Srikanth Gude – sgcnm@umsystem.edu

Git Hub Link - https://github.com/gudesrikanth/webcourse/tree/main/Webpart/ICP9

Gopi Nelluri – gng75@umsystem.edu

Git Hub Link - https://github.com/gopinelluri9/demo_remote_repository/tree/main/WebPart/ICP9

ICP – 9(Pizza ordering app)

Task Description:

In this Task, we will use Android Studio to create a Food Ordering App for Pizza.

- The Order Page displays a text area where we can input your name, type of pizza, and topping section where you can select from the checkboxes, and a quantity part where you can enter the number of pizzas that you want to order (i.e., the number of pizzas that you want to order).
- You have a choice between two Buttons. The first is named ORDER, while the second is named SUMMARY.
- When the user clicks on the ORDER button, it allows the user to send an email to the specified recipient, along with a description of the order and the order details.
- When the user clicks on the SUMMARY button, a new page is displayed, which includes the Summary of the Pizza order, an image of the Pizza, and the GOTO ORDER button.
- By clicking GOTO ORDER button, the user is taken to the order's main page.

Process:

ORDER PAGE => activity_main.xml

This page contains a text field, three check boxes in Pizza Type, two check boxes in Toppings, and two buttons to increment and decrease the quantity of pizza **activity_main.xml.**

First Text Field: Name

Here we have added an Edit Text Field with input type as userInput and we have also. The id is of the text field is @+id/user_input. This will allow the user to enter the user name.

```
<EditText
    android:id="@+id/user_input"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="16sp"
    android:inputType="textCapWords" />
```

Type of Pizza Check Boxes: Veggie, Chicken and Other

We have added three checkboxes one is the veggie selection, Second is Chicken selection and Other selection.

```
<CheckBox
    android:id="@+id/veggie_checked"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="16sp"
    android:paddingLeft="24dp"
    android:text="Veggie"
    android:textSize="16sp" />
<CheckBox
    android:id="@+id/chicken_checked"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="16sp"
    android:paddingLeft="24dp"
    android:text="Chicken"
    android:textSize="16sp" />
<CheckBox
    android:id="@+id/other_checked"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="16sp"
    android:paddingLeft="24dp"
    android:text="Other"
    android:textSize="16sp" />
```

@+id/veggie_checked is the id of the Veggie Checkbox, and @+id/chicken_checked is the id of the Chicken Checkbox. These id's can be retrieved in the java file using these id's.

Check Boxes: Mushroom and Extra Cheese (Topping)

We have added two checkboxes one is the Mushroom selection and the other is for Extra Cheeses selection.

```
<CheckBox
    android:id="@+id/mushroom_checked"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="16sp"
    android:paddingLeft="24dp"
    android:text="Mushroom"
    android:textSize="16sp" />

<CheckBox
    android:id="@+id/extra_cheese_checked"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="16sp"
    android:paddingLeft="24dp"
    android:text="Extra Cheese"
    android:textSize="16sp" />
```

@+id/mushroom_checked is the id of the Mushroom Checkbox, and @+id/extra_cheese_checked is the ID of the Extra Cheese Checkbox. These id's can be retrieved in the java file using these id's. There are two buttons after these two checkboxes.\

Two Buttons: Increment and Decrement Quantity:

Then you need include two buttons in the file. One is for increasing the quantity of pizza, while the other is for decreasing the quantity of pizza. The decrement button's id is @+id/buttonminus, and the increment button's id is @+id/buttonplus. The decrement button's id is @+id/buttonminus.

