1. Android App Components

1.1 four types of app components - purpose

1.1.1 Activities

Activities represent a screen on the app, which typically have some kind of **interface** with which the user can **perform** some **actions**. Thus, they are all about handling the interaction with the user (buttons, inputs) as well as displaying Views (e.g. TextView,..) on the screen.

1.1.2 Services

Services are used to run a **task** in the **background** of the app. Since Services run in the background the app doesn't have to be in the foreground, thus can be minimized in the tray, while the services still performs its task. Services are usually used for playing music, downloading content, or handling data (e.g., syncing) in the background.

1.1.3 Broadcast receivers

Broadcast receivers are used to handle or interact with **'global' events**, thus events, that don't necessarily relate to the app, such as "battery low", "phone ringing" or more. That means, that those events are generally system-wide events. Since the Broadcast receiver handles those events, displaying notifications, starting a service or other actions can be performed when such an event occurs.

1.1.4 Content providers

As the name suggests this component is used for handling data. Specifically, it handles the **shared data** between multiple apps, such as contacts, calendar events, media files or more.

1.2 App unique entry point or multiple? Deferrization on desktop apps?

1.2.1 App entry points

An android app can have multiple entry points which is commonly used. The "main" entry point could be triggered when the app icon is opened and other entry points like notifications, widgets or when web link references directly interact (open something) inside the app.

1.2.2 Desktop App entry points

A desktop application can also have multiple entry points, just like an android app. The "main" entry point could be the same, when opening the app using its icon. Another entry point could be through interacting with the application using its tray icon or through file associations on the pc, like opening a .json files can be automatically opened by visual studio code.

1.3 Explicit & Implicit Intents

Generally speaking intents are used to interact with other components. Hence, they can be used to transfer data between Activities, Services and so on.

1.3.1 Explicit Intents

This type of intent is used to start a **specific component by its name**, hence it is used when its clear which component is to be executed.

1.3.2 Implicit Intents

On the other hand an implicit intent is used when the component that is to be started is **not known**. Therefore, not the name, but the **action** of the component is **provided**, thus the system will resolve the intent and starts the corresponding component based on the given action, it handles.

1.4 Application Context - role of R

The application context, as its name suggests, relates to the app itself, thus provides access to its resources, preferences, system services and other information's of the application. Important to know is, that this context is created when ever the app is launched and removed when the app terminates.

The "R" class is used to access the resources programmatically using the application context, such as the layouts, strings, images, styles.

2. Hello World App

2.4 Investigation of AndroidManifest.xml

The AndroidManifest holds the apps permissions and components "information's", which are required by the Android system in order to execute the app.

Example manifest:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.discordbot">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".ListActivity">
            <intent-filter>
                <action android:name="android.intent.action.DEFAULT" />
            </intent-filter>
        </activity>
    </application>

    <uses-permission android:name="android.permission.INTERNET"></uses-
permission>

</manifest>
```

2.4.1 Package

This property is a unique identifier that is used to differentiate the app between other apps on the system, it usually contains the domain name of the company in which you develop the app.

2.4.2 Permissions

The permissions property is quite important, since some functionalities don't work by default, such as receiving / transmitting data over an TCP socket, when the *Internet permission* is not added to the application. Thus, allowing the app to access the internet.

2.4.3 Application property

This property describes the application itself, by declaring the icon, theme and so on. It also includes information's about its components. Like its Activities, Services, and so on. In order access a component using intents, the component has to be declared in this manifest first.

3. Logging and Debugging

3.1 Purposes, Pros/Cons of logs

3.2 Requirements for release variant

4. Android Lifecycle

4.3 State changes - circumstances