

## MCC MINI PROJECT REPORT

**Title:**        **Android app for mobile- Smart Calendar**

**Team members:**

**1. CRYSTAL CUTHINHO (7053)**

**2. LENHERD OLIVERA (7090)**

**Abstract:**

The project will be developed in the ANDROID studio development environment. Coding languages like java will be used. The basic objective of this project is to create an attractive calendar interface for the ANDROID operating system, which will include a *database* that will store all the events and an alarm that'll ring at the appropriate time specified by the user. The user can add multiple events, the events are just like reminders that can be added on any given date and time where the time and the date is specified by the user.

**Introduction:**

The app essentially uses the Alarm system service provided by Android to set alarms and Notification system service for notifications. The other highlight of the app is its simple yet elegant user interface which you will learn to design. One additional feature includes the datepicker service provided by the android that allows the user to select the date. The main screen is implemented using ListView with a custom layout for the rows. And the new reminder screen uses a ViewSwitcher and few other Android widgets.

The user initially selects the date via the date picker, then the user can select the date and the time for the reminder i.e the event . The event will be alarmed at a given time on the notification bar . The user at the same time can also edit the preferences in the

preferences screen which includes features like date format & time format.

## Code:

activity\_date\_picker.xml

```
<?xml version="1.0" encoding="utf"?>
<android.support.design.widget.CoordinatorLayout
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:fitsSystemWindows="true"
tools:context="com.approx.smartcalendar.DatePickerActivity">
<android.support.design.widget.AppBarLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:theme="@style/AppTheme.AppBarOverlay">
    <android.support.v7.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="?attr/colorPrimary"
        app:popupTheme="@style/AppTheme.PopupOverlay" />
    </android.support.design.widget.AppBarLayout>
    <include layout="@layout/content_main" />
    <!--
    <android.support.design.widget.FloatingActionButton
        android:id="@+id/fab"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom|end"
        android:src="@android:drawable/ic_dialog_email"
        android:layout_margin="@dimen/fab_margin"/>-->
</android.support.design.widget.CoordinatorLayout>
```

**main.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
```

```

android:orientation="vertical"
android:background="@drawable/bg_repeat" >
<RelativeLayout
    android:id="@+id/menu_bar"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:background="#5c755e" >
    <TextView
        android:id="@+id/heading_tv"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:padding="5dip"
        android:shadowDx="0"
        android:shadowDy="0"
        android:shadowRadius="1"
        android:shadowColor="#333333"
        android:textColor="#d3e397"
        android:textSize="18sp"
        android:text="Smart Calendar" />
    <ImageButton
        android:id="@+id/imageButton1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:paddingTop="5dip"
        android:paddingBottom="5dip"
        android:paddingLeft="5dip"
        android:paddingRight="10dip"
        android:onClick="onClick"
        android:background="@null"
        android:src="@drawable/preferences" />

    <ImageButton
        android:id="@+id/imageButton2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_toLeftOf="@id/imageButton1"
        android:paddingTop="5dip"
        android:paddingBottom="5dip"
        android:paddingLeft="10dip"
        android:paddingRight="5dip"
        android:onClick="onClick"
        android:background="@null"
        android:src="@drawable/add" />
</RelativeLayout>

```

```

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:background="@drawable/range"
    android:gravity="center_vertical" >
    <ImageButton
        android:id="@+id/imageButton3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingTop="5dip"
        android:paddingBottom="5dip"
        android:paddingLeft="5dip"
        android:paddingRight="10dip"
        android:onClick="onClick"
        android:background="@null"
        android:src="@drawable/left" />

    <TextView
        android:id="@+id/range_tv"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="12sp"
        android:textColor="#ffffff" />

    <ImageButton
        android:id="@+id/imageButton4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingTop="5dip"
        android:paddingBottom="5dip"
        android:paddingLeft="10dip"
        android:paddingRight="5dip"
        android:onClick="onClick"
        android:background="@null"
        android:src="@drawable/right" />

