20CSI605 - Mobile Application Development

User Interaction

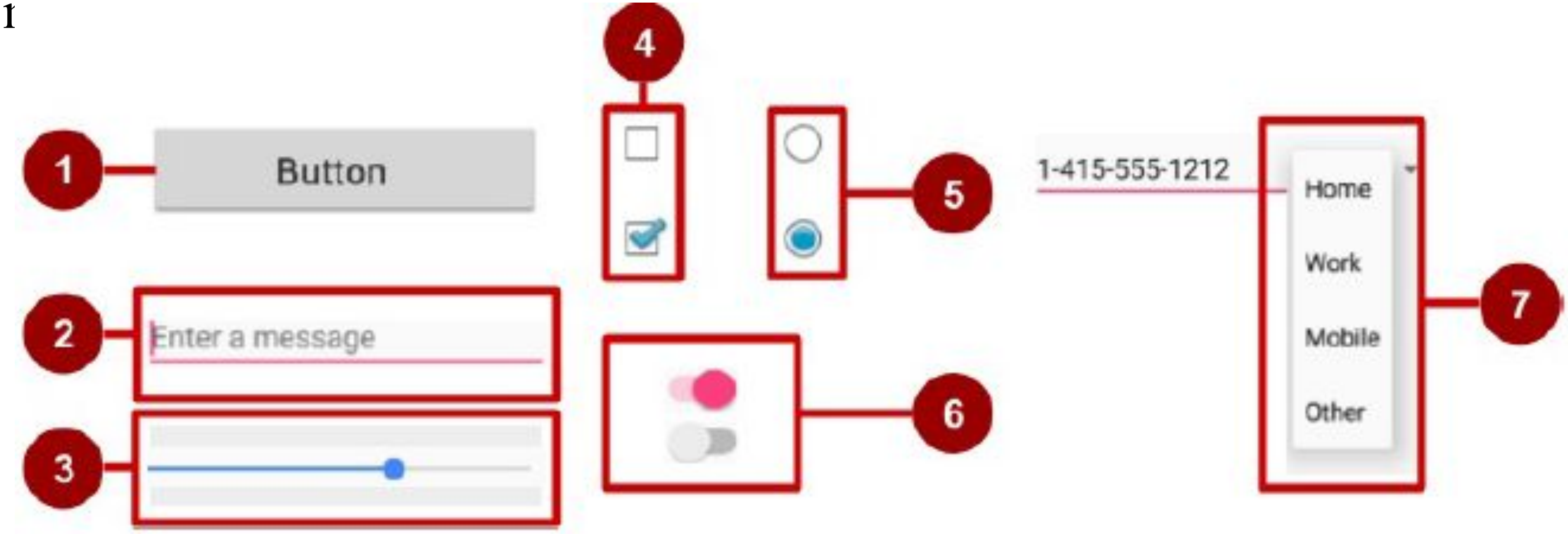
1.User Input Controls

Interaction design for user input

- Provide some function for a user to interact
- Interaction includes tapping, pressing, typing, or talking and listening
- The framework provides UI elements such as buttons, menus, keyboards, text entry fields and a mircophone.
- Android users will become familiar with UI elements by using these input control elements.

1.User Input Controls

You can use a wide variety of input controls in your UI, such as text fields, buttons, checkboxes, radio buttons, toggle buttons, spinners, and ı



1.Button 2.Text field 3.Seek bar 4.Checkboxes 5.Radio buttons

6.Toggle 7.Spinner

1.User Input Controls

Input Controls and view focus

- Android applies a common programmatic abstraction to all input controls called a *view.*
- *The View class represents the basic* building block for UI components, including input controls.
- View is the base class for classes that provide support for interactive UI components, such as buttons, text fields, and layout managers.

1.User Input Controls

- The View that "has the focus" will be the component that receives user input.
- *Focus indicates which view is currently selected to receive input. Focus can be initiated by the user by touching a View,* such as a TextView or an EditText object.
- You can define a focus order in which the user is guided from UI control to UI control using the Return key, Tab key, or arrow keys.
- Focus can also be programmatically controlled; a programmer can requestFocus() on any View that is focusable.

