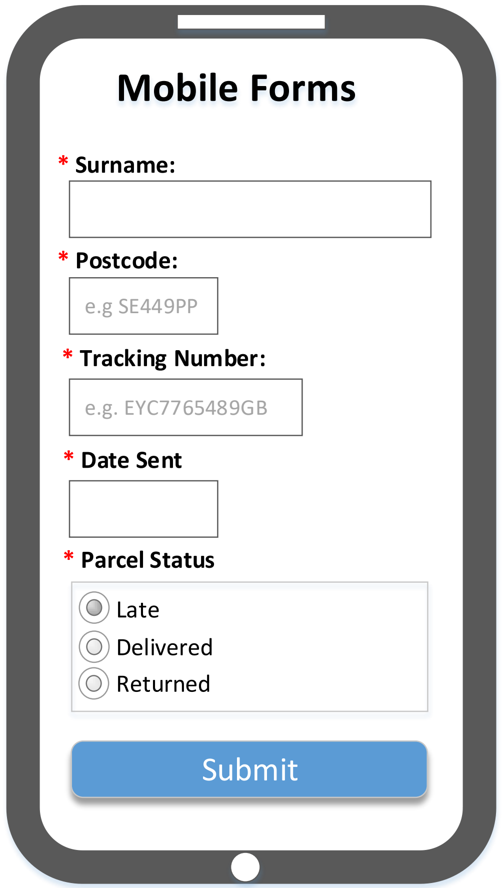
Mobile Web Apps – Design – Layout

- Extended content should be hidden to save screen space.
- Users should be given the opportunity to reveal only the content that is relevant to their current need.
- This can be achieved using bespoke CSS, or if using a framework, an appropriate widget (e.g. accordion, collapse, etc.).
- Content can be un-hidden for larger screens.


Hide and Reveal Content
Topic One
Topic Two
Topic Three
All about topic three. Some very interesting facts. Data about topic three. Maybe some graphics.
Topic Four

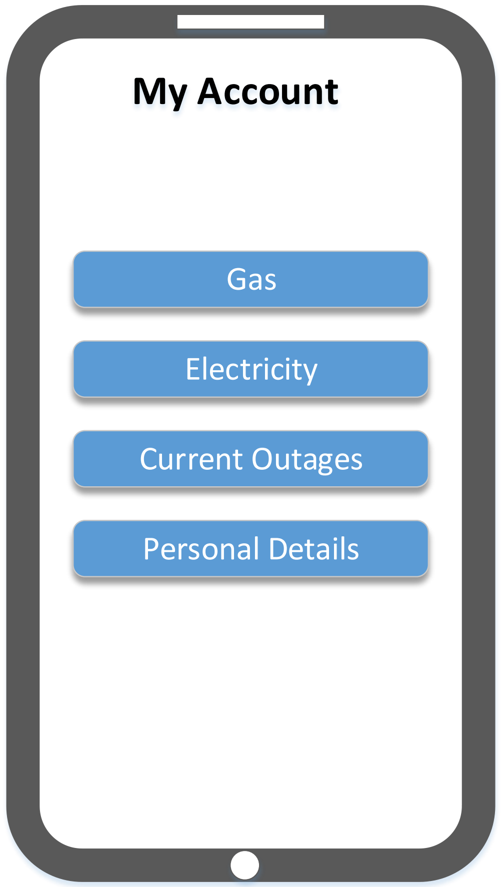
Mobile Web Apps – Design – Forms

- Form layout for mobile should:
  - Always be based on a single column.
  - Always select the correct control for the data being input (e.g. email, number, date, etc.).
  - Minimize the number of fields.
  - Clearly distinguish optional fields.
  - Provide a *show password* option.
  - Use placeholders, where appropriate.
  - Use inline form error messages together with field highlighting. Do not use JavaScript alerts.
- Forms can be multi column in larger displays; however, multi column forms should generally be avoided because of potential usability problems.



