

DEAKIN UNIVERSITY

MOBILE APPLICATION DEVELOPMENT

ONTRACK SUBMISSION

Pass Task 7.1

Submitted By:
Akashdeep AKASHDEEP
s223040483
2024/05/09 00:04

Tutor:
Shiva POKHREL

Outcome	Weight
Unit Learning Outcome 1	◆◆◆◆◇
Unit Learning Outcome 2	◆◆◆◆◇
Unit Learning Outcome 3	◆◆◆◆◇
Unit Learning Outcome 4	◆◆◆◆◇

task 7.1p

May 9, 2024

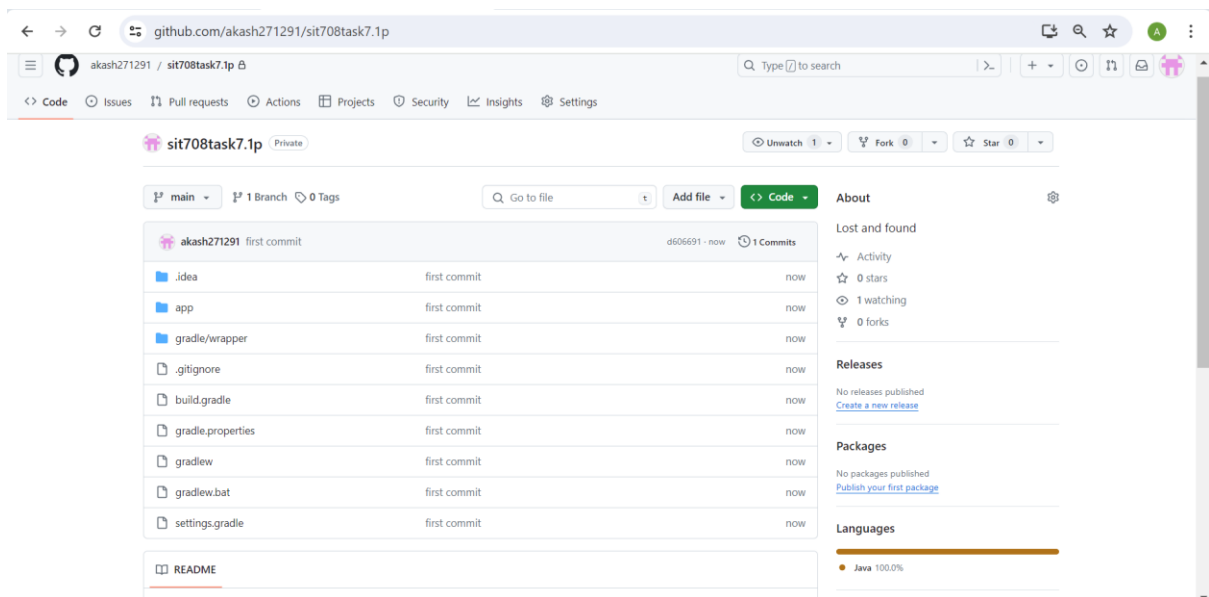


SIT708 TASK 7.1P

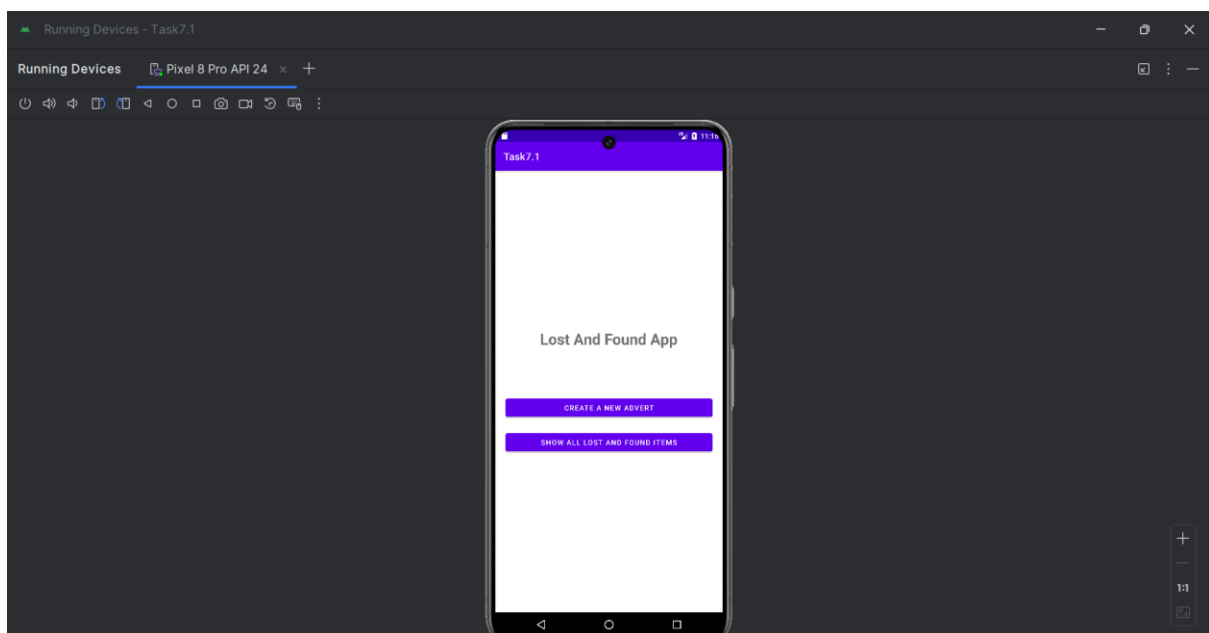
