

```
1  package edu.seaaddicts.brockbutler.scheduler;
2
3  import java.text.SimpleDateFormat;
4  import java.util.ArrayList;
5  import java.util.Calendar;
6  import java.util.Date;
7  import java.util.Locale;
8
9  import android.app.Activity;
10 import android.app.DatePickerDialog;
11 import android.app.Dialog;
12 import android.os.Bundle;
13 import android.util.Log;
14 import android.view.View;
15 import android.view.View.OnClickListener;
16 import android.view.WindowManager;
17 import android.widget.AdapterView;
18 import android.widget.Button;
19 import android.widget.DatePicker;
20 import android.widget.EditText;
21 import android.widget.Spinner;
22 import android.widget.TextView;
23 import edu.seaaddicts.brockbutler.R;
24 import edu.seaaddicts.brockbutler.coursemanager.Course;
25 import edu.seaaddicts.brockbutler.coursemanager.CourseHandler;
26
27 public class AddTaskActivity extends Activity {
28     private static final String TAG = "AddTaskActivity";
29     private static final int DATE_DIALOG_ID = 0;
30
31     private Button mDueDateButton;
32     private Button mSaveButton;
33     private Button mCancelButton;
34
35     private ArrayList<Course> mRegCourses;
36     private TextView mDueDateTextView;
37
38     private EditText mTaskTitle;
39     private EditText mTaskWeight;
40     private EditText mTaskBase;
41     private EditText mTaskMark;
42
43     private Spinner mCourseSpinner;
44     private Spinner mPrioritySpinner;
45
46     private CourseHandler mCourseHandle = null;
47
48     private int mYear;
49     private int mMonth;
50     private int mDay;
51
52     @Override
53     protected void onCreate(Bundle savedInstanceState) {
54         super.onCreate(savedInstanceState);
55         setContentView(R.layout.activity_add_task);
56         this.getWindow().setSoftInputMode(
57             WindowManager.LayoutParams.SOFT_INPUT_STATE_ALWAYS_HIDDEN);
58         init();
59     }
60
61     /*
62     * Initialize all views and sets Button OnClickListener.
63     */
64     private void init() {
65         /*
66         * CourseHandler
67         */
68         mCourseHandle = new CourseHandler(this);
69
70         /*
71         * Buttons
```

```

72      */
73      mDueDateButton = (Button) findViewById(R.id.add_task_due_date_button);
74      mSaveButton = (Button) findViewById(R.id.add_task_save_button);
75      mCancelButton = (Button) findViewById(R.id.add_task_cancel_button);
76
77      /*
78       * TextViews
79       */
80      mDueDateTextView = (TextView) findViewById(R.id.add_task_date_textview);
81
82      /*
83       * Spinners
84       */
85      mPrioritySpinner = (Spinner) findViewById(R.id.add_task_priority_spinner);
86      mCourseSpinner = (Spinner) findViewById(R.id.add_task_course_spinner);
87
88      /*
89       * EditTexts
90       */
91      mTaskTitle = (EditText) findViewById(R.id.add_task_title);
92      mTaskWeight = (EditText) findViewById(R.id.task_weight);
93      mTaskBase = (EditText) findViewById(R.id.task_base);
94      mTaskMark = (EditText) findViewById(R.id.task_mark);
95
96      mRegCourses = mCourseHandle.getRegisteredCourses();
97      ArrayList<String> cs = new ArrayList<String>();
98
99      /*
100       * ArrayList of Course Strings in format: SUBJECT CODE
101       */
102      for (int i = 0; i < mRegCourses.size(); i++) {
103          cs.add(mRegCourses.get(i).mSubject + " " + mRegCourses.get(i).mCode);
104      }
105
106      /*
107       * Set the course Spinner
108       */
109      mCourseSpinner.setAdapter(new ArrayAdapter<String>(this,
110          android.R.layout.simple_spinner_dropdown_item, cs));
111
112      /*
113       * DatePicker stuff.
114       */
115      final Calendar cal = Calendar.getInstance();
116      mYear = cal.get(Calendar.YEAR);
117      mMonth = cal.get(Calendar.MONTH);
118      mDay = cal.get(Calendar.DAY_OF_MONTH);
119
120      /*
121       * OnClickListener
122       */
123
124      // Deprecated but easier than alternative.