1.User Input Controls

- Another attribute of an input control is clickable. If this attribute is (boolean) true , then the View can react to click events.
- As it is with focus, clickable can be programmatically controlled.
- The difference between *clickable and focusable is that clickable means the view can be clicked or tapped, while focusable* means that the view is allowed to gain focus from an input device such as a keyboard.
- Input devices like keyboards can't determine which view to send their input events to, so they send them to the view that has focus.

1.User Input Controls

- □ Android device input methods are becoming quite diverse: directional pads, trackballs, touch screens, keyboards, and more.
- □ Some devices, like tablets and smartphones, are primarily navigated by touch.
- □ When a user is navigating through a user interface with an input device such as directional keys or a trackball, it is necessary to:
- ✓ Make it visually clear which view has focus, so that the user knows where the input goes.
- ✓ Explicitly set the focus in your code to provide a path for users to navigate through the input elements using directional keys or a trackball.

1.User Input Controls

□ You can also explicitly set the focus or find out which view has focus by using the following methods:

✔ Call onFocusChanged to determine where focus came from.

✔ To find out which view currently has the focus, call Activity.getCurrentFocus(), or use ViewGroup.getFocusedChild() to return the focused child of a view (if any).

✔ To find the view in the hierarchy that currently has focus, use findFocus().

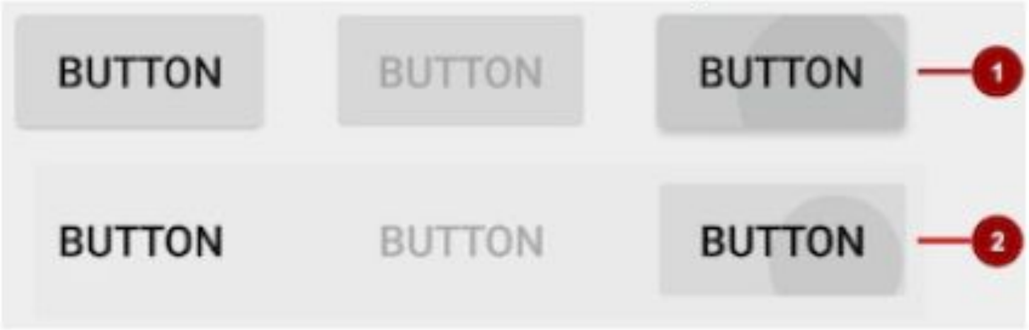
✔ Use requestFocus to give focus to a specific view.

✔ To change whether a view can take focus, call setFocusable.

✔ To set a listener that will be notified when the view gains or loses focus, use setOnFocusChangeListener.

1.User Input Controls

- Android offers several types of buttons, including raised buttons and flat buttons as shown in the figure below.
- These buttons have three states: normal, disabled, and pressed.

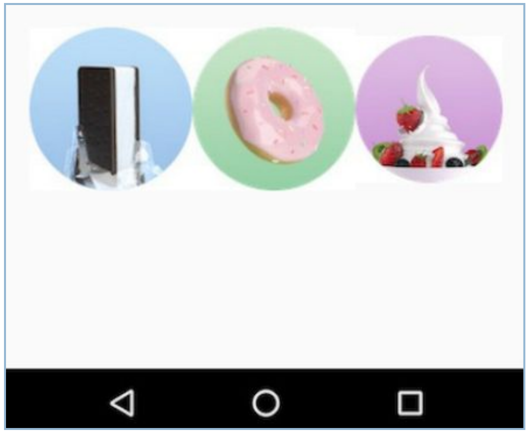


In the above figure:

1. Raised button in three states: normal, disabled, and pressed.
2. Flat button in three states: normal, disabled, and pressed.