Video link: <https://youtu.be/bKiAjvQnoJc>

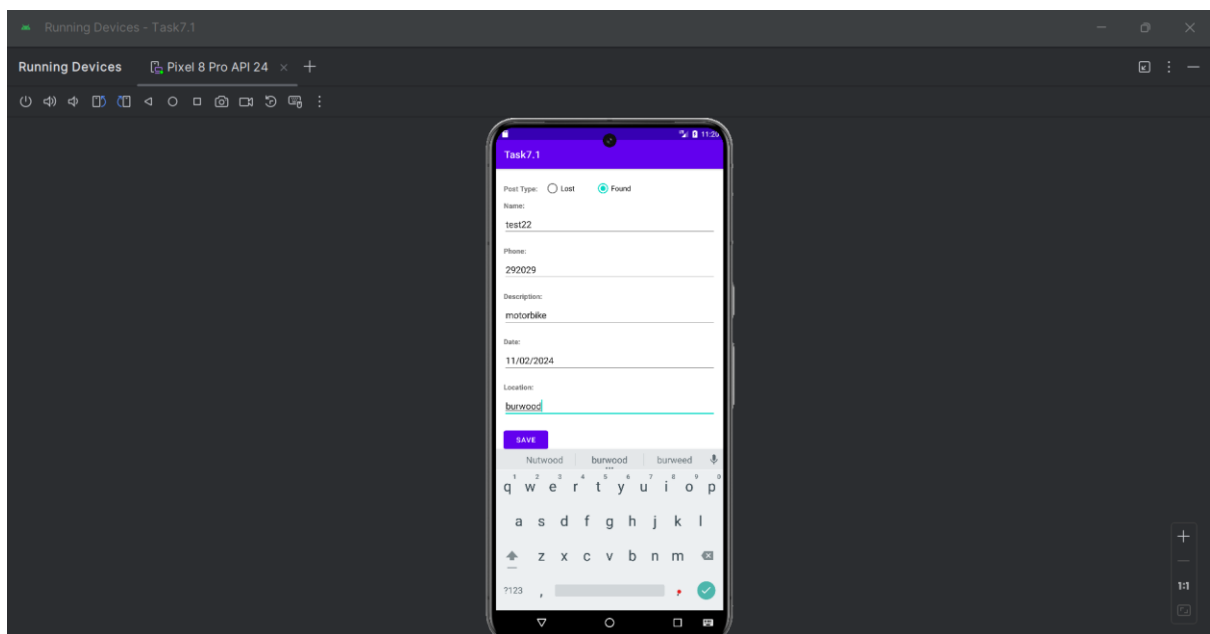
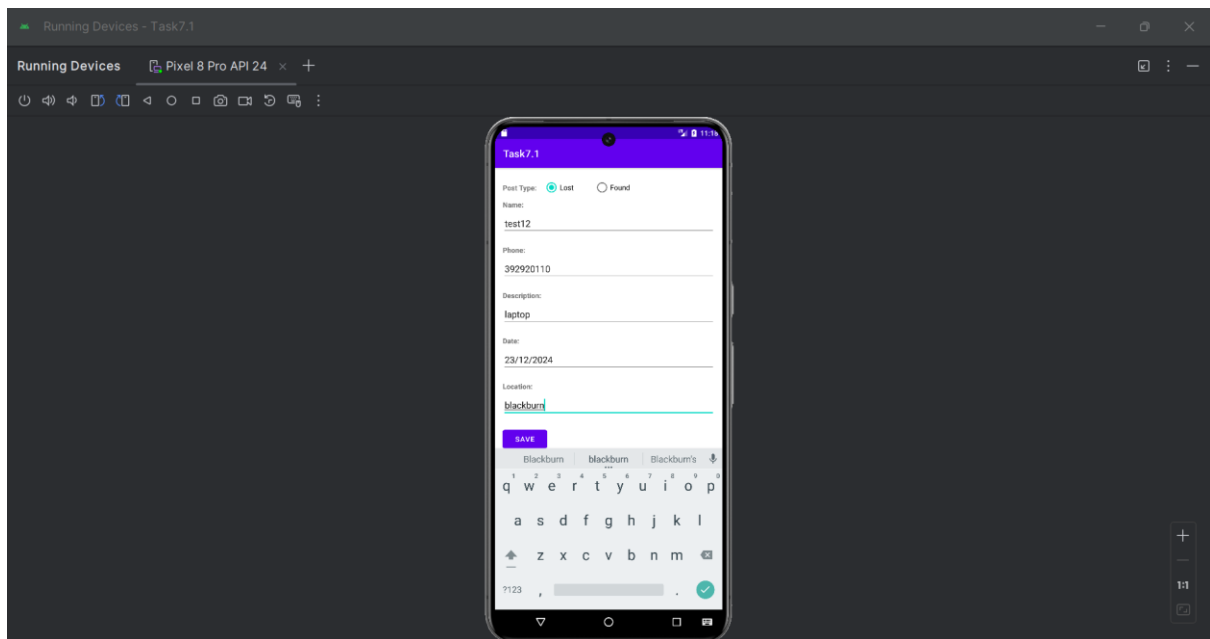
GitHub Link: <https://github.com/akash271291/sit708task7.1p.git>

