1. **Write a code example to show the use of brightness Adjuster broadcast by adjusting the phone brightness according to time of phone, for example, if the time is between 07:00PM to 05:00AM, it will adjust the brightness to minimum other wise, adjust the brightness to maximum.**

**Sol:**

**Xml**

**Add\_brighttime\_point.xml**

*<?***xml version="1.0" encoding="utf-8"***?>*<**FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent" android:layout\_height="match\_parent"**

**android:background="@drawable/pp"**>

<**FrameLayout  
 android:layout\_width="fill\_parent"  
 android:layout\_height="149dp"  
 android:layout\_gravity="center\_horizontal|bottom"**>  
  
 <**SeekBar  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/seekBrightness"  
 android:layout\_gravity="center"** />  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/BrightnessText"  
 android:layout\_gravity="center\_horizontal|top"  
 android:text="Set Brightness Level"** />  
  
 <**Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Confirm"  
 android:id="@+id/confirmAdd"  
 android:layout\_gravity="center\_horizontal|bottom"** />  
 </**FrameLayout**>  
  
 <**TimePicker  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/timePickerBrightness"  
 android:layout\_gravity="center\_horizontal|top"** />  
  
</**FrameLayout**>

**Java**

**addBrightPoint.java**

**package** teamunguided.brighttime;  
  
**import** android.app.Activity;  
**import** android.app.AlarmManager;  
**import** android.app.PendingIntent;  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.content.SharedPreferences;  
**import** android.net.Uri;  
**import** android.os.Bundle;  
**import** android.preference.PreferenceManager;  
**import** android.util.Log;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.SeekBar;  
**import** android.widget.TimePicker;  
**import** android.widget.Toast;  
**import** java.util.Calendar;  
**import** java.util.Collections;  
**import** java.util.HashSet;  
**import** java.util.Iterator;  
**import** java.util.Random;  
**import** java.util.Set;  
  
**public class** addBrightPoint **extends** Activity{  
 **static final** String ***SETTINGS\_HOUR*** = **"hour"**;  
 **static final** String ***SETTINGS\_MINUTES*** = **"minute"**;  
 **private static final** String ***alarmNames*** = **"alrmnam"**;  
 **private static final** String ***TAG*** = **"addBrightPoint"**;  
 **private** Set<String> **pointNames** = Collections.*emptySet*();  
 **private** Set<String> **\_pointNames** = Collections.*emptySet*();  
 **private** Set<String> **temp** = Collections.*emptySet*();  
 **private** AlarmManager **alarmgr**;  
 **private int randomID**;  
 **private** String **stringID**;  
  
 **private static** Context *mContext*;  
 **final int intmax** = 255;  
 **int brightnessToBeSet** = 0;  
  
 @Override  
 **protected void** onStop() {  
 Log.*w*(***TAG***, **"App stopped"**);  
  
 **super**.onStop();  
 }  
 **public static** Context getContext() {  
 **return** *mContext*;  
 }  
  
 CharSequence **text** = **"Point added!"**;  
 **int duration** = Toast.***LENGTH\_SHORT***;  
  
 @Override  
 **public void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 *mContext* = getApplicationContext();  
 setContentView(R.layout.***add\_brighttime\_point***);  
  
 SeekBar brightnessSeeker = (SeekBar) findViewById(R.id.***seekBrightness***);  
  
 **final** TimePicker brightnessTime = (TimePicker) findViewById(R.id.***timePickerBrightness***);  
 **final** Button confirmAdd = (Button) findViewById(R.id.***confirmAdd***);  
  
 brightnessSeeker.setMax(**intmax**);  
 brightnessTime.setIs24HourView(**false**);  
  
 brightnessSeeker.setProgress(0);  
 brightnessSeeker.setOnSeekBarChangeListener(**new** SeekBar.OnSeekBarChangeListener() {  
 **int currprogress** = 0;  
 @Override  
 **public void** onProgressChanged(SeekBar seekBar, **int** progress, **boolean** fromUser) {  
 **currprogress** = progress;  
 }  
  
 @Override  
 **public void** onStartTrackingTouch(SeekBar seekBar) { }  
  
 @Override  
 **public void** onStopTrackingTouch(SeekBar seekBar) {  
 **brightnessToBeSet** = **currprogress**;  
 }  
 });  
  
 confirmAdd.setOnClickListener(**new** View.OnClickListener(){  
 **public void** onClick(View v){  
 Calendar selectedTime = Calendar.*getInstance*();  
  
 selectedTime.set(Calendar.***MINUTE***, brightnessTime.getCurrentMinute());  
 selectedTime.set(Calendar.***HOUR\_OF\_DAY***, brightnessTime.getCurrentHour());  
 selectedTime.set(Calendar.***SECOND***,0);  
 SharedPreferences settings = PreferenceManager.*getDefaultSharedPreferences*(*getContext*());  
 **pointNames** = settings.getStringSet(***alarmNames***, **temp**);  
  