1.User Input Controls

- You can turn any View, such as an ImageView, into a button by adding the android:onClick attribute in the XML layout.
- The image for the ImageView must already be stored in the **drawables folder of your project.**
- Note: To bring images into your Android Studio project, create or save the image in JPEG format, and copy the image file into the app > src > main > res > drawables folder of your project



1.User Input Controls

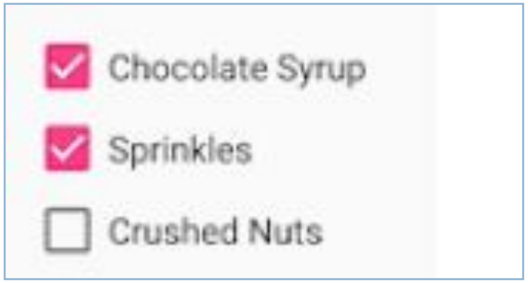
Using input controls for making choices

- Android offers ready-made input controls for the user to select one or more choices:
- **Checkboxes:** Select one or more values from a set of values by clicking each value's checkbox.
- **Radio buttons:** Select only one value from a set of values by clicking the value's circular "radio" buttonToggle button: Select one state out of two or more states. Toggle buttons usually offer two visible states, such as "on“ and "off".
- **Spinner:** Select one value from a set of values in a drop-down menu. Only one value can be selected.

1.User Input Controls

Checkboxes

- Use checkboxes when you have a list of options and the user may select *any number of choices, including no choices*.
- Each checkbox is independent of the other checkboxes in the list, so checking one box doesn't uncheck the others.
- Users expect checkboxes to appear in a vertical list, like a to-do list, or side-by-side horizontally if the labels are short.



☑ Chocolate Syrup

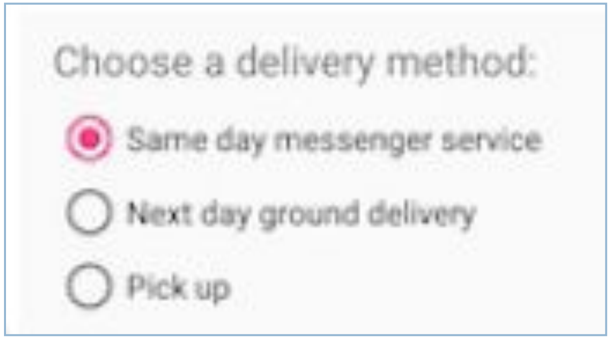
☑ Sprinkles

☐ Crushed Nuts

1.User Input Controls

Radio buttons

- Use radio buttons when you have two or more options that are mutually exclusive—the user must select only one of them.
- Users expect radio buttons to appear as a vertical list, or side-by-side horizontally if the labels are short



Choose a delivery method:

● Same day messenger service

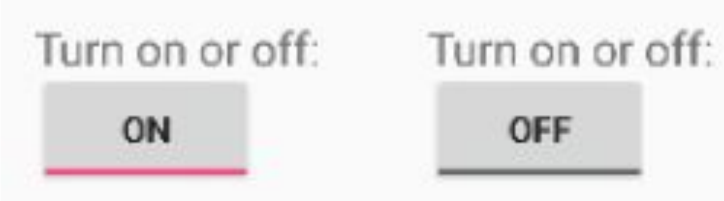
○ Next day ground delivery

○ Pick up

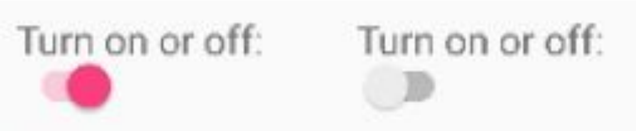
1.User Input Controls

Toggle buttons and switches

- A toggle input control lets the user change a setting between two states.
- Android provides the ToggleButton class, which shows a raised button with "OFF" and "ON".