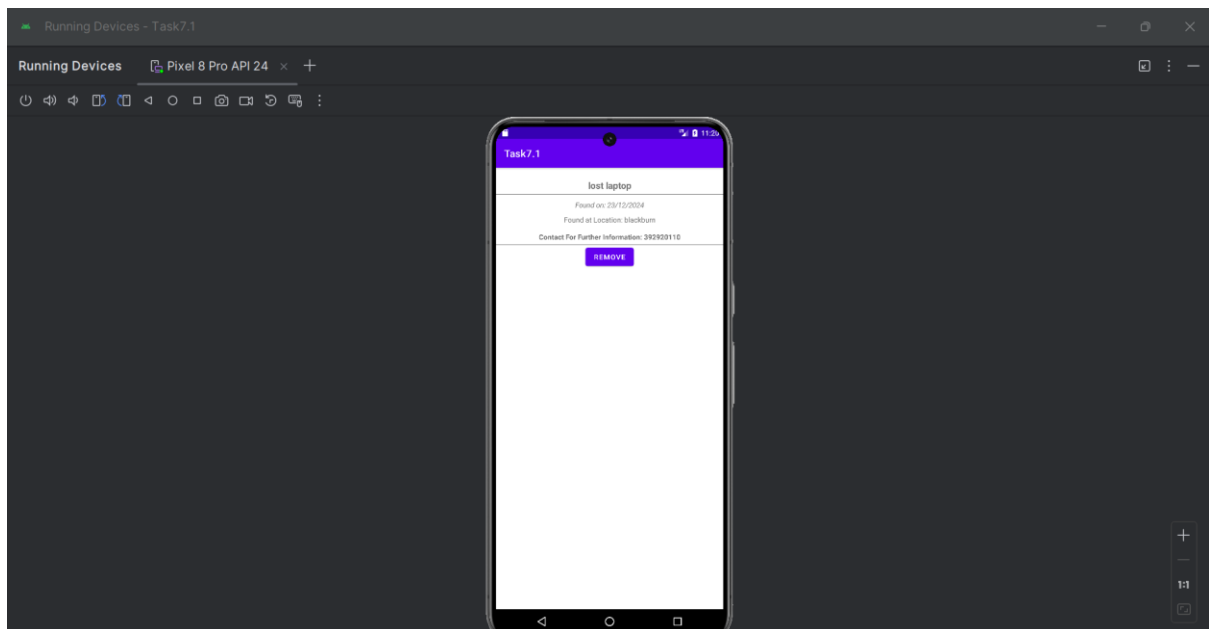
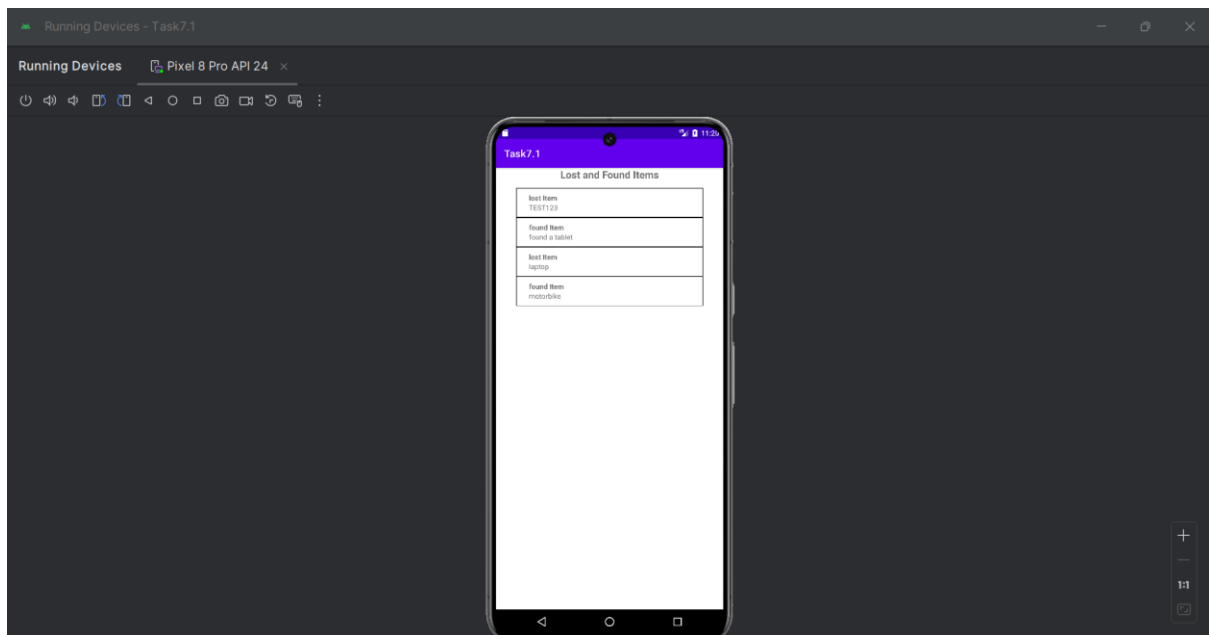
GitHub screenshot:



Output:







Below mentioned are few ideas for future directions

- collaboration with emergency service providers such as police departments, fire departments, and medical facilities will make it easier to report and retrieve lost property in an emergency situations which can improve response times and assist users in critical moments.
- By using community techniques to confirm the authenticity of lost item postings, the app's information can be made reliable. users might be encouraged to verify postings by

confirming details or providing additional information, helping to filter out false or misleading reports.

- Having multilingual support can increase the app's usability for a wider variety of users. The app can accommodate users with varying language backgrounds and promote communication amongst those who might not speak the same language by offering translations for multiple languages.
- Adding gaming components, such leaderboards, prizes, or badges, can encourage user involvement and improve the user experience when reporting lost or discovered objects. The app can motivate users to keep using it and give them a sense of achievement by making item recovery seem like a game.
- By integrating the app with well-known social media sites, you may grow the number of users and boost connection. Posts about lost items could be shared by users with their social networks, reaching a larger audience and increasing the chances that the item's rightful owner will be found.

```
1 package com.example.task71;
2
3 import androidx.appcompat.app.AppCompatActivity;
4
5 import android.content.Intent;
6 import android.os.Bundle;
7 import android.view.View;
8
9 import com.example.task71.databinding.ActivityMainBinding;
10
11 public class MainActivity extends AppCompatActivity {
12
13     ActivityMainBinding binding;
14
15     @Override
16     protected void onCreate(Bundle savedInstanceState) {
17         super.onCreate(savedInstanceState);
18
19
20         binding = ActivityMainBinding.inflate(getLayoutInflater());
21         setContentView(binding.getRoot());
22
23         binding.createButton.setOnClickListener(new View.OnClickListener() {
24             @Override
25             public void onClick(View view) {
26                 Intent newIntent = new Intent(MainActivity.this,
27                     ↪ CreateActivity.class);
28                 startActivityForResult(newIntent, 1);
29                 finish();
30             }
31         });
32
33         binding.showButton.setOnClickListener(new View.OnClickListener() {
34             @Override
35             public void onClick(View view) {
36                 Intent newIntent = new Intent(MainActivity.this,
37                     ↪ ListAllActivity.class);
38                 startActivityForResult(newIntent, 1);
39                 finish();
40             }
41         });
42     }
43
44     -----
45
46 package com.example.task71;
47
48
49 import android.content.Context;
50 import android.os.Bundle;
51 import android.view.LayoutInflater;
```

```
52 import android.view.View;
53 import android.view.ViewGroup;
54 import android.widget.TextView;
55
56 import androidx.annotation.NonNull;
57 import androidx.appcompat.app.AppCompatActivity;
58 import androidx.fragment.app.Fragment;
59 import androidx.recyclerview.widget.RecyclerView;
60
61 import java.io.Serializable;
62 import java.util.List;
63
64 public class AdaptorList extends RecyclerView.Adapter<AdaptorList.MyViewHolder> {
65
66     List<Item> items;
67     Context mContext;
68
69     public AdaptorList(Context mContext, List<Item> items) {
70         this.mContext = mContext;
71         this.items = items;
72     }
73
74     @Override
75     public void onBindViewHolder(@NonNull AdaptorList.MyViewHolder holder, int
76         ↪ position) {
77
78         holder.typeTextView.setText(items.get(position).getType() + " Item");
79         holder.descriptionTextView.setText(items.get(position).getDescription());
80     }
81
82     @Override
83     public int getItemCount() {
84
85         return items.size();
86     }
87
88     @NonNull
89     @Override
90     public AdaptorList.MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent,
91         ↪ int viewType) {
92
93         View view = LayoutInflater.from(mContext).inflate(R.layout.showallrecyclervj
94         ↪ ew_layout, parent,
95         ↪ false);
96
97         return new MyViewHolder(view);
98     }
99
100     public class MyViewHolder extends RecyclerView.ViewHolder implements
101         ↪ View.OnClickListener{
102
103         TextView descriptionTextView, typeTextView;
```

```
100
101     public MyViewHolder(@NonNull View itemView) {