 String currID = **null**;  
 Random rand = **new** Random();  
 **do** {  
 **randomID** = rand.nextInt((999999 - 1) + 1) + 1;  
 **stringID** = Integer.*toString*(**randomID**);  
 **if** (**pointNames**.size() != 0) {  
 **for** (Iterator<String> e = **pointNames**.iterator(); e.hasNext(); ) {  
 currID = e.next();  
 **if** (currID.equals(**stringID**)) {  
 **break**;  
 }  
 }  
 }  
 } **while** (**stringID**.equals(currID));  
  
 setBrightnessTimer(**brightnessToBeSet**, selectedTime, **randomID**);  
  
 **\_pointNames** = **new** HashSet<String>();  
 **for** (String currWord: **pointNames**){  
 **\_pointNames**.add(currWord);  
 }  
 **\_pointNames**.add(**stringID**);  
  
 *//pointNames.add(stringID); //broken add* SharedPreferences.Editor editStorage = settings.edit();  
 editStorage.remove(***alarmNames***);  
 editStorage.putStringSet(***alarmNames***, **\_pointNames**);  
 **int** hourSet = brightnessTime.getCurrentHour();  
 **int** minuteSet = brightnessTime.getCurrentMinute();  
 editStorage.putInt(**stringID** + ***SETTINGS\_HOUR***,hourSet);  
 editStorage.putInt(**stringID** + ***SETTINGS\_MINUTES***, minuteSet);  
 editStorage.putInt(**stringID**, **brightnessToBeSet**);  
 editStorage.commit();  
 Toast toast = Toast.*makeText*(*mContext*, **text**, **duration**);  
 toast.show();  
 Intent intent = **new** Intent(getApplicationContext(), BrightTime.**class**);  
 intent.addFlags(Intent.***FLAG\_ACTIVITY\_CLEAR\_TASK***);  
 startActivity(intent);  
 }  
 });  
 }  
  
 @Override  
 **protected void** onDestroy() {  
 Log.*w*(***TAG***, **"App destroyed"**);  
  
 **super**.onDestroy();  
 }  
 **public void** setBrightnessTimer(**int** userinputBrightness, Calendar userinputTimeset, **int** alarmID){  
 **alarmgr** = (AlarmManager) *mContext*.getSystemService(Context.***ALARM\_SERVICE***);  
 Intent brightnessIntent = **new** Intent(addBrightPoint.**this**, BrightTimeService.**class**);  
 String temp = Integer.*toString*(userinputBrightness);  
 brightnessIntent.setData(Uri.*parse*(temp));  
 PendingIntent setBrightness = PendingIntent.*getService*(addBrightPoint.**this**,alarmID,  
 brightnessIntent, PendingIntent.***FLAG\_UPDATE\_CURRENT***);  
 **alarmgr**.setRepeating(AlarmManager.***RTC***, userinputTimeset.getTimeInMillis(), AlarmManager.***INTERVAL\_DAY***, setBrightness);  
 }  
  
  
}

**xml**

**bright\_time.xml**

*<?***xml version="1.0" encoding="utf-8"***?>*<**FrameLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"**

**android:background="@drawable/pp"**>  
  
 <**ListView  
 android:id="@+id/pointlist"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**/>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="bottom"  
 android:orientation="horizontal"**>  
  
 <**FrameLayout  
 android:layout\_width="0dp"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="start|center\_vertical"  
 android:layout\_weight="1"**>  
  
 <**ImageButton  
 android:id="@+id/left\_button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="center\_horizontal"  
 android:background="?android:attr/selectableItemBackgroundBorderless"  
 android:scaleType="center"  
 android:contentDescription="@null"** />  
  
 </**FrameLayout**>  
  
 <**FrameLayout  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"**>  
  
 <**ImageButton  
 android:id="@+id/addbrighttimepoint"  
 android:tint="@color/white"  
 android:layout\_width="@dimen/footer\_button\_size"  
 android:layout\_height="@dimen/footer\_button\_size"  
 android:layout\_margin="@dimen/footer\_button\_layout\_margin"  
 android:layout\_gravity="center\_horizontal"  
 android:background="@drawable/floating\_action\_button"  
 android:src="@android:drawable/ic\_input\_add"  
 android:scaleType="center"  
 android:contentDescription="@null"** />  
  
 </**FrameLayout**>  
  
 <**FrameLayout  
 android:layout\_width="0dp"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="end|center\_vertical"  
 android:layout\_weight="1"**>  
  
 <**ImageButton  
 android:id="@+id/right\_button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="center\_horizontal"  
 android:background="?android:attr/selectableItemBackgroundBorderless"  
 android:scaleType="center"  
 android:contentDescription="@null"** />  
  