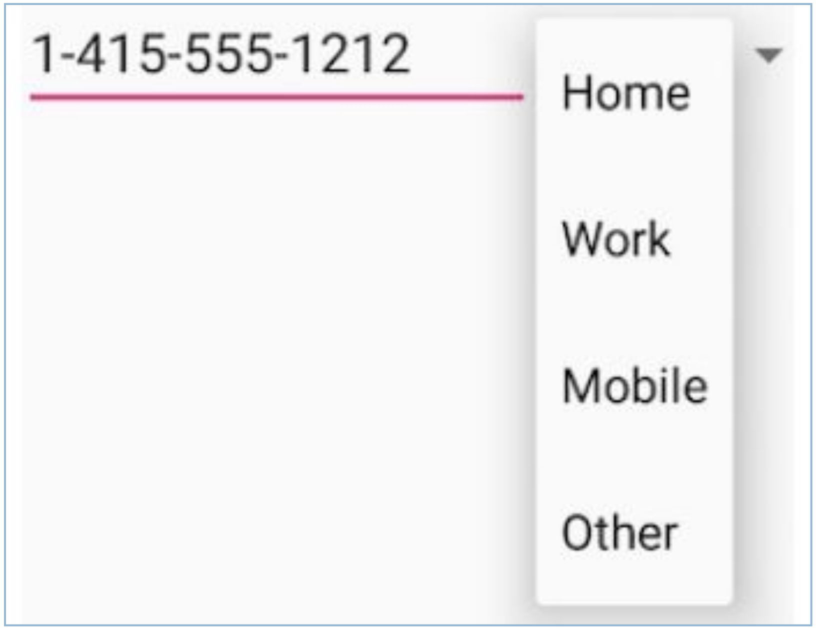
- Examples of toggles include the On/Off switches for Wi-Fi, Bluetooth, and other options in the Settings app.



1.User Input Controls

Spinners

- □ A *spinner provides a quick way to select one value from a set. Touching the spinner displays a drop-down list with all* available values, from which the user can select one.

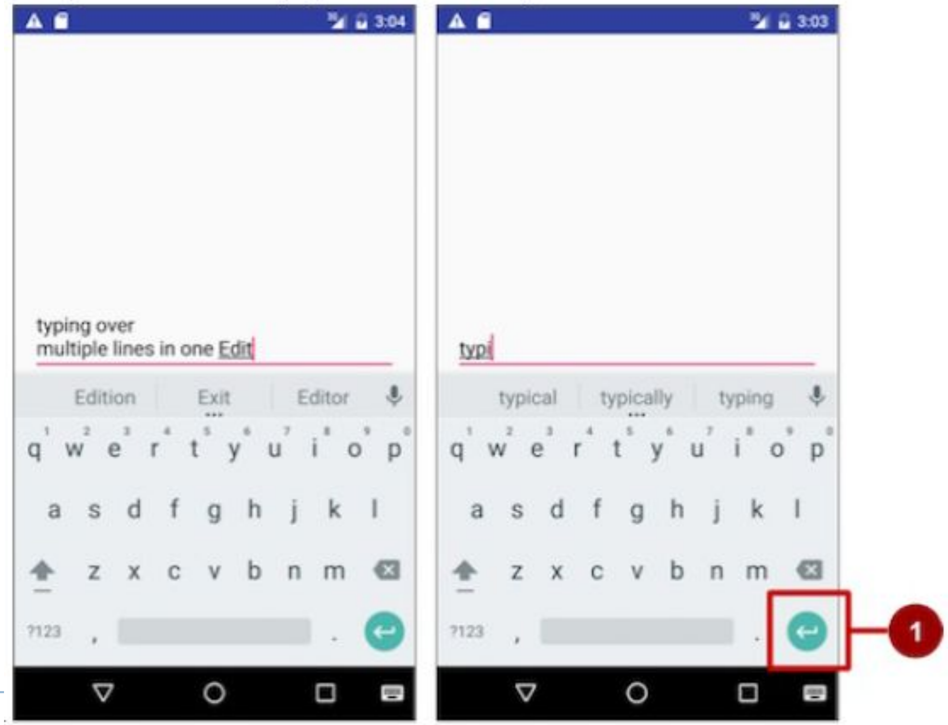


1-415-555-1212

1.User Input Controls

Spinners

- Use the EditText class to get user input that consists of textual characters, including numbers and symbols.
- EditText extends the TextView class, to make the TextView editable



