

Luke Talcott – ldt9gn@umsystem.edu

GitHub link:

https://github.com/LukeT11/WebMobileProgramming/tree/main/Mobile_Lessons/Mobile_ICP2

Mobile ICP2

Basic of Android Studio (Layouts, Views, Intents)

Introduction

Android Studio uses Layouts which defines the structure for a user interface within an app. All elements in the layouts are built using a hierarchy of View (user can see and interact with) and ViewGroup (invisible container defining layout structure) objects. Layout types include constraint, relative linear and grid.

Android Studio uses Views which represent the basic building block for user interface components which occupy an area on the screen with drawing and event handling. This is used for the base class for widgets (interactive UI components). View types include text, image, video, list, button, etc.

Android Studio uses Intents which is a messaging object that can be used to request an action from another app component. Three fundamental use cases include starting an activity, starting a service, and delivering a broadcast. These include Explicit (specify component by name) and Implicit (do not specify a component by name, general action to perform) intents.

Tasks

Mobile ICP2 tasks, description, and example uses.

Tasks (Android Studio Pizza Order Application):

- Main Activity should contain submit button, pizza type, quantity, text field, and checkboxes
- Add Two more options for the toppings
- Create 'ORDER' and 'SUMMARY' buttons beside each other
- When clicked, 'SUMMARY' button should display the order summary in a new activity in a list view
- When clicked, 'GO TO ORDER' button should navigate back to the Order screen
- When clicked, 'ORDER' button should have the option to send an email with the summary of order
- Include Image, Button, Spinner, Checkbox, Radio Views, and Relative/Linear Layouts
- Appropriate Icon for the application

Code

- Activity_main.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".MainActivity">
8     <!-- Layout Page -->
9     <LinearLayout
10         android:layout_width="match_parent"
11         android:layout_height="match_parent"
12         android:orientation="vertical"
13         android:paddingLeft="12dp"
14         android:paddingRight="12dp"
15         android:paddingBottom="12dp"
16         android:paddingTop="12dp"
17         tools:context="com.example.mypizza.MainActivity">
18
19         <!-- Image Pizza -->
20         <ImageView
21             android:id="@+id/pizzaImage"
22             android:layout_width="100sp"
23             android:layout_height="100sp"
24             android:layout_gravity="center"
25             android:src="@drawable/animatopizza"/>
26
27         <!-- Pizza Type Text -->
28         <TextView
29             android:id="@+id/pizzaTypeText"
30             android:layout_width="wrap_content"
31             android:layout_height="wrap_content"
32             android:layout_margin="5dp"
33             android:textSize="20sp"
34             android:text="Pizza Type"/>
35
36         <!-- Spinner select Pizza Type -->
37         <Spinner
38             android:id="@+id/pizzaTypeSpinner"
39             android:layout_width="wrap_content"
40             android:layout_height="wrap_content"/>
```

```
41 <!-- Text Toppings -->
42 <TextView
43     android:id="@+id/optionalToppings"
44     android:layout_width="wrap_content"
45     android:layout_height="wrap_content"
46     android:layout_margin="5dp"
47     android:textSize="20sp"
48     android:text="Toppings (optional)"/>
49 <!-- Layout for Topping Boxes -->
50 <LinearLayout
51     android:layout_width="match_parent"
52     android:layout_height="wrap_content"
53     android:paddingLeft="8dp"
54     android:paddingRight="8dp"
55     android:paddingBottom="8dp"
56     android:paddingTop="8dp">
57
58     <!-- Olives Checkbox -->
59     <CheckBox
60         android:id="@+id/olives"
61         android:layout_width="wrap_content"
62         android:layout_height="wrap_content"
63         android:layout_margin="2dp"
64         android:textSize="14sp"
65         android:text="Olives"/>
66
67     <!-- Green Pepper checkbox -->
68     <CheckBox
69         android:id="@+id/greenPepper"
70         android:layout_width="wrap_content"
71         android:layout_height="wrap_content"
72         android:layout_margin="2dp"
73         android:textSize="14sp"
74         android:text="Green Peppers"/>
75 </LinearLayout>
76
77 <!-- Text Quantity -->
78 <TextView
79     android:id="@+id/quantity"
```