102         super(itemView);
103
104         descriptionTextView = itemView.findViewById(R.id.descriptionTextView);
105         typeTextView = itemView.findViewById(R.id.typeTextView);
106
107         itemView.setOnClickListener(this);
108     }
109
110     @Override
111     public void onClick(View view) {
112
113         AppCompatActivity activity = (AppCompatActivity) view.getContext();
114         Fragment itemFragment = new FirstFragment();
115
116         Bundle args = new Bundle();
117         int position = getLayoutPosition();
118         args.putSerializable("data", (Serializable) items.get(position));
119         args.putInt("position", position);
120
121         itemFragment.setArguments(args);
122
123
124         activity.getSupportFragmentManager().beginTransaction()
125             .replace(R.id.frameLayout, itemFragment)
126             .addToBackStack(null)
127             .commit();
128
129     }
130 }
131 }
132
133 -----
134
135 package com.example.task71;
136
137 import androidx.appcompat.app.AppCompatActivity;
138
139 import android.content.Intent;
140 import android.os.Bundle;
141 import android.text.TextUtils;
142 import android.view.View;
143 import android.widget.Toast;
144
145 import com.example.task71.databinding.ActivityCreateBinding;
146
147 public class CreateActivity extends AppCompatActivity {
148
149     ActivityCreateBinding binding;
150     DatabaseHelper databaseHelper;
151
152     @Override
```



```
153     protected void onCreate(Bundle savedInstanceState) {
154         super.onCreate(savedInstanceState);
155         binding = ActivityCreateBinding.inflate(getLayoutInflater());
156         setContentView(binding.getRoot());
157
158         databaseHelper = new DatabaseHelper(this);
159
160         binding.saveButton.setOnClickListener(new View.OnClickListener() {
161             @Override
162             public void onClick(View view) {
163
164                 String date = binding.editTextDate.getText().toString();
165                 String description =
166                     ↳ binding.editTextDescription.getText().toString();
167                 String location = binding.editTextLocation.getText().toString();
168                 String name = binding.editTextName.getText().toString();
169                 String phone = binding.editTextPhone.getText().toString();
170                 String lostFound = "";
171                 //check if item radio button is lost or found
172                 if(binding.radioButtonLost.isChecked()) {
173                     lostFound = "lost";
174                 } else if (binding.radioButtonFound.isChecked()) {
175                     lostFound = "found";
176                 }
177
178                 if(TextUtils.isEmpty(name) || TextUtils.isEmpty(phone) ||
179                     ↳ TextUtils.isEmpty(description) ||
180                     ↳ TextUtils.isEmpty(date) || TextUtils.isEmpty(location) ||
181                     ↳ TextUtils.isEmpty(lostFound)) {
182                     Toast.makeText(CreateActivity.this, "Please fill in all
183                         ↳ information", Toast.LENGTH_SHORT).show();
184                 } else {
185                     Item newItem = new Item(0, name, lostFound, phone, description,
186                         ↳ date, location);
187                     databaseHelper.onInsert(newItem);
188                     Intent newIntent = new Intent(CreateActivity.this,
189                         ↳ MainActivity.class);
190                     startActivityForResult(newIntent, 1);
191                     finish();
192                 }
193             }
194         });
195     }
196
197
198
199     package com.example.task71;
```

```
200
201 import android.content.ContentValues;
202 import android.content.Context;
203 import android.database.Cursor;
204 import android.database.sqlite.SQLiteDatabase;
205 import android.database.sqlite.SQLiteOpenHelper;
206 import android.util.Log;
207
208 import androidx.annotation.Nullable;
209
210 import java.util.ArrayList;
211 import java.util.List;
212
213 public class DatabaseHelper extends SQLiteOpenHelper {
214
215     public static final String MY_DATABASE = "lostfound.db";
216     public static final int VERSION = 1;
217
218     public static final String COLUMN_ID = "id";
219     public static final String TABLE_NAME = "items";
220     public static final String COLUMN_NAME = "name";
221     public static final String COLUMN_TYPE = "type";
222     public static final String COLUMN_PHONE = "phone";
223     public static final String COLUMN_DESCRIPTION = "description";
224     public static final String COLUMN_DATE = "date";