 </**FrameLayout**>  
  
 </**LinearLayout**>  
  
</**FrameLayout**>

**BrightTime.java**

**package** teamunguided.brighttime;  
**import** android.app.Activity;  
**import** android.app.AlarmManager;  
**import** android.app.PendingIntent;  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.content.SharedPreferences;  
**import** android.net.Uri;  
**import** android.os.Build;  
**import** android.os.Bundle;  
**import** android.os.Handler;  
**import** android.preference.PreferenceManager;  
**import** android.support.annotation.RequiresApi;  
**import** android.view.View;  
**import** android.widget.AdapterView;  
**import** android.widget.ArrayAdapter;  
**import** android.widget.ImageButton;  
**import** android.widget.ListView;  
**import** android.widget.Toast;  
**import** java.util.ArrayList;  
**import** java.util.Arrays;  
**import** java.util.Calendar;  
**import** java.util.Collections;  
**import** java.util.HashMap;  
**import** java.util.HashSet;  
**import** java.util.List;  
**import** java.util.Set;  
**public class** BrightTime **extends** Activity {  
 **static final** String ***SETTINGS\_HOUR*** = **"hour"**;  
 **static final** String ***SETTINGS\_MINUTES*** = **"minute"**;  
 **private static final** String ***alarmNames*** = **"alrmnam"**;  
 **private** Set<String> **pointNames** = Collections.*emptySet*();  
 **private** Set<String> **temp** = Collections.*emptySet*();  
 **private static** Context *mContext*;  
 **private** AlarmManager **alarmgr**;  
 **private** ImageButton **addPoint**;  
 **private** ImageButton **mLeftButton**;  
 **private** ImageButton **mRightButton**;  
 **private static final float *FAB\_DEPTH*** = 20f;  
 **private static final int *UNKNOWN\_COLOR\_ID*** = 0;  
 **private boolean doubleBackToExitPressedOnce** = **false**;  
 StableArrayAdapter **adapter**;  
 **public static** Context getContext() {  
 **return** *mContext*;  
 }  
 @RequiresApi(api = Build.VERSION\_CODES.***LOLLIPOP***)  
  
 @Override  
 **public void** onCreate(Bundle savedInstanceState){  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***bright\_time***);  
 SharedPreferences settings = PreferenceManager.*getDefaultSharedPreferences*(getApplicationContext());  
 **if**(settings.getBoolean(**"isFirstRun"**,**true**)){  
 setDefaultPoints();  
 SharedPreferences.Editor editInitial = settings.edit();  
 editInitial.putBoolean(**"isFirstRun"**, **false**);  
 editInitial.commit();  
 }  
 **addPoint** = (ImageButton) findViewById(R.id.***addbrighttimepoint***);  
 **addPoint**.setTranslationZ(***FAB\_DEPTH***);  
 **mLeftButton** = (ImageButton) findViewById(R.id.***left\_button***);  
 **mRightButton** = (ImageButton) findViewById(R.id.***right\_button***);  
 **final** ListView mPointList = (ListView) findViewById(R.id.***pointlist***);  
 *mContext* = getApplicationContext();  
 **pointNames** = settings.getStringSet(***alarmNames***, **temp**);  
 **final** String[] arrPointNames = **pointNames**.toArray(**new** String[**pointNames**.size()]);  
 **final** ArrayList<String> list = **new** ArrayList<String>();  
 **for**(**int** i = 0; i < arrPointNames.**length**; ++i) {  
 String displayTime;  
 **boolean** isPM = **false**;  
 **int** displayHour = settings.getInt((arrPointNames[i] + ***SETTINGS\_HOUR***), -1);  
 **int** displayMin = settings.getInt((arrPointNames[i] + ***SETTINGS\_MINUTES***), -1);  
 **if** (displayHour == -1 || displayMin == -1)  
 list.add(**"Error: Unable to Retrieve Point"**);  
 **else** {  
 **if** (displayHour % 12 == 0) {  
 **if** (displayHour != 0)  
 isPM = **true**;  
 displayTime = **"12:"**;  
 } **else if** (displayHour < 12)  
 displayTime = displayHour + **":"**;  
 **else** {  
 displayTime = ((displayHour % 12)) + **":"**;  
 isPM = **true**;  
 }  
 **if** (isPM) {  
 **if** (displayMin < 10)  
 list.add(displayTime + **"0"** + displayMin + **" pm"**);  
 **else** list.add(displayTime + displayMin + **" pm"**);  
 } **else** {  
 **if** (displayMin < 10)  
 list.add(displayTime + **"0"** + displayMin + **" am"**);  
 **else** list.add(displayTime + displayMin + **" am"**);  
 }  
 }  
 }  
 **adapter** = **new** StableArrayAdapter(**this**,  
 R.layout.***settinglist***, list);  
 mPointList.setAdapter(**adapter**);  
 mPointList.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {  
 *//****TODO:change the click action to bring you to an edit screen*** @Override  
 **public void** onItemClick(AdapterView<?> parent, **final** View view,  
 **int** position, **long** id) {  
 **final int** tisPosition = position;  
 **final** String item = (String) parent.getItemAtPosition(position);  
 view.animate().setDuration(500).alpha(1)  
 .withEndAction(  
 **new** Runnable() {  
 @Override  
 **public void** run() {  
 Intent editIntent = **new** Intent(getApplicationContext(), editPoint.**class**);  
 editIntent.addFlags(Intent.***FLAG\_ACTIVITY\_CLEAR\_TASK***);  
 editIntent.addFlags(Intent.***FLAG\_ACTIVITY\_NO\_HISTORY***);  
 editIntent.putExtra(**"stringID"**, arrPointNames[tisPosition]);  
 startActivity(editIntent);  
 **adapter**.notifyDataSetChanged();  
 view.setAlpha(1);  
 }  
 });  
 }  
 });  
  