</LinearLayout>

<LinearLayout
    android:id="@+id/list_view"
    android:layout_width="fill_parent"
    android:layout_height="0dip"
    android:layout_weight="1"
    android:padding="5dip"
    android:orientation="vertical" >

    <ListView android:id="@android:id/list"
        android:layout_width="fill_parent"

```

```

        android:layout_height="fill_parent"
        android:divider="@drawable/line"
        android:dividerHeight="3dip"

        android:listSelector="@android:color/transparent"
        android:cacheColorHint="#00000000"
        android:drawSelectorOnTop="false" />

        <TextView android:id="@android:id/empty"
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            android:gravity="center"
            android:textColor="#777777"
            android:text="No data" />

    </LinearLayout>

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:paddingRight="3dip"
        android:background="#5c755e"
        android:gravity="right"
        android:textColor="#3d4935"
        android:textSize="10sp"
        android:textStyle="italic"
        android:text="(c) appsrox.com" />
</LinearLayout>

```

### AddAlarmActivity.java

```

package com.approx.smartcalendar;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import android.app.Activity;
import android.app.DatePickerDialog;
import android.app.Dialog;
import android.app.TimePickerDialog;
import android.content.Intent;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.CheckBox;
import android.widget.DatePicker;

```

```

import android.widget.EditText;
import android.widget.RadioGroup;
import android.widget.RadioGroup.OnCheckedChangeListener;
import android.widget.RelativeLayout;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.TimePicker;
import android.widget.Toast;
import android.widget.ViewSwitcher;
import com.appsrox.smartcalendar.model.Alarm;
import com.appsrox.smartcalendar.model.AlarmMsg;
import com.appsrox.smartcalendar.model.AlarmTime;
import com.appsrox.smartcalendar.model.DbHelper;
/**
 * @author appsrox.com
 *
 */
public class AddAlarmActivity extends Activity {
//    private static final String TAG = "AddAlarmActivity";
    private EditText msgEdit;
    private CheckBox soundCb;
    private DatePicker datePicker;
    private TimePicker timePicker;
    private TextView fromdateText;
    private TextView todaterText;
    private TextView attimeText;
    private ViewSwitcher vs;
    private RadioGroup rg;
    private RelativeLayout rl3;
    private RelativeLayout rl4;
    private Spinner spinner1;
    private Spinner spinner2;
    private Spinner spinner3;
    private EditText minsEdit;
    private EditText hoursEdit;
    private EditText daysEdit;
    private EditText monthsEdit;
    private EditText yearsEdit;
    private SQLiteDatabase db;
    private static final int DIALOG_FROMDATE = 1;
    private static final int DIALOG_TODATE = 2;
    private static final int DIALOG_ATTIME = 3;
    private static final SimpleDateFormat sdf = new
SimpleDateFormat();
    private AdapterView.OnItemClickListener spinnerListener =
new OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parentView,

```

```

View selectedItemView, int position, long id) {
    if (spinner1.getSelectedItemPosition() > 0 &&
spinner2.getSelectedItemPosition() > 0)
        spinner1.setSelection(0);
    }
    @Override
    public void onNothingSelected(AdapterView<?> parentView)
{
    }
};
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setTitle("New Reminder");
    setContentView(R.layout.add);
    findViews();
    db = RemindMe.db;
    rg.setOnCheckedChangeListener(new
OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(RadioGroup group, int
checkedId) {
        switch(checkedId) {
            case R.id.radio0:
                r13.setVisibility(View.VISIBLE);
                r14.setVisibility(View.GONE);
                break;
            case R.id.radio1:
                r14.setVisibility(View.VISIBLE);
                r13.setVisibility(View.GONE);
                break;
        }
    }
});
    spinner1.setOnItemSelectedListener(spinnerListener);
    spinner2.setOnItemSelectedListener(spinnerListener);
}
@Override
protected void onSaveInstanceState(Bundle outState) {
    super.onSaveInstanceState(outState);
    outState.putInt("vs", vs.getDisplayedChild());
    // outState.putInt("date", datePicker.getDayOfMonth());
    // outState.putInt("month", datePicker.getMonth());
    // outState.putInt("year", datePicker.getYear());
    outState.putInt("hour", timePicker.getCurrentHour());
    // outState.putInt("min", timePicker.getCurrentMinute());
    outState.putCharSequence("fromdate",
fromdateText.getText());
    outState.putCharSequence("todate", todateText.getText());
}

```

