

# CHAPTER-10:

# CUSTOMIZING ANDROID UI VIEW ELEMENTS

*BBC 3, Business Telecommunications*  
*Academic Year, 2023/ 2024, Semester – I*

*By Ddamba Abdul*  
*0703348992, addamba@mubs.ac.ug*  
*@MUBS*

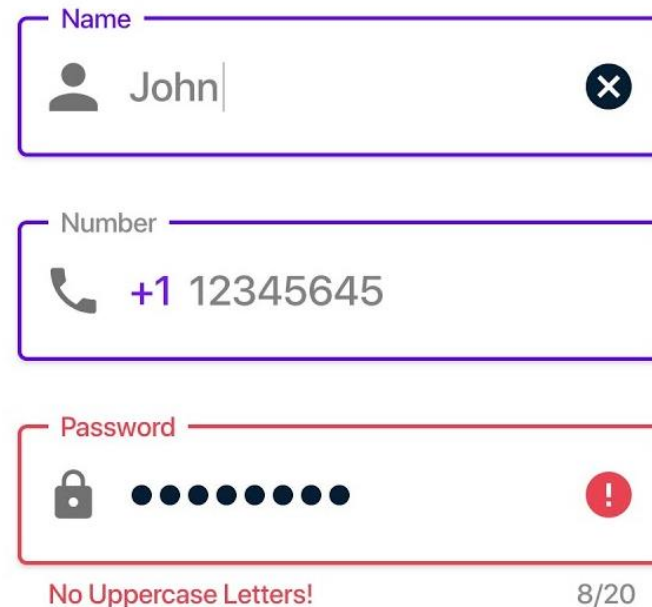
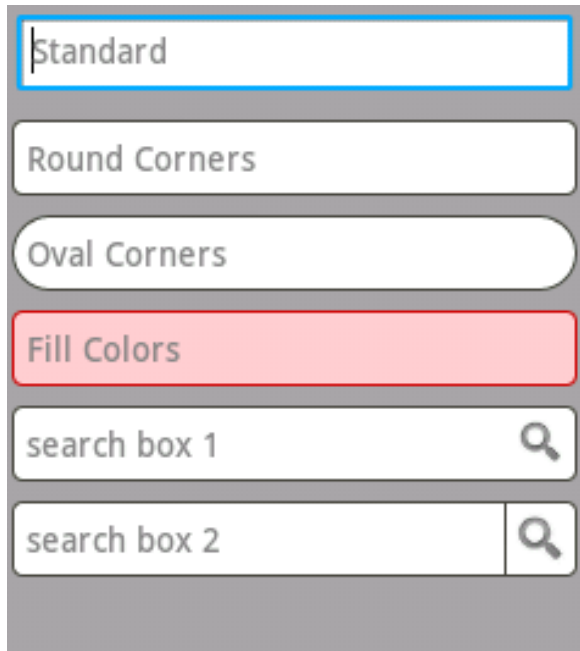
# Introduction

## ❑ Customizing Android UI view elements

- Involves altering the appearance & behavior of standard UI elements to match the design & functionality of an app
- Customizing Android UI view elements gives you the flexibility to create visually appealing and unique user interfaces that match your app's design and functionality requirements
- You can use XML attributes, styles, theming, and programmatic code to achieve the desired customization

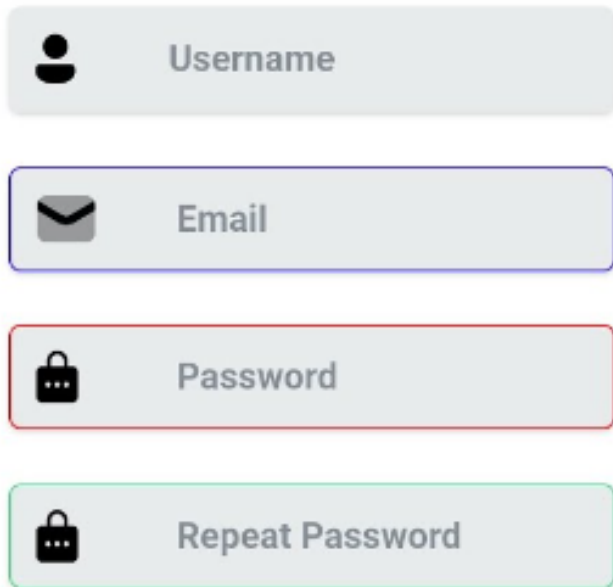
# EditText View

- ❑ A UI element in Android that provides a text input field where users can enter or edit text
  - It is commonly used in Android app forms, search bars, or any interaction that requires text input.



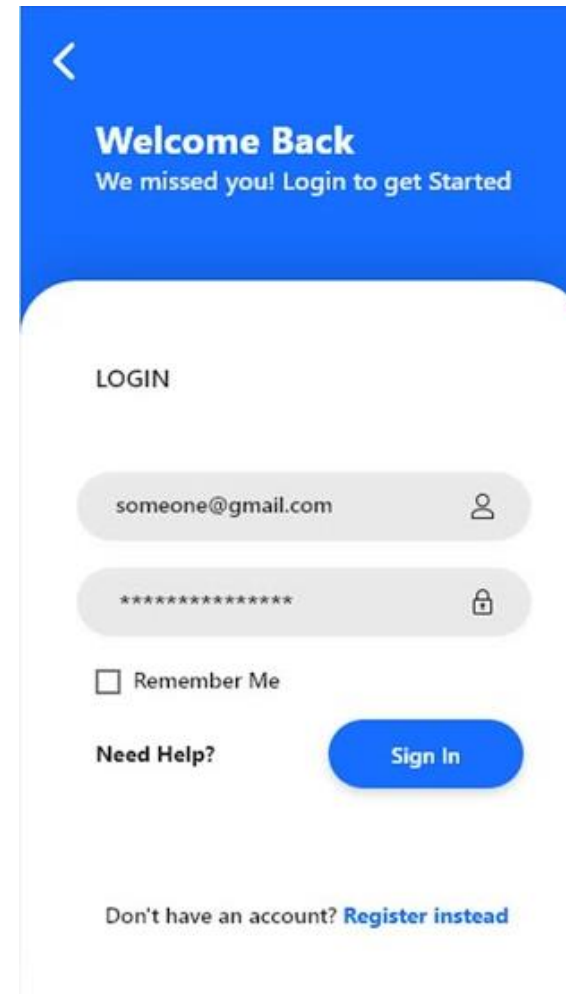
# EditText View

## ❑ More Images Showing a Customized EditText View



Four customized EditText views are shown, each with a unique icon and border color:

