# Unit II:
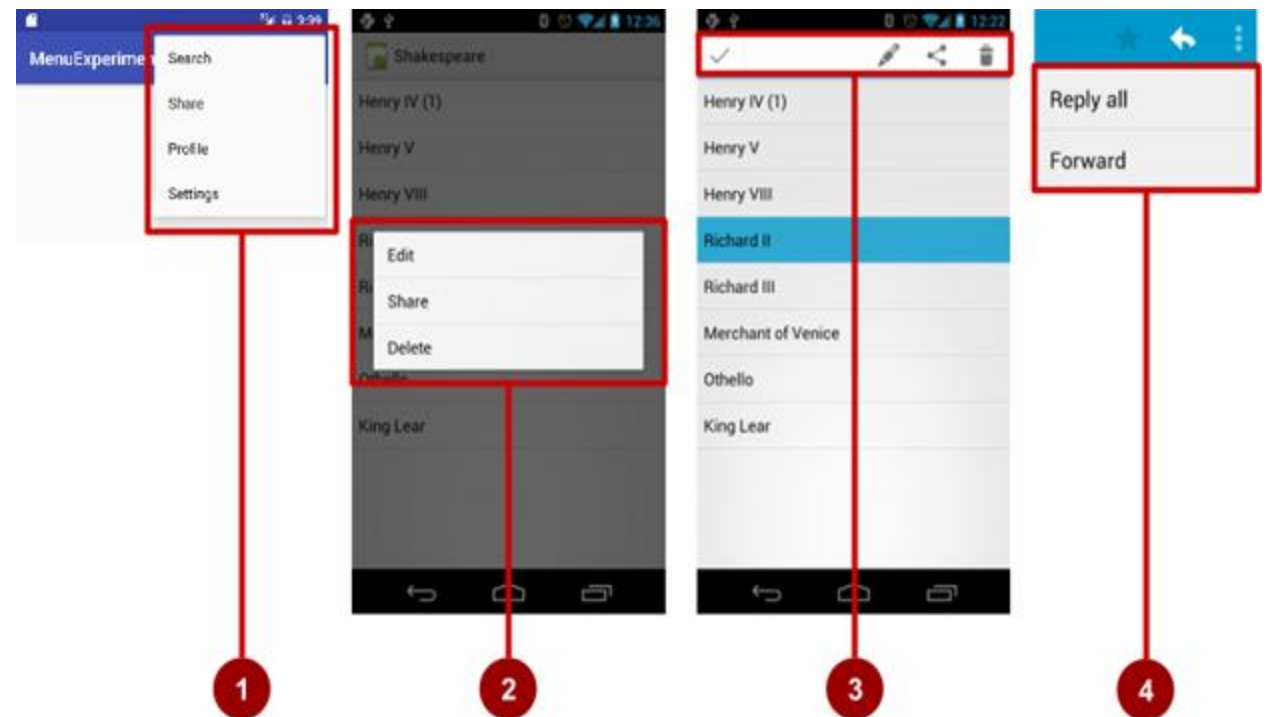# Menu

## CTE308- AS2025

Tutor: Pema Galey

#17682761

# Outline

- App Bar with Options Menu

- Contextual menus

- Popup menus
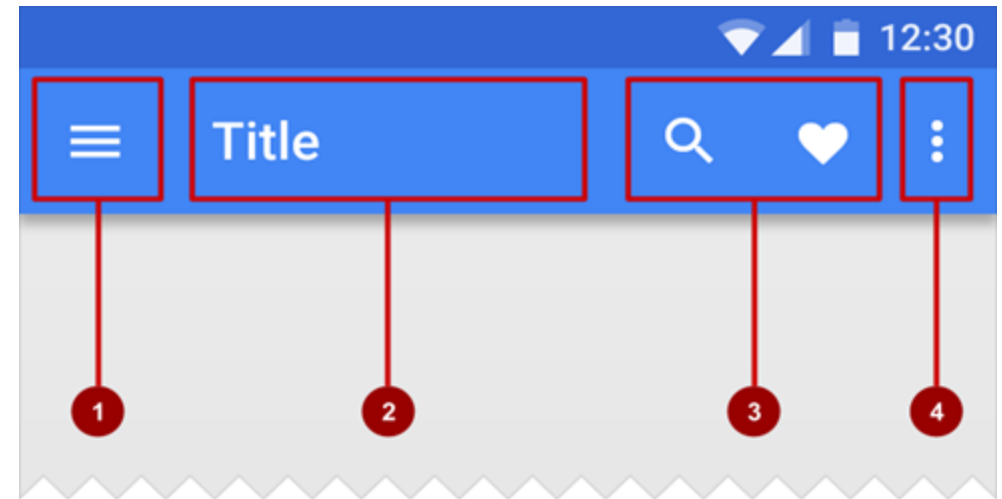
# Different Types of Menu (Four Types)