```

79         android:layout_width="wrap_content"
80         android:layout_height="wrap_content"
81         android:layout_margin="3dp"
82         android:textSize="28sp"
83         android:text="Quantity"/>
84
85     <!-- Layout Quantity Buttons-->
86     <LinearLayout
87         android:layout_width="match_parent"
88         android:layout_height="wrap_content"
89         android:paddingLeft="8dp"
90         android:paddingRight="8dp"
91         android:paddingBottom="8dp"
92         android:paddingTop="8dp">
93
94         <!-- Button decrement quantity -->
95         <Button
96             android:id="@+id/decrement"
97             android:layout_width="40dp"
98             android:layout_height="40dp"
99             android:layout_margin="5dp"
100             android:onClick="decrement"
101             android:text="-"/>
102
103         <!--Button increment quantity -->
104         <Button
105             android:id="@+id/increment"
106             android:layout_width="40dp"
107             android:layout_height="40dp"
108             android:layout_margin="5dp"
109             android:onClick="increment"
110             android:text="+"/>
111
112         <!-- Text quantity amount-->
113         <TextView
114             android:id="@+id/quantityAmount"
115             android:layout_width="wrap_content"
116             android:layout_height="wrap_content"
117             android:layout_margin="5dp"

```

```

118             android:textSize="24sp"
119             android:text="1"/>
120
121     </LinearLayout>
122
123     <!-- EditText Input Name -->
124     <EditText
125         android:id="@+id/userInput"
126         android:layout_width="match_parent"
127         android:layout_height="wrap_content"
128         android:layout_margin="5dp"
129         android:hint="Name"/>
130
131     <!-- Button submit-->
132     <Button
133         android:id="@+id/submit"
134         android:layout_width="100dp"
135         android:layout_height="50dp"
136         android:layout_margin="3dp"
137         android:text="Submit"
138         android:onClick="submitOrder"/>
139
140     <!-- Layout Buttons order and summary-->
141     <LinearLayout
142         android:layout_width="wrap_content"
143         android:layout_height="wrap_content">
144
145         <!-- Button Order -->
146         <Button
147             android:id="@+id/order"
148             android:layout_width="100dp"
149             android:layout_height="50dp"
150             android:layout_margin="3dp"
151             android:text="Order"
152             android:visibility="gone"/>
153
154         <!-- Button Summary Page -->
155         <Button
156             android:id="@+id/summary"

```

```

157         android:layout_width="128dp"
158         android:layout_height="50dp"
159         android:layout_margin="3dp"
160         android:text="summary"
161         android:visibility="gone"/>
162     </LinearLayout>
163
164     <!-- Layout Email and Email Button -->
165     <LinearLayout
166         android:layout_width="match_parent"
167         android:layout_height="wrap_content">
168
169         <!-- EditText Input Email -->
170         <EditText
171             android:id="@+id/emailInput"
172             android:layout_width="275dp"
173             android:layout_height="wrap_content"
174             android:layout_margin="1dp"
175             android:hint="Email Optional. If not, click finish."
176             android:visibility="gone"/>
177
178         <!-- Button Send Email -->
179         <Button
180             android:id="@+id/sendBtn"
181             android:layout_width="80dp"
182             android:layout_height="50dp"
183             android:text="Email"
184             android:visibility="gone"/>
185     </LinearLayout>
186
187     <!-- Button Finish Page -->
188     <Button
189         android:id="@+id/finish"
190         android:layout_width="wrap_content"
191         android:layout_height="wrap_content"
192         android:layout_margin="5dp"
193         android:text="Finish"
194         android:visibility="gone"/>
195 </LinearLayout>

```

- MainActivity.java