- Username:** Features a person icon and a light gray border.
- Email:** Features an envelope icon and a purple border.
- Password:** Features a padlock icon and a red border.
- Repeat Password:** Features a padlock icon and a green border.



A mobile app login screen with a blue header and white body. The header contains a back arrow, the text "Welcome Back", and the message "We missed you! Login to get Started". The body contains a "LOGIN" section with two input fields: "someone@gmail.com" (with a person icon) and "\*\*\*\*\*" (with a padlock icon). Below the input fields is a "Remember Me" checkbox. At the bottom of the login section is a "Need Help?" link and a blue "Sign In" button. At the very bottom of the screen is a link: "Don't have an account? [Register instead](#)".

# Customizing an EditText View

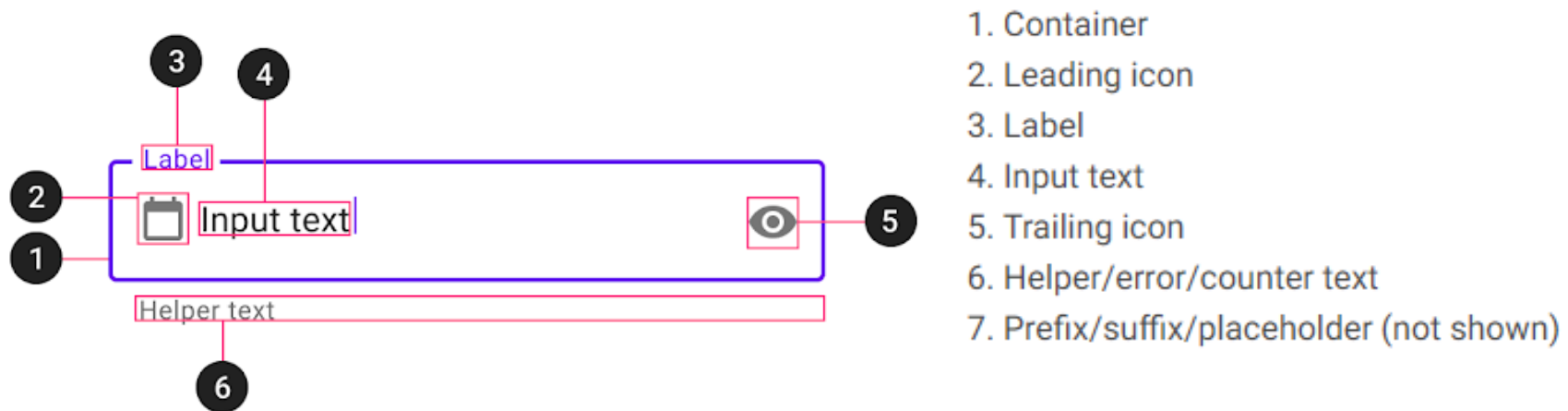
## ❑ What is a Material Design EditText Input View?

- It is a UI element in Android that follows the Material Design guidelines provided by Google
  - It is a specialized form of an EditText used for text input but with a distinctive visual and interactive style consistent with the Material Design principles
  - It is implemented by a **TextInputLayout**
    - It is a view container that is used to add more features to an EditText. It acts as a wrapper for EditText and has some features like: *Floating hint*, *animation* that can be disabled or enabled, *error labels* that display error messages when an error occurs, *character counter* that displays the number of characters that the user enters, and a *password visibility toggle*

# Adding a Material EditText Field

## □ Anatomy and key properties of a material EditText

- An outlined text field has a stroked container, input text, a label, optional helper/error text and optional leading/trailing icons.



# Adding a Material EditText Field

## □ Steps to creating a Material EditText Field

1. Add a dependency to the Material Components Library in the project's **build.gradle** file (added automatically)

```
implementation 'com.google.android.material:material:1.10.0'
```

- Then, sync your project with the updated Gradle files
2. Set the **MaterialComponent** theme in your layout file

```
android:theme="@style/Theme.MaterialComponents.Bridge">
```
  3. Add '**TextInputLayout**' to the '**layout**' file
    - It is a child element inside the ConstraintLayout
  4. Add a **TextInputEditText** inside a **TextInputLayout**
    - This creates a Material Design input field used to enter data

# Adding a Material EditText Field

## ❑ Steps to creating a Material EditText Field

5. Customize the appearance of the material EditText by adding the following attributes in the **TextInputLayout**

- Add the **ID**, **height** and **width** attributes in both views
- Add a hint attribute: `android:hint="Enter Your Name:"`
- Add an outline to the view: `app:boxBackgroundMode="outline"`
- Add an outline stroke color: `app:boxStrokeColor="@color/blue"`
- Set a box background color:

`app:boxBackgroundColor="@color/lightbrown"`

- Set a **DenseFilledBox** style to the **TextInputLayout** view  
`style="@style/Widget.MaterialComponents.TextInputLayout.FilledBox.Dense"`



