# Model paper - 2

I Answer the following questions:-

1. Name two key mobile application services in Android

i) Foreground Services:-

* services that natify the user about its ongoing oper -ations are termed as foreground services.
* users can interact with the service by the notificati -ons provided about the ongoing task

ii) Background services:-

* Background services do not require any user interven -tion.
* These services do not notify the user about ongoing background tasks & users also cannot access them

2. What is the purpose of an Intent object in Android?

→ An intent is a messaging object used to request any action from another app component.

→ Intents facilitate communication b/w different compon -ents in several ways.

→ The intent is used to launch an activity, start the services, broadcast receivers, display a web page, dial a phone call, send messages from one activity to another activity.

3. List two tools required for Android development.

i) Kotlin & Android studio:-

* This is the official integrated development environment for Android development, provided by Google
* Android studio offers a comprehensive suite of features including code editing, debugging and testing tools

ii) Android SDK:-

* The Android SDK is a collection of tools, libraries and resources that developers use to build, test and debug Android applications
* It includes the Android platform libraries, system images, emulators and various other components necessary for Android development.

4. what is an Activity in Android ?

→ An activity represents a single screen with a user interface,

→ If an application has more than one activity, then one of them should be marked as the activity that is presented when the application is launched

→ Eg:- An email application might have one activity that shows a list of new emails, another activity to compose an email, and another activity for reading emails.

→ An activity is implemented as a subclass of Activity class as follows-

```
public class MainActivity extends Activity {
}
```

5. Mention two types of views used in Android user interfaces.

i) Text view:-

* This is used to display text to the user
* It can be styled with different fonts, colors and sizes

n) Image view:-

* This is used to display images in the UI
* It can load images from various sources and apply transformations or effects.

6. what is SQLite used for in Android?

→ In Android, SQLite is used as a lightweight, embedded database engine for storing and managing structured data

→ It provides a relational database management system within the application, allowing developers to perform sql queries

→ By integrating SQLite, Android apps can efficiently store, query and manage data locally without needing a separate database server.

II Answer the following questions:-

07. Describe the process of adapting an application to display changes.

i) understanding the change requirements :-

* Gather requirements :- Identify what needs to change
* Analyze impact :- Determine how these changes will affect the application, including the user interface, backend and any integrations

ii) Design the update :-

* update the UI/UX Design :- create design mockup that reflect the new changes
* Define Data flow :- plan how the data will flow thro -ugh the application to support the changes, inclu -ding any modifications to the data model

iii) Implement the changes :-

* update Backend code :- Modify the server-side logic to support the new functionality

* update frontend code:- change the client-side code to reflect the new design and functionality

iv) Ensure Real-time updates:-

* State Management:- Implement state management strategies to manage the application state and ensure UI components reflect changes promptly

* Data Binding:- use data binding techniques to automatically update the UI when the underlying data changes

v) optimize performance:-

* Minimize Re-renders:- optimize the components to avoid necessary re-renders, which can slow down the application

* Lazy loading:- implement lazy loading for resources and components to improve initial load times and reduce bandwidth usage

vi) Testing:-

* unit testing:- test individual components and functions to ensure they work as expected

* user Acceptance testing :- conduct testing with actual users to gather feedback and identify any issues not caught during internal testing

vii) Deployment :-

* Staging Environment :- Deploy the changes to staging environment to verify them in a production-like setting
* production Release :- once tested, deploy the changes to the live environment, ensuring minimal downtime

8. Explain how to use the intent object to invoke built-in applications.

→ The Intent object is a fundamental tool used to request actions from other applications within the same application

→ Steps to use the intent object are :-

i) create an Intent object :-

* Instantiate an Intent with the action you want to perform

* Android provides several predefined actions for common tasks like viewing a webpage, taking a photo or sending an email.

ii) Set Additional Data:-

* Some actions require additional data, like a URI or specific extras, to perform the intended task
* This could include URIs for a browser, phone numbers for the dialer, or email addresses for the email client

iii) Invoke the intent:-

* use the startActivity method to launch the activity associated with the Intent.
* For some actions, you may need to handle the result, in which case you use startActivityForResult

iv) Handle the Result:-

* If you need a result back from the activity, override the onActivityResult method to process the returned data.

9. How to create a listView to display long lists in an Android application?

→ ListView is default scrollable so we do not need to use scroll view or anything else with listView.

→ There are two types of listView
i) List view and Spinner view

→ A very common example of listView is your phone contact book, where you have a list of your contacts displayed in a ListView and if you click on it then user information is displayed

→ The list view is a type a AdapterView

→ In Android, Adapter view is an abstract class that acts as a parent for several view classes like listView, GridView and spinner

→ For displaying the items in the list method setAdaptor() is used

→ The main purpose of adapter is to fetch data from an array or database and insert each item that placed into the list of the desired result.