- Different Types of Menu (Four Types)
    1. App bar with options menu.
    2. Contextual menu.
    3. Contextual action bar.
    4. Popup menu.

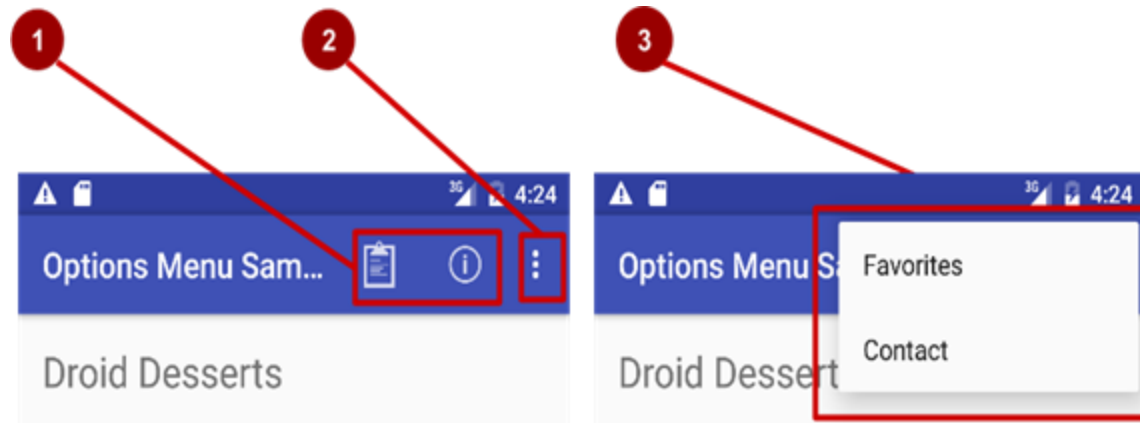# App Bar with Options Menu

Bar at top of each screen—(usually) the same for all screens.

1. Navigation icon to open navigation drawer

2. Title of current activity

3. Icons for options menu items

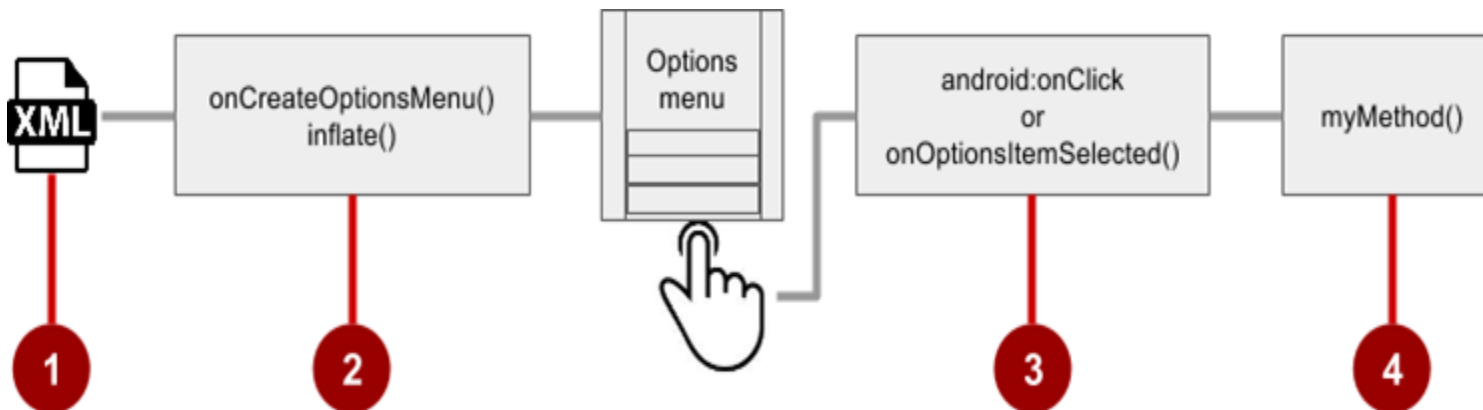4. Action overflow button for the rest of the options menu