# Adding a Material EditText Field

## ❑ Steps to creating a Material EditText Field

5. Customize the appearance of the material EditText by adding the following attributes in the **TextInputLayout**

- Add **boxCornerRadius** attributes

- It is needed to change the corner radius of the box outline

- ```
app:boxCornerRadiusBottomEnd="16dp"
```

- ```
app:boxCornerRadiusBottomStart="16dp"
```

- ```
app:boxCornerRadiusTopEnd="16dp"
```

- ```
app:boxCornerRadiusTopStart="16dp"
```

- Add:

- ```
app:helperText="Required"
```

- ```
app:helperTextTextColor="@color/red"
```

- Though optional, it provides additional information about a field's input, such as how it will be utilized

# Adding a Material EditText Field

## ❑ Steps to creating a Material EditText Field

### 5. Customize the appearance of the material EditText

- Add

```
app:counterEnabled="true"
```

```
app:counterMaxLength="5"
```

- Optional, it provides a character count validation for text input

- Add

```
app:endIconMode="password_toggle"
```

```
app:endIconTint="@color/black"
```

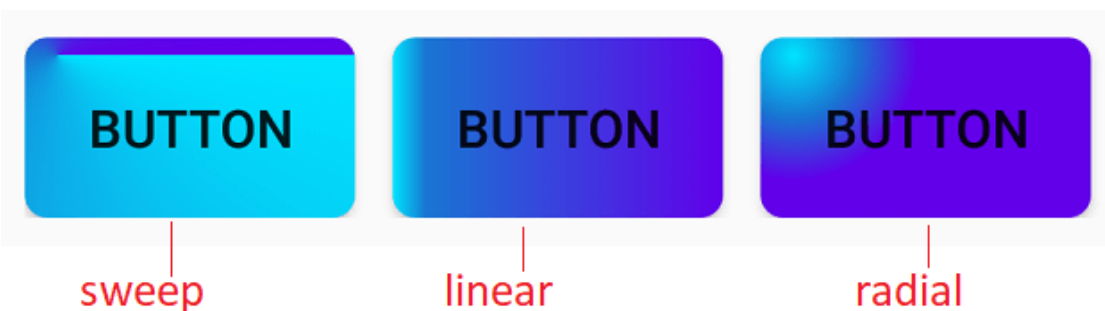
- Used to specify the mode of the end icon in the input field, it can also be set to **clear\_text** or **custom**

- Add start icon

```
app:startIconDrawable="@drawable/user"
```

# Using Shape Drawables

- ❑ In addition to graphical files, Android supports shape drawables
  - A shape drawable is an XML file that defines a geometric shape such as **rectangle**, **oval**, **line**, and **ring** with size, style e.g. color, borders and gradients which can be assigned to views
    - For example, you can use a shape drawable to change the shape, border, and gradient of a Button background



# Types of Shape Drawables

❑ Shape drawables provide four basic shapes:

- Rectangle,
  - Oval,
  - Line and
  - Ring
- 
- From these shapes you can create an almost unlimited number of effects and styles for your app

# Creating a Rectangle Shape Drawable

## ❑ Steps to creating a Rectangle Drawable Shape

1. In your Android Studio project, navigate to the “**res**” folder.
2. Right-click on the “**drawable**” directory (or create one if it doesn’t exist) and choose “New” -> “**Drawable resource file.**”
3. Give your drawable resource file a name, for example, “**rectangle\_shape.xml.**”
4. Specify the root element as **shape**
5. Open the “**rectangle\_shape.xml**” file in the XML editor

# Creating a Rectangle Shape Drawable

## ❑ Steps to creating a Rectangle Drawable Shape

6. Within the root **shape element**, define shape attribute and set it to the shape you are creating;

```
android:shape="rectangle"
```

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<shape
```

```
    xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    android:shape="rectangle">
```

```
</shape>
```

# Creating a Rectangle Shape Drawable

## ❑ Steps to creating a Rectangle Drawable Shape

6. Add the following child tags that further customizes the drawable shape

- The **stroke**, of the shape with stroke width and color

```
<stroke  
    android:width="2dp"  
    android:color="@color/black">  
</stroke>
```

- The **padding**, that defines the view padding all round

```
<padding  
    android:bottom="16dp" android:top="16dp"  
    android:right="16dp" android:left="16dp"  
>  
</padding>
```

# Creating a Rectangle Shape Drawable

## ❑ Steps to creating a Rectangle Drawable Shape

6. Add the following child tags that further customizes the drawable shape

- The **size**, of the shape by width and height

```
<size  
    android:width="200dp"  
    android:height="150dp">  
</size>
```

- The **corner**, that defines the corners all round

```
<corners  
    android:radius="45dp"  
>  
</corners>
```



# Creating a Rectangle Shape Drawable

## ❑ Steps to creating a Rectangle Drawable Shape

6. Add the following child tags that further customizes the drawable shape
  - The **solid**, of the shape defines the color filled on to the shape

```
<solid
    android:color="@color/red"
>
</solid>
```

# Creating a Rectangle Shape Drawable

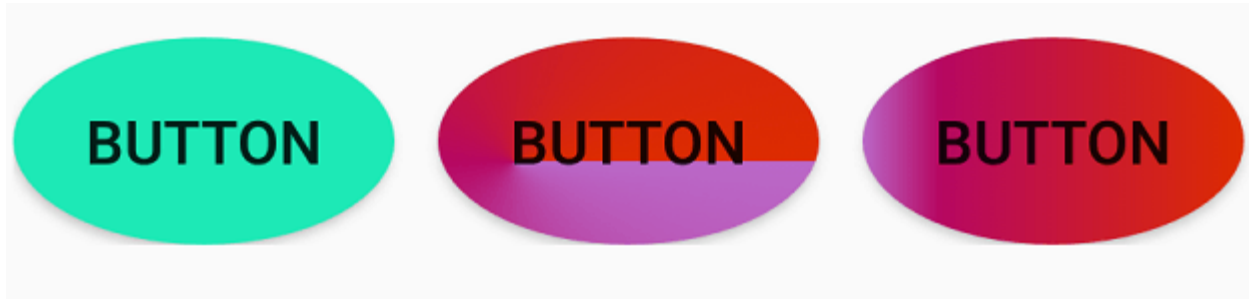
## □ Steps to creating a Rectangle Drawable Shape

6. Add the following child tags that further customizes the drawable shape
  - The **gradient**, that defines a gradient color effect, giving a smooth transition between two or more colors

