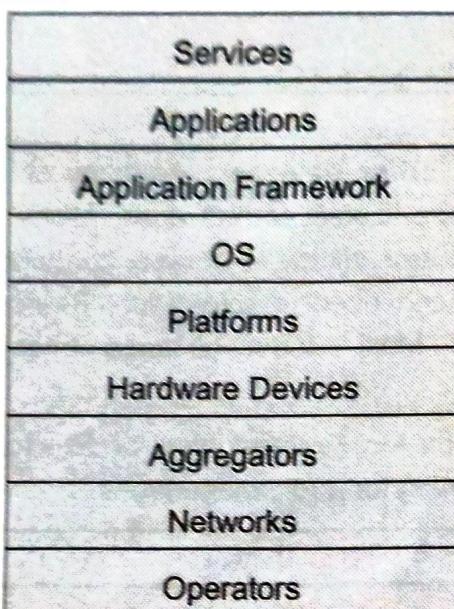


## 6.1 Anatomy of Mobile App

- Mobile devices do not exist in a vacuum - a series of networks and interconnected systems exist to support modern mobility. The utility of modern mobile devices is greatly enhanced by software applications and their supporting cloud services.
- Fig. 6.1.1 shows layers of mobile ecosystem. Each layer is reliant on the others to create a seamless, end-to-end experience.



**Fig. 6.1.1 : Layers of mobile ecosystem**

- Mobile ecosystem consists of operators, networks, aggregators, hardware devices, platforms, OS, application framework, applications and services.

### 1. Operators :

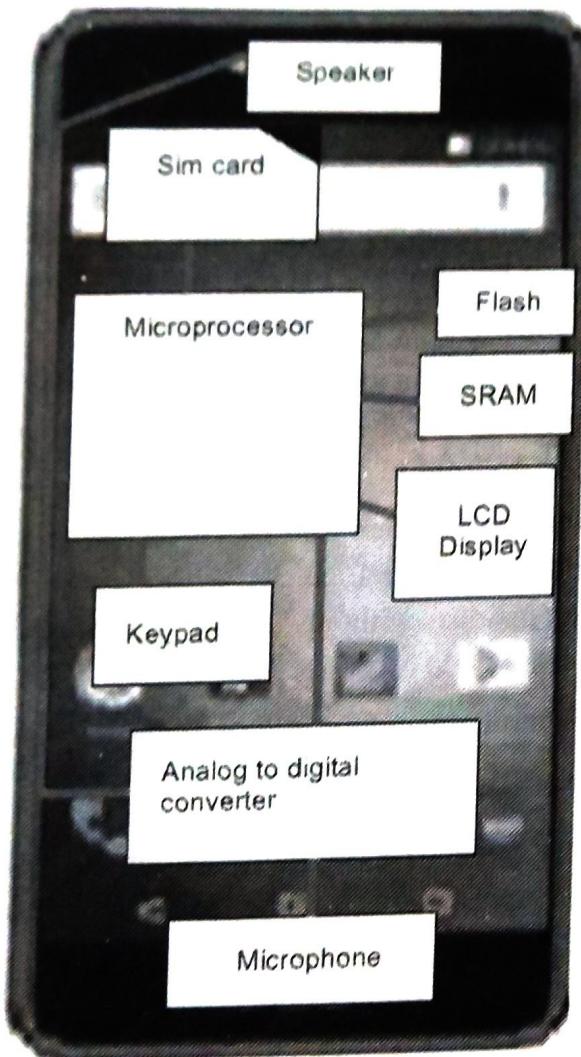
- Operators are the bottom layer of mobile ecosystem. Operators is also referred as mobile network operators, mobile service provider, mobile phone operators or cellular companies. Role of operators in ecosystem is to create and maintain a specific set of wireless services over a reliable cellular network. Various network operators are BSNL, MTNL, Airtel, Reliance communications, Idea and Vodafone etc.

### 2. Networks :

- Operator operates wireless network. A cellular network is a radio telecommunications network that provides wireless communications infrastructure to support mobile radio transceivers (cell phones) located throughout a geographical coverage area.

