

Experiment - 1

Aim:- Installing and Running Applications on Android studio.

Step-1: System Requirements

The required tools to develop Android Applications are open source and can be downloaded from the web. Following is the list of software's you will need before you start your Android application programming.

Java JDK 5 or later version

Java Runtime Environment (JRE) 6

Android studio

Step-2: Set up Android studio

Android studio is the official IDE for android application development. It works based on IntelliJ IDEA, You can download the latest version of android studio from Android studio 2.2 Download . If you are new to installing Android studio on windows , you will find a file , which is named as android-studio-bundle-143.310438-windows.exe . So just download and run on windows machine according to android studio wizard guideline.

If you are installing Android studio on Mac or Linux , You can download the latest version from Android studio Mac Download , or Android studio Linux Download , check the instructions provided along with the downloaded file for Mac os and linux . This Tutorial will consider that you are going to set up your environment on Windows Machine having windows 8.1 operating system.

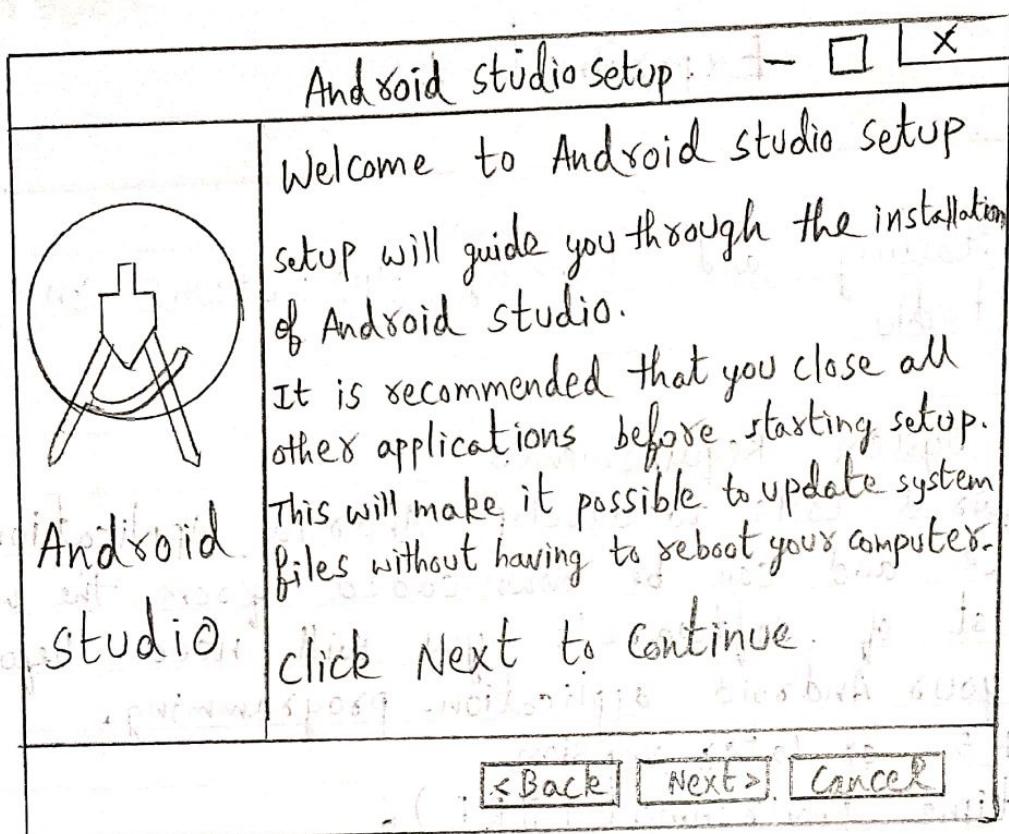


Figure -1

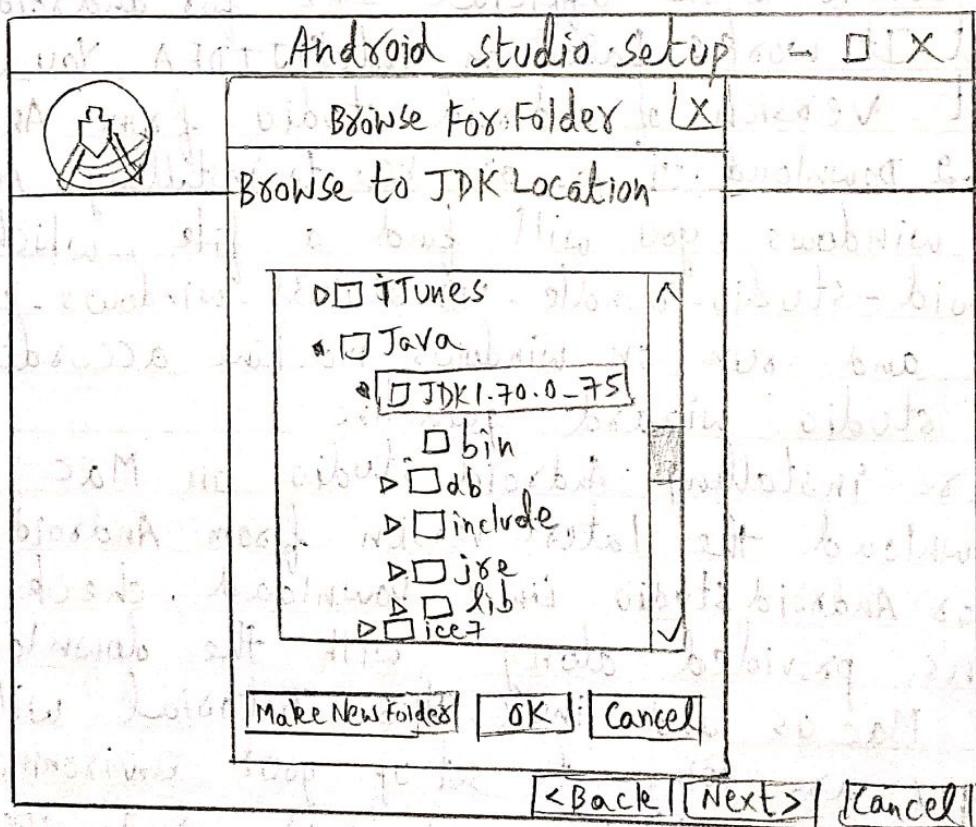


Figure -2

Installation

So let's launch Android studio.exe, Make sure before launch Android studio, our Machine should required installed Java JDK. To install Java JDK. To install Java JDK, take a references of Android environment setup.

Below the initiating JDK to android SDK.