```

        outState.putCharSequence("attime", attimeText.getText());
    }
    @Override
    protected void onRestoreInstanceState(Bundle state) {
        super.onRestoreInstanceState(state);
        vs.setDisplayedChild(state.getInt("vs"));
        // datePicker.updateDate(state.getInt("year"),
state.getInt("month"), state.getInt("date"));
        timePicker.setCurrentHour(state.getInt("hour"));
        // timePicker.setCurrentMinute(state.getInt("min"));
        fromdateText.setText(state.getCharSequence("fromdate"));
        todatetimeText.setText(state.getCharSequence("todatetime"));
        attimeText.setText(state.getCharSequence("attime"));
    }
    @Override
    protected void onResume() {
        super.onResume();
        sdf.applyPattern(RemindMe.getDateFormat());
    }
    private void findViews() {
        msgEdit = (EditText) findViewById(R.id.msg_et);
        soundCb = (CheckBox) findViewById(R.id.sound_cb);
        datePicker = (DatePicker) findViewById(R.id.datePicker);
        timePicker = (TimePicker) findViewById(R.id.timePicker);
        fromdateText = (TextView) findViewById(R.id.fromdate_tv);
        todatetimeText = (TextView) findViewById(R.id.todatetime_tv);
        attimeText = (TextView) findViewById(R.id.attime_tv);
        vs = (ViewSwitcher) findViewById(R.id.view_switcher);
        rg = (RadioGroup) findViewById(R.id.radioGroup);
        rl3 = (RelativeLayout)
findViewById(R.id.relativeLayout3);
        rl4 = (RelativeLayout)
findViewById(R.id.relativeLayout4);
        spinner1 = (Spinner) findViewById(R.id.spinner1);
        spinner2 = (Spinner) findViewById(R.id.spinner2);
        spinner3 = (Spinner) findViewById(R.id.spinner3);
        minsEdit = (EditText) findViewById(R.id.mins_et);
        hoursEdit = (EditText) findViewById(R.id.hours_et);
        daysEdit = (EditText) findViewById(R.id.days_et);
        monthsEdit = (EditText) findViewById(R.id.months_et);
        yearsEdit = (EditText) findViewById(R.id.years_et);
    }
    private boolean validate() {
        if (TextUtils.isEmpty(msgEdit.getText())) {
            msgEdit.requestFocus();
            Toast.makeText(this, "Enter a message",
Toast.LENGTH_SHORT).show();
            return false;
        }
    }

```



```

        if (vs.getDisplayedChild() == 1) {
            if (TextUtils.isEmpty(fromdateText.getText())) {
                Toast.makeText(this, "Specify from date",
Toast.LENGTH_SHORT).show();
                return false;
            }
            if (TextUtils.isEmpty(todateText.getText())) {
                Toast.makeText(this, "Specify to date",
Toast.LENGTH_SHORT).show();
                return false;
            }
            try {
                if
(sdf.parse(fromdateText.getText().toString()).after(sdf.parse(todateT
ext.getText().toString()))) {
                    Toast.makeText(this, "From date is
after To date!", Toast.LENGTH_SHORT).show();
                    return false;
                }
            } catch (ParseException e) {}
            if (TextUtils.isEmpty(attimeText.getText())) {
                Toast.makeText(this, "Specify time",
Toast.LENGTH_SHORT).show();
                return false;
            }
        }
        return true;
    }
    /* Save */
    public void create(View v) {
        if (!validate()) return;
        Alarm alarm = new Alarm();
        alarm.setName(msgEdit.getText().toString());
        alarm.setSound(soundCb.isChecked());
        AlarmTime alarmTime = new AlarmTime();
        long alarmId = 0;
        switch(vs.getDisplayedChild()) {
            case 0: //one time