## Options Menu?

- Action icons in the app bar for important items (1)
- Tap the three dots, the "action overflow button" to see the options menu (2)
- Appears in the right corner of the app bar (3)
- For navigating to other activities and editing app settings

# Options Menu

- Steps to implement options menu:
  1. XML menu resource (menu_main.xml)
  2. *onCreateOptionsMenu()* to inflate the menu
  3. onClick attribute or *onOptionsItemSelected()*
  4. Method to handle item click

## Creating XML menu resource(menu_main.xml)

Steps to create XML menu resource:

1. Create menu resource directory
2. Create XML menu resource (menu_main.xml)
3. Add an entry for each menu item

```
<item  android:id="@+id/item1"
        android:title="Item One" />


<item  android:id="@+id/item2"
        android:title="Item Two" />
```
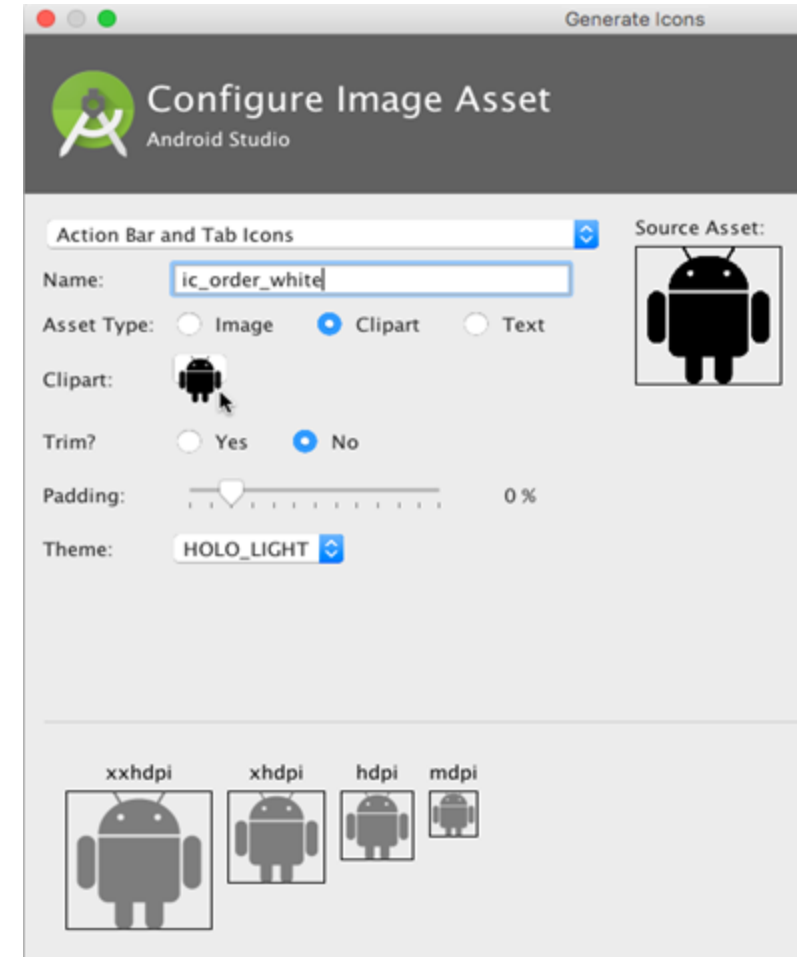
## Inflate Options Menu

Override onCreateOptionsMenu() in main activity

```java
@Override
public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.menu_main, menu);
        return true;
}
```