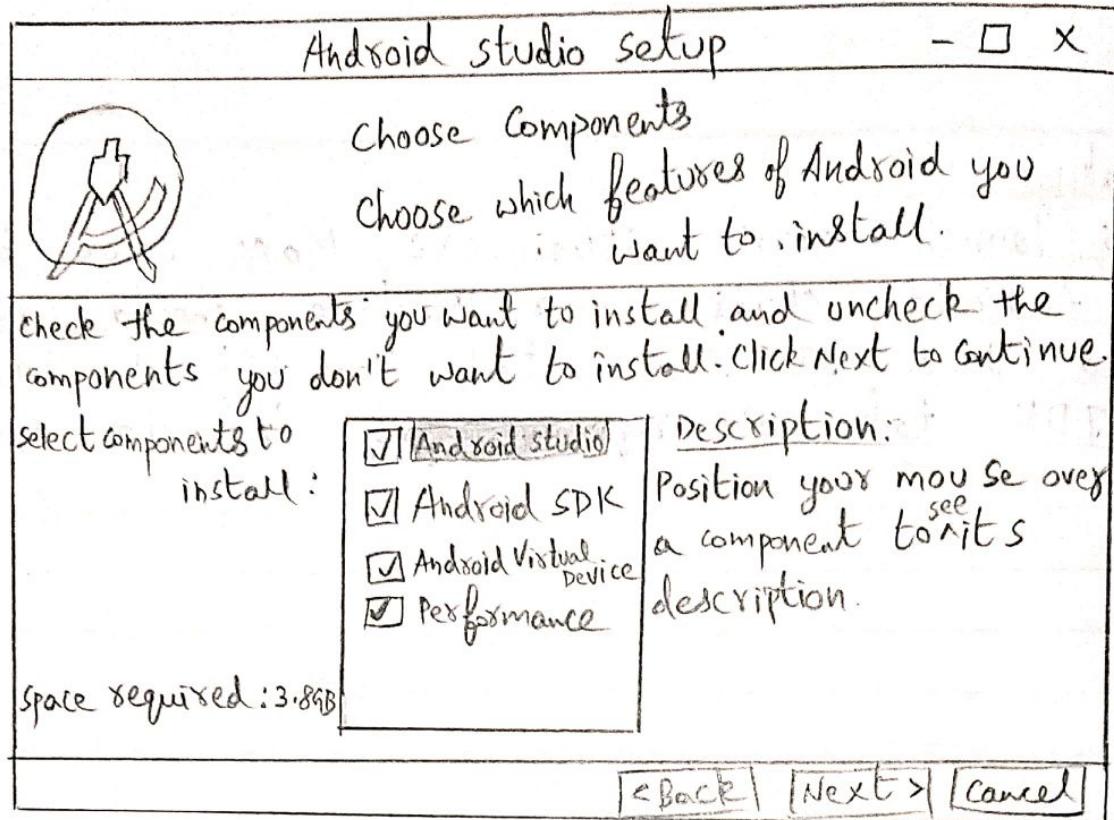


Figure -3

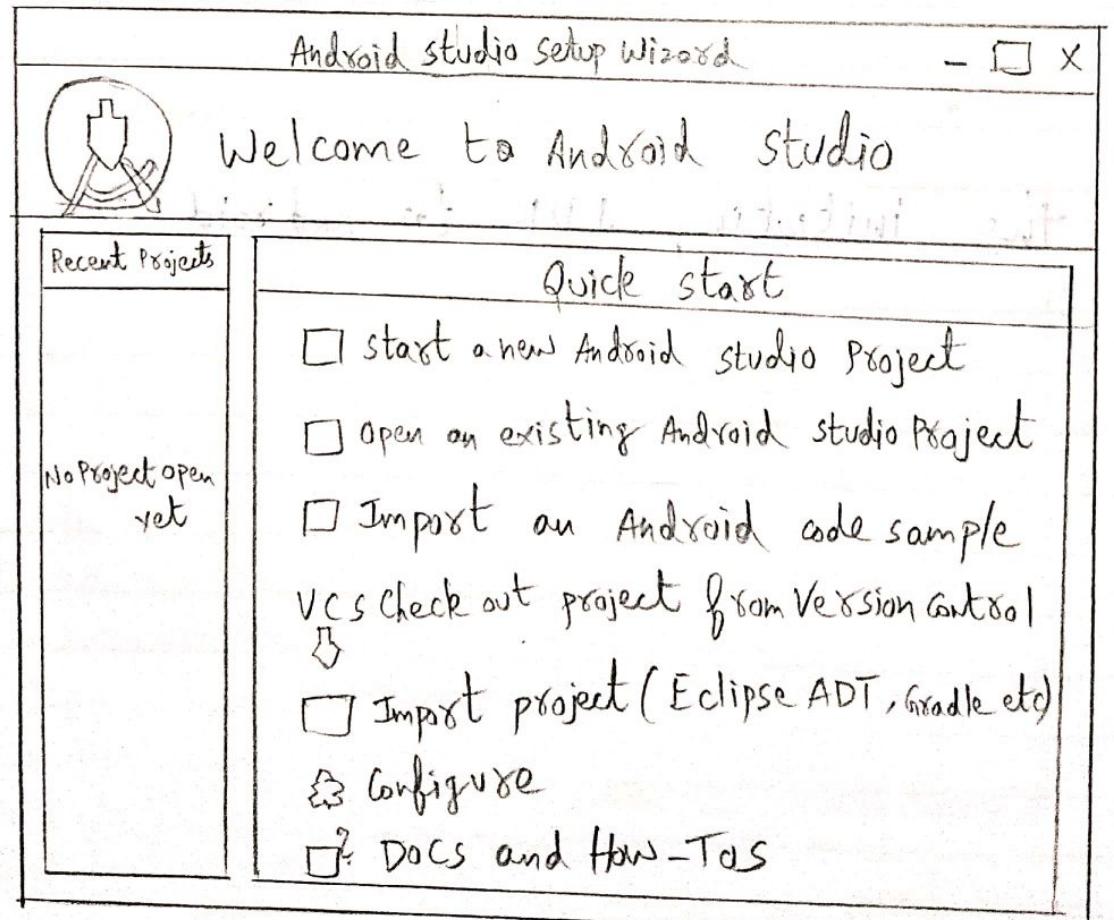


Figure -4

e:

Page No. : 3

Need to check the components, which are required to create applications, below the image has selected Android studio, Android SDK, Android Virtual Machine and performance (Intel chip)

After done all above steps perfectly, you must get finish button and it gonna be open android studio project with Welcome to android studio message as shown below.