                alarm.setFromDate(DbHelper.getDateStr(datePicker.getYear(),
datePicker.getMonth()+1, datePicker.getDayOfMonth()));
                alarmId = alarm.persist(db);

                alarmTime.setAt(DbHelper.getTimeStr(timePicker.getCurrentHour()
, timePicker.getCurrentMinute()));
                alarmTime.setAlarmId(alarmId);
                alarmTime.persist(db);
                break;
            case 1: //repeating

```

```

        alarm.setFromDate(Util.toPersistentDate(fromdateText.getText().
toString(), sdf));

        alarm.setToDate(Util.toPersistentDate(todateText.getText().toSt
ring(), sdf));

        switch(rg.getCheckedRadioButtonId()) {
            case R.id.radio0: //rule

                alarm.setRule(Util.concat(spinner1.getSelectedItemPosition(), "
",

                spinner2.getSelectedItemPosition(), " ",

                spinner3.getSelectedItemPosition()));
                break;
            case R.id.radio1: //interval

                alarm.setInterval(Util.concat(minsEdit.getText(), " ",
                hoursEdit.getText(), "
",
                daysEdit.getText(), "
",
                monthsEdit.getText(),
                " ",
                yearsEdit.getText()));
                break;
        }
        alarmId = alarm.persist(db);

        alarmTime.setAt(Util.toPersistentTime(attimeText.getText().toSt
ring()));

        alarmTime.setAlarmId(alarmId);
        alarmTime.persist(db);
        break;
    }
    Intent service = new Intent(this, AlarmService.class);
    service.putExtra(AlarmMsg.COL_ALARMID,
String.valueOf(alarmId));
    service.setAction(AlarmService.POPULATE);
    startService(service);
    finish();
}

public void onClick(View v) {
    switch (v.getId()) {
        case R.id.toggleButton1:
            vs.showNext();
            break;
        case R.id.fromdate_tv:

```

```

        case R.id.fromdate_lb:
            showDialog(DIALOG_FROMDATE);
            break;
        case R.id.todate_tv:
        case R.id.todate_lb:
            showDialog(DIALOG_TODATE);
            break;
        case R.id.attime_tv:
        case R.id.attime_lb:
            showDialog(DIALOG_ATTIME);
            break;
    }
}

@Override
protected Dialog onCreateDialog(final int id) {
    Calendar cal = Calendar.getInstance();
    switch(id) {
        case DIALOG_ATTIME:
            TimePickerDialog.OnTimeSetListener
mTimeSetListener =
                new
TimePickerDialog.OnTimeSetListener() {
                    public void
onTimeSet(TimePicker view, int hourOfDay, int minute) {
                        attimeText.setText(Util.getActualTime(hourOfDay, minute));
                    }
                };
            return new TimePickerDialog(this,
mTimeSetListener, cal.get(Calendar.HOUR_OF_DAY),
cal.get(Calendar.MINUTE), RemindMe.is24Hours());
        case DIALOG_FROMDATE:
        case DIALOG_TODATE:
            DatePickerDialog.OnDateSetListener
dateListener =
                new
DatePickerDialog.OnDateSetListener() {
                    public void
onDateSet(DatePicker view, int year, int monthOfYear, int dayOfMonth)
{
                        String txt =
DbHelper.getDateStr(year, monthOfYear+1, dayOfMonth);
                        try {
                            txt =
sdf.format(DbHelper.sdf.parse(txt));
                        } catch
(ParseException e) {}
                    }
                };
            if (id ==
DIALOG_FROMDATE) {

```