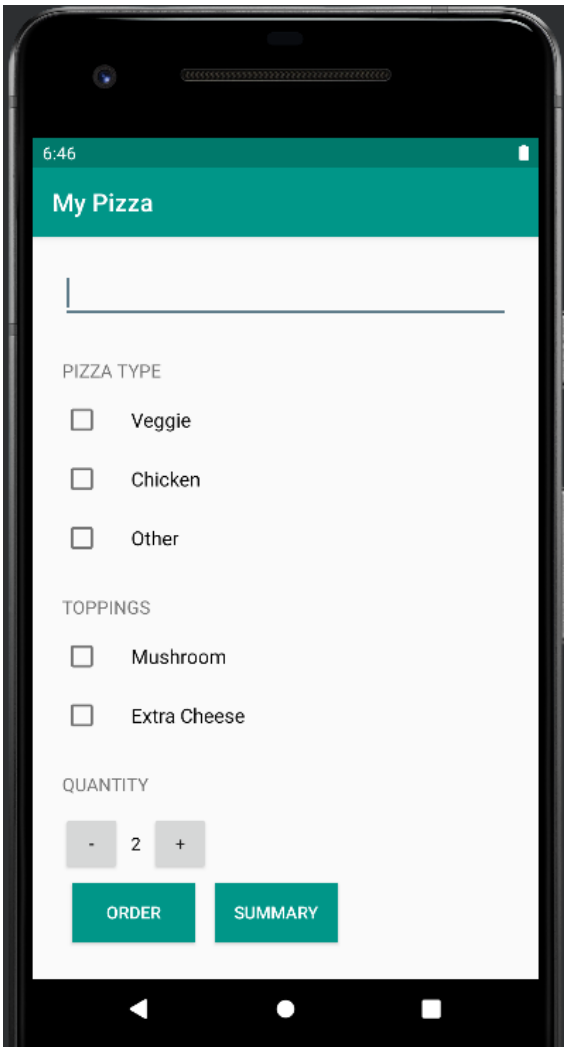
```
<Button
    android:id="@+id/buttonminus"
    android:layout_width="48dp"
    android:layout_height="wrap_content"
    android:layout_marginRight="8dp"
    android:onClick="decrement"
    android:text="-" />
<TextView
    android:id="@+id/quantity_text_view"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginRight="8dp"
    android:text="2"
    android:textColor="#000000" />
<Button
    android:id="@+id/buttonplus"
    android:layout_width="48dp"
    android:layout_height="48dp"
    android:onClick="increment"
    android:text="+" />
```

Two Buttons: ORDER and SUMMARY

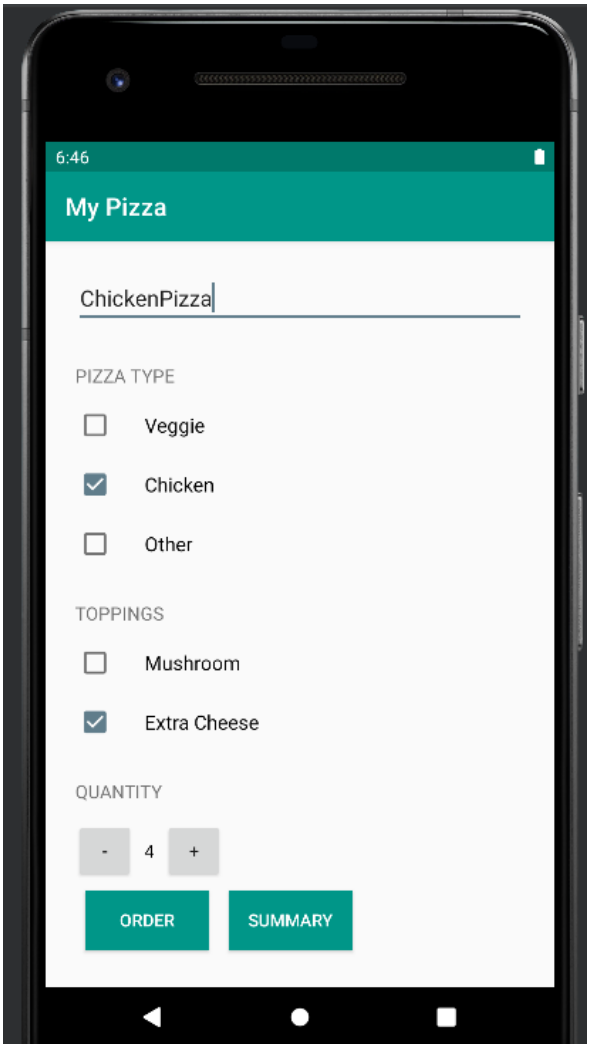
When the order button is clicked, it enables the user to send the email to the intended recipient more quickly. When the order button is selected, the user is prompted to share the information, and when the user selects Gmail as the method of sharing the information, the order details and topic of the email are automatically filled in. The id of the order is @+id/buttonorder, and it is a string.