→ To display a list, you can include a list view in your layout XML file:

```
<ListView
    android:id="@+id/list_view"
    android:layout_width="match_parent"
    android:layout_height="match_parent"/>
```

10. Discuss the process of sending an SMS message in Android.

i) User Interaction

* The process typically starts with user interaction, where the user inputs the recipient's phone number and the message content into a user interface provided by an app

ii) permissions

* To send an SMS, the app must have the appropriate permissions. In Android, this requires the app to declare the SEND_SMS permission in its AndroidManifest.xml file

* `<uses-permission android:name="android.permission.SEND_SMS"/>`

iii) creating the SMS Message

* once the permission is granted, the app can create an SMS message using the SmsManager class, which provides the method to send text messages

* SmsManager SmsManager = SmsManager.getDefault();

iv) SMS Delivery:-

* The SmsManager forwards the message to the Android system's SMS service. this service interfaces with the cellular radio layer to send the SMS over the mobile network.

v) Satus feedback:-

* Sent status:- The pendingIntent specified for sent messages can be used to track the sending status

* Delivery status:- The pendingIntent specified for delivery can be used to track whether the message was delivered to the receipient.

vi) Receiving SMS messages:-

* Receiving SMS messages is handled by the system's default SMS app, but apps can register to receive sms messages by declaring a BroadcastReceiver for the SMS_RECEIVED action

11. outline the steps to create a custom content provi-der in Android

i) Define your Data structure

* Determine the type of data your content provider will manage and its structure.

ii) create the content provider class

* Extend the ContentProvider class & implement the required methods:-

* onCreate(), query(), insert(), update(), delete(), getType()

iii) Define the URI structure

* This involves specifying the base URI and the path patter-ns for different types of data requests

iv) create a contract class

* Define a contract class that specifies the structure of the data, including table names, column names and content URIs

v) Implement Database Management:-

* If your content provider uses an SQLite database, create a SQLiteOpenHelper class to manage database creation and version management.

vi) implement the CRUD operations

* In your contentprovider class, implement the methods to perform the CRUD operations using the database helper

vii) Register the content provider in the manifest:-

* <provider
android:name = ".MyContentProvider"
android:authorities = "com.example.provider"
android:exported = "true"/>

viii) handle permissions

* Define the permissions for accessing your content provider if needed. you can specify readpermission and writeperm-ission attributes in the manifest declaration to control access

ix) Access the content provider

* clients can access the content provider using ContentResolver and URIs defined in the contract class

* Eg:- ContentResolver resolver = getContentResolver();
Cursor cursor = resolver.query(MyContentContract.Items.CONTENT_URI, null, null, null, null);

x) Testing:-

* Test your content provider thoroughly to ensure it correctly handles data operations and respects permissions and URI patterns.

12. what are the key considerations for deploying APK files to the Google play store?

i) App preparation:-

* App signing: Sign your app with a valid certificate using Google play App signing or your own Keystore

* Target API level:- Ensure your app targets a recent API level as required by Google.

* permissions:- Request only the permissions necessary for your app's functionality
* Format:- prefer Android App bundle over Apk for more efficient distribution

2. Compliance with policies

* Content and privacy:- Adhere to Google play's content policies and include a privacy policy if your app handles user data
* Data Safety:- complete the data safety section detailing data handling practices
* Family policy:- Ensure compliance if your app is designed for children, adhering to the families policy
* Billing compliance:- use Google play's billing system for any in-app purchases

3. Technical Requirements :-

* App size :- Optimize the size of your APK ; use expans -ion files if the APK exceeds 100MB

* Testing :- Testing your app on multiple devices and Android versions

* Localization :- Translate your app and store listing for different languages if targeting multiple regions

* Security :- use tools like proguard or R8 to obfus -cate your code and protect against reverse engineering

4. Google play console setup

* Developer Account :- set up and verify your Google play Developer account

* App listing :- prepare your store listing, including title, description, screenshots and feature graphic

* Distribution settings:- set your app's pricing and select the countries where it will be available
* content Rating:- complete the content rating questionnaire to get a rating for your app

5. Release Management:-

* versioning:- Increment the version code and version name for each new release
* Release track:- choose the appropriate track for your release
* Staged Rollout:- consider releasing your app gradually to a percentage of users to monitor for issues
* Release Nates:- provide detailed release notes to inform users about new features and updates.

III Answer the following question :-

13. Explain how to use fragments in Android and provide an example of their usage

→ Fragments are modular sections of an activity's user interface that enable a more flexible and reusable approach to UI design in Android

→ They allow for better management of UI components, especially on larger screens like tablets, where you might want to show multiple fragments side-by-side

→ steps to use fragments

1. create a fragment class :-

* Extend fragment and override necessary lifecycle methods

* Implement onCreateView() to define the fragments UI

2. Add the fragment to an Activity:

* use xml layout files or programatically add the fragment using the fragmentManager

3. Handle Fragment Transactions:-

* use fragmentManager and fragmentTransaction to add, replace or remove fragments dynamically

14. Discuss the steps involved in persisting data to files in an Android application.

1. choose the Appropriate storage type

* Internal storage:- private storage within the app's directory, accessible only by the app

* External storage:- public or app-specific directories on external storage, accessible by other apps if permissions are granted

2. Request Necessary permissions

* Internal storage :- No special permissions are required
* External storage :- Requires permissions such as READ-EXTERNAL-STORAGE AND WRITE-EXTERNAL-STORAGE

3. Determine the file path

* Internal storage :- use getFilesDir() or getCacheDir()
* External storage :- getExternalFilesDir() for app-specific files or Environment.getExternalStoragePublicDirectory() for public files

4. Write Data to the file

* use java's fileOutputStream or BufferedWriter to write data to the file

5. Read Data from the file

* use FileInputStream or BufferedReader to read data from the file