```

1  package com.example.mypizza;
2
3  import ...
4
19
20  //Actions for Pizza Order page for creating order with summary, email summary.
21  public class MainActivity extends AppCompatActivity implements AdapterView.OnItemClickListener {
22
23      //Variables/Objects
24      private static final String MAIN_ACTIVITY_TAG = "MainActivity";
25      final int pizzaPrice = 10;
26      final int toppingsPrice = 1;
27      int quantity = 1;
28      Spinner pizzaType;
29      ArrayAdapter<CharSequence> adapter;
30      CheckBox olives;
31      CheckBox greenPepper;
32      EditText userInputNameView;
33      EditText emailInput;
34      Button sendBtn;
35      Button orderButton;
36      Button summaryButton;
37      Button finish;
38
39      @Override
40      protected void onCreate(Bundle savedInstanceState) {
41          super.onCreate(savedInstanceState);
42          setContentView(R.layout.activity_main);
43
44          // Assigning Objects used
45          olives = (CheckBox) findViewById(R.id.olives);
46          greenPepper = (CheckBox) findViewById(R.id.greenPepper);
47          userInputNameView = (EditText) findViewById(R.id.userInput);
48          pizzaType = findViewById(R.id.pizzaTypeSpinner);
49
50          //Assigning array to spinner to select pizza types
51          adapter = ArrayAdapter.createFromResource( context: this, R.array.pizzaOptions, android.R.layout.s
52          adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
53          pizzaType.setAdapter(adapter);
54          //pizzaType.setOnItemClickListener(this);
55      }

```

```

58     @Override
59     public void onSaveInstanceState(Bundle savedInstanceState) {
60         savedInstanceState.putBoolean("olivesKey", olives.isChecked());
61         savedInstanceState.putBoolean("greenPepperKey", greenPepper.isChecked());
62         savedInstanceState.putString("nameInputKey", userInputNameView.getText().toString());
63         savedInstanceState.putInt("pizzaTypeKey", pizzaType.getSelectedItemPosition());
64         savedInstanceState.putInt("pizzaQuantity", quantity);
65
66         super.onSaveInstanceState(savedInstanceState);
67     }
68
69     //Restore instance state with saved bundle of order
70     @Override
71     protected void onRestoreInstanceState(Bundle savedInstanceState) {
72         Boolean savedOlivesChecked = savedInstanceState.getBoolean( key: "olivesKey");
73         olives.setChecked(savedOlivesChecked);
74
75         Boolean savedGreenPeppersChecked = savedInstanceState.getBoolean( key: "greenPepperKey");
76         greenPepper.setChecked(savedGreenPeppersChecked);
77
78         String savedNameInput = savedInstanceState.getString( key: "nameInputKey");
79         userInputNameView.setText(savedNameInput);
80
81         int savedPizzaType = savedInstanceState.getInt( key: "pizzaTypeKey");
82         pizzaType.setSelection(savedPizzaType);
83
84         int savedPizzaQuantity = savedInstanceState.getInt( key: "pizzaQuantity");
85         quantity = savedPizzaQuantity;
86     }
87
88     //Displays message of selected Pizza type spinner
89     @Override
90     public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {
91         String text = adapterView.getItemAtPosition(i).toString();
92         Toast.makeText(adapterView.getContext(), text, Toast.LENGTH_SHORT).show();
93     }
94
95     //On nothing selected from Pizza type spinner
96     @Override
97     public void onNothingSelected(AdapterView<?> adapterView) {
98     }

```