# Procedures to ADD icons for menu items

1. Right-click drawable
2. Choose **New > Image Asset**
3. Choose Action Bar and Tab Items
4. Edit the icon name
5. Click clipart image, and click icon
6. Click Next, then Finish

## Attributes of the Options Menu

```
<item
    android:id="@+id/action_order"
    android:icon="@drawable/ic_toast_dark"
    android:title="@string/toast"
    android:orderInCategory="1"
    app:showAsAction="ifRoom" />
```

# Handling Click Event

When user selects an item, system calls your activity's **onOptionsItemSelected(MenuItem item)** method.

```java
@Override
public boolean onOptionsItemSelected(@NonNull MenuItem item) {
    if (item.getItemId()==R.id.item1)
        Toast.makeText(this,item.getTitle(),Toast.LENGTH_LONG).show();
    else if (item.getItemId()==R.id.item2)
        Toast.makeText(this,item.getTitle(),Toast.LENGTH_LONG).show();
    else if (item.getItemId()==R.id.item3)
        Toast.makeText(this,item.getTitle(),Toast.LENGTH_LONG).show();
    return super.onOptionsItemSelected(item);
}
```

## Code conversion (if-else-if to switch)

```java
@Override

public boolean onOptionsItemSelected(@NonNull MenuItem item) {

    switch (item.getItemId()) {

        case R.id.item1:

            Toast.makeText(this, item.getTitle(), Toast.LENGTH_LONG).show();

            break;

        case R.id.item2:

            Toast.makeText(this, item.getTitle(), Toast.LENGTH_LONG).show();

            break;

        case R.id.item3:

            Toast.makeText(this, item.getTitle(), Toast.LENGTH_LONG).show();

            break;

         default:

            return super.onOptionsItemSelected(item);

    }

    }
```

# Override onOptionsItemSelected()

```java
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.action_order:
            showOrder();
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}
```

# Contextual Menu

- **Contextual Menu**
  - Allow users to perform an action on a selected view or content
  - Can be deployed on any View object, but most often used for items in a **RecyclerView**, **GridView**, or other view collection.
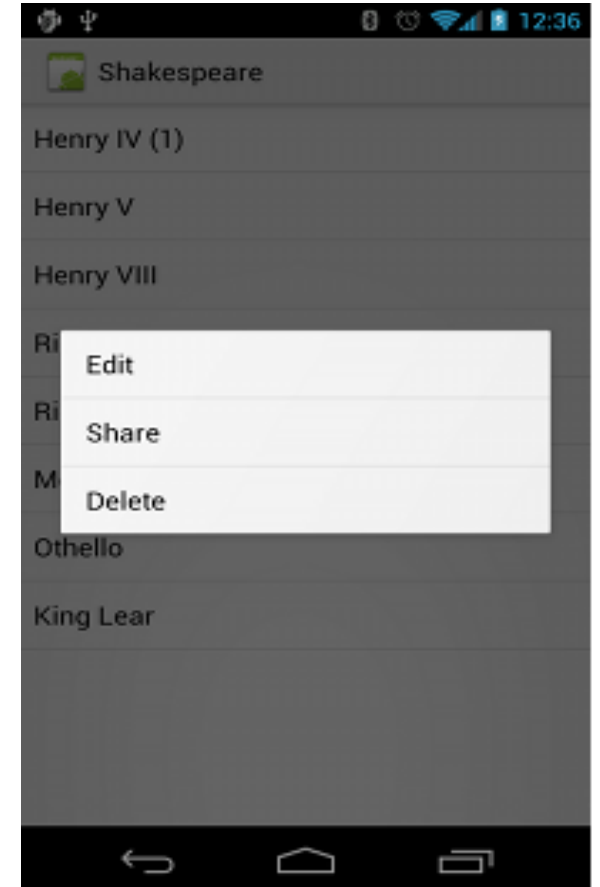
- **Two Ways Contextual Actions**
  - Floating Context Menu
  - Contextual Action Mode

# Contextual Menu

- **Creating Floating Context Menu**
  - Floating list of menu items is displayed when long-presses on a view element.
    - User can modify the view element or use it in some fashion.
    - Users perform a contextual action on one view element at a time.
  - To provide floating context menu:
    - Register View by calling registerForContextMenu()
    - Implement the onCreateContextMenu() method in your Activity or Fragment