### 3. Device :

- Cell-phone handset is composed of two components : **Radio Frequency (RF) and Baseband**. RF is the mode of communication for wireless technologies of all kinds, including cordless phones, radar, ham radio, GPS and radio and television broadcasts. Fig. 6.1.2 shows cell phone block diagram.



**Fig. 6.1.2 : Cell phone block diagram**

### 4. Platforms :

- Mobile platforms primary duty is to provide access to the devices. To run software and services on each of these devices. Platforms are of three types : licensed, proprietary and open source.
- Open source is an approach to the design, development and distribution of software, offering practical accessibility to software's source code.

- Open source licenses must permit non-exclusive commercial exploitation of the licensed work, must make available the work's source code and must permit the creation of derivative works from the work itself.
- Proprietary software is computer software which is the legal property of one party. The terms of use for other parties is defined by contracts or licensing agreements. These terms may include various privileges to share, alter, disassemble and use the software and its code.

#### ■ 5. Operating system :

- Mobile OSs provide dedicated application stores for end users offering a convenient and customized means of adding functionality.
- Application stores pose an additional threat vector for attackers to distribute malware or other harmful software to end users. This is especially true of third - party application stores not directly supervised by mobile OS vendors.
- Android is a software environment built for mobile devices. It is not a hardware platform. While components of the underlying OS are written in C or C++, user applications are built for Android in Java.
- Android has been available as open source since October 2008. Google opened the entire source code under an Apache License. With the Apache License, vendors are free to add proprietary extensions without submitting those back to the open source community.
- Other mobile OS are Symbian, windows CE, Palm OS, Linux and Mac OS X etc.

#### ■ 6. Application framework :

- First layer the developer can access is the application framework or API released by one of the companies. An application framework is a software library that provides a fundamental structure to support the development of applications for a specific environment. An application framework acts as the skeletal support to build an application.

Application frameworks often run on top of operating system sharing core services such as communications, messaging, graphics, location, security, authentication etc.

**The Binary Runtime Environment for Wireless (BREW)** gives developers a mechanism for creating portable applications that will work on Code Division Multiple Access (CDMA) - based handsets.

BREW manages all of the telephony functions on the device, thereby allowing application developers to create advanced wireless applications without having to program to the device's system - level interface.

- **S60** : S60 platform is formerly known as Series 60. It is application platform for devices that run the Symbian OS. S60 applications can be created in Java, the Symbian C++ framework etc.
- **Adobe flash lite** is a lightweight version of adobe flash player, a software application published by Adobe Systems for viewing flash content. Flash Lite operates on devices that flash player cannot, such as mobile phones and other portable electronic devices like Wii, Chumby and Iriver.
- **Cocoa touch** is the application development environment for building software programs to run on iOS for the iPhone and iPod Touch, iPadOS for the iPad, watchOS for the Apple Watch and tvOS for the fourth - generation Apple TV, from Apple Inc.

## 7. Applications :

- For creating applications, application framework is used. Applications are web browser, camera or media player.

## 6.2 Mobile Form Factors

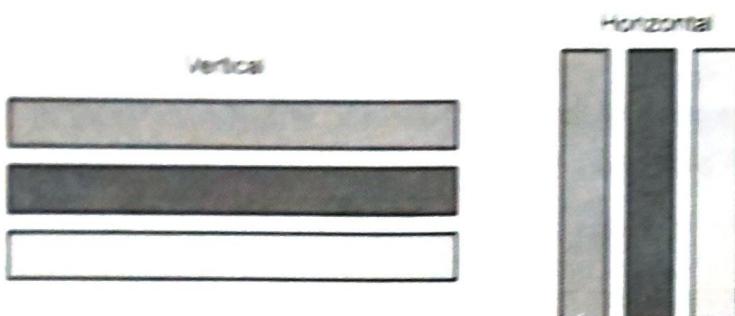
- The form factor of the mobile device has a significant effect on the navigation, the layout and even the behavioural strategies and patterns employed.
- Modern multi - touch mobile devices fall primarily into three form factor categories :
  1. Handhelds consist of phones and media devices like the iPod touch. They are characterized by tall, narrow screens that are 4 to 6 inches diagonally and are used most frequently in portrait orientation.
  2. Tablets consist of devices sporting 9 - to - 10-inch diagonal screens. Android and windows tablets seem biased toward landscape use in their design, while Apple tablets seem to be used more frequently in both orientations.
  3. Mini-tablets consist of devices sporting screens that are 7 to 8 inches diagonally. Like their larger cousins, they have either a 4 : 3 aspect ratio (Apple) or 16 : 9 aspect ratio (Google, Microsoft).