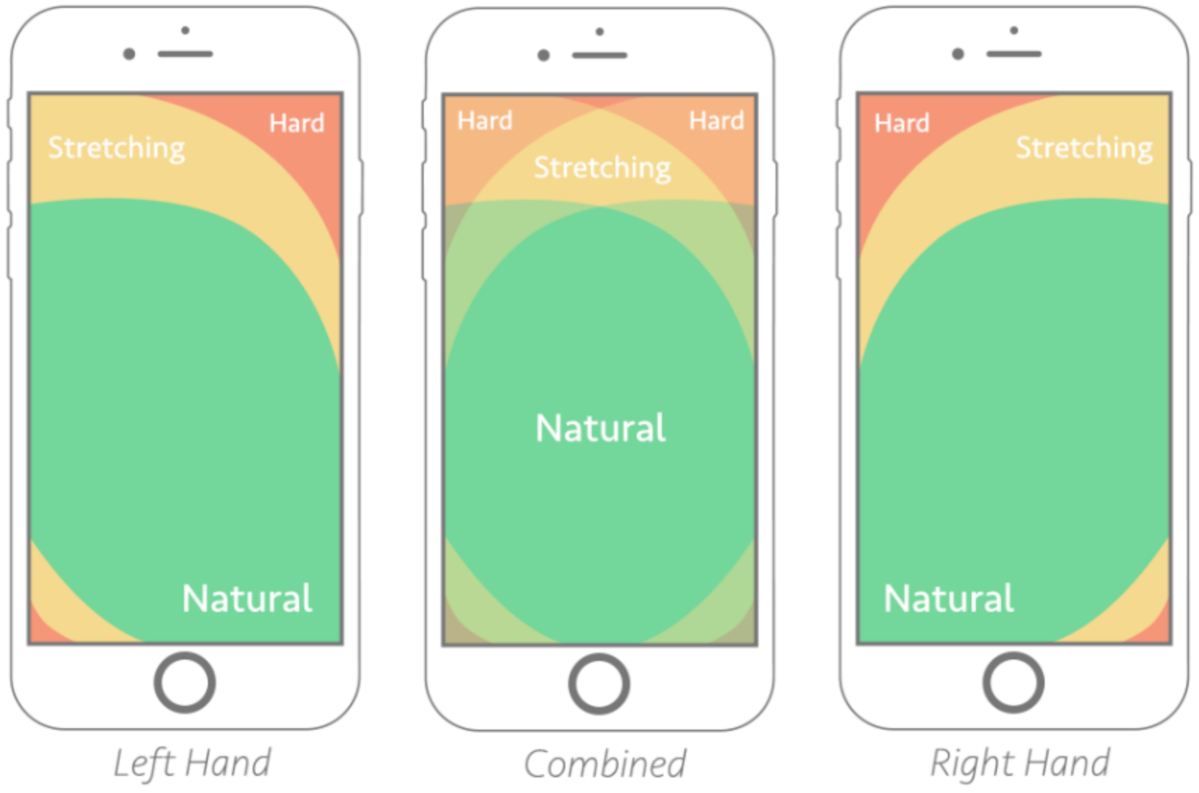
Mobile Web Apps – Design – Touch Screen

- In small displays clickables should be large enough to be easily hit by a thumb.
  - Minimum 72px x 48px.
- Space clickables adequately to avoid mis-hits.
  - Minimum 8px.
- Use buttons rather than links for important clickable items.
- Minimise scrolling
  - Design mobile content that requires little or no scrolling.
- Forget hover
  - It does not work in touch screens.



Mobile Web Apps – Design – Touch Screen

- Place important targets inside *thumb zones.* This includes buttons, navigation, search, toolbars, etc.



Mobile Web App – Typography

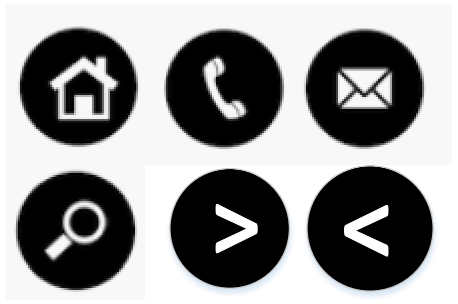
- Readability is crucial for successful mobile application design.
    - Fonts should not be too small (cannot be read) or too large (take up too much space). The base font should be 16px.
    - Use a single font family with a high readability rating (Android uses Roboto and Noto).
    - Optimise color contrast between font and background. Minimum is 4.5 to 1 (WCAG).
    - Limit line length to 30 – 40 characters per line for smaller screens.