```

100     /**
101      * This method is called when the order button is clicked.
102      */
103
104     //Submit Order function get selected data for the pizza order. Used with 'submit' button when click
105     public void submitOrder(View view) {
106
107         // get user input
108         String userInputName = userInputNameView.getText().toString();
109
110         //Make sure name EditText box filled out. If not, triggers message.
111         if (TextUtils.isEmpty(userInputName)) {
112             Toast.makeText( context: this, text: "Must fill out Name", Toast.LENGTH_SHORT).show();
113         }
114
115         else {
116             //Get selected pizza type
117             String spinnerText = pizzaType.getSelectedItem().toString();
118             Toast.makeText(pizzaType.getContext(), spinnerText, Toast.LENGTH_SHORT).show();
119
120             // check if olives is selected
121             boolean hasOlives = olives.isChecked();
122
123             // check if greenPepper is selected
124             boolean hasGreenPeppers = greenPepper.isChecked();
125
126             // calculate and store the total price
127             float totalPrice = calculatePrice(hasOlives, hasGreenPeppers);
128
129             // create and store the order summary
130             String orderSummaryMessage = createOrderSummary(userInputName, spinnerText, hasOlives, has
131
132             // Code for making the buttons work(i.e implement the implicit and explicit intents)
133             orderButton = findViewById(R.id.order);
134             summaryButton = findViewById(R.id.summary);
135
136             showButton(orderButton);
137             showButton(summaryButton);
138
139             //Button click listener for Order Button to trigger sending an email function
140             orderButton.setOnClickListener(new View.OnClickListener() {
141                 @Override

```

```

139 //Button click listener for Order Button to trigger sending an email function
140 orderButton.setOnClickListener(new View.OnClickListener() {
141     @Override
142     public void onClick(View view) { sendEmail(userInputName, orderSummaryMessage); }
143 });
144
145 //Button click listener for Summary Button to trigger going to summary page/activity,
146 //takes the summary input in form of a string
147 summaryButton.setOnClickListener(new View.OnClickListener() {
148     @Override
149     public void onClick(View view) {
150         Intent summaryIntent = new Intent( packageContext.MainActivity.this, MainActivity2.class);
151         summaryIntent.putExtra( name: "Summary", orderSummaryMessage);
152         startActivity(summaryIntent);
153     }
154 });
155
156 //Function to make a button visible
157 public void showButton(Button btn) { btn.setVisibility(View.VISIBLE); }
158
159 //Function to make a EditText visible
160 public void showEditText(EditText txtBox) {txtBox.setVisibility(View.VISIBLE);}
161
162 //Function send email with summary to the user
163 public void sendEmail(String name, String output) {
164     // Write the relevant code for triggering email
165     emailInput = (EditText) findViewById(R.id.emailInput);
166     sendBtn = (Button) findViewById(R.id.sendBtn);
167     finish = (Button) findViewById(R.id.finish);
168
169     showEditText(emailInput);
170     showButton(sendBtn);
171     showButton(finish);
172
173     //Button click listener to send email with user email, summary and name
174     sendBtn.setOnClickListener(new View.OnClickListener() {
175         @Override
176         public void onClick(View view) {
177             if (TextUtils.isEmpty(emailInput.getText().toString())) {
178                 Toast.makeText(MainActivity.this, "Email is Empty", Toast.LENGTH_SHORT).show();
179             } else {
180                 String email = emailInput.getText().toString();
181                 Intent intent = new Intent(Intent.ACTION_VIEW);
182                 intent.putExtra(Intent.EXTRA_EMAIL, email);
183                 intent.putExtra(Intent.EXTRA_SUBJECT, "Summary for: " + name);
184                 intent.putExtra(Intent.EXTRA_TEXT, output);
185                 if (intent.resolveActivity(getPackageManager()) != null) {
186                     startActivity(intent);
187                     Toast.makeText(MainActivity.this, "Email", Toast.LENGTH_SHORT).show();
188                 }
189             }
190         }
191     });
192 }
193
194 //Return string for boolean
195 private String boolToString(boolean bool) {
196     return bool ? ("Yes") : ("No");
197 }
198
199 //Returns String of Pizza Order Summary
200 private String createOrderSummary(String userInputName, String spinnerText, boolean hasOlives, boolean hasGreenPeppers, float price) {
201     String orderSummaryMessage = "Name: (Vijaya)" + "\n" +
202         "Pizza Type: " + spinnerText + "\n" +
203         "Add Olives? (yes)" + "\n" +
204         "Add Green Peppers? (yes)" + "\n" +
205         "Quantity: (2)" + "\n" +
206         "Total: $ (10)";
207     return orderSummaryMessage;
208 }
209
210 //Method to calculate the total price
211 //Returns total Price
212 }

```