### 6.2.1 Handheld Format Apps

- Mobile apps are very different; the width and height of elements are set by points. The layout does have the ability to wrap, but not re - stack or re - arrange unless it's specifically coded to do so for a particular device size and that can lead to way too much development effort.
- Modern handheld - format devices also continue to use some of the same basic layout patterns employed in these early systems.

### □ 1. Stack

- Primary pattern used by non-game mobile application is stack. It is handheld device. Narrow and tall form factor of smartphones and other handheld mobile device dictates a list-like display for most types of content or control.
- Fig. 6.2.1 shows mobile app use a stack layout pattern including content, control and navigation elements.



**Fig. 6.2.1 : Stack layout**

- Stacks are vertically organized structures with a content area, usually arranged in a list or grid. Most iOS, Android and Windows Phone apps follow this top-level pattern.
- Stack layout positions child elements in a stack either horizontally or vertically based on the orientation property

### □ 2. Screen carousels

- Alternative to top-level pattern and used in dashboard like display. It has multiple instances or variants between which the user can quickly navigate via a swipe gesture to the left or right. The iOS weather app is an example of screen carousels.
- Carousels may or may not have top or bottom bars associated with them, but they usually do have a page marker widget that shows the user's place in the carousel content.
- Carousels don't provide circular flow, but rather disallow further swiping at the far left and right.

### □ 3. Orientation and layout

- Latest mobile devices can detect their screen orientation i.e. portrait or landscape. This helps the app to rearrange its layout to better suit the current orientation. The majority of apps stick with portrait orientation even when rotated.
- For list-and-grid-based content browsing assuming portrait orientation is a good because users usually operate the phone one-handed in portrait orientation. For applications such as photo or video capture and editing, it makes sense to allow rotation to landscape orientation, since the medium itself can be in that orientation.

## 6.2.2 Tablet Format Apps

### 1. Stacks and index panes

- Tablet format apps also rely on the stack pattern, vertically stacking a primary area and one or more tab, navigation and action bars.
- The additional pane is an index pane that lists content items, such as your e - mail inbox or search results. Search and filter widgets are also common controls in tablet index panes.
- In portrait mode, index panes typically are launched from a button and overlap the main content area.

### 2. Pop-up control panels

- Tablet screens are large enough to support pop - up panels that don't overlay the entire screen and that can replace navigation to a full - screen control panel screen, as is usually required in handheld format apps.
- Pop - up panels are different from dialogs, in that they are attached to a particular control or content object and are used to make changes to parameters associated with that object.

## 6.2.3 Mini - Tablet Format Apps

- Mini - tablets such as Google's Nexus 7 and the Kindle Fire are popular, inexpensive mobile devices that handily fit into a large purse or pocket, making them popular with consumers.

Navigation and layout strategies employed by handheld and tablet format apps will work for mini - tablets, with some cautions :

1. Generally, not a good idea on full - sized tablets in portrait orientation, adjacent panes are usually far too cramped to consider on mini - tablets. In landscape, at most two adjacent panes can be supported.
2. Tool bars : In portrait view, these can feel distant from the action due to the tall, narrow form factor and increased screen size over handhelds. In landscape orientation, tool bars stacked with navigation bars leave little vertical space for content. Vertical adjacent panes tool bars may sometimes make more sense on mini - tablets.
3. Lists : Single - column lists tend to look out of proportion on mini - tablets, even in portrait orientation.
4. Pop - up versus full - screen dialogs : Mini - tablets are big enough that using phone - style full - screen idioms for menus and dialogs won't work.

