Date :

Page No.
Roll No.

Expeeiment No: 2

Title:
Use of fragments.

Problem Statement:
Design an app to demonstrate reuse of visible space using fragments.

Theory:

• Introduction to Android SDK
- It is a s/w develop kit & it is a collection of s/w development tool in one installable package.
- It helps to download new tools & latest version of android. Every time google releases a new version the corresponding SDK is also used.
- Developers must download each version of SDK the android SDK is the set of development tools that are used by developers to develop application for android platform.

Features of android SDK:

1) ~~offline mapping~~: Dynamic markers:
- In previous versions of SDK you could not have map position without a callback but in

new SDK making is Dynamic

2) offline mapping:
- SDK helps in dynamically downloading the map for different Contries in more than 60 languages & you can view this map offline.

3) Improvised API Compatibility:
- It is easy to migrate from the google map to the android API.

Android SDK Tools:
- It is a Component includes Complete Set of development & debugging tools of android appln development.
- They are also provided with android Studio.

Android SDK manager:
- Android SDK manager is used to manage different tools & different versions of SDK we can add or remove tools & SDK according to our requirement.

SDK platform:
- Each android SDK platform includes the Software development kit ones installed the android Studio will automatically Check for updates.

Date :

Page No.
Roll No.

• Android Emulator:

- The android Emulator mimics all the hardware & Software features of typical mobile device but we cannot make phone call by using this Emulator.
- It provides variety of nevigation & different Control keys which you can press by using mouse & keyboard to generate event for your mobile application.
- It also provides a Screen in which your application displayed together with any other active android application
- It is only used for development purpose developer Can Create & Run his application into the emulator it is only for testing Purpose.

Creating AVD
If you want to emulate a Real device, first Create an AVD with the Same device Configuration as Real device, that launch this AVD from AVD manager.

Changing orientation:
Usually by default when you launch the emulator its orientation is vertical, but you can change

it orientation by pressing ctrl+f11 key from keyboard

Emulator Commands:

1. Home:
   Shifts to main Screen.

2. F2:
   Toggles Context Sensitive menu.

3. F3:
   Bring out call log

4. F4
   End call

5. F5.
   Search

6. F6
   Toggle trackball mode

7. F7
   power button

8. F8.
   Toggle data network.

Date :

Page No.
Roll No.

Creating Project:

- Android Studio makes it easy to Create Android apps for various form factors, Such as phones, tablets, Tv,s and wear devices.
- If you don't have a project opened, Create a new project by Clicking Start a new Android Studio project on the Android Studio welcome Screen.
- If you do have a project opened, Create anew project by Selecting File > New > New project from the main menu.

Choose your project type:

- In the New project Screen that appears, you Can Select the type of project you want to Create from Categories of device from factors, Shown in the Templates pane.
- Selecting the type of project you want to Create lets Android Studio include Sample Code and resources in your project to help you get Started.
- once you Select your project type, Click Next.

Configure your project:

- The next Step in Creating your project is to Configure Some Settings, If you Creating a Native

C++ project, Read Create a new project with C/C++ Support to learn more about options you need to Configure.

1. Specify the Name of your project
2. Specify the package name.
3. Specify the the Save location where you want to locally Store your project.
4. Select the language, kotlin or java, you want Android Studio to use when Creating Sample Code of your new project
5. Select the Minimum API level you want your app to Support.
6. your project is Configured to use Androidx libraries by default, which replace the Android Support libraries. To use legacy Support libraries insted, Select Use legacy android. Support libraries.
7. When you're redy to Create your project, Click finish.

- Import an existing project:-

1) Click file > New > Import project.
2) In the window that appears, nevigate to the root directory of the project you want to import.
3) Click ok.

Date :

Page No.
Roll No.