```

        fromdateText.setText(txt);
    } else if (id ==
DIALOG_TODATE) {
        todateText.setText(txt);
    }
    };
    return new DatePickerDialog(this,
dateListener, cal.get(Calendar.YEAR), cal.get(Calendar.MONTH),
cal.get(Calendar.DATE));
    }
    return super.onCreateDialog(id);
}
@Override
protected void onPrepareDialog(int id, Dialog dialog) {
    super.onPrepareDialog(id, dialog);
    switch(id) {
        case DIALOG_ATTIME:
            if (!TextUtils.isEmpty(attimeText.getText()))
{
                String[] arr =
Util.toPersistentTime(attimeText.getText().toString()).split(":");

                ((TimePickerDialog)dialog).updateTime(Integer.parseInt(arr[0]),
Integer.parseInt(arr[1]));
            }
            break;
        case DIALOG_FROMDATE:
            if
(!TextUtils.isEmpty(fromdateText.getText())) {
                String[] arr =
Util.toPersistentDate(fromdateText.getText().toString(),
sdf).split("-");

                ((DatePickerDialog)dialog).updateDate(Integer.parseInt(arr[0]),
Integer.parseInt(arr[1])-1, Integer.parseInt(arr[2]));
            }
            break;
        case DIALOG_TODATE:
            if (!TextUtils.isEmpty(todateText.getText()))
{
                String[] arr =
Util.toPersistentDate(todateText.getText().toString(), sdf).split("-
");

                ((DatePickerDialog)dialog).updateDate(Integer.parseInt(arr[0]),
Integer.parseInt(arr[1])-1, Integer.parseInt(arr[2]));
            }
    }
}

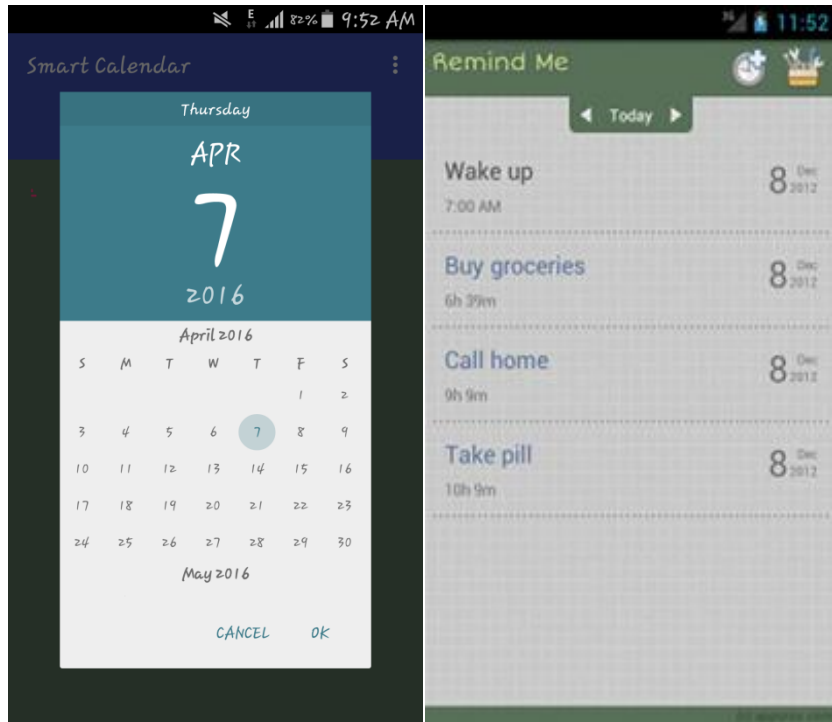
```

```

    }
    break;
}
}
}

```

## Screenshots:



New Reminder

Message

☐ Play ringtone

Once Repeat

+ + +

Dec 07 2012

- - -

+ +

11 38 AM

- -

✓

Remind Me

Display Settings

Display Time

Remaining

Date Range

Daily

Date Format

MMMM dd yyyy

Time Format

Use 24-hour format ☐

Notification Settings

Vibrate

Vibrate on notification ☒

Set Ringtone

Default