 **addPoint**.setOnClickListener(**new** View.OnClickListener(){  
 **public void** onClick(View v) {  
 Intent intent = **new** Intent(getApplicationContext(), addBrightPoint.**class**);  
 intent.addFlags(Intent.***FLAG\_ACTIVITY\_CLEAR\_TASK***);  
 startActivity(intent);  
 }  
 });  
 }  
  
 @Override  
 **public void** onResume(){  
 **super**.onResume();  
 SharedPreferences settings = PreferenceManager.*getDefaultSharedPreferences*(*getContext*());  
 **pointNames** = settings.getStringSet(***alarmNames***, **temp**);  
 }  
  
 @Override  
 **public void** onBackPressed() {  
 **if** (**doubleBackToExitPressedOnce**) {  
 **super**.onBackPressed();  
 **this**.finish();  
 Intent intent = **new** Intent(Intent.***ACTION\_MAIN***);  
 intent.addCategory(Intent.***CATEGORY\_HOME***);  
 intent.setFlags(Intent.***FLAG\_ACTIVITY\_CLEAR\_TASK***);  
 startActivity(intent);  
 }  
 **this**.**doubleBackToExitPressedOnce** = **true**;  
 Toast.*makeText*(**this**, **"Please click BACK again to exit"**, Toast.***LENGTH\_SHORT***).show();  
  
 **new** Handler().postDelayed(**new** Runnable() {  
  
 @Override  
 **public void** run() {  
 **doubleBackToExitPressedOnce**=**false**;  
 }  
 }, 2000);  
 }  
 **private class** StableArrayAdapter **extends** ArrayAdapter<String> {  
  
 HashMap<String, Integer> **mIdMap** = **new** HashMap<String, Integer>();  
 **public** StableArrayAdapter(Context context, **int** textViewResourceId,  
 List<String> objects) {  
 **super**(context, textViewResourceId, objects);  
 **for** (**int** i = 0; i < objects.size(); ++i) {  
 **mIdMap**.put(objects.get(i), i);  
 }  
 }  
 @Override  
 **public long** getItemId(**int** position) {  
 String item = getItem(position);  
 **return mIdMap**.get(item);  
 }  
 @Override  
 **public boolean** hasStableIds() {  
 **return true**;  
 }  
 }  
 **private void** setDefaultPoints(){  
 SharedPreferences settings = PreferenceManager.*getDefaultSharedPreferences*(getApplicationContext());  
 Set<String> initialPoints = **new** HashSet<String>(Arrays.*asList*( **"0"**, **"1"**, **"2"** ,**"3"** , **"4"**));  
 String[] initPoints = {**"0"**,**"1"**,**"2"**,**"3"**,**"4"**};  
 **int**[] initialTimeHOUR = {6,8,12,14,19};  
 **int**[] initialsetBrightness = {64,153,255,128,51};  
 SharedPreferences.Editor editInitial = settings.edit();  
 **for**(**int** i = 0; i < 5 ; ++i){  
 Calendar setTime = Calendar.*getInstance*();  
 setTime.set(Calendar.***HOUR\_OF\_DAY***, initialTimeHOUR[i]);  
 setTime.set(Calendar.***MINUTE***, 0);  
 setTime.set(Calendar.***SECOND***, 0);  
 setBrightnessTimer(initialsetBrightness[i], setTime, Integer.*parseInt*(initPoints[i]));  
 editInitial.putInt(initPoints[i], initialsetBrightness[i]);  
 editInitial.putInt(initPoints[i] + ***SETTINGS\_HOUR***, initialTimeHOUR[i]);  
 editInitial.putInt(initPoints[i] + ***SETTINGS\_MINUTES***, 0);  
 }  
 editInitial.putStringSet(***alarmNames***, initialPoints);  
 editInitial.commit();  
 }  
 **public void** setBrightnessTimer(**int** userinputBrightness, Calendar userinputTimeset, **int** alarmID){  
 **alarmgr** = (AlarmManager) getApplicationContext().getSystemService(Context.***ALARM\_SERVICE***);  
 Intent brightnessIntent = **new** Intent(BrightTime.**this**, BrightTimeService.**class**);  
 String temp = Integer.*toString*(userinputBrightness);  
 brightnessIntent.setData(Uri.*parse*(temp));  
 PendingIntent setBrightness = PendingIntent.*getService*(BrightTime.**this**,alarmID,  
 brightnessIntent, PendingIntent.***FLAG\_UPDATE\_CURRENT***);  
 **alarmgr**.setRepeating(AlarmManager.***RTC***, userinputTimeset.getTimeInMillis(), AlarmManager.***INTERVAL\_DAY***, setBrightness);  
 }  
}  
**class** TimeAdapter **extends** ArrayAdapter<String> {  
 HashMap<String, Integer> **mIdMap** = **new** HashMap<String, Integer>();  
 **public** TimeAdapter(Context context, **int** textViewResourceId,  
 List<String> objects) {  
 **super**(context, textViewResourceId,objects);  
 **mIdMap**.put(**"testing!!"**, 1);  
 }  
  
