

Department of Information Technology
Academic -2021-22

Assignment No: 02

Subject: Mobile Application Development

Roll No.: 18IT1050

Name of Student: Soham Salkar

Class/Batch: A/A2

Date of Performance:

Date of Submission:

Grade:

Signature:

Q1

Ans. SQLite is a very light version of SQL supporting many features of SQL. Basically it has been developed for small devices like mobile phones, tablets etc. SQLite is a third party open-sourced and in-process database engine.

SQL server compact is from Microsoft and is a stripped-down version of SQL server.

They are two competing database engines. SQL is a query language. SQLite is an embeddable relational database management system. SQLite also doesn't require a special database server or anything. It's just a direct filesystem engine that SQL syntax. Technically, SQLite is not open-source software but rather public domain.

```

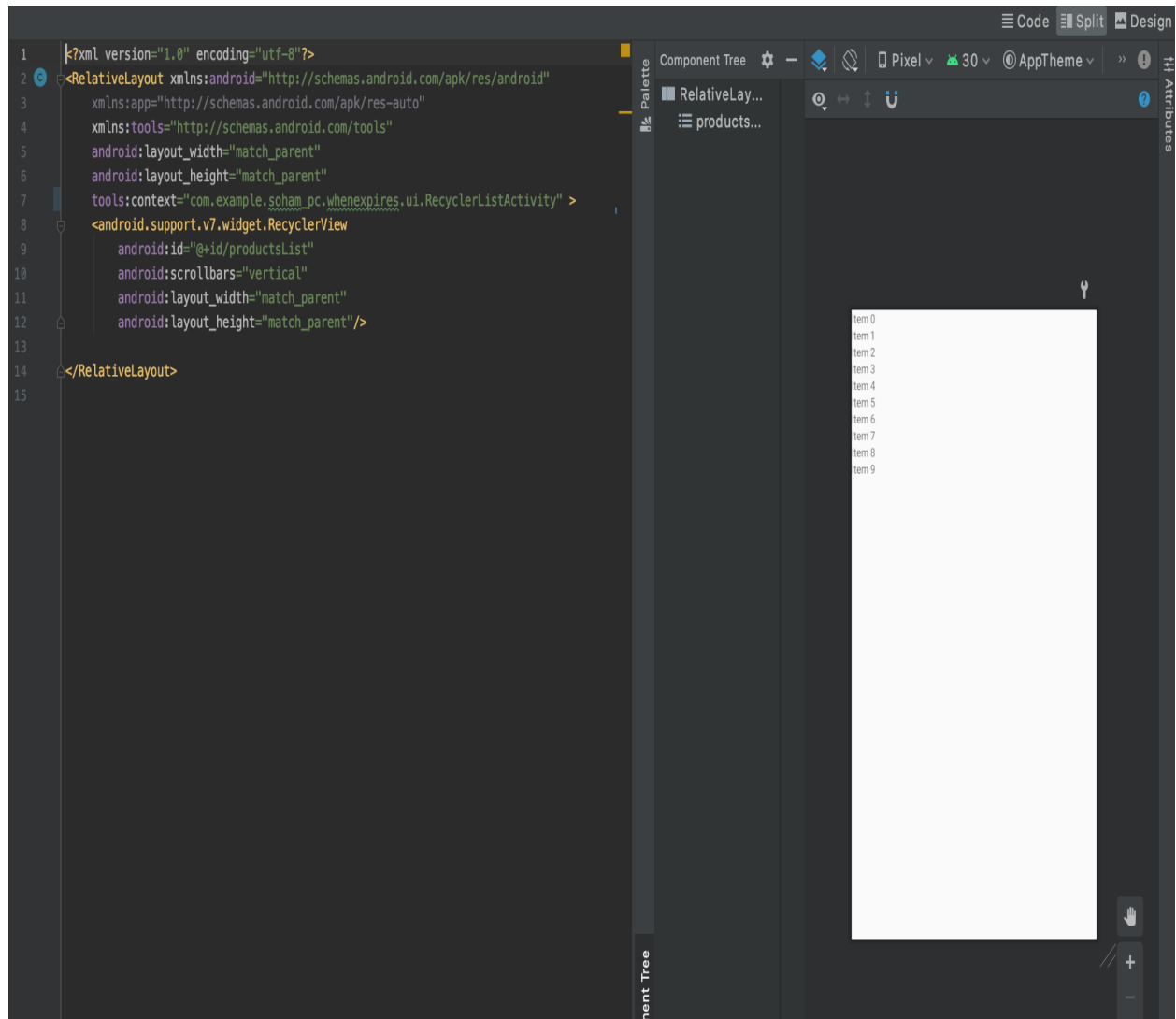
1  package com.example.soham_pc.whenexpires.database.dao;
2
3  import android.arch.lifecycle.LiveData;
4  import android.arch.persistence.room.Dao;
5  import android.arch.persistence.room.Delete;
6  import android.arch.persistence.room.Insert;
7  import android.arch.persistence.room.OnConflictStrategy;
8  import android.arch.persistence.room.Query;
9
10 import com.example.soham_pc.whenexpires.database.entity.ProductEntity;
11
12 import java.util.List;
13
14
15
16 @Dao
17 public interface ProductDao {
18
19     @Query("select * from products_table group by ` product_barcode`" +
20         "")
21     LiveData<List<ProductEntity>> getAllProducts();
22
23     @Insert(onConflict = OnConflictStrategy.REPLACE)
24     long insertProduct(ProductEntity product);
25
26     @Query("SELECT COUNT(*) from products_table")
27     int countProducts();
28
29     @Query("SELECT COUNT( 'product_barcode') from products_table WHERE ` product_barcode`=:barcode")
30     int getProductCount(String barcode);
31
32
33     @Delete
34     int delete(ProductEntity productModel);
35
36 }
37
38