# Creating Floating Context Menu

```java
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    et1=(EditText)findViewById(R.id.et1);
    et2=(EditText)findViewById(R.id.et2);
    registerForContextMenu(et1);
    registerForContextMenu(et2);}
@Override
public void onCreateContextMenu(ContextMenu menu, View v,
ContextMenu.ContextMenuInfo menuInfo){
    switch (v.getId()){
        case R.id.et1:
            getMenuInflater().inflate(R.menu.et1_menu,menu);
            break;
        case R.id.et2:
            getMenuInflater().inflate(R.menu.et2_menu,menu);
            break;
    }}
```
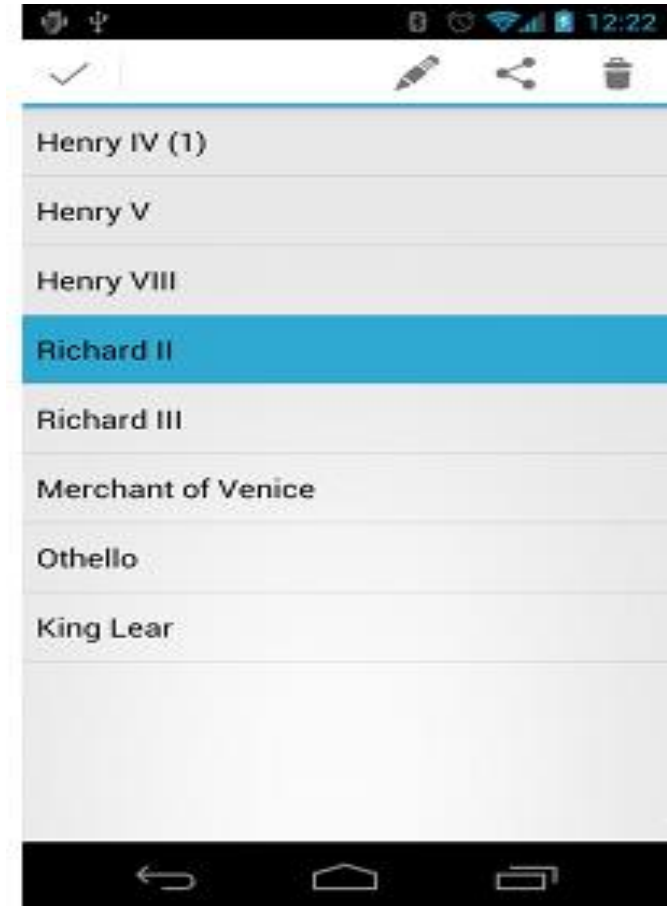
## Handling Click Event on Floating Context Menu

```java
public boolean onContextItemSelected(@NonNull MenuItem item) {
    switch(item.getItemId()){
        case R.id.i1:
        Toast.makeText(this, item.getTitle() + " Clicked/Selected",Toast.LENGTH_LONG).show();
            break;
        case R.id.i2:
            Toast.makeText(this, item.getTitle() + " Clicked/Selected",Toast.LENGTH_LONG).show();
            break;
        case R.id.s1:
            Toast.makeText(this, item.getTitle() + " Clicked/Selected",Toast.LENGTH_LONG).show();
            break;
        case R.id.s2:
            Toast.makeText(this, item.getTitle() + " Clicked/Selected",Toast.LENGTH_LONG).show();
            break;
    }
    return true;
}
```
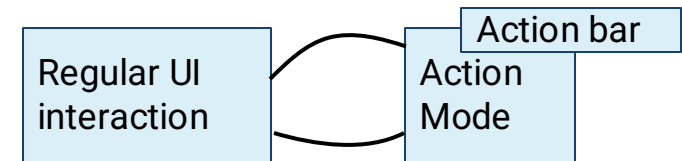
# Contextual Action Mode