## 6.3 Mobile Navigation, Content and Control Idioms

### 6.3.1 Browse Controls

- Most mobile apps are optimized for browsing. Whether it's music, videos, social networking updates, restaurant reviews, e-mail, shopping or search results, we do a lot of casual surveying in our mobile apps. Due to the limitations of the form factor and input options, it is much easier on mobile devices to browse and select content than to input data.
- Given this situation, it's unsurprising that mobile apps have developed a rich set of patterns around browsing through content.
  1. **Lists** : Lists are the most frequently used pattern for organizing content on handheld format touchscreen devices. List content often includes line items or blocks of text, controls and their labels and image or video thumbnails. Lists can either be finite in length or allow infinite scrolling
  2. **Grids** : Grids are used to organize content such as apps, thumbnails and function icons into regular rows and columns. Grids also can scroll horizontally, as in Apple's Music app when the iPhone is rotated to landscape orientation. grids can be either finite or infinitely scrolling.
  3. **Content carousels** : Content carousels live within a single screen layout but use the same type of gesture to allow navigation between different content objects that are presented within that screen. Content carousels are used to present a relatively small set of objects that the app is meant to feature or highlight.
  4. **Swimlanes** : Swimlanes are a clever mash - up of the carousel concept with a grid. They combine the carousel's natural browsability with the data density that only a grid can permit. Swimlanes are a vertical stack of carousels, each of which can be scrolled horizontally, independent of the others.

### 6.3.2 Navigation and Tool Bars

- Bars are the primary mechanism for navigating to the different functional and content areas of handheld mobile apps. Bars are narrow horizontal regions at the top or bottom of the screen that consist of tab - like or button - like controls with either icons or text labels.

#### 1. Tab bars

- Tab bars contain a set of text and / or icon buttons. Tapping a tab button switches to a different list or grid view in the main content area.

- Each tab in a tab bar maintains its own content hierarchy of associated lists and grids and typically preserves the state of that hierarchy while the app is running.
- Tab bars are frequently found at the bottom of iOS screens and at or near the top of Android and Windows phone screens.

## □ 2. Tab carousels

- Tab carousels uses control addresses. Tabs are shown in the tab bar as usual but extend off the edges of the screen.
- The selected tab is centered or otherwise highlighted in the tab bar. Swiping the tab bar selects the adjacent tab on the left or right and slides the contents into view.

## □ 3. Nav bars and action bars

- Nav bars, located at the top of the screen, provide a way to navigate a list or grid hierarchy.
- Android calls this set of controls an action bar. Frequently, function menus or buttons are included on the right.
- Most versions of Android have a system - level navigation bar at the bottom. It contains a back control, a home control and a “recents” control.
- The presence of a ubiquitous bar at the bottom means that Android apps typically place most of their app navigation at the top of the screen.

## □ 4. Tool bars and palettes

- Tool bars contain buttons that execute functions on the current or selected app content.
- Tool palettes, a variant of tool bars similar to their desktop brethren, use iconic buttons as a way to access tools that operate on a document.

## □ 5. Tool carousels

- It seems particularly popular with image processing apps such as Google's snapseed.
- Each item in the tool carousel is a labeled thumbnail that both describes and shows a small example of the filter or effect applied to an image.



### 6.3.3 Drawers

- Drawers are a clever idiom that provides access to a vertical list of navigational elements similar to tabs.
- They use minimal screen real estate by hiding in a panel that lives in a layer under the main content area. The drawer icon is also called the hamburger menu icon due to its shape: three short, stacked lines.

- Tapping a drawer item simultaneously swaps what is displayed in the content area and snaps the drawer back shut. Items in the drawer are usually textual, but may have icons and other adornments

#### Secondary - action drawers

- Drawers can be used to replace a navigational tab bar or can be used to interact with a secondary set of objects in the app. Drawers usually slide open from the left, but not always.
- Secondary - action drawers offers a right - hand drawer that gives you access to a list of online friends for chatting.