Create New Project X

 New Project
Android Studio

Configure your new project

Application name:

Company Domain:

Package name:

Project Location:

Please enter an application name (shown in launcher)

Figure - 5

Create New Project X

 Target Android Devices

Select the form factors your app will run on

Different platforms may require separate SDKs.

Phone and Tablet Wear TV Android Auto Glass

Phone and Tablet Minimum SDK

Wear Minimum SDK

TV Minimum SDK

Android Auto

Glass Minimum SDK

Figure - 6

You can start your application development by calling start a new android studio project. In a new installation form should ask Application name, package information and location of the project.

After entered application name, it going to be called select the form factors your application runs on, here need to specify Minimum SDK, In our tutorial, I have declared as API23: Android 6.0 (Marshmallow).

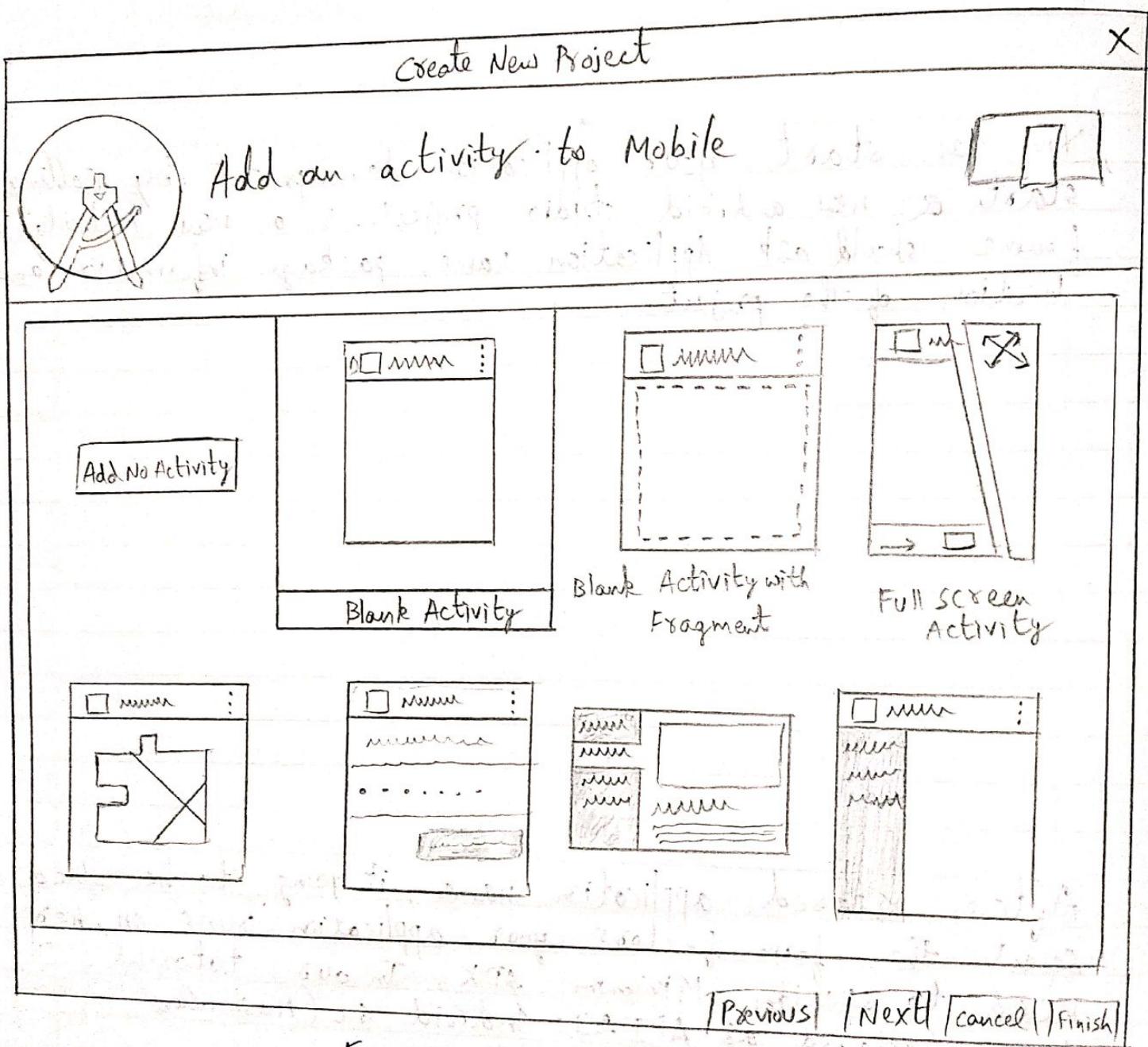


Figure : 7

Date :