- Project Directory Structure:

▼ ▭ app
  ▼ manifests
    AndRoid Manifest. xml
  ▼ java
    ▶ Com. example. tutorial point 7. my application
    ▶ Com. example. tutorial point 7. my application (androidTest)
    ▶ Com. example. tutorial point 7. my application (test)
  ▼ res
    ▭ drawable
  ▼ layout
    activity - main. xml
  ▼ mipmap
    ▶ ic. launcher. png
  ▼ Values
    Colors. xml
    ▶ dimens. xml
    Strings. xml
    Styles. xml
▼ ⊙ Gradle Scripts
  build. gradles (project. my Application)
  build. gradle (Moudle . app)
  proguard. rules (proGuard Rules for app)
  gradle. properties (Project properties)
  Setting . gradle (project Settings)
  local. properties (SDk Location)

• DDMS (Dalvik Debug Monitor Service) :-

- The Dalvik Debug Monitor Service (DDMS) is a debugging tool used in the Android platform. The Dalvik Debug monitor Service is downloaded as a part of Android SDK. Some of the Services provided by the DDMS are port forwarding, on device Screen capture, on device thread & heap monitoring, & radio State information.
- The Dalvik Debug monitor Service allows developers to Spot bug in applications running on either an emulator or an actual Android device.
  For example, by using the DDMS' Logcat feature, developers can view log messages regarding the State of the application and the device. Log cat can pinpoint the exact line number on which an error occured.

• Logging in Android (Logcat):

- Logcat is a Command-line tool that dumps a log of System messages including messages that you have written from your app with the log class.

Logging System overview:
- The Android loging System is a Set of Structured Circular buffers maintained by the System process

Date :

Page No.
Roll No.

logd. The set of available buffers is fixed and defined by the System. The most relevant buffers are:

- main: Stores most Application logs.
- System: Stores messages orginating from the Android OS.
- Crash: Stores crash logs. Each log entry has a priority, a tag that identifies the origin of the log, and the actual log message.

Logs displayed by adb logcat undergo four levels of filtering:

Compile time filtering:
- Depending on Compilation Settings, Some logs may be Completely removed from the binary. for example, ProGuard can be Configured to remove calls to Log.d from java Code.

System property filtering:
liblog queries a set of System properties to determine the minimum Severity level to be sent to logd. If your logs have the tag myApp, the following properties are checked and are expected to contain the first letter of the minimum Severity (V, D, I, W, or S to disable all logs).
- log. tag. MyApp

- peesist . log. tag. myApp.
- log. tag
- peesist. log. tag.

Application filteeing:

If none of the propeeties aee set, liblog uses the minimum peopeety set by
--andeoid-- log set- minimum-peioeity. The default setting is INFO

Display filteeing:

adb logcat suppoets additional filtees that can eeduce the amount of logs shown feom logd. See the

Command-line Syntax:

To eun logcat theough the adb shell, the geneeal usage is

```
[adb] shell logcat [<option>] ----[<filtee-spec>]...
```

Date :

Page No.
Roll No.

· Andeoid Manifest file:-

- Manifest file is Andeoid manifest.xml file
- We must declaee all the Components used in application in this andeoid manifest file.
- This file works as an inteefaie between andeoid OS & application.

eg:

```
  <? xml veesion="1.0" encoding =" utf-8 "?>
 <manifest xmlns : andeoid="http:// Schemas.
    andeoid . com / apk / ees/ andeoid ">
 < application
    andeoid: allowbackup ="teue"
   andeoid : icon = "@ deawable / ic_launchee "
   andeoid: label="@ Steing / app- name"
   andeoid: theme="@ Style / AppTheme">
 < activity
    andeoid: name=" MainActivity">
    andeoid: label =" @ Steing / App-name">
 <intent - filtee
 < action andeoid: name = andeoid. intent. action. MAIN>
 < Categoey andeoid: name= andeoid. intent. Categoey
                                        LAUNCHER"/>
 </intent - filtee>
 </ activity>
 </ application>
 </ manifest>
```

• permission:

- The purpose of permission is to protect the privacy of user. Android app must request for permission to access sensitive information such as contact, SMS as well as some system features such as camera & audio system.
- Depending on the feature the system can grant the permission automatically or the system can promt the user to approve the request.
- we can implement permission in your android app using <user- permission> tag in mainifest file. for eg:-
  if app have give permission of sending sms then odd following tag into manifest file.

```
<user- permission:name = "android. permission.
                          SEND - SMS "/>
```