

Historical Places in MAHARASTRA

Objective: Our state MAHARASTRA is rich in history and culture. This fact is known from ages. But many of us residing in Maharashtra are actually un-aware of its culture and historical places.

Keeping this requirement in mind, you have to develop a **HistoricalApp**, which allows a user to

1. Know about the Maharashtra State in brief.
2. View Google Map of Maharashtra , highlighting at least 10 prominent places to visit
3. On selection of any place *brief details of the place, best time to visit and How to reach* should be displayed.

Technical Specifications:

1st Activity: Should be a splash screen for 5 seconds which can be created using thread object. After 5 seconds control goes to **2nd Activity**.



Figure 1 : Splash Screen

2nd Activity : Is the **HomepageActv**. It should provide brief about Information about the State, as shown in **figure 2 below**.



Figure 2

On Continue we go to the **3rd Activity**. This shows **Google Map of Maharastra with markers** indicating important places (or must visit) places. See figure 3

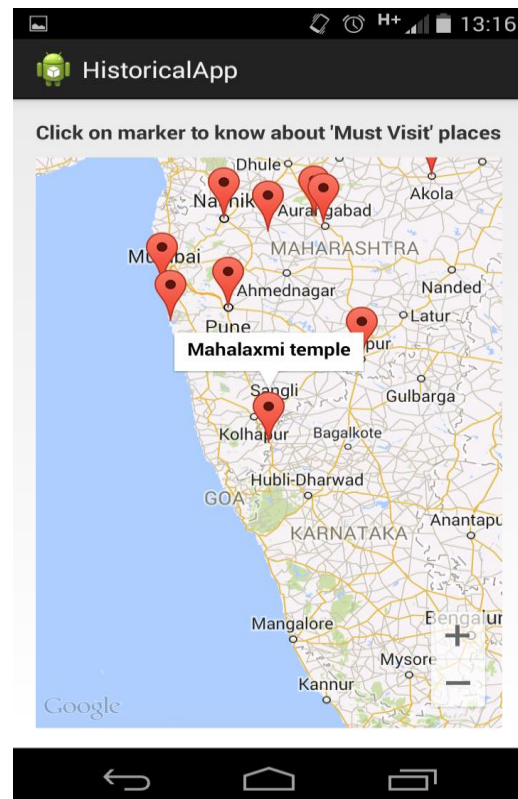
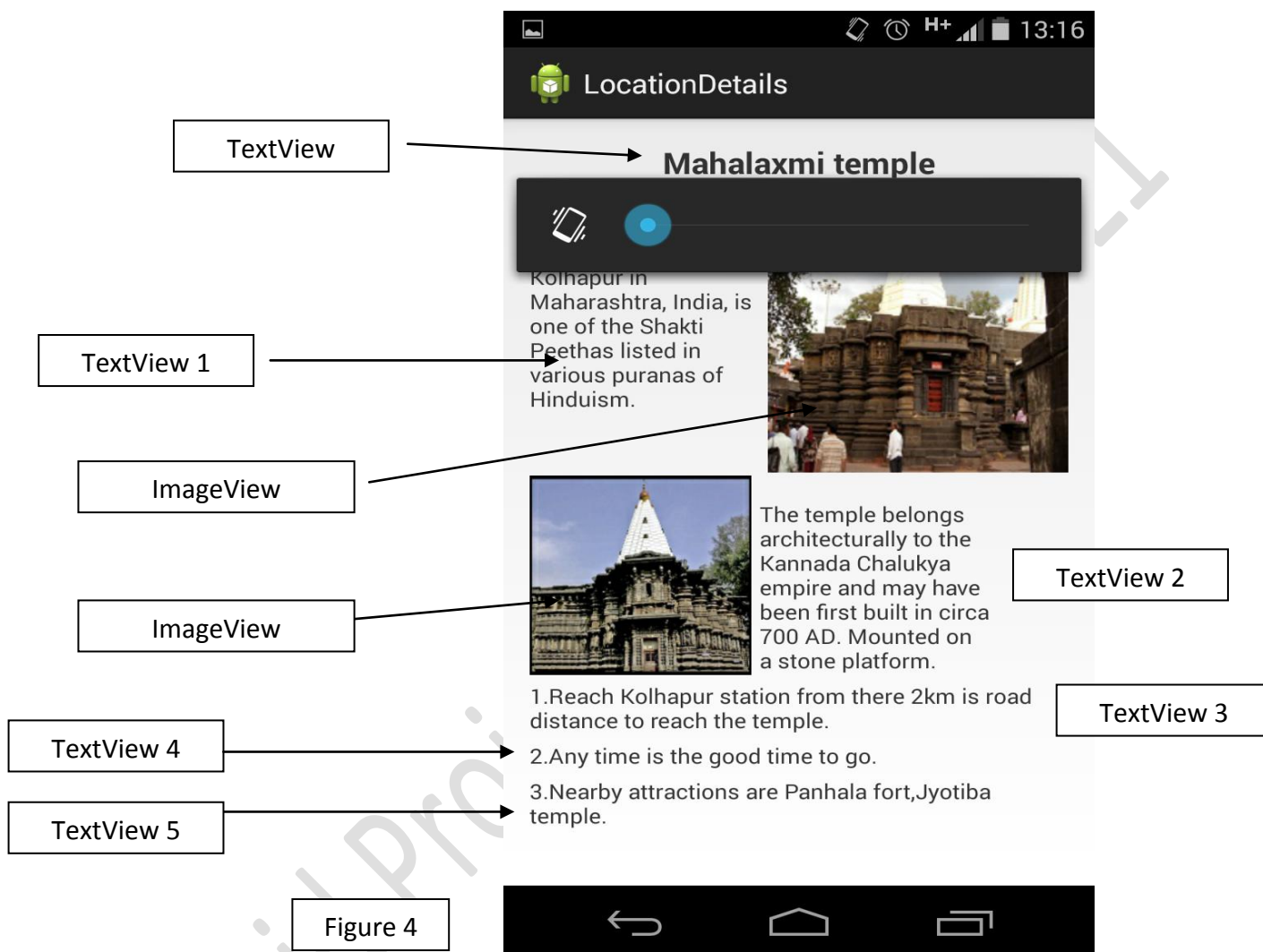