225     public static final String COLUMN_LOCATION = "location";
226
227
228
229     public DatabaseHelper(@Nullable Context context) {
230         super(context, MY_DATABASE, null, VERSION);
231     }
232
233     @Override
234     public void onCreate(SQLiteDatabase db) {
235         //create table
236         String CREATE_TABLE = "CREATE TABLE " + TABLE_NAME + " (" +
237             COLUMN_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
238             COLUMN_NAME + " TEXT, " +
239             COLUMN_TYPE + " TEXT, " +
240             COLUMN_PHONE + " TEXT, " +
241             COLUMN_DESCRIPTION + " TEXT, " +
242             COLUMN_DATE + " TEXT, " +
243             COLUMN_LOCATION + " TEXT)";
244         db.execSQL(CREATE_TABLE);
245
246     }
247
248
249     public long onInsert(Item item) {
250         //get the database
251         SQLiteDatabase db = this.getWritableDatabase();
252         //get the values from the input item variable
```

```
253     ContentValues contentValues = new ContentValues();
254     contentValues.put(COLUMN_NAME, item.getName());
255     contentValues.put(COLUMN_TYPE, item.getType());
256     contentValues.put(COLUMN_PHONE, item.getPhone());
257     contentValues.put(COLUMN_DESCRIPTION, item.getDescription());
258     contentValues.put(COLUMN_DATE, item.getDate());
259     contentValues.put(COLUMN_LOCATION, item.getLocation());
260     //create new row in the database with the values
261     long newRowId = db.insert(TABLE_NAME, null, contentValues);
262     db.close();
263     return newRowId;
264
265 }
266
267 @Override
268 public void onUpgrade(SQLiteDatabase db, int i, int i1) {
269
270     String DROP_TABLE = "DROP TABLE IF EXISTS " + TABLE_NAME;
271     db.execSQL(DROP_TABLE);
272     onCreate(db);
273
274 }
275
276 public List<Item> getAllItems() {
277     Log.v("getallitems", "true");
278
279     List<Item> items = new ArrayList<>();
280     SQLiteDatabase db = this.getReadableDatabase();
281     //get all items from database query
282     String selectQuery = "SELECT * FROM " + TABLE_NAME;
283     Cursor cursor = db.rawQuery(selectQuery, null);
284
285     String name, type, phone, description, date, location;
286     int id;
287
288     //iterate over the database adding to the items List to be used within the
289     ↪ program
290     if(cursor.moveToFirst()) {
291         do {
292             Log.v("database",
293                 ↪ cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_NAME)));
294             id = cursor.getInt(cursor.getColumnIndexOrThrow(COLUMN_ID));
295             name = cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_NAME));
296             type = cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_TYPE));
297             phone =
298                 ↪ cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_PHONE));
299             description = cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_
300                 ↪ DESCRIPTION));
301             date = cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_DATE));
302             location =
303                 ↪ cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_LOCATION));
304             Item item = new Item(id, name, type, phone, description, date,
305                 ↪ location);
```

```

300         items.add(item);
301     } while (cursor.moveToNext());
302
303     }
304     cursor.close();
305     return items;
306 }
307
308 public void deleteItem(int id)
309 {
310     //get the database and delete by column id
311     SQLiteDatabase db = this.getWritableDatabase();
312     db.execSQL("DELETE FROM " + TABLE_NAME + " WHERE " + COLUMN_ID + "=" + id
313         ↪ + "'");
314     db.close();
315 }
316
317 }
318 -----
319
320 package com.example.task71;
321
322 import android.os.Bundle;
323
324 import androidx.fragment.app.Fragment;
325
326 import android.view.LayoutInflater;
327 import android.view.View;
328 import android.view.ViewGroup;
329 import android.widget.Button;
330 import android.widget.TextView;
331
332 import java.io.Serializable;
333 public class FirstFragment extends Fragment {
334
335     private int position;
336     private Item item;
337
338     DatabaseHelper databaseHelper;
339
340     public FirstFragment() {
341
342         this.item = item;
343     }
344
345     @Override
346     public void onCreate(Bundle savedInstanceState) {
347         super.onCreate(savedInstanceState);
348         if (getArguments() != null) {
349             //when fragment entered, get the item clicked data and the position in
350             ↪ the recyclerview
351             item = (Item) getArguments().getSerializable("data");