}

**TimeAdaptor.java**

**package** teamunguided.brighttime;  
**import** android.app.Activity;  
**import** android.app.AlarmManager;  
**import** android.app.PendingIntent;  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.content.SharedPreferences;  
**import** android.net.Uri;  
**import** android.os.Build;  
**import** android.os.Bundle;  
**import** android.os.Handler;  
**import** android.preference.PreferenceManager;  
**import** android.support.annotation.RequiresApi;  
**import** android.view.View;  
**import** android.widget.AdapterView;  
**import** android.widget.ArrayAdapter;  
**import** android.widget.ImageButton;  
**import** android.widget.ListView;  
**import** android.widget.Toast;  
  
**import** java.util.ArrayList;  
**import** java.util.Arrays;  
**import** java.util.Calendar;  
**import** java.util.Collections;  
**import** java.util.HashMap;  
**import** java.util.HashSet;  
**import** java.util.List;  
**import** java.util.Set;  
**public class** BrightTime **extends** Activity {  
 **static final** String ***SETTINGS\_HOUR*** = **"hour"**;  
 **static final** String ***SETTINGS\_MINUTES*** = **"minute"**;  
 **private static final** String ***alarmNames*** = **"alrmnam"**;  
 **private** Set<String> **pointNames** = Collections.*emptySet*();  
 **private** Set<String> **temp** = Collections.*emptySet*();  
 **private static** Context *mContext*;  
 **private** AlarmManager **alarmgr**;  
 **private** ImageButton **addPoint**;  
 **private** ImageButton **mLeftButton**;  
 **private** ImageButton **mRightButton**;  
 **private static final float *FAB\_DEPTH*** = 20f;  
 **private static final int *UNKNOWN\_COLOR\_ID*** = 0;  
 **private boolean doubleBackToExitPressedOnce** = **false**;  
 StableArrayAdapter **adapter**;  
 **public static** Context getContext() {  
 **return** *mContext*;  
 }  
 @RequiresApi(api = Build.VERSION\_CODES.***LOLLIPOP***)  
  
 @Override  
 **public void** onCreate(Bundle savedInstanceState){  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***bright\_time***);  
 SharedPreferences settings = PreferenceManager.*getDefaultSharedPreferences*(getApplicationContext());  
 **if**(settings.getBoolean(**"isFirstRun"**,**true**)){  
 setDefaultPoints();  
 SharedPreferences.Editor editInitial = settings.edit();  
 editInitial.putBoolean(**"isFirstRun"**, **false**);  
 editInitial.commit();  
 }  
 **addPoint** = (ImageButton) findViewById(R.id.***addbrighttimepoint***);  
 **addPoint**.setTranslationZ(***FAB\_DEPTH***);  
 **mLeftButton** = (ImageButton) findViewById(R.id.***left\_button***);  
 **mRightButton** = (ImageButton) findViewById(R.id.***right\_button***);  
 **final** ListView mPointList = (ListView) findViewById(R.id.***pointlist***);  
 *mContext* = getApplicationContext();  
 **pointNames** = settings.getStringSet(***alarmNames***, **temp**);  
 **final** String[] arrPointNames = **pointNames**.toArray(**new** String[**pointNames**.size()]);  
 **final** ArrayList<String> list = **new** ArrayList<String>();  
 **for**(**int** i = 0; i < arrPointNames.**length**; ++i) {  
 String displayTime;  
 **boolean** isPM = **false**;  
 **int** displayHour = settings.getInt((arrPointNames[i] + ***SETTINGS\_HOUR***), -1);  
 **int** displayMin = settings.getInt((arrPointNames[i] + ***SETTINGS\_MINUTES***), -1);  
 **if** (displayHour == -1 || displayMin == -1)  
 list.add(**"Error: Unable to Retrieve Point"**);  
 **else** {  
 **if** (displayHour % 12 == 0) {  
 **if** (displayHour != 0)  
 isPM = **true**;  
 displayTime = **"12:"**;  
 } **else if** (displayHour < 12)  
 displayTime = displayHour + **":"**;  
 **else** {  
 displayTime = ((displayHour % 12)) + **":"**;  
 isPM = **true**;  
 }  
 **if** (isPM) {  
 **if** (displayMin < 10)  
 list.add(displayTime + **"0"** + displayMin + **" pm"**);  
 **else** list.add(displayTime + displayMin + **" pm"**);  
 } **else** {  
 **if** (displayMin < 10)  
 list.add(displayTime + **"0"** + displayMin + **" am"**);  
 **else** list.add(displayTime + displayMin + **" am"**);  
 }  
 }  
 }  
 **adapter** = **new** StableArrayAdapter(**this**,  
 R.layout.***settinglist***, list);  
 mPointList.setAdapter(**adapter**);  
 mPointList.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {  
@Override  
 **public void** onItemClick(AdapterView<?> parent, **final** View view,  
 **int** position, **long** id) {  
 **final int** tisPosition = position;  
 **final** String item = (String) parent.getItemAtPosition(position);  
 view.animate().setDuration(500).alpha(1)  
 .withEndAction(  
 **new** Runnable() {  
 @Override  
 **public void** run() {  
 Intent editIntent = **new** Intent(getApplicationContext(), editPoint.**class**);  
 editIntent.addFlags(Intent.***FLAG\_ACTIVITY\_CLEAR\_TASK***);  
 editIntent.addFlags(Intent.***FLAG\_ACTIVITY\_NO\_HISTORY***);  
 editIntent.putExtra(**"stringID"**, arrPointNames[tisPosition]);  
 startActivity(editIntent);  
 **adapter**.notifyDataSetChanged();  
 view.setAlpha(1);  
 }  
 });  
 }  
 });  
  