Mobile Web App – Images

- Images should be appropriately sized for the current display size.
- There are two methods to achieve this:
  - Media queries
    - Uses a single image. Scales the image to adapt to the current display size.
  - HTML picture element.
    - Uses multiple images. Selects the most appropriate image from a set of images, depending on the current display size.
- Be judicious with your use of images in smaller screens. Consider using less graphical content.

Mobile Web App – Icons

- Icons should be used in smaller displays to replace navigation related text that would otherwise take up significant screen real estate.
- Icons are particularly useful for creating horizontal toolbars and navigation bars.
- Use standard icons only.
- Icon services such as Font Awesome provide a huge range of easy to use, high quality icons.

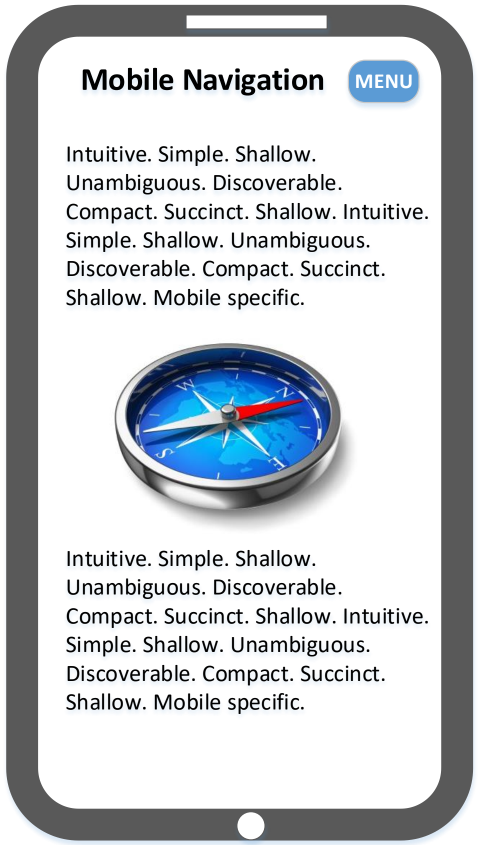


Mobile Web App – Navigation – Guidelines

- **Mobile navigation schemas** differ significantly from traditional desktop navigation schemas.
- Mobile navigation should be:
  - Accessible (Located within the thumb zones).
  - Compact (Fits comfortably into limited screen space).
  - Highly intuitive (Learnable in a single use).
  - Succinct (Uses single short words or icons).
  - Shallow (Ideally be no more than two levels deep).

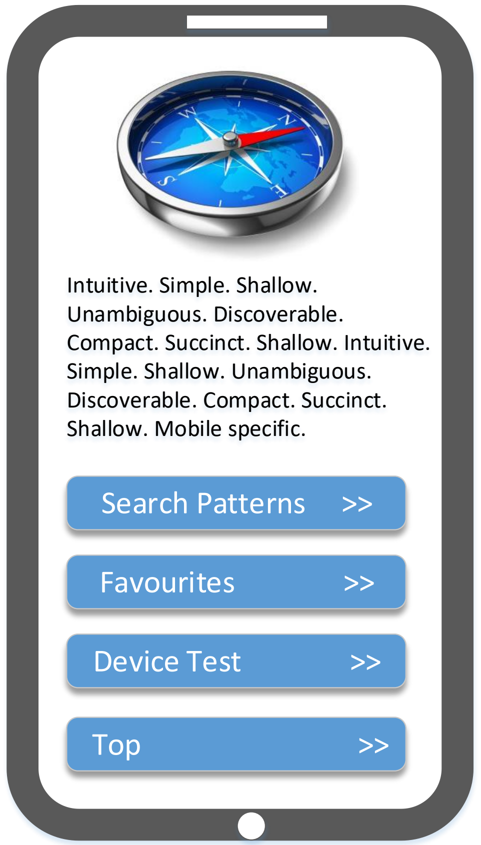
Mobile Web App – Navigation – Menu

- Primary navigation options are located towards the bottom of the page.
- The navigation options are accessed via a menu button located towards the top of the page.
- Works well for small sites and applications. Easy to implement.