```
<gradient
    android:startColor="@color/red"
    android:endColor="@color/black"
    android:centerColor="@color/white"
    android:angle="50"
    android:type="linear"
>
</gradient>
```

# Creating a Circular Shape Drawable

- ❑ You can create oval shape drawable as shown below and apply it to button to create oval shaped button



- You can specify similar child tags used in the case of a rectangle shape, such as; **corners**, **gradient**, **padding**, **size**, **solid** and **stroke** of the shape to define a drawable

# Creating a Ring Shape Drawable

- ❑ To create ring shape drawable, use the xml codes below;

```
<?xml version="1.0" encoding="utf-8"?>
<shape
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:innerRadius="45dp"
    android:shape="ring"
    android:thickness="50dp"
    android:useLevel="false">

    <solid
        android:color="@color/red"
        >
    </solid>
</shape>
```

# Creating a Line Shape Drawable

❑ To define line shape drawable, you need to;

1. Set shape attribute of shape element to line
2. Add stroke element with color and size element

```
<?xml version="1.0" encoding="utf-8"?>
<shape
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="line">

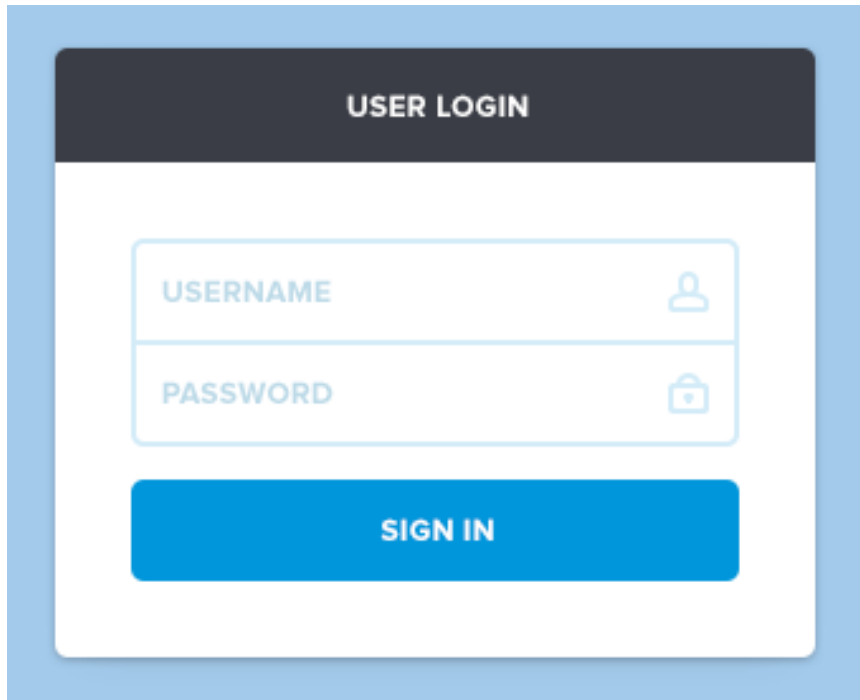
    <stroke
        android:width="10dp"
        android:color="#b71c1c">