```

213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

```

```

224 */
225
226 //Calculates and return the price of the pizza order with the quantity
227 private float calculatePrice(boolean hasOlives, boolean hasGreenPeppers) {
228     int basePrice = pizzaPrice;
229
230     if (hasOlives) {
231         basePrice += toppingsPrice;
232     }
233     if (hasGreenPeppers) {
234         basePrice += toppingsPrice;
235     }
236     return quantity * basePrice;
237 }
238
239 /**
240  * This method displays the given quantity value on the screen.
241  */
242
243 //Display quantity of Pizzas with TextView when incremented or decremented
244 private void display(int number) {
245     TextView quantityTextView = (TextView) findViewById(R.id.quantityAmount);
246     quantityTextView.setText("" + number);
247 }
248
249 /**
250  * This method increments the quantity of Pizzas by one
251  *
252  * @param view on passes the view that we are working with to the method
253  */
254
255 //Increment the quantity of pizzas and update number shown
256 public void increment(View view) {
257     if (quantity < 100) {
258         quantity = quantity + 1;
259         display(quantity);
260     } else {
261         Log.i(tag, "MainActivity", msg, "Please select less than 50 Pizzas");
262         Context context = getApplicationContext();
263         String lowerLimitToast = "Must Select less than 50 Pizzas";
264         int duration = Toast.LENGTH_SHORT;
265         Toast toast = Toast.makeText(context, lowerLimitToast, duration);

```

```

266 toast.show();
267         return;
268     }
269 }
270
271 /**
272  * This method decrements the quantity of Pizzas by one
273  *
274  * @param view passes on the view that we are working with to the method
275  */
276
277 //Decrement the quantity of pizzas and update number shown
278 public void decrement(View view) {
279     if (quantity > 1) {
280         quantity = quantity - 1;
281         display(quantity);
282     } else {
283         Log.i(tag, "MainActivity", msg, "Please select at least 1 Pizza");
284         Context context = getApplicationContext();
285         String upperLimitToast = "Must Select at least 1 Pizza";
286         int duration = Toast.LENGTH_SHORT;
287         Toast toast = Toast.makeText(context, upperLimitToast, duration);
288         toast.show();
289         return;
290     }
291 }
292 }

```

- Acitivity_main2.xml

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/a
3  xmlns:app="http://schemas.android.com/apk/res-auto"
4  xmlns:tools="http://schemas.android.com/tools"
5  android:layout_width="match_parent"
6  android:layout_height="match_parent"
7  tools:context=".MainActivity2">
8
9  <!-- Layout Page -->
10 <LinearLayout
11     android:layout_width="match_parent"
12     android:layout_height="match_parent"
13     android:orientation="vertical"
14     android:paddingLeft="12dp"
15     android:paddingRight="12dp"
16     android:paddingBottom="12dp"
17     android:paddingTop="12dp"
18     tools:context="com.example.mypizza.MainActivity2">
19
20     <!-- ListView Summary of Order -->
21     <ListView
22         android:id="@+id/summaryList"
23         android:layout_width="match_parent"
24         android:layout_height="wrap_content"/>
25
26     <!-- Button Back to Order Page -->
27     <Button
28         android:id="@+id/orderPage"
29         android:layout_width="180dp"
30         android:layout_height="60dp"
31         android:layout_margin="10dp"
32         android:text="Go To Order"
33         android:onClick="orderScreen"/>
34
35 </LinearLayout>
36 </androidx.constraintlayout.widget.ConstraintLayout>