125      mDueDateButton.setOnClickListener(new OnClickListener() {
126          @SuppressWarnings("deprecation")
127          public void onClick(View v) {
128              showDialog(DATE_DIALOG_ID);
129          }
130      });
131
132      mSaveButton.setOnClickListener(new OnClickListener() {
133          public void onClick(View v) {
134              Task t = new Task();
135              t.mSubj = mRegCourses.get(mCourseSpinner
136                  .getSelectedItemPosition()).mSubject;
137              t.mCode = mRegCourses.get(mCourseSpinner
138                  .getSelectedItemPosition()).mCode;
139              t.mName = mTaskTitle.getText().toString();
140              // Lower == higher priority
141              t.mPriority = mPrioritySpinner.getSelectedItemPosition();
142              t.mDueDate = mDueDateTextView.getText().toString();

```

```

143
144     try {
145         t.mWeight = Float.parseFloat(mTaskWeight.getText()
146             .toString());
147         t.mBase = Float.parseFloat(mTaskBase.getText().toString());
148         t.mMark = Float.parseFloat(mTaskMark.getText().toString());
149     } catch (NumberFormatException e) {
150         Log.e(TAG,
151             "It is OK, we just had a blank field when parsing a float.");
152     }
153
154     // Get current date
155     SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd", Locale.
156         CANADA);
157     Date date = new Date();
158     t.mCreationDate = dateFormat.format(date);
159     mCourseHandle.addTask(t);
160     onBackPressed();
161 }
162 }};
163
164 mCancelButton.setOnClickListener(new OnClickListener() {
165     public void onClick(View v) {
166         // Do nothing.
167         onBackPressed();
168     }
169 });
170
171 /**
172  * (non-Javadoc)
173  *
174  * @see android.app.Activity#onCreateDialog(int)
175  *
176  * Overridden method to handle multiple dialogs that may be created
177  * from this activity.
178  */
179 @Override
180 protected Dialog onCreateDialog(int id) {
181     switch (id) {
182         case DATE_DIALOG_ID:
183             return new DatePickerDialog(this, mDatePickerListener, mYear,
184                 mMonth, mDay);
185     }
186     return null;
187 }
188
189 /**
190  * Retrieves date set from DatePickerDialog and sets TextView in this
191  * activity.
192  */
193 private DatePickerDialog.OnDateSetListener mDatePickerListener = new
194     DatePickerDialog.OnDateSetListener() {
195
196     public void onDateSet(DatePicker view, int selectedYear,
197         int selectedMonth, int selectedDay) {
198         mYear = selectedYear;
199         mMonth = selectedMonth + 1; // Since index starts at 0 and there is
200             // no 0th month.
201         mDay = selectedDay;
202
203         // Set TextView in this activity.
204         if (mMonth < 10) {
205             if (mDay < 10) {
206                 mDueDateTextView.setText(new StringBuilder().append(mYear)
207                     .append("/").append(0).append(mMonth).append("/")
208                     .append(0).append(mDay));
209             } else
210                 mDueDateTextView.setText(new StringBuilder().append(mYear)
211                     .append("/").append(0).append(mMonth).append("/")
212                     .append(mDay));

```

```
212     } else if (mMonth > 9) {
213         if (mDay < 10) {
214             mDueDateTextView.setText(new StringBuilder().append(mYear)
215                                     .append("/").append(mMonth).append("/").append(0)
216                                     .append(mDay));
217         } else
218             mDueDateTextView.setText(new StringBuilder().append(mYear)
219                                     .append("/").append(mMonth).append("/")
220                                     .append(mDay));
221     }
222
223 }
224 };
225
226 }
227
```

```
1  package edu.seaaddicts.brockbutler.scheduler;
2
3  import java.text.SimpleDateFormat;
4  import java.util.ArrayList;
5  import java.util.Date;
6  import java.util.Locale;
7
8  import android.app.Activity;