Intuitive. Simple. Shallow. Unambiguous. Discoverable. Compact. Succinct. Shallow. Intuitive. Simple. Shallow. Unambiguous. Discoverable. Compact. Succinct. Shallow. Mobile specific.

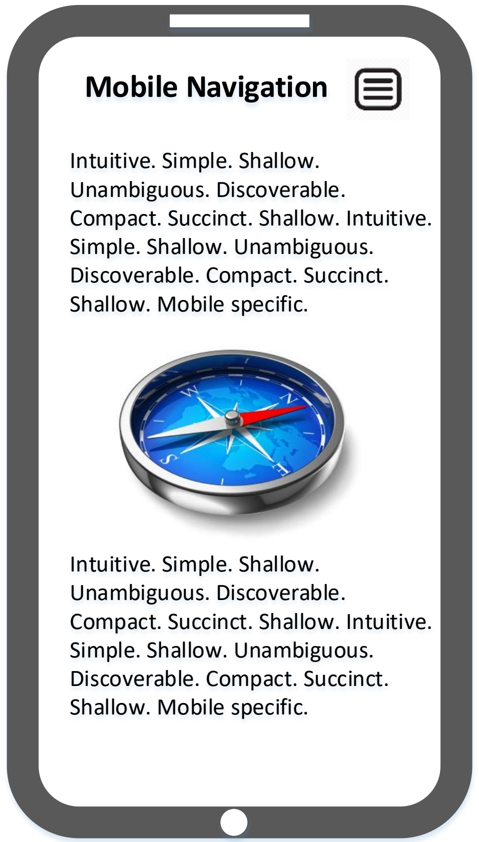
Intuitive. Simple. Shallow. Unambiguous. Discoverable. Compact. Succinct. Shallow. Intuitive. Simple. Shallow. Unambiguous. Discoverable. Compact. Succinct. Shallow. Mobile specific.



Intuitive. Simple. Shallow. Unambiguous. Discoverable. Compact. Succinct. Shallow. Intuitive. Simple. Shallow. Unambiguous. Discoverable. Compact. Succinct. Shallow. Mobile specific.

Mobile Web App – Navigation – Hamburger Menu

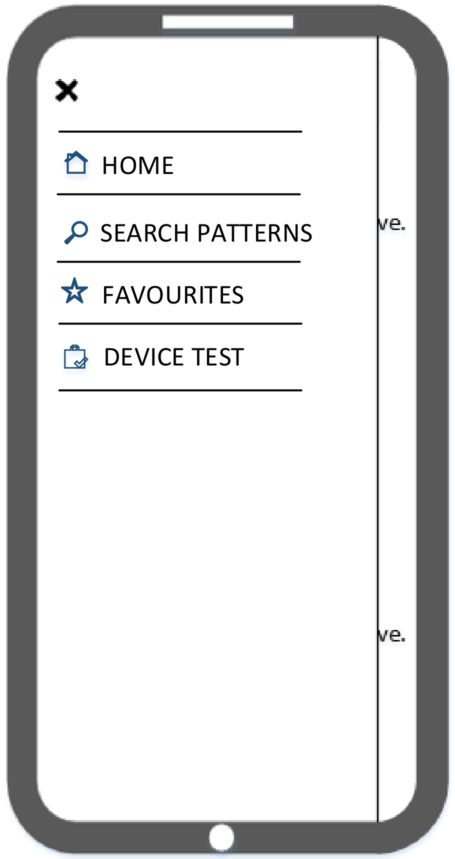
- Navigation choices are contained within a hidden panel that reveals horizontally or vertically.
- The panel is revealed by clicking a hamburger icon located towards the top of the page.
- This option works better for more complex sites and applications.