#### Double drawers

- The Path design uses two drawers, a standard left - hand drawer for primary navigation between views and a Facebook - like right - hand drawer for messaging friends.

### 6.4 Tap - to - Reveal and Direct Manipulation

#### Tap - to - reveal controls

- The iDraw app provides a good example tap - to - reveal control. Tap an object and the manipulation tools are revealed.
- Streaming video apps use the tap - to - reveal idiom for controls that normally are hidden during playback. Tapping anywhere on the video playback area of the You - Tube app launches transport, volume and other controls.
- Direct manipulation : Rapid feedback is just one feature of the interaction technique known as direct manipulation.
- Here users act on displayed objects of interest using physical, incremental, reversible actions whose effects are immediately visible on the screen.
- Features of a direct manipulation interface :
  1. Visibility of the objects of interest
  2. Incremental action at the interface with rapid feedback on all actions
  3. Reversibility of all actions, so that users are encouraged to explore without severe penalties
  4. Syntactic correctness of all actions, so that every user action is a legal operation
  5. Replacement of complex command languages with actions to manipulate directly the visible objects.

## 6.5 Searching, Sorting and Filtering

- Searching is a key user activity on mobile apps. A smart app might keep track of the kinds of things the user has viewed, liked or purchased in the past.

### Building search queries

- The challenge in a mobile app context is to allow sufficient expression of search terms, but with a minimum of data entry for the user. Here are a few of the most useful approaches :
  1. **Voice search** : The three major mobile platforms support in - app voice search. Voice search can certainly ease entry for simple searches in supported domains.
  2. **Auto - complete** : As the user types, displaying a list of popular options matching the entered letters can dramatically decrease keyboard time and user frustration.
  3. **Tap - ahead** : This is a refinement on top of auto - complete. Tap - ahead allows users to take any auto - completed term option the app provides as the result of auto - suggest, load it into the search box and run a new auto - complete query.
  4. **Recent / frequent searches** : Humans are creatures of habit who typically search for the same things repeatedly. Any search functionality should remember past searches and present them as soon as the user taps the search box.
- **Sorting and filtering** : On mobile devices, sorting and filtering often amount to the same thing.

## 6.6 Welcome and Help Screens

- Welcome and help interfaces are most often two sides of the same coin in mobile apps.
- Welcome screens provide guidance on what the important activities are in the app and how to perform them.
- Help in a mobile context provides much the same and often identical information, but on demand when the user requests it.
- **Guided tours** usually consist of a carousel of cards, each of which contains text and images or video that describes the use of a particular function or set of related app functions.
- **Overlay** covers the entire screen with a semi - transparent layer on which instructions often rendered to look hand - drawn and employing arrows to indicate gestures or highlight controls are displayed. Tapping anywhere on the screen dismisses the overlay.
- **ToolTip** overlays are an overlay variation that attempts to provide a ToolTip - like display of all primary functions on a single overlay screen.

## 6.7 Multi - Touch Gestures

1. Tap to select, activate or toggle : The tap is used to select objects and toggle the activation state of controls. Tapped items should get an appropriate selection highlight or activation / deactivation state or animation.
2. Tap - and - hold : Tap - and - hold is a gestural idiom that is falling out of favor and probably rightfully so. It is typically used to open a contextual pop - up menu on an object.
3. Drag to scroll can work horizontally or vertically and is a fundamental direct manipulation gesture.
4. Drag to move : Dragging also can be used to move or copy an object from one list, pane or container to another or to move an object arbitrarily within a canvas or grid.
5. Drag to control : Dragging also can be used to control knobs, switches, sliders, virtual x - y control pads and contextual touch controls and to operate palette tools on a canvas.
6. Swiping left usually is synonymous with dragging left. Swiping left also can open a right - hand drawer or close a left - hand drawer.
7. Swipe right usually is synonymous with dragging right. Swiping right also can open a left - hand drawer or close a right - hand drawer.
8. Pinch in / out : The pinch - in gesture is used to shrink or zoom out on objects physically. The pinch - out gesture is used to expand or zoom in on objects physically.
9. Rotate is a gesture employing the thumb and forefinger twisted clockwise or counter clockwise together on the touchscreen surface.