- It is a temporary action bar in place of or underneath the app bar.
  - Action items affect the selected view element(s).
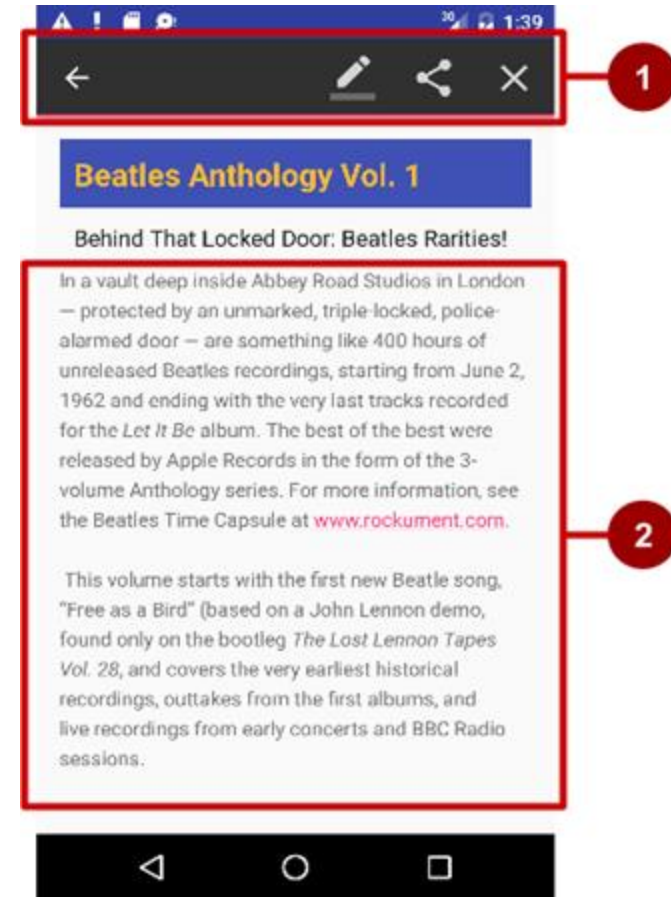  - Users can perform action on multiple view elements at once.

# Action Mode

- What is Action Mode?

  o Action Mode is a UI mode that lets you replace parts of the normal UI interactions temporarily.

  o For example, selecting a section of text or long-pressing an item could trigger action mode.
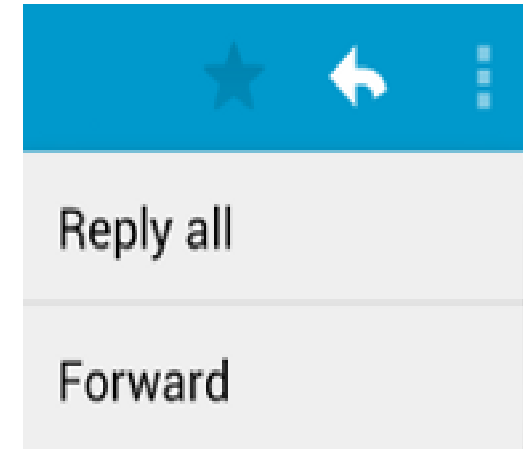
# Action Mode

- What is Contextual Action Bar?
  - Long-tap on the view shows contextual action bar.

1. Contextual action bar with actions
   - ✓ Edit, Share, and Delete
   - ✓ Done (left arrow icon) on the left side
2. View on which long press triggers the contextual action bar.
   - ✓ Action bar is available until user taps Done

## Popup menu

○ What is the popup menu?

- Vertical list of items anchored to a view.

- Typically anchored to a visible icon.

- Actions should not directly affect view content.

  ● The options menu overflow that opens Settings.

  ● For example, in an email app, Reply All and Forward are related to the email message, but don't affect or act on the message.

# Class Work

- To enhance your student semester registration application with menu functionalities, consider the following improvements:

   1. **Floating Action Button (FAB):** Add a FAB on the login page that provides quick access to basic information about the app or registration process.
   2. **Menu Options:** On the registration page, implement a menu that offers multiple options, such as saving progress, viewing help, or accessing settings.
   3. **Confirmation Dialog:** Before submitting the registration form, include a confirmation dialog that allows the user to review and confirm their information.
   4. **Additional Features:** Feel free to incorporate any other features that enhance the user experience and functionality of the application.

# Thank you!