Page No. :

3

The next level of installation should contain selecting the activity to mobile, it specifies the default layout for Applications.

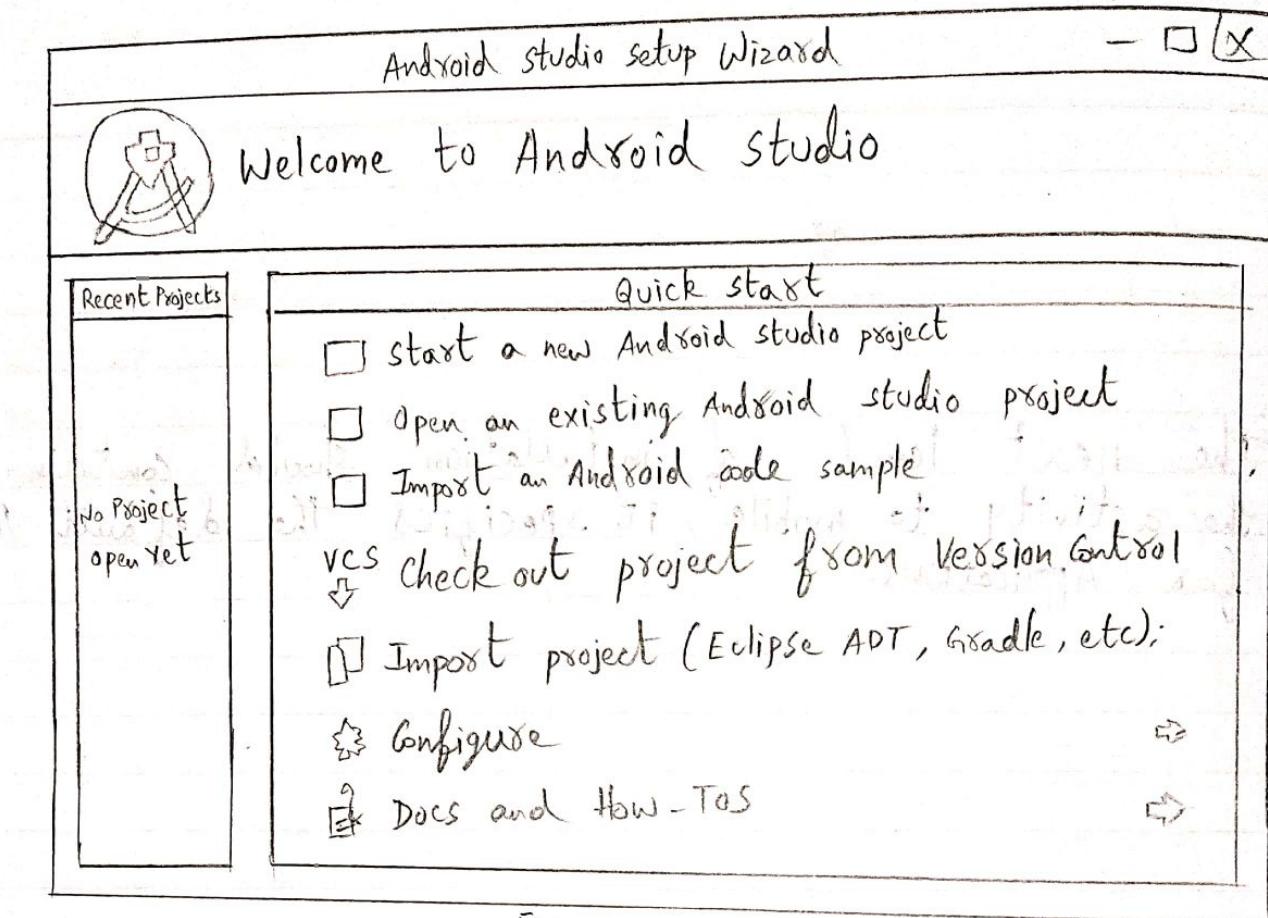


Fig:1

2)

The screenshot shows the 'Create New Project' configuration screen. It has a sidebar on the left with 'configure your project' and 'Empty Activity' sections, and a main configuration area on the right.