Mobile Navigation

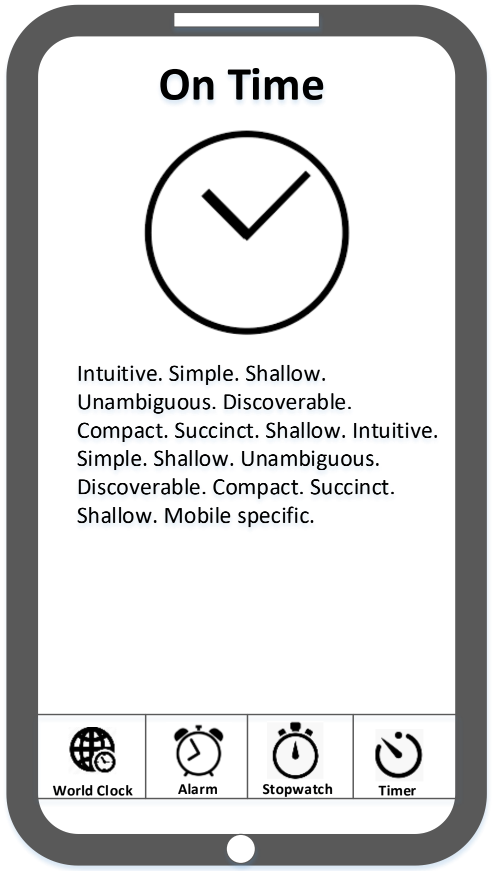
Intuitive. Simple. Shallow. Unambiguous. Discoverable. Compact. Succinct. Shallow. Intuitive. Simple. Shallow. Unambiguous. Discoverable. Compact. Succinct. Shallow. Mobile specific.

Intuitive. Simple. Shallow. Unambiguous. Discoverable. Compact. Succinct. Shallow. Intuitive. Simple. Shallow. Unambiguous. Discoverable. Compact. Succinct. Shallow. Mobile specific.