9  import android.app.DatePickerDialog;
10 import android.app.Dialog;
11 import android.os.Bundle;
12 import android.util.Log;
13 import android.view.View;
14 import android.view.View.OnClickListener;
15 import android.view.WindowManager;
16 import android.widget.AdapterView;
17 import android.widget.Button;
18 import android.widget.DatePicker;
19 import android.widget.EditText;
20 import android.widget.Spinner;
21 import android.widget.TextView;
22 import edu.seaaddicts.brockbutler.R;
23 import edu.seaaddicts.brockbutler.coursemanager.Course;
24 import edu.seaaddicts.brockbutler.coursemanager.CourseHandler;
25
26 public class ModifyTaskActivity extends Activity {
27     private static final String TAG = "ModifyTaskActivity";
28     private static final int DATE_DIALOG_ID = 0;
29
30     private Button mDueDateButton;
31     private Button mSaveButton;
32     private Button mCancelButton;
33
34     private ArrayList<Course> mRegCourses;
35     private TextView mDueDateTextView;
36
37     private EditText mTaskTitle;
38     private EditText mTaskWeight;
39     private EditText mTaskBase;
40     private EditText mTaskMark;
41
42     private Spinner mCourseSpinner;
43     private Spinner mPrioritySpinner;
44
45     private CourseHandler mCourseHandle = null;
46
47     private int mYear;
48     private int mMonth;
49     private int mDay;
50     private int mAssign;
51     private int mIsDone;
52
53     @Override
54     protected void onCreate(Bundle savedInstanceState) {
55         super.onCreate(savedInstanceState);
56         setContentView(R.layout.activity_add_task);
57         this.getWindow().setSoftInputMode(WindowManager.LayoutParams.
58             SOFT_INPUT_STATE_ALWAYS_HIDDEN);
59         init();
60     }
61     /*
62     * Initialize all views and sets Button OnClickListener.
63     */
64     private void init() {
65         Bundle bundle = this getIntent().getExtras();
66         mAssign = bundle.getInt("assign");
67         mIsDone = bundle.getInt("is_done");
68         /*
69         * CourseHandler
70         */
```

```

71     mCourseHandle = new CourseHandler(this);
72
73     /*
74     * Buttons
75     */
76     mDueDateButton = (Button) findViewById(R.id.add_task_due_date_button);
77     mSaveButton = (Button) findViewById(R.id.add_task_save_button);
78     mCancelButton = (Button) findViewById(R.id.add_task_cancel_button);
79
80     /*
81     * TextViews
82     */
83     mDueDateTextView = (TextView) findViewById(R.id.add_task_date_textview);
84
85     /*
86     * Spinners
87     */
88     mPrioritySpinner = (Spinner) findViewById(R.id.add_task_priority_spinner);
89     mPrioritySpinner.setSelection(bundle.getInt("priority"));
90     mCourseSpinner = (Spinner) findViewById(R.id.add_task_course_spinner);
91
92     /*
93     * EditTexts
94     */
95     mTaskTitle = (EditText) findViewById(R.id.add_task_title);
96     mTaskTitle.setText(bundle.getString("title"));
97     mTaskWeight = (EditText) findViewById(R.id.task_weight);
98     mTaskWeight.setText(bundle.getFloat("weight")+"");
99     mTaskBase = (EditText) findViewById(R.id.task_base);
100    mTaskBase.setText(bundle.getFloat("base")+"");
101    mTaskMark = (EditText) findViewById(R.id.task_mark);
102    mTaskMark.setText(bundle.getFloat("mark")+"");
103
104    mRegCourses = mCourseHandle.getRegisteredCourses();
105    ArrayList<String> cs = new ArrayList<String>();
106
107    /*
108    * ArrayList of Course Strings in format: SUBJECT CODE
109    */
110    String course = bundle.getString("course");
111    int pos = 0;
112    for (int i = 0; i < mRegCourses.size(); i++) {
113        String c = mRegCourses.get(i).mSubject + " " + mRegCourses.get(i).mCode;
114        cs.add(c);
115        if (c.equals(course))
116            pos = i;
117    }
118
119    /*
120    * Set the course Spinner
121    */
122    mCourseSpinner.setAdapter(new ArrayAdapter<String>(this,
123        android.R.layout.simple_spinner_dropdown_item, cs));
124
125    mCourseSpinner.setSelection(pos);
126
127    /*
128    * DatePicker stuff.