Name: Hello World Application

Package name: com.example.helloworldapplication

Save Location: Desktop/MainsourceCode_360VideoPlayer/HelloworldApplication

Language: java

Empty Activity: Minimum API Level [API 15: Android 4.0.3]

Create a new activity:

① Your app will run on approximately 100% of devices
Help me choose

This project will support instant apps
 Use androidx artifacts

At the bottom are buttons for Previous, Next, Cancel, and Finish.

Fig:2

Date :

Experiment - 2

Page No. : 6

Development of Hello World Application

Aim:- To Design an android application to display Hello World.

1) First step is to create a simple Android Application using Android studio. When you click on Android studio icon, it will show screen as shown below.

2) Configure the Hello World Project Details
We'll finish creating the project by configuring some details about its name, location, and the API version it.

create New Project

X



Target Android Devices

Select the form factors your app will run on

Different platforms may require separate SDKs.

Phone and Tablet

Minimum SDK: API 23: Android 6.0

Help me choose

Wear

Minimum SDK API 21: Android 5.0 (Lollipop)

TV

Minimum SDK API 21: Android 5.0 (Lollipop)

Android Auto

Glass

Minimum SDK Glass Development Kit Preview (API 19)

[Previous]

[Next]

[Cancel] [Finish]

Fig: 3

Date :

3) Change the name of the application. Change the default Project Location to your preferred directory or just leave it as the default location.

On the minimum API Level ensure that API 15: Android 4.0.3 Ice Cream Sandwich is set as the Minimum SDK. This ensures that your application runs on almost all devices.

SOURCE CODE:-

The Main Activity File

```
package com.example.helloworldapplication;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
public class MainActivity extends AppCompatActivity;
```

```
@Override
protected void onCreate(Bundle savedInstanceState)
```

```
{  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
}
```

```
}
```

```
}
```

The Layout File

```
<RelativeLayout xmlns:android = "http://schemas.android.com/apk/res/android"  
    xmlns:tools = "http://schemas.android.com/tools"  
    android:layout_width = "match-parent"  
    android:layout_height = "match-parent" >
```