```

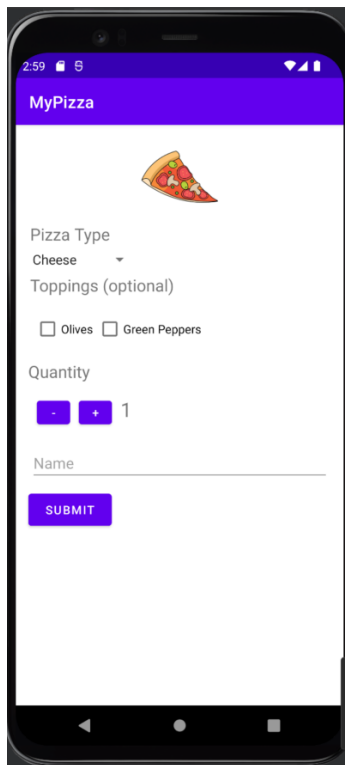
- MainActivity2.java

```

1  package com.example.mypizza;
2
3  import ...
4
12
13 public class MainActivity2 extends AppCompatActivity {
14
15     ArrayAdapter arrayAdapter;
16     ListView listView;
17
18     //Creates a ListView with summary from the order page, taking string from Order Page with Summary
19     @Override
20     protected void onCreate(Bundle savedInstanceState) {
21         super.onCreate(savedInstanceState);
22         setContentView(R.layout.activity_main2);
23
24         listView = findViewById(R.id.summaryList);
25         ArrayList<String> summaryArray = new ArrayList<>();
26
27         Intent i = getIntent();
28
29         String summaryText = i.getStringExtra( name: "Summary");
30         String[] splitText = summaryText.split( regex: "\n");
31
32         for (int l=0; l < splitText.length; l++ ) {
33             summaryArray.add(splitText[l]);
34         }
35
36         ArrayAdapter arrayAdapter = new ArrayAdapter( context: this, android.R.layout.simple_list_item_1, summaryArray
37         listView.setAdapter(arrayAdapter);
38     }
39
40     //Function with Button click, back to MainActivity1 Order Page
41     public void orderScreen(View view) {
42         Intent orderIntent = new Intent( packageContext: MainActivity2.this, MainActivity.class);
43         startActivity(orderIntent);
44     }
45 }


```


Output



2:59

MyPizza



Pizza Type
Cheese ▾

Toppings (optional)

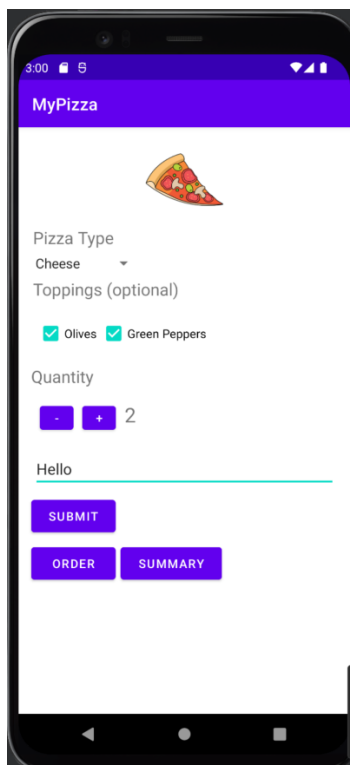
☐ Olives ☐ Green Peppers

Quantity

- + 1


Name

SUBMIT



3:00

MyPizza



Pizza Type
Cheese ▾

Toppings (optional)