129    */
130    mYear = bundle.getInt("year");
131    mMonth = bundle.getInt("month");
132    mDay = bundle.getInt("day");
133
134    /*
135    * OnClickListener
136    */
137
138    // Deprecated but easier than alternative.
139    mDueDateButton.setOnClickListener(new OnClickListener() {
140        @SuppressWarnings("deprecation")
141        public void onClick(View v) {

```

```

142         showDialog(DATE_DIALOG_ID);
143     }
144 });
145
146 mSaveButton.setOnClickListener(new OnClickListener() {
147     public void onClick(View v) {
148         Task t = new Task();
149         t.mAssign = mAssign;
150         t.mIsDone = mIsDone;
151         t.mSubj = mRegCourses.get(mCourseSpinner
152             .getSelectedItemPosition()).mSubject;
153         t.mCode = mRegCourses.get(mCourseSpinner
154             .getSelectedItemPosition()).mCode;
155         t.mName = mTaskTitle.getText().toString();
156         // Lower == higher priority
157         t.mPriority = mPrioritySpinner.getSelectedItemPosition();
158         t.mDueDate = mDueDateTextView.getText().toString();
159
160         try {
161             t.mWeight = Float.parseFloat(mTaskWeight.getText().toString());
162             t.mBase = Float.parseFloat(mTaskBase.getText().toString());
163             t.mMark = Float.parseFloat(mTaskMark.getText().toString());
164         } catch (NumberFormatException e) {
165             Log.e(TAG, "It is OK, we just had a blank field when parsing a float.");
166         }
167
168         // Get current date
169         SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd", Locale.
170             CANADA);
171         Date date = new Date();
172         t.mCreationDate = dateFormat.format(date);
173         mCourseHandle.addTask(t);
174         onBackPressed();
175     }
176 });
177
178 mCancelButton.setOnClickListener(new OnClickListener() {
179     public void onClick(View v) {
180         // Do nothing.
181         onBackPressed();
182     }
183 });
184
185 setDueDateText();
186 }
187
188 /**
189  * (non-Javadoc)
190  * @see android.app.Activity#onCreateDialog(int)
191  *
192  * Overridden method to handle multiple dialogs that may be created
193  * from this activity.
194  */
195 @Override
196 protected Dialog onCreateDialog(int id) {
197     switch (id) {
198         case DATE_DIALOG_ID:
199             return new DatePickerDialog(this, mDatePickerListener, mYear,
200                 mMonth, mDay);
201     }
202     return null;
203 }
204
205 /**
206  * Retrieves date set from DatePickerDialog and sets TextView in this
207  * activity.
208  */
209 private DatePickerDialog.OnDateSetListener mDatePickerListener = new
210     DatePickerDialog.OnDateSetListener() {

```

```
211     public void onDateSet(DatePicker view, int selectedYear,
212         int selectedMonth, int selectedDay) {
213         mYear = selectedYear;
214         mMonth = selectedMonth + 1; // Since index starts at 0 and there is
215             // no 0th month.
216         mDay = selectedDay;
217         setDueDateText();
218     }
219 }
220 };
221
222 private void setDueDateText(){
223     // Set TextView in this activity.