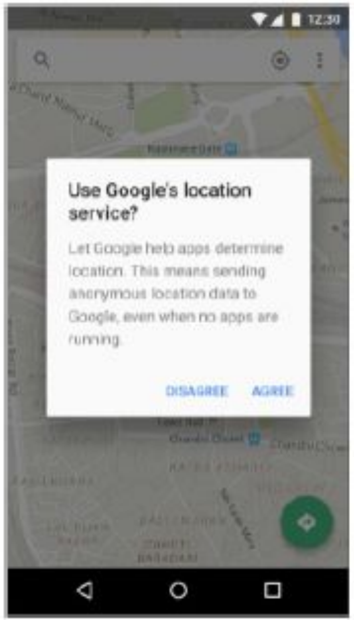
typing over
multiple lines in one Edit

typi

1.User Input Controls

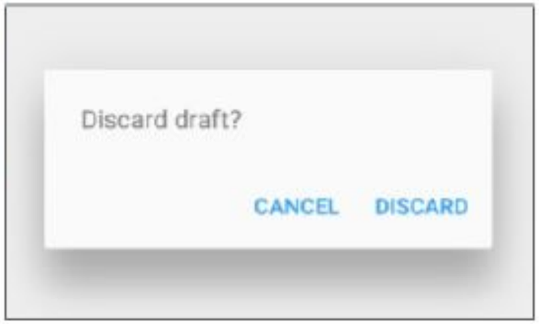
Using dialogs and pickers

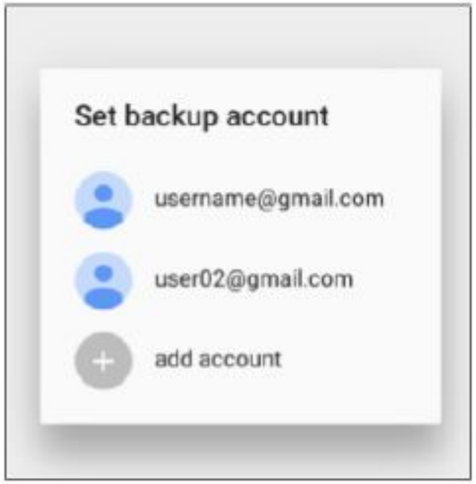
- A *dialog is a window that appears on top of the display or fills the display, interrupting the flow of activity.*
- *Dialogs inform* users about a specific task and may contain critical information, require decisions, or involve multiple tasks.



Use Google's location service?

Let Google help apps determine location. This means sending anonymous location data to Google, even when no apps are running.


Discard draft?