 **addPoint**.setOnClickListener(**new** View.OnClickListener(){  
 **public void** onClick(View v) {  
 Intent intent = **new** Intent(getApplicationContext(), addBrightPoint.**class**);  
 intent.addFlags(Intent.***FLAG\_ACTIVITY\_CLEAR\_TASK***);  
 startActivity(intent);  
 }  
 });  
 }  
  
 @Override  
 **public void** onResume(){  
 **super**.onResume();  
 SharedPreferences settings = PreferenceManager.*getDefaultSharedPreferences*(*getContext*());  
 **pointNames** = settings.getStringSet(***alarmNames***, **temp**);  
 }  
  
 @Override  
 **public void** onBackPressed() {  
 **if** (**doubleBackToExitPressedOnce**) {  
 **super**.onBackPressed();  
 **this**.finish();  
 Intent intent = **new** Intent(Intent.***ACTION\_MAIN***);  
 intent.addCategory(Intent.***CATEGORY\_HOME***);  
 intent.setFlags(Intent.***FLAG\_ACTIVITY\_CLEAR\_TASK***);  
 startActivity(intent);  
 }  
 **this**.**doubleBackToExitPressedOnce** = **true**;  
 Toast.*makeText*(**this**, **"Please click BACK again to exit"**, Toast.***LENGTH\_SHORT***).show();  
  
 **new** Handler().postDelayed(**new** Runnable() {  
  
 @Override  
 **public void** run() {  
 **doubleBackToExitPressedOnce**=**false**;  
 }  
 }, 2000);  
 }  
 **private class** StableArrayAdapter **extends** ArrayAdapter<String> {  
  
 HashMap<String, Integer> **mIdMap** = **new** HashMap<String, Integer>();  
 **public** StableArrayAdapter(Context context, **int** textViewResourceId,  
 List<String> objects) {  
 **super**(context, textViewResourceId, objects);  
 **for** (**int** i = 0; i < objects.size(); ++i) {  
 **mIdMap**.put(objects.get(i), i);  
 }  
 }  
 @Override  
 **public long** getItemId(**int** position) {  
 String item = getItem(position);  
 **return mIdMap**.get(item);  
 }  
 @Override  
 **public boolean** hasStableIds() {  
 **return true**;  
 }  
 }  
 **private void** setDefaultPoints(){  
 SharedPreferences settings = PreferenceManager.*getDefaultSharedPreferences*(getApplicationContext());  
 Set<String> initialPoints = **new** HashSet<String>(Arrays.*asList*( **"0"**, **"1"**, **"2"** ,**"3"** , **"4"**));  
 String[] initPoints = {**"0"**,**"1"**,**"2"**,**"3"**,**"4"**};  
 **int**[] initialTimeHOUR = {6,8,12,14,19};  
 **int**[] initialsetBrightness = {64,153,255,128,51};  
 SharedPreferences.Editor editInitial = settings.edit();  
 **for**(**int** i = 0; i < 5 ; ++i){  
 Calendar setTime = Calendar.*getInstance*();  
 setTime.set(Calendar.***HOUR\_OF\_DAY***, initialTimeHOUR[i]);  
 setTime.set(Calendar.***MINUTE***, 0);  
 setTime.set(Calendar.***SECOND***, 0);  
 setBrightnessTimer(initialsetBrightness[i], setTime, Integer.*parseInt*(initPoints[i]));  
 editInitial.putInt(initPoints[i], initialsetBrightness[i]);  
 editInitial.putInt(initPoints[i] + ***SETTINGS\_HOUR***, initialTimeHOUR[i]);  
 editInitial.putInt(initPoints[i] + ***SETTINGS\_MINUTES***, 0);  
 }  
 editInitial.putStringSet(***alarmNames***, initialPoints);  
 editInitial.commit();  
 }  
 **public void** setBrightnessTimer(**int** userinputBrightness, Calendar userinputTimeset, **int** alarmID){  
 **alarmgr** = (AlarmManager) getApplicationContext().getSystemService(Context.***ALARM\_SERVICE***);  
 Intent brightnessIntent = **new** Intent(BrightTime.**this**, BrightTimeService.**class**);  
 String temp = Integer.*toString*(userinputBrightness);  
 brightnessIntent.setData(Uri.*parse*(temp));  
 PendingIntent setBrightness = PendingIntent.*getService*(BrightTime.**this**,alarmID,  
 brightnessIntent, PendingIntent.***FLAG\_UPDATE\_CURRENT***);  
 **alarmgr**.setRepeating(AlarmManager.***RTC***, userinputTimeset.getTimeInMillis(), AlarmManager.***INTERVAL\_DAY***, setBrightness);  
 }  
}  
**class** TimeAdapter **extends** ArrayAdapter<String> {  
 HashMap<String, Integer> **mIdMap** = **new** HashMap<String, Integer>();  
 **public** TimeAdapter(Context context, **int** textViewResourceId,  
 List<String> objects) {  
 **super**(context, textViewResourceId,objects);  
 **mIdMap**.put(**"testing!!"**, 1);  
 }  
  