```

```

1 package com.example.soham_pc.whenexpires.ui;
2
3 import ...
4
29
30 public class RecyclerViewActivity extends AppCompatActivity {
31     Context mContext;
32     List<ProductEntity> product_models;
33     private ProductViewModel productViewModel;
34     private RecyclerView mRecyclerView;
35     private ProductsRecyclerViewAdapter mAdapter;
36     private RecyclerView.LayoutManager mLayoutManager;
37
38     @Override
39     protected void onCreate(Bundle savedInstanceState) {
40         super.onCreate(savedInstanceState);
41         setContentView(R.layout.activity_recycler_list);
42         mRecyclerView = findViewById(R.id.productsList);
43         mRecyclerView.setHasFixedSize(true);
44         mContext = this;
45
46         // use a linear layout manager
47         mLayoutManager = new LinearLayoutManager(context: this);
48         mRecyclerView.setLayoutManager(mLayoutManager);
49
50         productViewModel = ViewModelProviders.of(activity: this).get(ProductViewModel.class);
51
52
53         productViewModel.getItems().observe(owner: RecyclerViewActivity.this, (products) -> {
54             if (!products.isEmpty()) {
55                 product_models = products;
56                 mAdapter = new ProductsRecyclerViewAdapter(products, getApplicationContext());
57                 mRecyclerView.addItemDecoration(new SimpleDecorator(getApplicationContext()));
58                 mRecyclerView.setAdapter(mAdapter);
59                 processNotificationDate(product_models);
60             }
61         });
62
63
64
65
66
67
68         mRecyclerView.addOnItemTouchListener(new RecyclerViewTouchListener(getApplicationContext(), mRecyclerView, new RecyclerViewTouchListener.ClickListener() {
69
70
71     @Override

```



Q.2 Location API

Location-aware features in apps can provide much more help than simply detect your current whereabouts. Without geolocation in smartphones and tablets, it would have never been possible to search for a nearby movie theater, restaurant or show, choose an interesting event to attend or a place to visit, surf through a dating or social-networking app to find a new acquaintance in your neighborhood, or receive any real-time services through an app. With geolocation you can request taxi or produce delivery and enjoy a vast number of other non-conventional services like booking a home massage and getting a dog-sitter or your pet.