Set backup account

username@gmail.com

user02@gmail.com

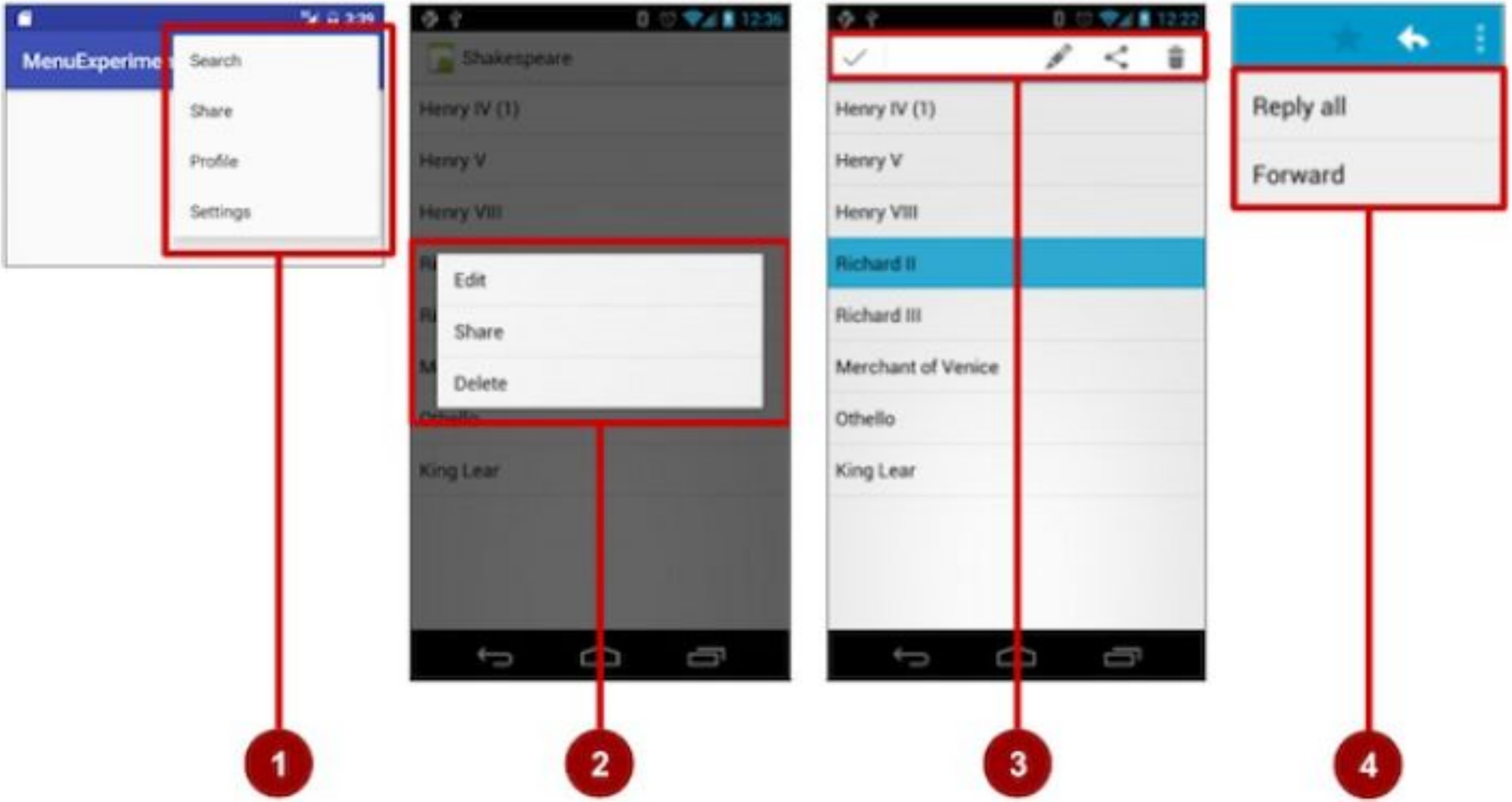
add account

2.Menus

Types of menus

- A menu is a set of options the user can select from to perform a function, such as searching for information, saving information, editing information, or navigating to a screen.
- Android offers the following types of menus, which are useful for different situations

2.Menus


MenuExperime
Search
Share
Profile
Settings
Shakespeare
Henry IV (1)
Henry V
Henry VIII
Edit
Share
Delete
King Lear
Henry IV (1)
Henry V
Henry VIII
Richard II
Richard III
Merchant of Venice
Othello
King Lear
Reply all
Forward
1
2
3
4

2.Menus

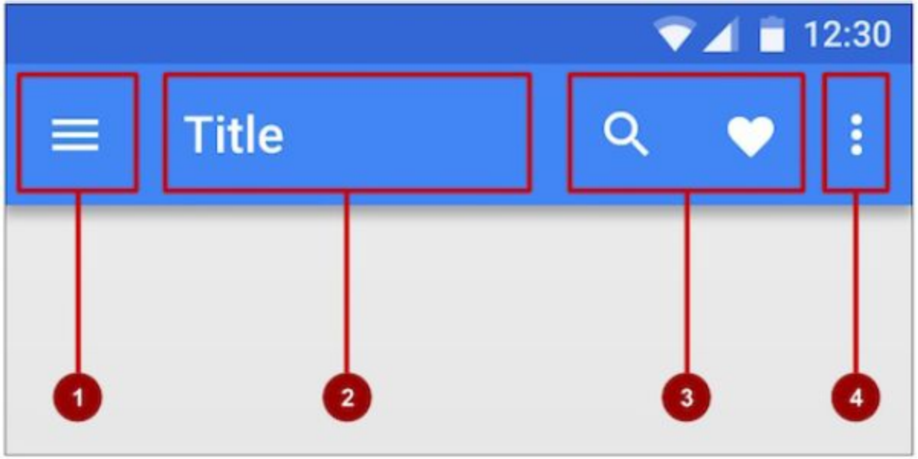
Types of menus

1. ***Options menu:*** *Appears in the app bar and provides the primary options that affect using the app itself.*
2. ***Context menu:*** *Appears as a floating list of choices when the user performs a long tap on an element on the screen.*
3. ***Contextual action bar:*** *Appears at the top of the screen overlaying the app bar, with action items that affect the selected* element(s).
4. ***Popup menu:*** *Appears anchored to a view such as an ImageButton, and provides an overflow of actions or the second* part of a two-part command.