}

**DummyBrightnessActivity.java**

**package** teamunguided.brighttime;  
  
**import** android.app.Activity;  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.os.Handler;  
**import** android.os.Message;  
**import** android.view.WindowManager;  
  
**public class** DummyBrightnessActivity **extends** Activity {  
  
 **private static final int *DELAYED\_MESSAGE*** = 1;  
  
 **private** Handler **handler**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 **handler** = **new** Handler() {  
 @Override  
 **public void** handleMessage(Message msg) {  
 **if**(msg.**what** == ***DELAYED\_MESSAGE***) {  
 DummyBrightnessActivity.**this**.finish();  
 }  
 **super**.handleMessage(msg);  
 }  
 };  
 Intent brightnessIntent = **this**.getIntent();  
 **float** brightness = brightnessIntent.getFloatExtra(**"brightness value"**, 0);  
 WindowManager.LayoutParams lp = getWindow().getAttributes();  
 lp.**screenBrightness** = brightness;  
 getWindow().setAttributes(lp);  
  
 Message message = **handler**.obtainMessage(***DELAYED\_MESSAGE***);  
 **handler**.sendMessageDelayed(message,500);  
 }  
  
}

**edit\_point.xml**

*<?***xml version="1.0" encoding="utf-8"***?>*<**FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent" android:layout\_height="match\_parent"**

**android:background="@drawable/pp"**>

<**TimePicker  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/timePickerEdit"  
 android:layout\_gravity="center\_horizontal|top"** />  
  
 <**FrameLayout  
 android:layout\_width="fill\_parent"  
 android:layout\_height="201dp"  
 android:layout\_gravity="center\_horizontal|bottom"**>  
  
 <**Button  
 android:layout\_width="191dp"  
 android:layout\_height="wrap\_content"  
 android:text="delete"  
 android:id="@+id/rmPoint"  
 android:layout\_gravity="left|bottom"** />  
  
 <**SeekBar  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/brightnessEdit"  
 android:layout\_gravity="center"** />  
  
 <**Button  
 android:layout\_width="191dp"  
 android:layout\_height="wrap\_content"  
 android:text="comfirm"  
 android:id="@+id/comfirmEdit"  
 android:layout\_gravity="right|bottom"** />  
  
 </**FrameLayout**>  
</**FrameLayout**>

**editPoint.java**

**package** teamunguided.brighttime;  
  
**import** android.app.Activity;  
**import** android.app.AlarmManager;  
**import** android.app.PendingIntent;  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.content.SharedPreferences;  
**import** android.net.Uri;  
**import** android.os.Bundle;  
**import** android.preference.PreferenceManager;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.SeekBar;  
**import** android.widget.TimePicker;  
**import** java.util.Calendar;  
**import** java.util.Collections;  
**import** java.util.Set;  
  
**public class** editPoint **extends** Activity {  
 **static final** String ***SETTINGS\_HOUR*** = **"hour"**;  
 **static final** String ***SETTINGS\_MINUTES*** = **"minute"**;  
 **private static final** String ***alarmNames*** = **"alrmnam"**;  
 **private** Set<String> **temp** = Collections.*emptySet*();  
  
 **private** AlarmManager **alarmgr**;  
 **private static** Context *mContext*;  
  
 **private int brightnessToBeSet**;  
  
 @Override  
 **public void** onCreate(Bundle savedInstanceState){  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***edit\_point***);  
 **final** String alarmID = getIntent().getStringExtra(**"stringID"**);  
 *mContext* = getApplicationContext();  
  
 **final** SharedPreferences settings = PreferenceManager.*getDefaultSharedPreferences*(getApplicationContext());  
 **int** currBrightness = settings.getInt(alarmID,-1);  
  
 **final** TimePicker brightnessTime = (TimePicker) findViewById(R.id.***timePickerEdit***);  
 brightnessTime.setIs24HourView(**false**);  
  
 brightnessTime.setCurrentHour(settings.getInt(alarmID + ***SETTINGS\_HOUR***, -1));  
 brightnessTime.setCurrentMinute(settings.getInt(alarmID + ***SETTINGS\_MINUTES***, -1));  
  
 Button confirmEdit = (Button) findViewById(R.id.***comfirmEdit***);  
 Button removeEdit = (Button) findViewById(R.id.***rmPoint***);  
  