The Summary button directs the user to a new page that contains the order summary, an image of a pizza, and a GOTO ORDER button. Summary buttons have the id @+id/buttonsummary, which stands for "summary."

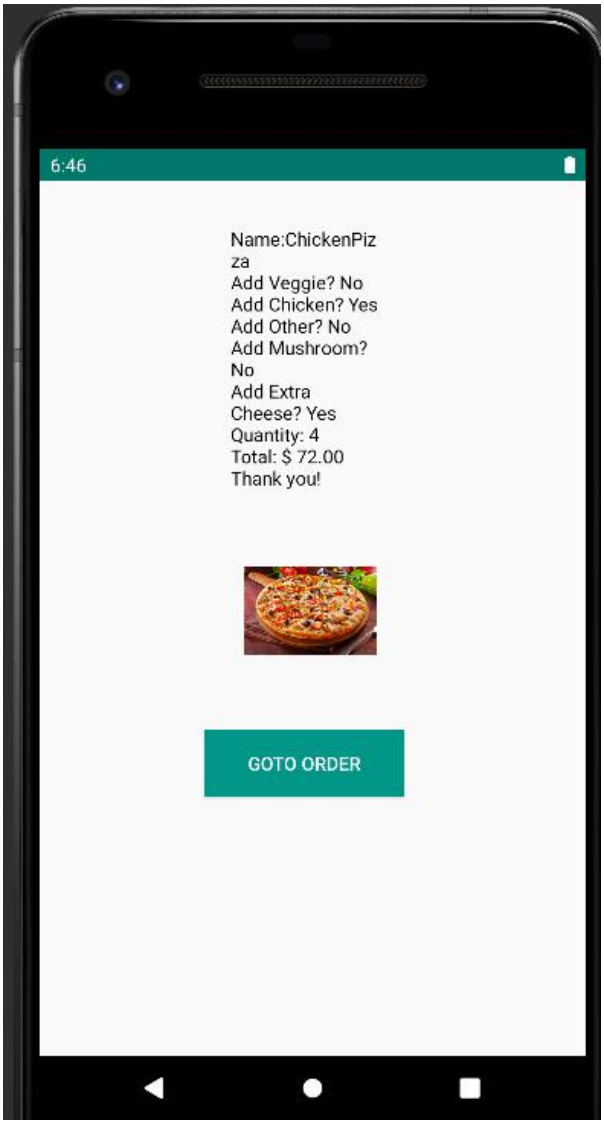
The UI functionality of the app looks like the following images