2.Menus

The app bar and options menu

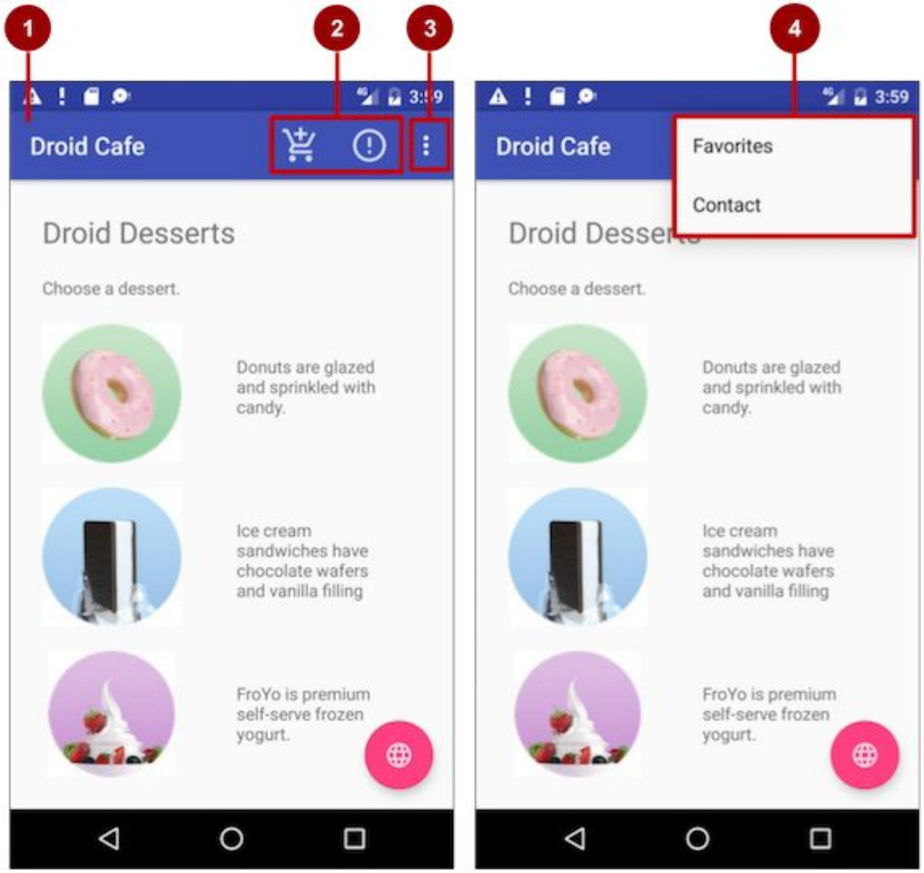
- The app bar (also called the action bar) is a dedicated space at the top of each activity screen.
- The *options menu in the app bar provides navigation to other activities in the app, or the primary options that affect using* the app itself — but not ones that perform an action on an element on the scr



1.Navigation button
2.Title
3.Action icons
4.Overflow options

2.Menus

- Frequently-used options menu items should appear as icons in the app bar. The overflow options menu shows the rest of the menu:



Donuts are glazed and sprinkled with candy.

Ice cream sandwiches have chocolate wafers and vanilla filling

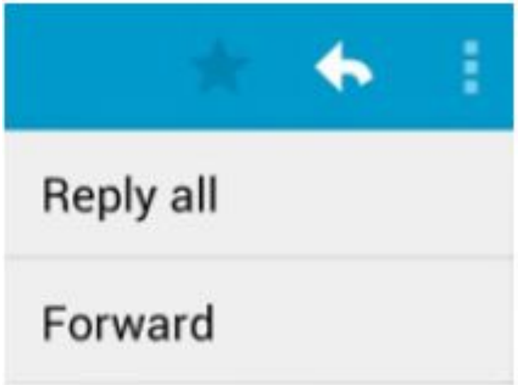
FroYo is premium self-serve frozen yogurt.

