SLIIT
Discover Your Future

BSc (Hons) in Information Technology
Year 2

Additional Lab sheet

IT2010 – Mobile Application Development

Android Services

- Create a new project in Android Studio to do the following.

- In the first Activity, add the following code to its onCreate() method.

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    MediaPlayer player =
MediaPlayer.create(this,Settings.System.DEFAULT_NOTIFICATION_URI);
    player.setLooping(true);
    player.start();
}
```

- Import MediaPlayer library.

```
import android.media.MediaPlayer;
```

- Now, run the project. You will here a sound. You can change the sound instead of using "*DEFAULT_NOTIFICATION_URI*".

- Go to Home screen / another app in your device or emulator. Now the music will get stop.

- If you want to play a music as a background feature of your app, that means even though you switched to another app/ another window / home screen of your device, still if you want to continue playing music, then the option is using Android **Services**. Not only for playing music, but if you want to perform another task as a background service also, can be done accordingly.

Go through the following steps to make it as a service.

- Create a java class called MyServices.java and extend the Service class which is the parent class for Services.

```
public class MyServices extends Service {
```

- You will see an error message that asks to override the methods available in Service class. Override onStartCommand() and onDestroy() methods.

- Now cut the above code snippet you used to play a music before and paste it inside onStartCommand() method and modify it as following.

```
@Override
public int onStartCommand(Intent intent, int flags, int startId) {
    player = MediaPlayer.create(this,
Settings.System.DEFAULT_NOTIFICATION_URI);
    player.setLooping(true);
    player.start();
    return START_STICKY;
    //return super.onStartCommand(intent, flags, startId);
}
```

- Create MediaPlayer type object as a private object to your MyServices.java class.

```
private MediaPlayer player;
```

- Modify onDestroy() method as following to stop playing music.

```
@Override
public void onDestroy() {
    super.onDestroy();
    player.stop();
}
```

- Add your service class as a service in AndroidManifest.xml file before closing the `</application>` tag.

```
<service android:name=".MyServices"></service>
```

- Now the service is ready to perform. You can use this whenever you want in your mobile application by calling to above methods.

Sample App:

- We will use this service from the activity we created at the beginning. Add two buttons called start and stop to the layout. Button click event of the Start button will start playing the music and Stop button will stop it.
- Implement the button click events as following.

```
btnStart.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        startService(new Intent(MainActivity.this, MyServices.class));
    }
});

btnStop.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        stopService(new Intent(MainActivity.this, MyServices.class));
    }
});
```

- Now run the application again and control music by using above two buttons.

- Start playing music and go for another application / Home screen of your device as you did before and see what will happen. You will here the music even though you are interacting with any other application. This is because the music is playing as a background service in your app.