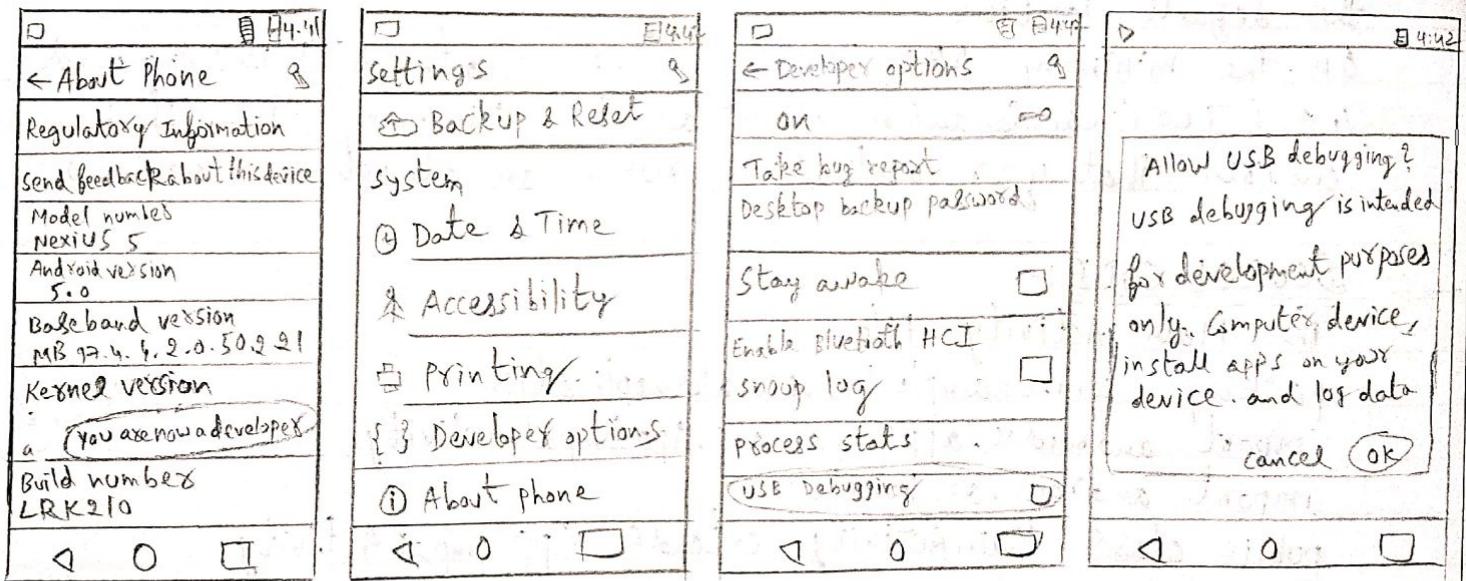


Fig: 4

<Text View

```
    android:layout_width = "wrap_content"  
    android:layout_weight = "wrap_content"  
    android:layout_centerHorizontal = "true"  
    android:layout_centerVertical = "true"  
    android:text = "@string/hello_world"  
    tools:context = ".MainActivity" />
```

i) Running app on Phone:

Connect your phone to computer

Plug in your device to your computer with a USB cable. If you're developing on Windows, you might need to install this universal ADB USB driver or find your specific USB driver from your device.

The next step is to enable USB debugging so your phone can interact with your computer in a developer mode.

The following steps are needed:

- 1) (Windows only) Install the ADB Driver.
- 2) Plug-in your Android Device to Computer via USB.
- 3) Open the "settings" App on the Device.
- 4) Scroll Down to Bottom to find "About Phone" item.
- 5) Scroll Down to Bottom to find "Build number" section.
- 6) Tap on "Build number" 7 times in quick succession.
- 7) You should see the message "you are now a developer!"
- 8) Go back to main "settings" page.
- 9) Scroll down bottom to find "Developer options" item.
- 10) Turn on "USB Debugging" switch and hit "OK".
- 11) Unplug and re-plug the device.

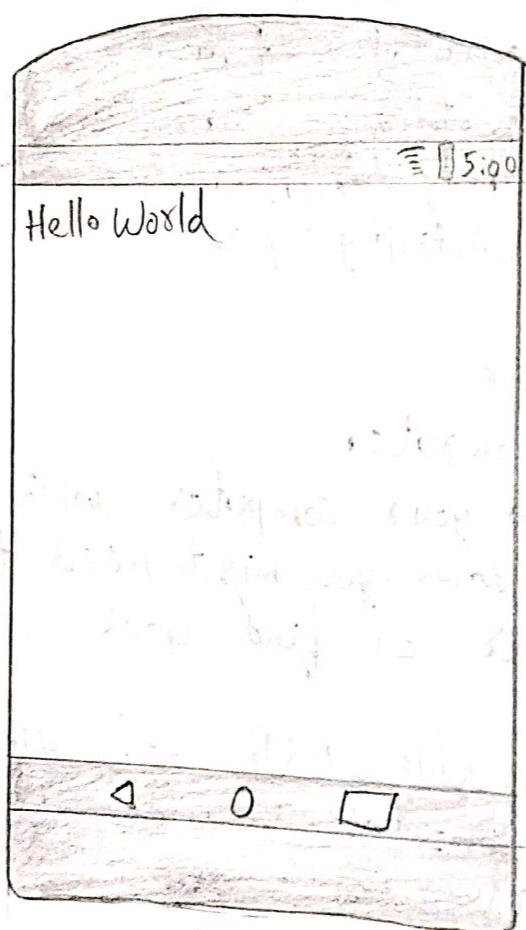


Fig 75

12) Dialog appears "Allows USB Debugging?"

13) Check "Always allow from this computer" and then hit "OK".

Running your App

Now, we can launch apps from Android studio onto our device:

- 1) Select one of your projects and click "Run" from the toolbar.
- 2) In the "Choose Device" window that appears, select the "choose a running device" radio button, select the device and click OK.

ii) Running app on Emulator(AVD)

To run the app from Android studio, open one of your project's activity files and click Run (play icon) from the tool bar. Android studio installs the app on your AVD and starts it and if everything is fine with your set-up and application, it will display following Emulator window - once Gradle finishes building, Android studio should install the app on your connected device and start it.