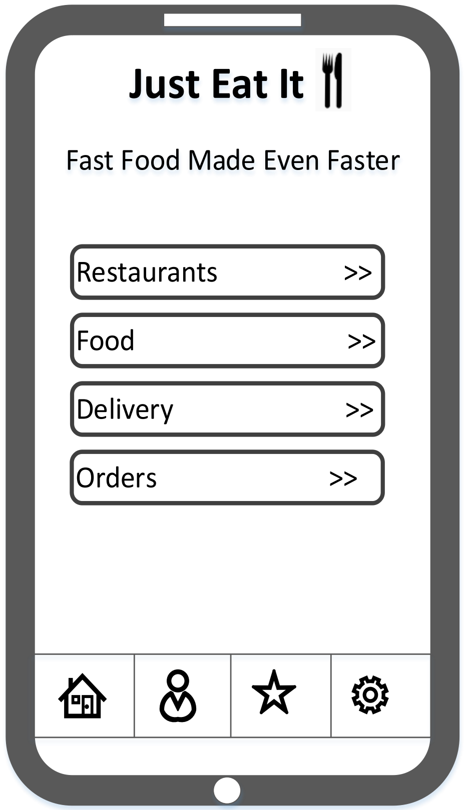
Mobile Web App – Navigation – Tab Bar

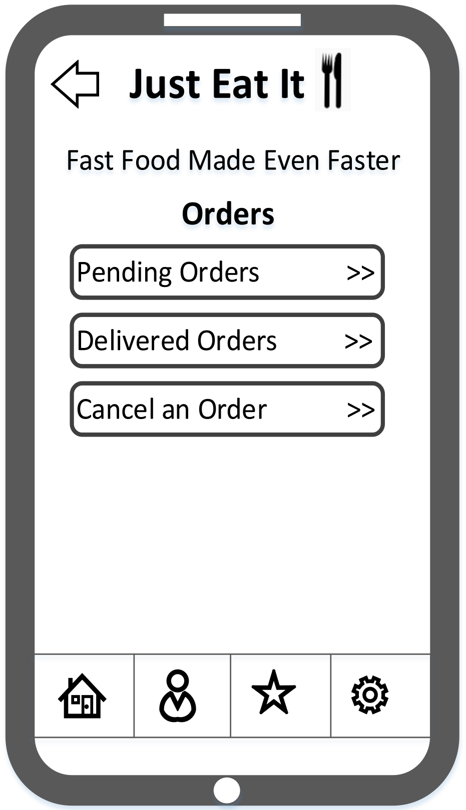
- Best positioned towards the bottom of a page: not towards the top as in desktop.
- Can be fixed or floating. Can also be made to auto hide.
- Works well for apps with few navigation choices, or for toolbars.
- Number of navigation options that can be included is limited by the limited screen width.


On Time
Intuitive. Simple. Shallow. Unambiguous. Discoverable. Compact. Succinct. Shallow. Intuitive. Simple. Shallow. Unambiguous. Discoverable. Compact. Succinct. Shallow. Mobile specific.
World Clock
Alarm
Stopwatch
Timer

Mobile Web App – Navigation – Navigation Hub

- Devotes most of the landing page to primary navigation.
- Further levels of navigation can be created on sub-pages.
- Reverse navigation is enabled by a *back* button.
- Works well for apps with lots of navigation choices and deeper navigation structure.



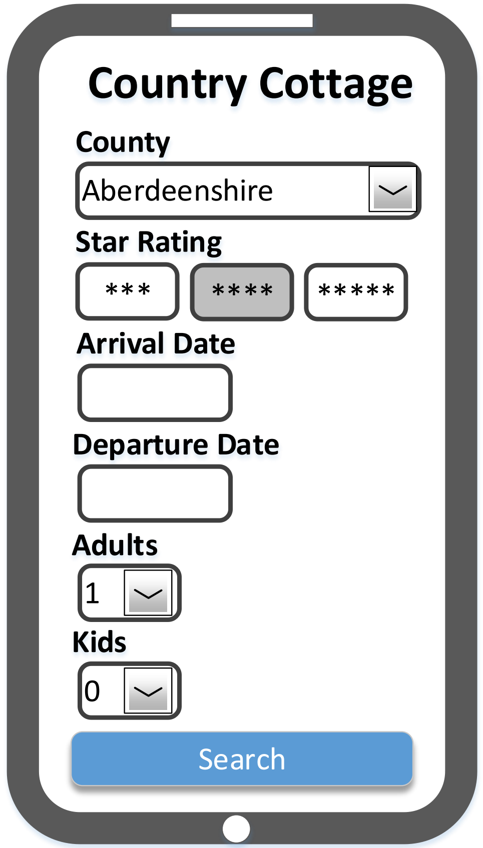


Mobile Web App – Navigation – Search Driven

- Search driven navigation is used for data intensive applications to allow users to quickly find the product or service they require.
- Search functionality should normally be on the landing page (See Zoopla).
- Design of search form is critical to success.
  - Users should have to type in as little data as possible.
  - Form should be intuitive.
  - Form should be easily navigable.

Mobile Web App – Navigation – Search Driven

- A user search returns a list of results in summary format.
- Full details of each search result can be accessed by clicking the result title or a *details* button.