    </stroke>

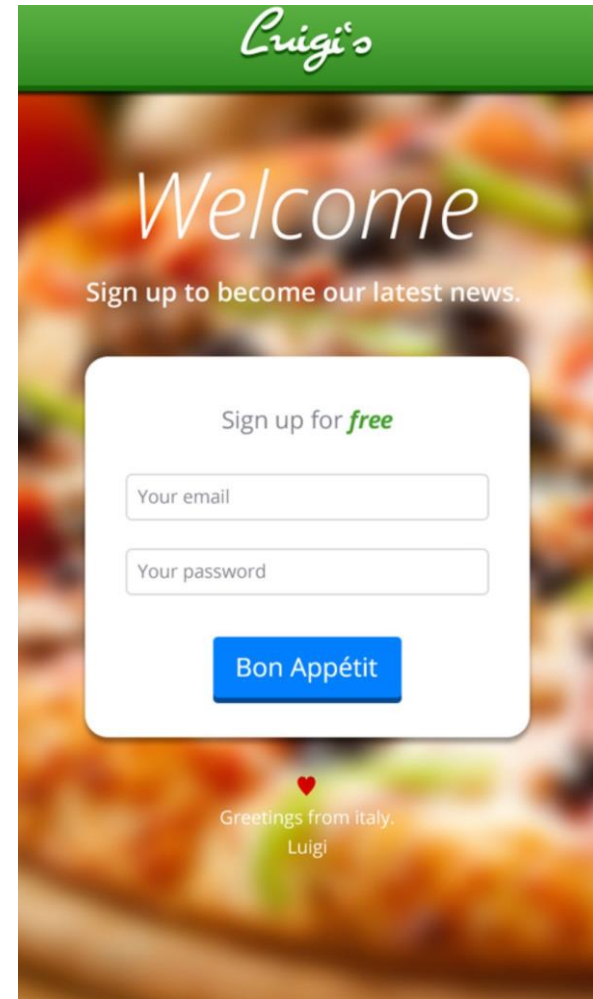
    <size android:height="8dp" />
</shape>
```

# ASSIGNMENT

❑ Create the following User Interfaces



A user login interface mockup. It features a dark gray header with the text "USER LOGIN". Below the header is a white card with a light blue border. Inside the card, there are two input fields: "USERNAME" with a user icon and "PASSWORD" with a lock icon. Below the input fields is a large blue button labeled "SIGN IN".





A welcome screen UI mockup for "Luigi's". It has a green header with the "Luigi's" logo. Below the header is a blurred background image of a pizza. The text "Welcome" is displayed in a large, white, serif font. Below "Welcome" is the text "Sign up to become our latest news." in a smaller, white, sans-serif font. A white card with rounded corners is centered on the screen. Inside the card, there is a text "Sign up for *free*" in green. Below this are two input fields: "Your email" and "Your password". At the bottom of the card is a blue button labeled "Bon Appétit". Below the card, there is a small red heart icon and the text "Greetings from Italy. Luigi" in a small, white, sans-serif font.