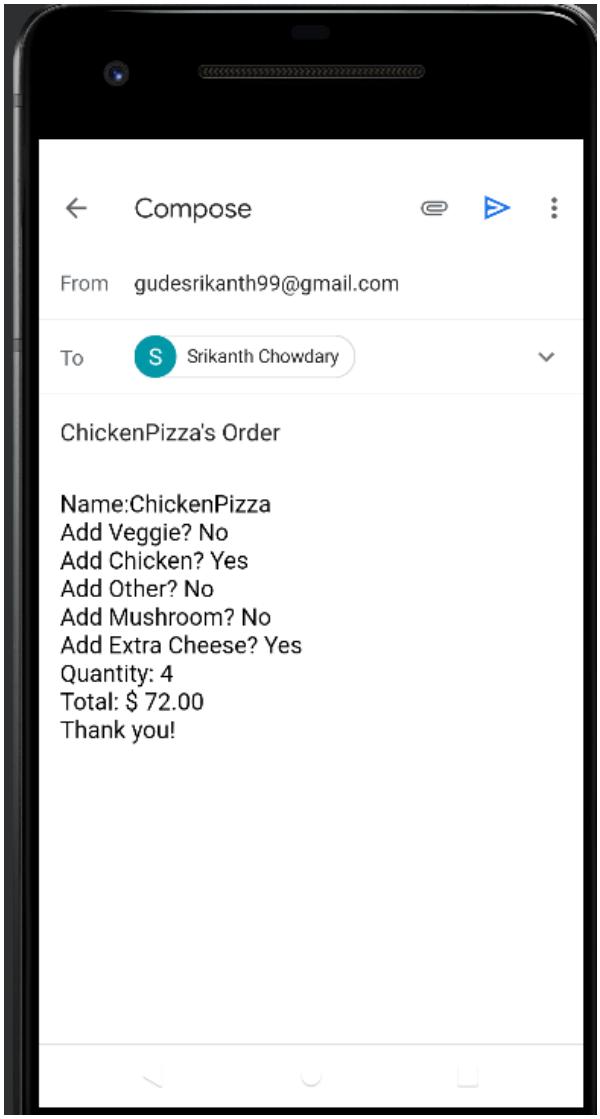


Home Page



User Input





Summary Page **Clicking on Order**

MainActivity.java: The Functionality of the app is described in this file.

SumbitOrder: The functionality of submitOrder is to allow users to send email with the order description along with the subject of the mail.

Parameters: View view. This method takes view as the input. The code looks as follows

This method internally invokes the sendEmail method.

```
public void submitOrder(View view) {
    String orderSummaryMessage = getOrderSummary(view);
    Intent i = new Intent(Intent.ACTION_SEND);
    EditText userInputNameView = (EditText) findViewById(R.id.user_input);
    String userInputName = userInputNameView.getText().toString();
    sendEmail(userInputName,orderSummaryMessage);
```

sendEmail: The functionality of the sendEmail is to set the Subject of the mail, body of the mail, and recipient of the mail and send the mail to the recipient.

Parameters: This method has two parameters name and output string. These two parameters are used to set the description of the Pizza order.

```
public void sendEmail(String name, String output) {
    Intent intent = new Intent(Intent.ACTION_VIEW);
    if (intent.resolveActivity(getPackageManager()) !=null){
        startActivity(intent);
    }
    String subject = name + "'s Order";
    Intent sendIntent = new Intent();
    sendIntent.putExtra(Intent.EXTRA_EMAIL  , new String[]{"gudesrikanth99@gmail.com"});
    sendIntent.putExtra(Intent.EXTRA_SUBJECT, subject);
    sendIntent.setAction(Intent.ACTION_SEND);
    sendIntent.putExtra(Intent.EXTRA_TEXT, output);
    sendIntent.setType("text/plain");
    // Try to invoke the intent.
    try {
        startActivity(sendIntent);
    } catch (ActivityNotFoundException e) {
        // Define what your app should do if no activity can handle the intent.
    }
}
```

submitSummary(): The functionality of the submitSummary is to navigate to the new page to summary Screen where it will display the description of the pizza order.

Parameters: View view. The parameter to this method is view.

```java
public void submitSummary(View view) {
    String orderSummaryMessage = getOrderSummary(view);
    // get user input
    Intent summary = new Intent( packageContext: this,Summary.class);
    summary.putExtra( name: "message",orderSummaryMessage);
    startActivity(summary);
}
```

Summary Layout:

In this we have created a linear layout with a textview, an image, and a button on the screen.

Textview:

This textview is responsible for showing the order summary on the website; its id is textView, which will allow us to refer to it in the java class and add the string we need to display to it.

```xml
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_marginStart="120dp"
    android:layout_marginLeft="120dp"
    android:layout_marginTop="10dp"
    android:layout_marginEnd="128dp"
    android:layout_marginRight="128dp"
    android:layout_marginBottom="0dp"
    android:text="TextView"
    android:textColor="#000000"
    tools:ignore="MissingConstraints" />
```

```xml
<Button
    android:id="@+id/buttonret"
    android:layout_width="150dp"
    android:layout_height="50dp"
    android:layout_marginStart="100dp"
    android:layout_marginLeft="100dp"
    android:layout_marginTop="0dp"
    android:layout_marginEnd="128dp"
    android:layout_marginRight="128dp"
    android:layout_marginBottom="248dp"
    android:background="#009688"
    android:onClick="homeNav"
    android:text="Goto Order"
    android:textColor="#FAF9F9" />
```

Image:

We have taken the pizza image and placed it on the screen in a 100x100 resolution. Immediately below the textview and above the Goto Order Button will be displayed this graphic.