```

```
351         position = getArguments().getInt("position");
352     }
353 }
354
355 public static FirstFragment newInstance(Item item) {
356     FirstFragment fragment = new FirstFragment();
357     Bundle args = new Bundle();
358     args.putSerializable("data", (Serializable) item);
359     fragment.setArguments(args);
360     return fragment;
361 }
362
363
364 @Override
365 public View onCreateView(LayoutInflater inflater, ViewGroup container,
366     Bundle savedInstanceState) {
367     // Inflate the layout for this fragment
368     View view = inflater.inflate(R.layout.fragment_item, null);
369
370     if(getArguments() != null) {
371
372         TextView textViewLocation = view.findViewById(R.id.textViewLocation);
373         TextView textViewPhone = view.findViewById(R.id.textViewPhone);
374         TextView textViewTitle = view.findViewById(R.id.textViewTitle);
375         TextView textViewDate = view.findViewById(R.id.textViewDate);
376         Button removeButton = view.findViewById(R.id.removeButton);
377
378         textViewTitle.setText(item.getType() + " " + item.getDescription());
379         textViewDate.setText("Found on: " + item.getDate());
380         textViewLocation.setText("Found at Location: " + item.getLocation());
381         textViewPhone.setText("Contact For Further Information: " +
382             ↵ item.getPhone());
383
384         removeButton.setOnClickListener(new View.OnClickListener() {
385             @Override
386             public void onClick(View view) {
387                 //delete the item from the database
388                 //pass the postition reference back to the activity to update
389                 ↵ the recyclerview that an item has been removed
390                 databaseHelper = new DatabaseHelper(getContext());
391                 databaseHelper.deleteItem(item.getId());
392                 ((ListAllActivity) getActivity()).dataChanged(position);
393                 getParentFragmentManager().popBackStack();
394             }
395         });
396     }
397
398     return view;
399 }
400 }
401
```

```
402 -----
403
404 package com.example.task71;
405
406 import java.io.Serializable;
407
408 public class Item implements Serializable {
409
410     String name, type, phone, description, date, location;
411     int id;
412
413     public Item(int id, String name, String type, String phone, String description,
414 ↪ String date, String location) {
415         this.id = id;
416         this.name = name;
417         this.type = type;
418         this.phone = phone;
419         this.description = description;
420         this.date = date;
421         this.location = location;
422     }
423
424     public String getName() {
425         return name;
426     }
427
428     public void setName(String name) {
429         this.name = name;
430     }
431
432     public String getType() {
433         return type;
434     }
435
436     public void setType(String type) {
437         this.type = type;
438     }
439
440     public String getPhone() {
441         return phone;
442     }
443
444     public void setPhone(String phone) {
445         this.phone = phone;
446     }
447
448     public String getDescription() {
449         return description;
450     }
451
452     public void setDescription(String description) {
453         this.description = description;
454     }
455 }
```

```
454
455     public String getDate() {
456         return date;
457     }
458
459     public void setDate(String date) {
460         this.date = date;
461     }
462
463     public String getLocation() {
464         return location;
465     }
466
467     public void setLocation(String location) {
468         this.location = location;
469     }
470
471     public int getId() {
472         return id;
473     }
474
475     public void setId(int id) {
476         this.id = id;
477     }
478
479 }
480
481
482
483 -----
484
485 package com.example.task71;
486
487 import androidx.appcompat.app.AppCompatActivity;
488 import androidx.recyclerview.widget.LinearLayoutManager;
489 import androidx.recyclerview.widget.RecyclerView;
490
491 import android.os.Bundle;
492 import android.util.Log;
493
494 import com.example.task71.databinding.ActivityShowAllBinding;
495
496 import java.util.List;
497
498 public class ListAllActivity extends AppCompatActivity {
499
500     ActivityShowAllBinding binding;
501     DatabaseHelper databaseHelper;
502
503     RecyclerView.LayoutManager layoutManager;
504     AdaptorList listAllAdaptor;
505
506     List<Item> itemList;
```

```
507
508     public void dataChanged(int position) {
509         itemList.remove(position);
510         Log.v("onresume", String.valueOf(itemList.size()));
511         listAllAdaptor.notifyItemRemoved(position);
512     }
513
514     @Override
515     protected void onCreate(Bundle savedInstanceState) {
516         super.onCreate(savedInstanceState);
517         binding = ActivityShowAllBinding.inflate(getLayoutInflater());
518         setContentView(binding.getRoot());
519
520         databaseHelper = new DatabaseHelper(this);
521         itemList = databaseHelper.getAllItems();
522         layoutManager = new LinearLayoutManager(this);
523         listAllAdaptor = new AdaptorList(this, itemList);
524         databaseHelper = new DatabaseHelper(this);
525
526
527         binding.recyclerView.setAdapter(listAllAdaptor);
528         binding.recyclerView.setLayoutManager(layoutManager);
529
530     }
531
532 }
533 }
```