Adding current location of user in android app.


```

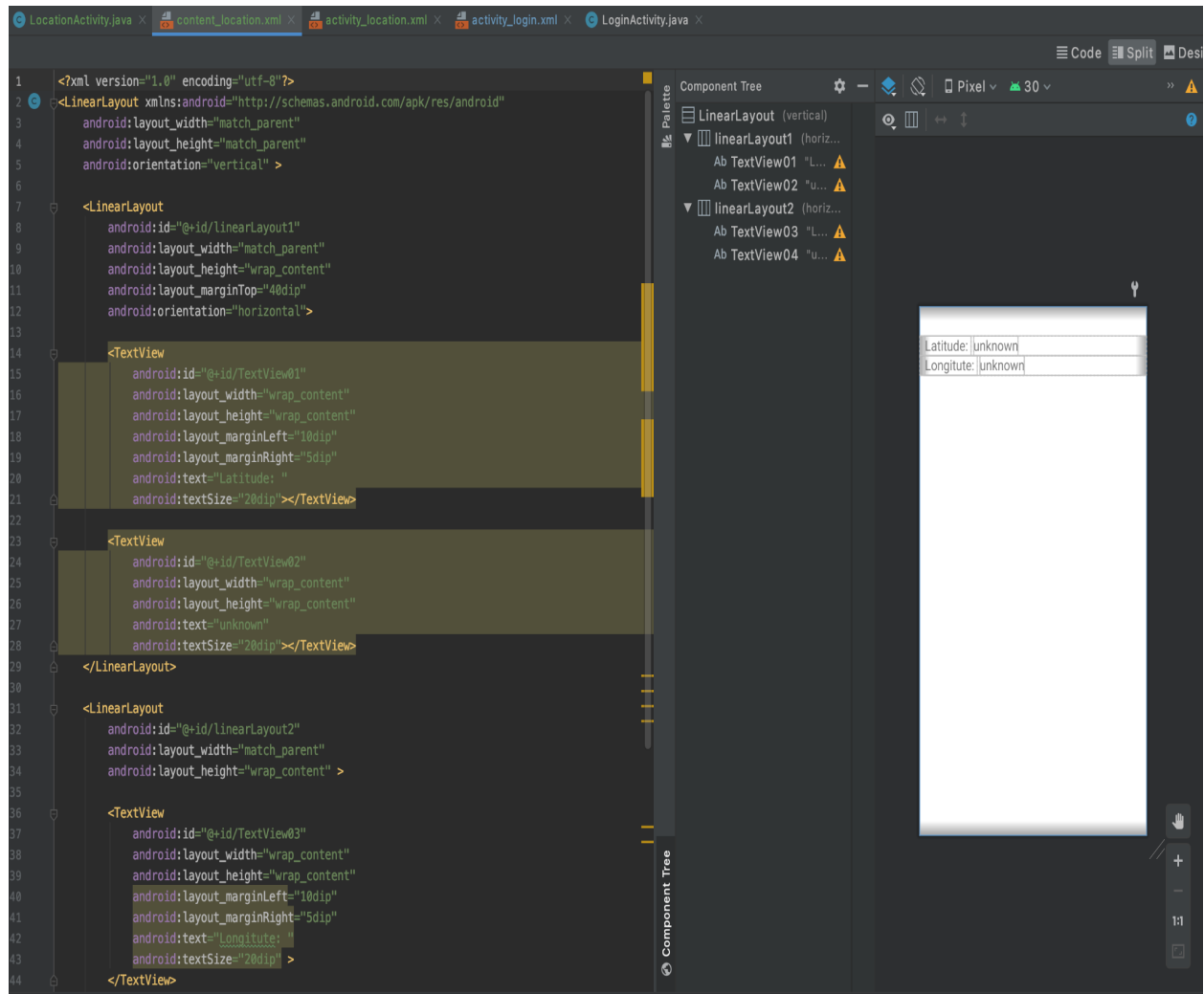
1 package com.soml.whenexpires;
2
3 import ...
4
13 public class LocationActivity extends Activity implements LocationListener {
16     private TextView latitudeField;
17     private TextView longitudeField;
18     private LocationManager locationManager;
19     private String provider;
20
21     /** Called when the activity is first created. */
22     @Override
23     public void onCreate(Bundle savedInstanceState) {
24         super.onCreate(savedInstanceState);
25         setContentView(R.layout.content_location);
26         latitudeField = (TextView) findViewById(R.id.TextView02);
27         longitudeField = (TextView) findViewById(R.id.TextView04);
28         locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
29         Criteria criteria = new Criteria();
30         provider = locationManager.getBestProvider(criteria, enabledOnly: false);
31         if (ActivityCompat.checkSelfPermission( context: this, android.Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED
32             && ActivityCompat.checkSelfPermission( context: this, android.Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
33             return;
34         }
35         Location location = locationManager.getLastKnownLocation(provider);
36
37         if (location != null) {
38             System.out.println("Provider " + provider + " has been selected.");
39             onLocationChanged(location);
40         } else {
41             latitudeField.setText("Location not available");
42             longitudeField.setText("Location not available");
43         }
44     }
45
46     @Override
47     protected void onResume() {
48         super.onResume();
49         if (ActivityCompat.checkSelfPermission( context: this, android.Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED
50             && ActivityCompat.checkSelfPermission( context: this, android.Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
51             return;
52         }
53         locationManager.requestLocationUpdates(provider, minTimeMs: 400, minDistanceM: 1, listener: this);
54     }
55