1.App bar
2.Option menu
3.Overflow button
4.Options overflow menu

2.Menus

Popup menu

- A PopupMenu is a vertical list of items anchored to a View. It appears below the anchor view if there is room, or above the view otherwise.



3.Screen Navigations

- Providing users with a path through your app
- Back-button navigation
- Hierarchical navigation patterns
- Ancestral navigation (the Up button)
- Descendant navigation
- Lateral navigation with tabs and swipes

3.Screen Navigations

Providing users with a path through your app

- Each path enables users to navigate across, into, and back out from the different tasks and pieces of content within the app.

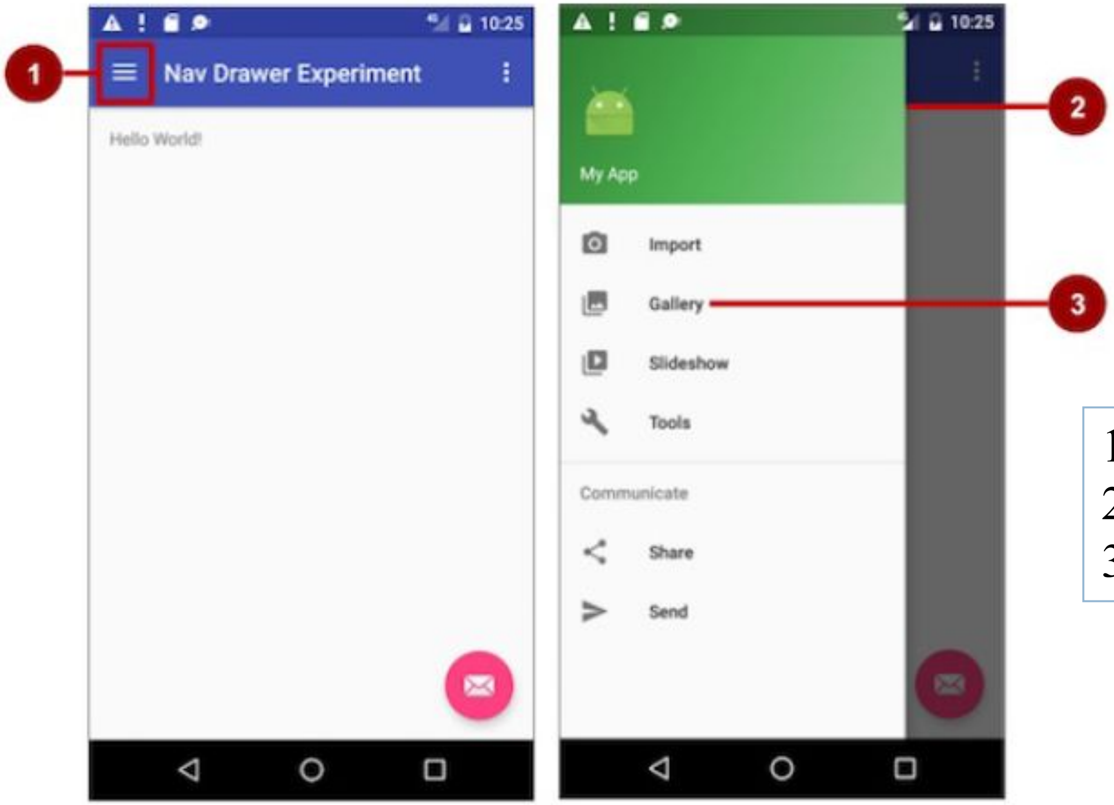
In many cases you will need several different paths through your app that offer the following types of navigation:

- ✔ *Back navigation: Users can navigate back to the previous screen using the Back button.*
- ✔ *Hierarchical navigation: Users can navigate through a hierarchy of screens organized with a parent screen for every* set of *child screens.*

3.Screen Navigations

Navigation drawer

- A *navigation drawer is a panel that usually displays navigation options on the left edge of the screen, as shown on the right* side of the figure b



1.Navigation icon
2.Navigation drawer
3. Navigation drawer menu

THANK YOU