Figure 3

On selecting any marker over the google Map we call an activity to show the details of the place. **See figure 4**



Kindly note : For all places only 1 Activity has to be made.

How to show details of a respective place in the 4th activity ?

Depending upon the marker selected in the 3rd Activity, pass a marker title to 4th activity (**LocationDetails**) . In **LocationDetails** we have to populate the Activity from strings.xml file. **(Refer all code snippets)**

Code snippet 1 : Defining Marker positions and adding them to Map :

```
double lat []={
18.921876000000000000,
20.026318700000000000,
19.901497500000000000,
18.299669400000000000,
19.997453300000000000,
19.783140000000000000,
20.706244500000000000,
18.518720400000000000,
17.680000000000000000,
16.254930900000000000
};
```

Use these coordinates in your project. These are exact coordinates of the places mentioned in the String name[] array

```
double lon[]={
72.834677999999940000,
75.177640900000030000,
75.320119900000010000,
72.964880100000070000,
73.789802300000020000,
74.466506999999980000,
76.990047600000030000,
73.856654599999950000,
75.920000000000070000,
```

```
74.479062300000010000
```

```
};
```

```
String name[]={
```

```
"Gatewayofindia",
```

```
"Ajanta-Ellora Caves",
```

```
"Bibi Ka Maqbara",
```

```
"Janjira Fort",
```

```
"Nashik",
```

```
"Shirdi",
```

```
"Akola fort",
```

```
"Lal mahal",
```

```
"Mallikarjuna temple",
```

```
"Mahalaxmi temple"
```

```
};
```

```
for(int i=0;i<10;i++)
```

```
{
```

```
// making Marker Object, specifying Lat , Lng and title
```

```
mark=new MarkerOptions()
```

```
    .position(new LatLng(lat[i], lon[i]))
```

```
    .title(name[i]);
```

// adding Marker to the map

```
Marker loc = map.addMarker(mark);
```

// to show the title Information

```
loc.showInfoWindow(); // only the last title would be viewed.
```

```
// Others would be viewed on Touch over the marker
```

```
}
```

Code Snippet 2: Handle Click over Marker and transfer control to activity showing details of the place.

```
map.setOnMarkerClickListener(new OnMarkerClickListener() {
```

```
@Override
```

```
public boolean onMarkerClick(Marker arg0) {
```

```
String title=arg0.getTitle();
```

```
Bundle b=new Bundle();
```

```
b.putString("file", title);
```

```
Intent i=new Intent(getBaseContext(), LocationDetails.class);
```

```
i.putExtras(b);
```

```
startActivity(i);
```

```
return false;
```

```
}
```

```
});
```

Bundle is used for holding the **title** of the Marker location. So that in the next activity (LocationDetails) , you can use it to fetch information from the strings.xml file.

Code Snippet 3: How to fetch details of the place from strings.xml file in the 4th activity (LocationActivity)

```
Bundle b = getIntent().getExtras();
```

```
String file = b.getString("file");
```

```
tv.setText(file); // TextView showing the Heading , top center
```

```
if (file.equals("Gatewayofindia")) {
```

```
    tv1.setText(getString(R.string.gatetv1));
```

```
    tv2.setText(getString(R.string.gatetv2));
```

```
    tv3.setText(getString(R.string.gatetv3));
```

```
    tv4.setText(getString(R.string.gatetv4));
```

```
    tv5.setText(getString(R.string.gatetv5));
```

```
    i1.setImageResource(R.drawable.gof1);
```

```
    i2.setImageResource(R.drawable.gof2);
```

```
}
```

Separate TextViews are needed as \n or
 does not work in xml file. Hence we make different TextViews to put space between them.

Images are fetched from the drawable folder. The Student must download (at his end) and use most appropriate images.