## 6.8 Other Device

- Kiosks and embedded systems, like TVs, microwave ovens, automobile dashboards, cameras, bank machines and laboratory equipment, are unique platforms with their own opportunities and limitations.

### General design principles

- When designing any embedded system, whether it is a smart appliance, kiosk system or handheld device, following principles are considered :
  1. Integrate hardware and software design.
  2. Let context drive the design.
  3. Use modes judiciously, if at all.
  4. Specify the scope.

5. Balance navigation with display density.
6. Minimize input complexity.

### 6.8.1 Designing for Kiosks

- Kiosks may appear to have much in common with desktop interfaces : Large, colorful screens and reasonably processors behind them.
- Kiosk users will have either a specific goal in mind when approaching a kiosk or no readily definable goal at all.
- Kiosk users typically don't have access to keyboards or pointing devices and often they would be unable to use either effectively even if they did.
- Kiosk users typically are in a public environment, full of noise and distractions.
- Kiosks are of two types : Transactional and explorational.
- Transactional kiosks provide a tightly scoped transaction or service. These include ATM, ticketing machines such as those used in airports, train and bus depots.
- Gasoline pumps and vending machines can be considered a simple type of transactional kiosk. Users of transactional kiosks have specific goals in mind : To get cash, a ticket, specific piece of information.
- Explorational kiosks are most often found in museums or as an information display in shopping mall.

### Android kiosks

- Android kiosks are customer facing devices that serve a single purpose by running only a single app or a specific set of apps.
- Mobile devices like smartphones and tablets can now be used as Android kiosks by locking down devices to run a single app or a specific set of apps, using Android Kiosk Mode.
- Organizations use Android kiosks as customer facing kiosks, digital signage, patient check - in devices etc. Generally, a dedicated Kiosk app is used to enable Kiosk mode on these Android devices.

### 6.8.2 Designing for 10 - Foot Interfaces

- Television - based interfaces is also called 10 - foot interfaces. Users typically operate it when they are sitting across the room from the television. Most remote controls use inexpensive infrared light to communicate.

- Lists, grids, carousels and swimlanes all map reasonably well to 10 - foot interfaces, because D-pad navigation consists of up - down and left - right movements.
- Use a screen layout and visual design that can be easily read from across the room. Keep onscreen navigation simple. Keep remote controls as simple as possible.

### 6.8.3 Designing for Automotive Interfaces

- Automotive interfaces, especially those that offer sophisticated navigation and entertainment functionality, have the particular challenge of driver safety.
- Following parameters are consider :
  - a) Choose input mechanisms carefully
  - b) Use direct control mappings when possible
  - c) Enforce consistent layout from screen to screen
  - d) Minimize the amount of time hands are off the wheel
  - e) Keep mode/context switching simple and predictable.

### 6.8.4 Designing for Audible Interfaces

- Principles for designing usable audible interfaces
  - a) Organize and name functions according to user mental models
  - b) Always signpost the currently available functions
  - c) Always provide a way to go back one step and to return to the top level
  - d) Always provide a means to speak with a human
  - e) Give the user enough time to respond.

## 6.9 Multiple Choice Questions

**Q.1** Mobile ecosystem consists of \_\_\_\_\_.

- |  |   |
|--|---|
| <input type="checkbox"/> a operators, networks       | <input type="checkbox"/> b aggregators, hardware devices, |
| <input type="checkbox"/> c OS, application framework | <input type="checkbox"/> d all of these                   |

**Q.2** The form factor of the \_\_\_\_\_ has a significant effect on the navigation, the layout and even the behavioural strategies and patterns employed.

- |                                     |  |
|-------------------------------------|--|
| <input type="checkbox"/> a computer | <input type="checkbox"/> b mobile device |
| <input type="checkbox"/> c TV       | <input type="checkbox"/> d mouse         |