Country Cottage

County
Aberdeenshire

Star Rating
*** | **** | *****

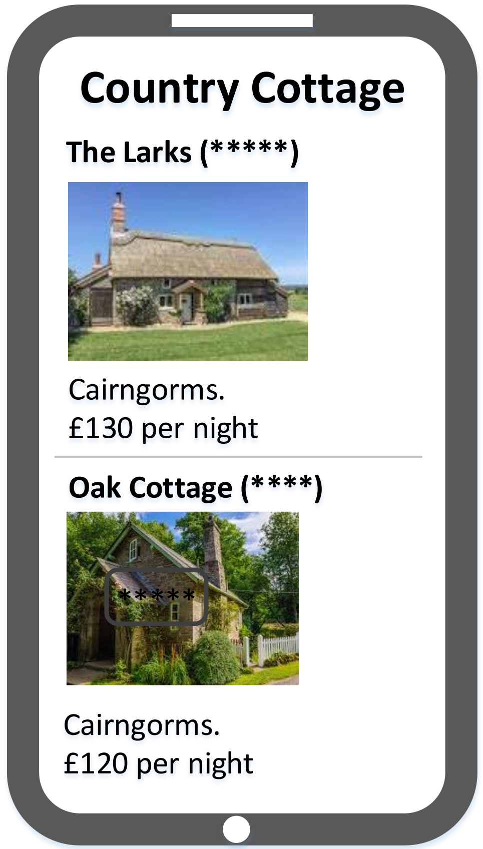
Arrival Date

Departure Date

Adults
1

Kids
0

Search



Country Cottage

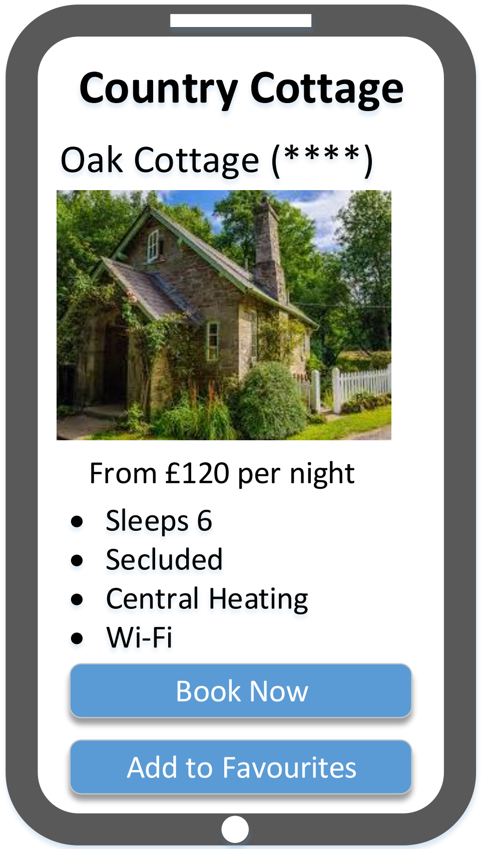
The Larks (*****)

Cairngorms.
£130 per night

Oak Cottage (****)

*****

Cairngorms.
£120 per night



Country Cottage

Oak Cottage (****)

From £120 per night

- Sleeps 6
- Secluded
- Central Heating
- Wi-Fi

Book Now

Add to Favourites

Mobile Web App – Navigation Schemas

Menu	Good for small scale sites. Easy to implement.	Lower discoverability. Requires click to reveal. Less accessible, as menu button tends to be outside thumb zone. Takes up significant screen real estate.
Hamburger menu	Better for larger sites with more complex navigation. Saves screen space.	More difficult to implement: can require JS. Lower discoverability. Less accessible, as hamburger button tends to be outside thumb zone.
Tab bar	Easily accessible when positioned correctly. High discoverability and accessibility.	Allows fewer navigation choices. Takes up significant screen real estate unless auto hide implemented, which typically needs JS.
Navigation hub	Easy to implement. Results in highly intuitive navigation structure. High accessibility and discoverability.	Can prove difficult to scale up to desktop.
Search driven	Good for data intensive sites. Can save significant space. High accessibility and discoverability.	Requires user input to find elements .

Mobile Web App – Implementation Approaches

- There are two approaches to implementing mobile first web applications:
  - Bespoke code
    - Full control over layout. Easier to customise to specific needs. Leaner code.
    - Longer time to implement. Higher development costs. Must be built from scratch. More susceptible to errors and bugs. Greater technical expertise needed.
  - Framework
    - Faster implementation time. Lower development costs. Less susceptible to bugs and errors. Host of predefined components available. Pre-tested in all browsers and screen sizes. Regular updates.
    - Less control over app layout. More difficult to customise to specific needs. Code tends to be more verbose.
- There is no correct approach. The approach you use will depend on your budget, the technical expertise available to you, and the specific technical demands of your project.