Experiment - 3

Create a screen that has input boxes for UserName, Password, Address, Gender (radio buttons for male and female), Age (numeric) and a Submit button. On clicking the submit button, print all the data below the Submit Button (use any layout).

Code for MainActivity.java

```
package com.example.nkreddy.mrcet;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
public class MainActivity extends AppCompatActivity
{
    EditText editName, editPassword, editAddress, editAge;
    TextView result;
    RadioGroup radioGenderGroup;
    private RadioButton radioGenderButton;
    Button buttonSubmit;
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        editName = findViewById(R.id.editName);
        editPassword = findViewById(R.id.editPassword);
        editAddress = findViewById(R.id.editAddress);
```

```

editAge = (EditText) findViewById(R.id.editAge)
radioSexGender = (RadioGroup) findViewById(R.id.radioGender);
result = (TextView) findViewById(R.id.tvResult);
buttonSubmit = (Button) findViewById(R.id.buttonSubmit);
buttonSubmit.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v)
    {
        String name = editName.getText().toString();
        String password = editPassword.getText().toString();
        String address = editAddress.getText().toString();
        gen = (RadioButton) findViewById(selectedId);
        result.setText("Name: " + name + " Password: " + password +
        " Address: " + address + " AGE: " + age +
        " Gender: " + gen);
    }
});
}

```

activity_main.xml

```

<?xml version='1.0' encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFFF8D"/>

```

< TextView

```
    android:id = "@+id/textView"
    android:layout_width = "wrap_content"
    android:layout_height = "wrap_content"
    android:layout_alignParentLeft = "true"
    android:layout_alignParentStart = "true"
    android:layout_alignParentTop = "true"
    android:text = "Name"
    android:textSize = "20sp"
    android:layout_margin = "20dp" />
```

< TextView

```
    android:id = "@+id/textView2"
    android:layout_width = "wrap_content"
    android:layout_height = "wrap_content"
    android:textSize = "20sp"
    android:text = "Password"
    android:layout_marginTop = "38dp"
    android:layout_below = "@+id/textView"
    android:layout_alignLeft = "@+id/textView"
    android:layout_alignStart = "@+id/textView" />
```

< EditText

```
    android:id = "@+id/editName"
    android:layout_width = "wrap_content"
    android:layout_height = "wrap_content"
    android:ems = "10"
    android:inputType = "textPersonName"
    android:hint = "Enter Name"
    android:layout_alignParentTop = "true"
    android:layout_alignParentRight = "true"
    android:layout_alignParentEnd = "true" />
```

<EditText

```
    android:id = "@+id/editAddress"
    android:layout_width = "wrap_content"
    android:layout_height = "wrap_content"
    android:layout_marginRight = "18dp"
    android:layout_marginEnd = "18dp"/>
```

<EditText

```
    android:id = "@+id/editAge"
    android:layout_width = "wrap_content"
    android:layout_height = "wrap_content"
    android:layout_marginRight = "18dp"
    android:layout_marginEnd = "18dp"/>
```

<RadioGroup

```
    android:id = "@+id/radioGender"
    android:layout_width = "wrap_content"
    android:layout_height = "wrap_content"
```

<RadioButton

```
    android:id = "@+id/radioMale"
    android:layout_width = "wrap_content"
    android:layout_height = "wrap_content"
    android:text = "@string/radio-male"
    android:checked = "true"/>
```

<RadioButton

```
    android:id = "@+id/radioFemale"
    android:layout_width = "wrap_content"
    android:layout_height = "wrap_content"
    android:text = "@string/radio-female"/>
```

</RadioGroup>

<Button

```
    android:id = "@+id/buttonSubmit"
    android:layout_width = "wrap_content"
    android:layout_height = "wrap_content"
    android:layout_alignParentLeft = "true"
    android:layout_alignParentStart = "true"
    android:layout_below = "@+id/textView2"
    android:layout_marginTop = "20dp"
    android:text = "SUBMIT" />
```

<TextView>

```
    android:id = "@+id/tvResult"
    android:layout_width = "wrap_content"
    android:layout_width = "wrap_content"
    android:layout_alignParentBottom = "true"
    android:layout_alignParentLeft = "true"
    android:layout_alignParentStart = "true"
    android:layout_marginBottom = "143dp"
    android:textsize = "30sp" /> </RelativeLayout>
```