# ASSIGNMENT

## ❑ Create the following User Interfaces

**LOGIN FORM**










**LOGIN**


☒ Remember me | [Forgot password](#)

Login to your account

 Sign in with facebook

 Sign in with twitter

 Sign in with linkedin

 Sign in with google+

OR

USERNAME

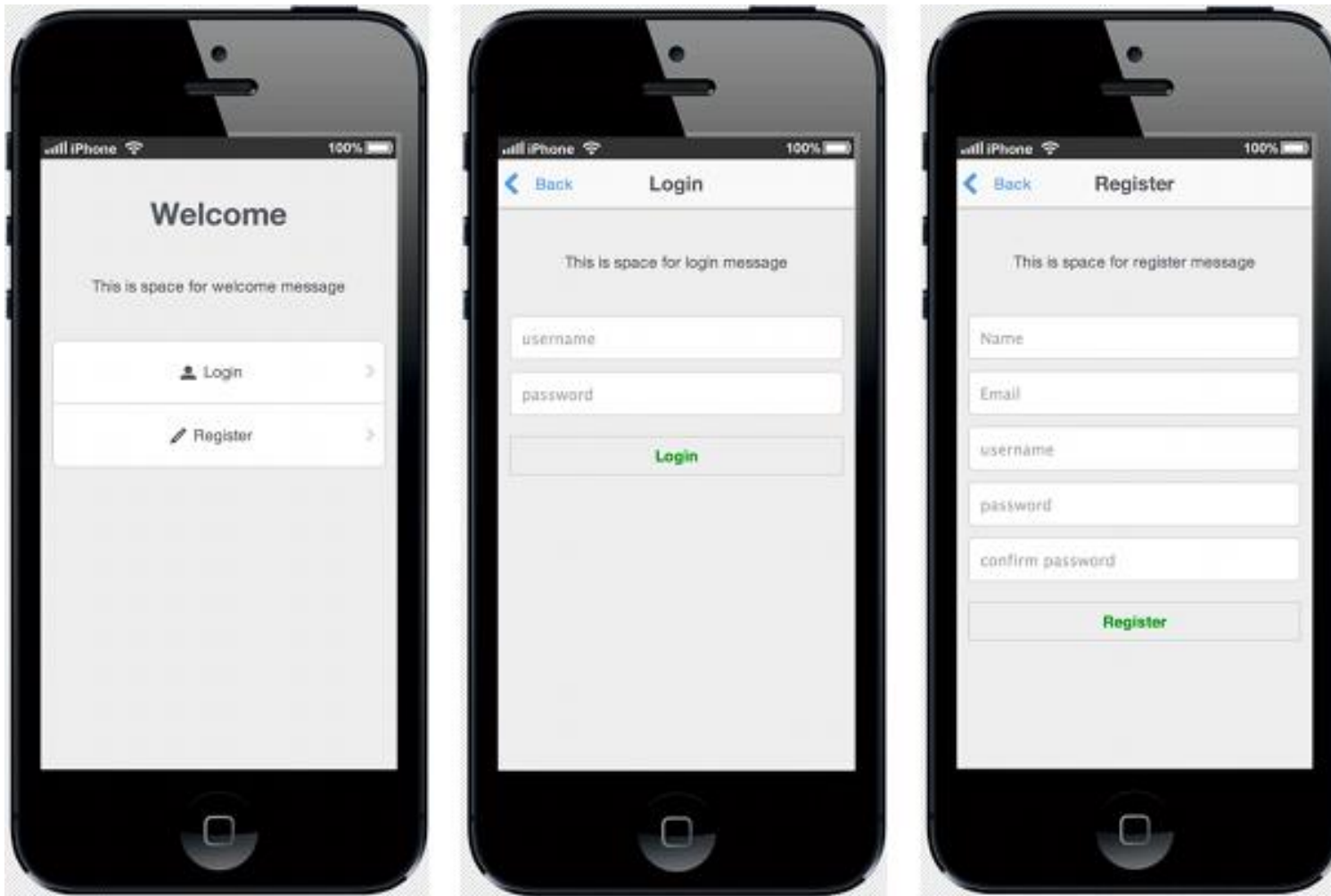
PASSWORD

Not a Member yet?

Sign in

# ASSIGNMENT

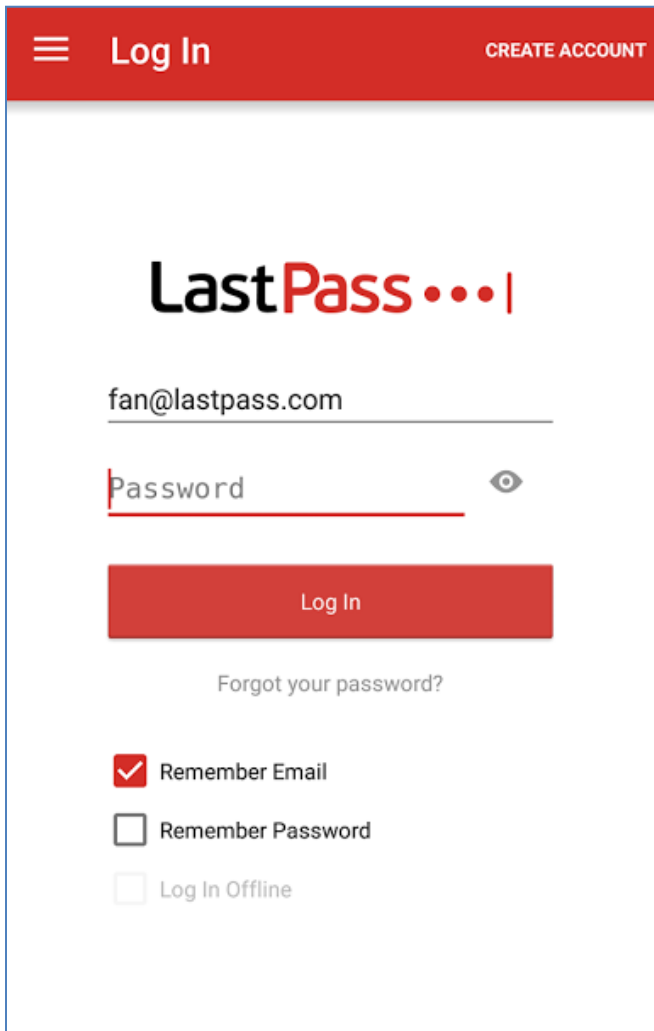
❑ Create the following User Interfaces



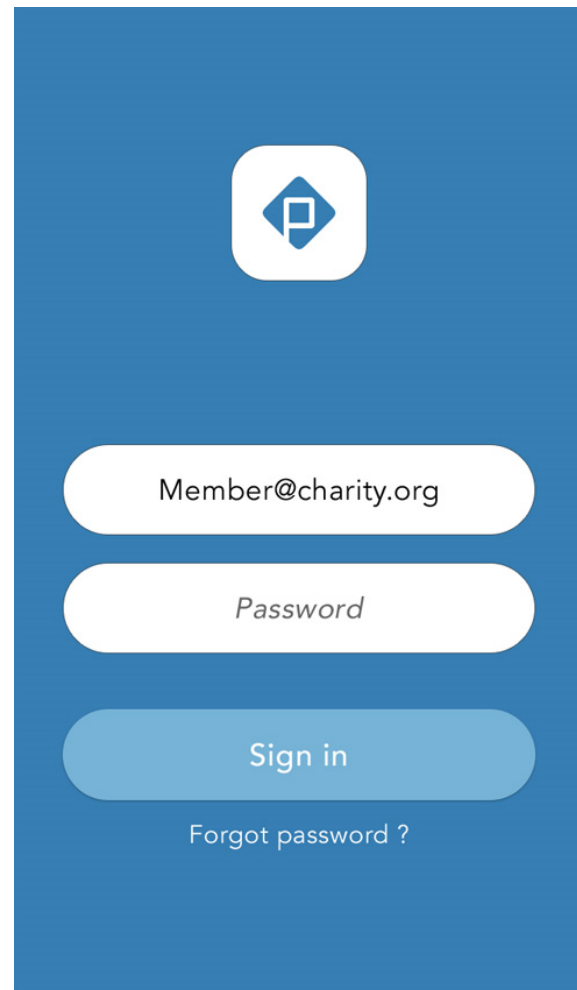


# ASSIGNMENT

## ❑ Create the following User Interfaces



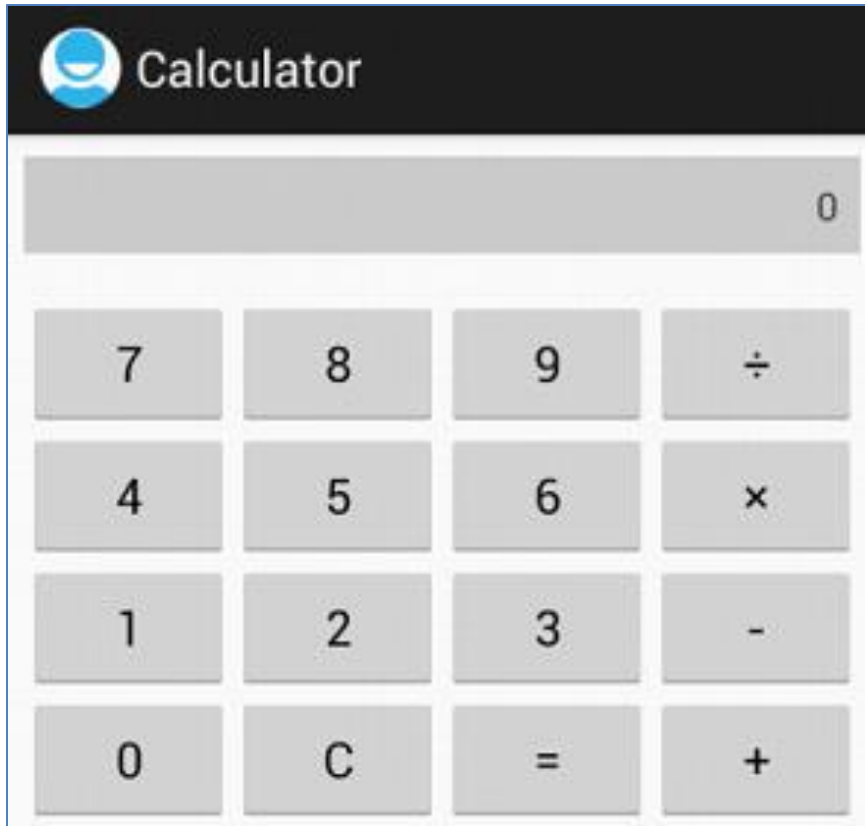
The image shows a web-based login interface for LastPass. At the top, there is a red header bar with a hamburger menu icon on the left, the text "Log In" in the center, and "CREATE ACCOUNT" on the right. Below the header, the "LastPass" logo is displayed. Underneath the logo is a text input field containing the email "fan@lastpass.com". Below the email field is a password input field with the placeholder text "Password" and a toggle icon (an eye) to its right. A red "Log In" button is positioned below the password field. Under the button is a link that says "Forgot your password?". At the bottom, there are three checkboxes: "Remember Email" (checked), "Remember Password" (unchecked), and "Log In Offline" (unchecked).