 SeekBar brightnessSeeker = (SeekBar) findViewById(R.id.***brightnessEdit***);  
 brightnessSeeker.setMax(255);  
 brightnessSeeker.setProgress(currBrightness);  
  
  
 brightnessSeeker.setOnSeekBarChangeListener(**new** SeekBar.OnSeekBarChangeListener() {  
 **int currprogress** = 0;  
 @Override  
 **public void** onProgressChanged(SeekBar seekBar, **int** progress, **boolean** fromUser) {  
 **currprogress** = progress;  
 *//FUTURE: make it so that as they change they will see the corresponding brightness* }  
  
 @Override  
 **public void** onStartTrackingTouch(SeekBar seekBar) { }  
  
 @Override  
 **public void** onStopTrackingTouch(SeekBar seekBar) {  
 *//FUTURE: could get rid of currprogress and just use brightnessToBeSet in onProgress* **brightnessToBeSet** = **currprogress**;  
 }  
 });  
  
 confirmEdit.setOnClickListener(**new** View.OnClickListener(){  
 **public void** onClick(View v){  
 Calendar selectedTime = Calendar.*getInstance*();  
 selectedTime.set(Calendar.***MINUTE***, brightnessTime.getCurrentMinute());  
 selectedTime.set(Calendar.***HOUR\_OF\_DAY***, brightnessTime.getCurrentHour());  
 selectedTime.set(Calendar.***SECOND***,0);  
  
 removeBrightnessTimer(**brightnessToBeSet**, Integer.*parseInt*(alarmID));  
 setBrightnessTimer(**brightnessToBeSet**, selectedTime, Integer.*parseInt*(alarmID));  
  
 SharedPreferences.Editor editStorage = settings.edit();  
 editStorage.putInt(alarmID, **brightnessToBeSet**);  
 editStorage.putInt(alarmID + ***SETTINGS\_HOUR***, brightnessTime.getCurrentHour());  
 editStorage.putInt(alarmID + ***SETTINGS\_MINUTES***, brightnessTime.getCurrentMinute());  
 editStorage.commit();  
  
 Intent intent = **new** Intent(getApplicationContext(), BrightTime.**class**);  
 intent.addFlags(Intent.***FLAG\_ACTIVITY\_CLEAR\_TASK***);  
 startActivity(intent);  
 }  
 });  
  
 removeEdit.setOnClickListener(**new** View.OnClickListener(){  
 **public void** onClick(View v){  
 Calendar selectedTime = Calendar.*getInstance*();  
 selectedTime.set(Calendar.***MINUTE***, brightnessTime.getCurrentMinute());  
 selectedTime.set(Calendar.***HOUR\_OF\_DAY***, brightnessTime.getCurrentHour());  
 selectedTime.set(Calendar.***SECOND***,0);  
  
 removeBrightnessTimer(**brightnessToBeSet**, Integer.*parseInt*(alarmID));  
  
 Set<String> pointNames = settings.getStringSet(***alarmNames***,**temp** );  
  
 Set<String> \_pointNames = pointNames;  
 \_pointNames.remove(alarmID);  
 SharedPreferences.Editor editStorage = settings.edit();  
 editStorage.remove(***alarmNames***);  
 editStorage.putStringSet(***alarmNames***, \_pointNames);  
 editStorage.remove(alarmID);  
 editStorage.remove(alarmID + ***SETTINGS\_HOUR***);  
 editStorage.remove(alarmID + ***SETTINGS\_MINUTES***);  
 editStorage.commit();  
 Intent intent = **new** Intent(getApplicationContext(), BrightTime.**class**);  
 intent.addFlags(Intent.***FLAG\_ACTIVITY\_CLEAR\_TASK***);  
 startActivity(intent);  
 }  
 });  
 }  
 **public void** removeBrightnessTimer(**int** userinputBrightness, **int** alarmID){  
 **alarmgr** = (AlarmManager) *mContext*.getSystemService(Context.***ALARM\_SERVICE***);  
 Intent brightnessIntent = **new** Intent(editPoint.**this**, BrightTimeService.**class**);  
 String temp = Integer.*toString*(userinputBrightness);  
 brightnessIntent.setData(Uri.*parse*(temp));  
 PendingIntent setBrightness = PendingIntent.*getService*(editPoint.**this**,alarmID,  
 brightnessIntent, PendingIntent.***FLAG\_UPDATE\_CURRENT***);  
 **alarmgr**.cancel(setBrightness);  
 }  
 **public void** setBrightnessTimer(**int** userinputBrightness, Calendar userinputTimeset, **int** alarmID){  
 **alarmgr** = (AlarmManager) *mContext*.getSystemService(Context.***ALARM\_SERVICE***);  
 Intent brightnessIntent = **new** Intent(editPoint.**this**, BrightTimeService.**class**);  
 String temp = Integer.*toString*(userinputBrightness);  
 brightnessIntent.setData(Uri.*parse*(temp));  
 PendingIntent setBrightness = PendingIntent.*getService*(editPoint.**this**,alarmID,  
 brightnessIntent, PendingIntent.***FLAG\_UPDATE\_CURRENT***);  
 **alarmgr**.setRepeating(AlarmManager.***RTC***, userinputTimeset.getTimeInMillis(), AlarmManager.***INTERVAL\_DAY***, setBrightness);  
 }  
}

**Output:-**

****