☒ Olives ☒ Green Peppers

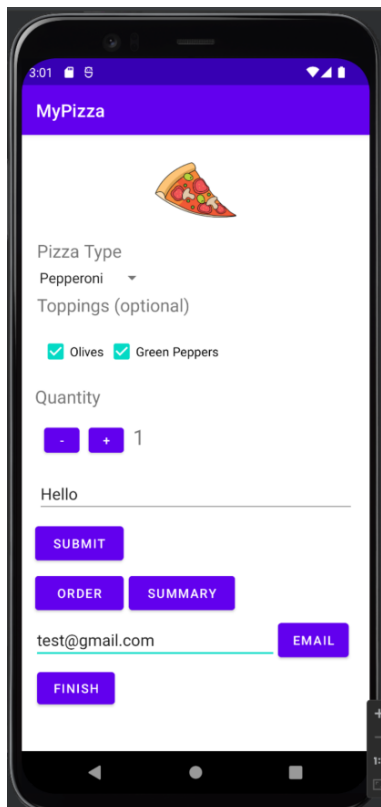
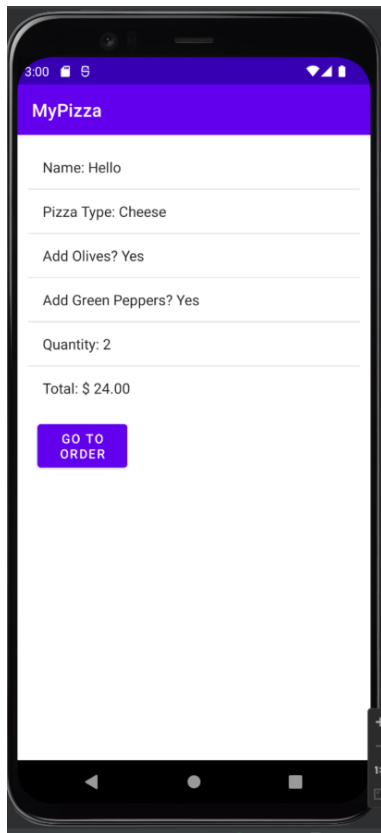
Quantity

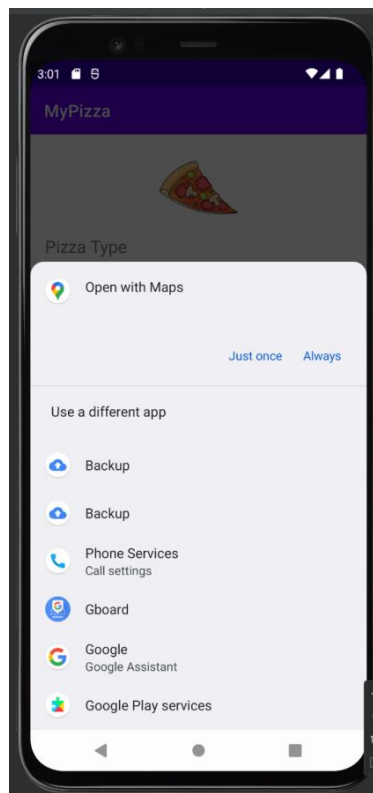
- + 2

Hello

SUBMIT

ORDER SUMMARY





- Image



- Button

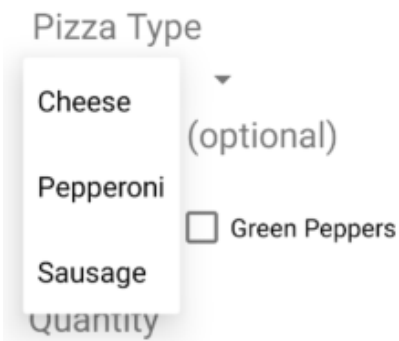


- Checkbox

Toppings (optional)

☒ Olives ☒ Green Peppers

- Spinner



- Icon



Contributors

I worked independently, so I am the sole contributor.

Conclusion

I used android studio to develop a basic pizza ordering application while learning to use layouts, views, intents, and other widgets along the way. The application contains a main activity screen which requires the user to create a pizza order. I used a spinner for the pizza types, checkboxes for the topping choices, and buttons to choose the quantity. The main activity also includes a name box for the order using an edit text, an order button which allows the user the option to send a summary of there order by email and a summary button. The summary button takes the user to a second activity, a summary screen, which displays a summary of their order using a list view and another button to return the user back to the order screen. I used different layouts to design the activity screens and intents to switch between activities or screens and for the user to send the summary by email. Lastly, I included an image on the main activity and changed the icon of the application to a pizza icon.