```

```

43     }
44 }
45
46
47 @Override
48 protected void onResume() {
49     super.onResume();
50     if (ActivityCompat.checkSelfPermission( context: this, android.Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED
51         && ActivityCompat.checkSelfPermission( context: this, android.Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
52         return;
53     }
54     locationManager.requestLocationUpdates(provider, minTimeMs: 400, minDistanceM: 1, listener: this);
55 }
56
57 @Override
58 protected void onPause() {
59     super.onPause();
60     locationManager.removeUpdates( listener: this);
61 }
62
63 @Override
64 public void onLocationChanged(Location location) {
65     int lat = (int) (location.getLatitude());
66     int lng = (int) (location.getLongitude());
67     latitudeField.setText(String.valueOf(location.getLatitude()));
68     longitudeField.setText(String.valueOf(location.getLongitude()));
69 }
70
71 @Override
72 public void onStatusChanged(String provider, int status, Bundle extras) {
73     // TODO Auto-generated method stub
74 }
75
76 @Override
77 public void onProviderEnabled(String provider) {
78     Toast.makeText( context: this, text: "Enabled new provider " + provider,
79         Toast.LENGTH_SHORT).show();
80 }
81
82 @Override
83 public void onProviderDisabled(String provider) {
84     Toast.makeText( context: this, text: "Disabled provider " + provider,
85         Toast.LENGTH_SHORT).show();
86 }
87 }

```



2:59

VoLTE+ LTE1 78%

Latitude: 19.06075228256053

Longitude: 72.88641240692213



Q.3 short notes

(1) Android security issue.

A- The security flaws allows hackers to exfiltrate media files such as photos, videos and call-recordings. Additionally, criminals could access real-time data, such as mic, GPS and location data. The attackers could also freeze the affected Android phone, rendering them unresponsive. The attackers could also freeze the affected Android phone rendering them unresponsive. Location data.

(2) Android App publishing

A- Publishing is the general process that makes your Android application available to users. You can publish an Android application you perform two main tasks.

- You prepare the application for release during the preparation step you build a release version of your application, which users can download and install on their Android-powered devices.

- You release the application to users. If you do not want to release your app on a marketplace like Google Play, you can make app available for downloading on your website or server. Then all you need to do is host the release-ready APK file on your website and provide a download link to users.