The image shows a mobile login interface with a solid blue background. At the top center is a white rounded square containing a blue icon of a document with a lock. Below this icon are three white rounded rectangular input fields. The first field contains the email "Member@charity.org", the second field contains the placeholder text "Password", and the third field is empty. Below the input fields is a blue rounded rectangular button with the text "Sign in". At the bottom, there is a link that says "Forgot password ?".

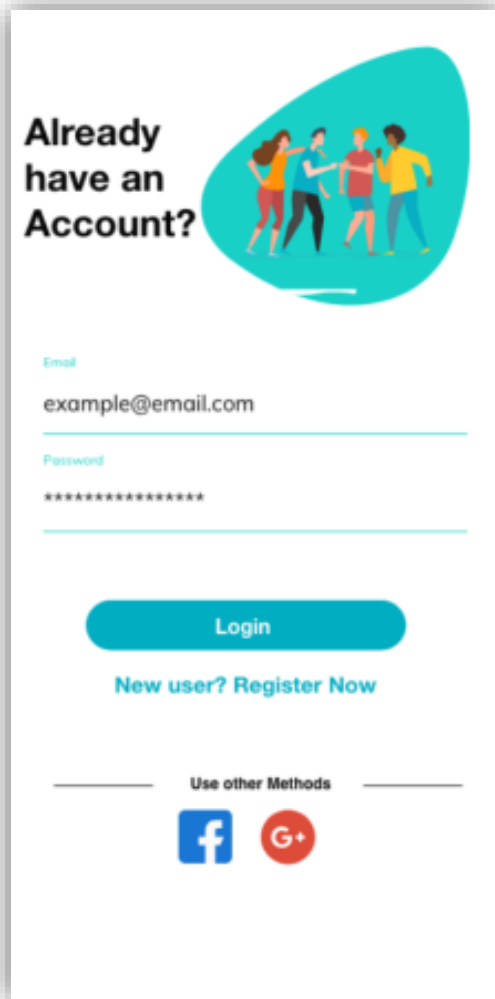
# ASSIGNMENT

❑ Create the following User Interfaces




# ASSIGNMENT

## ❑ Create the following User Interfaces



Already have an Account?



Email

example@email.com



Password

\*\*\*\*\*

Login

New user? Register Now

Use other Methods



Here's your first step with us!



Name

kapil

Email

example@email.com

Mobile

8129\*\*\*\*

Password

\*\*\*\*\*

Register

Use other Methods

# ASSIGNMENT

❑ Create the following User Interfaces