Goto Order Button:

When the Goto Order button is clicked, the screen is redirected to the order screen, and navHome is the value supplied to the onclick function, therefore functionality for the same function is written in the java file.

Summary.java

The onCreate function of the Summary class is overridden to get the string message, i.e. the order description. Then we construct a Textview object and add the message to it using the bundle method. So the order description appears on screen.

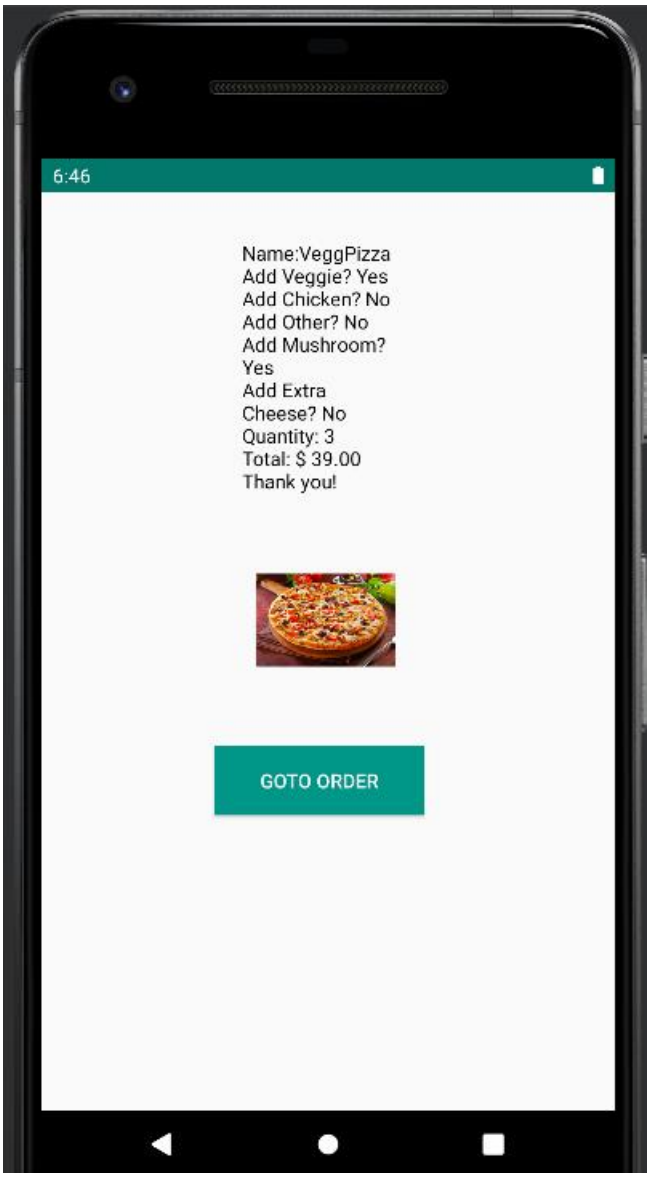
homeNav:

When the Goto Order button is clicked, an intent is created using this and MainActivity parameters.

```
public class Summary extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.summary);
        Bundle bundle = getIntent().getExtras();
        String message = bundle.getString( key: "message");
        TextView txtView = (TextView) findViewById(R.id.textView);
        txtView.setText(message);
        System.out.print("Text View" + txtView);
    }
    public void homeNav(View v){
        Intent i = new Intent( packageContext: this,MainActivity.class);
        startActivity(i);
    }
}
```

The UI for the Summary screen as below

The order summary screen displays all of the order details, including the type of toppings, the number of pizzas requested, the total amount, and the name of the user. When the Goto Order button is clicked, you'll be sent to the order page.


6:46
Name:VeggPizza
Add Veggie? Yes
Add Chicken? No
Add Other? No
Add Mushroom?
Yes
Add Extra
Cheese? No
Quantity: 3
Total: $ 39.00
Thank you!
GOTO ORDER