The image displays two mobile application user interface screens side-by-side. Both screens have a blue header with a white curved bottom edge.

**Left Screen (Log in):**

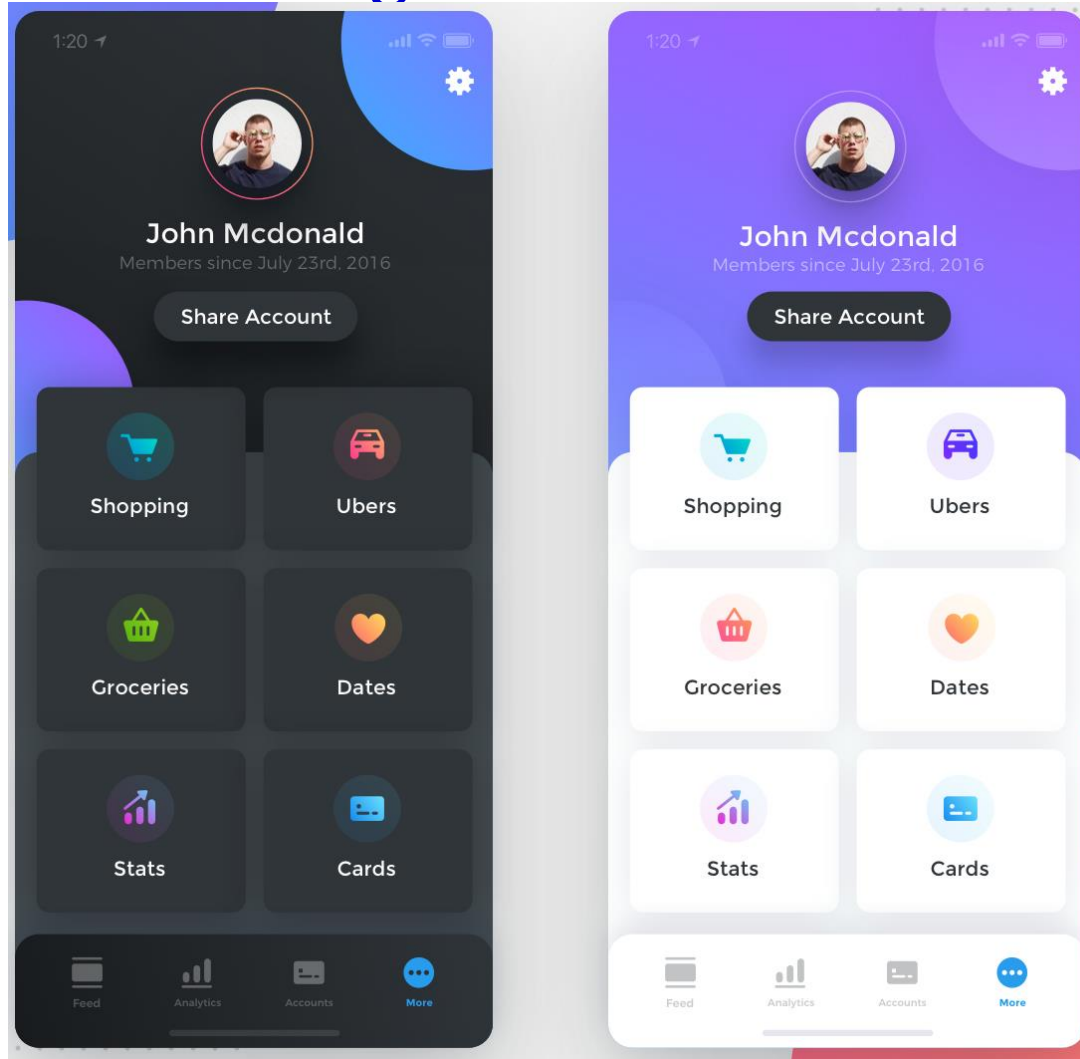
- Title: "Log in"
- Fields: "Email" (with an envelope icon) and "Password" (with a lock icon).
- Options: A checkbox for "Remember Me?" and a red link for "Forget password?".
- Button: A blue button labeled "Log in with your account".
- Section: "Log In with" followed by two buttons: a blue "Facebook" button with the 'f' logo and a red "G+ Google" button with the 'G+' logo.
- Footer: A blue button labeled "New user ? Create account".

**Right Screen (Create New account):**

- Back arrow: A white left-pointing arrow in the top left corner.
- Title: "Create New account"
- Fields: "Name" (with a person icon), "Email" (with an envelope icon), and "Password" (with a lock icon).
- Options: A checkbox for "I agree to the" followed by a red link for "Term & Conditions".
- Button: A blue button labeled "Create account".
- Footer: A blue button labeled "Already a User ? Log in".

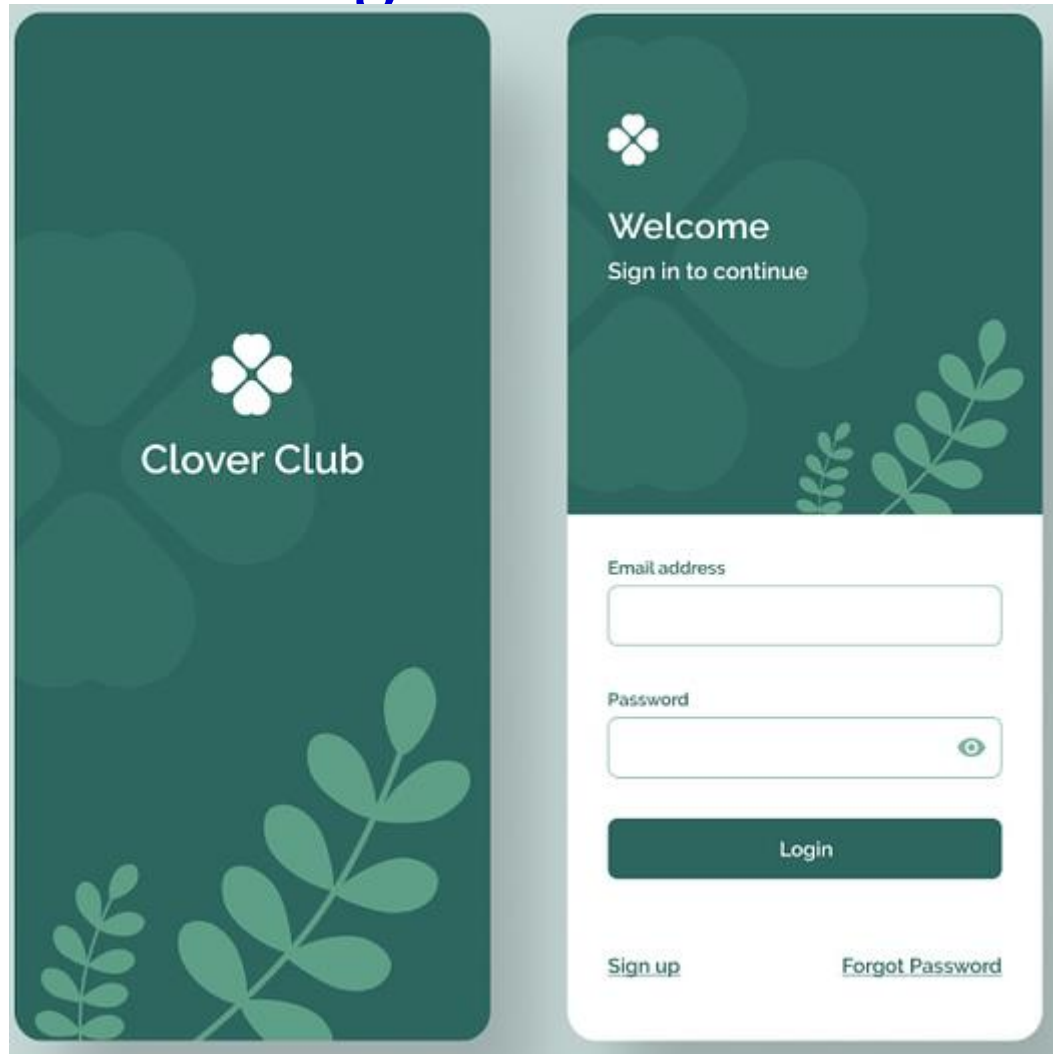
# ASSIGNMENT

## ❑ Create the following User Interfaces



# ASSIGNMENT

❑ Create the following User Interfaces



**-THE END-**