224     if (mMonth < 10) {
225         if (mDay < 10) {
226             mDueDateTextView.setText(new StringBuilder().append(mYear)
227                 .append("/").append(0).append(mMonth).append("/")
228                 .append(0).append(mDay));
229         } else
230             mDueDateTextView.setText(new StringBuilder().append(mYear)
231                 .append("/").append(0).append(mMonth).append("/")
232                 .append(mDay));
233     } else if (mMonth > 9) {
234         if (mDay < 10) {
235             mDueDateTextView.setText(new StringBuilder().append(mYear)
236                 .append("/").append(mMonth).append("/").append(0)
237                 .append(mDay));
238         } else
239             mDueDateTextView.setText(new StringBuilder().append(mYear)
240                 .append("/").append(mMonth).append("/")
241                 .append(mDay));
242     }
243 }
244 }
245
```



```
1 package edu.seaaddicts.brockbutler.scheduler;
2
3 import java.util.ArrayList;
4 import java.util.Calendar;
5
6 import android.app.Activity;
7 import android.app.AlertDialog;
8 import android.content.Context;
9 import android.content.DialogInterface;
10 import android.content.Intent;
11 import android.graphics.Paint;
12 import android.net.Uri;
13 import android.os.Bundle;
14 import android.util.Log;
15 import android.view.Menu;
16 import android.view.MenuItem;
17 import android.view.View;
18 import android.view.View.OnClickListener;
19 import android.view.ViewGroup;
20 import android.widget.AdapterView;
21 import android.widget.AdapterView.Adapter;
22 import android.widget.AdapterView.OnItemClickListener;
23 import android.widget.AdapterView.OnItemSelectedListener;
24 import android.widget.AdapterView.OnItemClickListener;
25 import android.widget.AdapterView.OnItemSelectedListener;
26 import android.widget.AdapterView.OnItemClickListener;
27 import android.widget.AdapterView.OnItemSelectedListener;
28 import android.widget.AdapterView.OnItemClickListener;
29 import android.widget.AdapterView.OnItemSelectedListener;
30 import android.widget.AdapterView.OnItemClickListener;
31 import android.widget.AdapterView.OnItemSelectedListener;
32 import edu.seaaddicts.brockbutler.R;
33 import edu.seaaddicts.brockbutler.animation.ExpandAnimation;
34 import edu.seaaddicts.brockbutler.coursemanager.Course;
35 import edu.seaaddicts.brockbutler.coursemanager.CourseHandler;
36 import edu.seaaddicts.brockbutler.coursemanager.Offering;
37 import edu.seaaddicts.brockbutler.coursemanager.OfferingTime;
38 import edu.seaaddicts.brockbutler.help.HelpActivity;
39
40 public class SchedulerActivity extends Activity {
41     private static final String TAG = "SchedulerActivity";
42
43     private static final int VISIBLE = 0;
44     private static final int GONE = 8;
45
46     private ArrayList<Course> mRegisteredCoursesList = null;
47     private CourseHandler mCourseHandle = null;
48     private ListView mRegisteredCoursesListView = null;
49     private ArrayAdapter<String> mListAdaptor;
50     private View mView;
51     private int mCurTaskTextViewId;
52     private int mCurTaskCheckBoxId;
53     private int mCurTaskNum;
54     private ArrayList<Task> mCurTasks;
55
56     @Override
57     protected void onCreate(Bundle savedInstanceState) {
58         super.onCreate(savedInstanceState);
59         setContentView(R.layout.activity_scheduler);
60         mCourseHandle = new CourseHandler(this);
61         populateCoursesLayout();
62     }
63
64     @Override
65     protected void onResume() {
66         super.onResume();
67         populateCoursesLayout();
68     }
69
70     /**
71     *
```

```

72 private void populateCoursesLayout() {
73     TextView tvNoCourses = (TextView) findViewById(R.id.tv_no_courses);
74     mRegisteredCoursesListView = (ListView) findViewById(R.id.sched_list);
75     mRegisteredCoursesList = mCourseHandle.getRegisteredCourses();
76     if (mRegisteredCoursesList.size() == 0) {
77         // There are no registered courses so set message.
78         mRegisteredCoursesListView.setVisibility(GONE);
79         tvNoCourses.setVisibility(VISIBLE);
80     } else {
81         // We have registered courses so populate ListView.
82         // Creating the list adapter and populating the list
83         mListAdaptor = new CustomListAdapter(this, R.layout.sched_list_item);
84         for (int i = 0; i < mRegisteredCoursesList.size(); i++)
85             mListAdaptor.add(mRegisteredCoursesList.get(i).mSubject + " "
86                             + mRegisteredCoursesList.get(i).mCode);
87         mRegisteredCoursesListView.setAdapter(mListAdaptor);
88         tvNoCourses.setVisibility(GONE);
89         mRegisteredCoursesListView.setVisibility(VISIBLE);
90         mRegisteredCoursesListView
91             .setOnClickListener(new AdapterView.OnItemClickListener() {
92                 public void onItemClick(AdapterView<?> parent,
93                     final View view, int position, long id) {
94                     showHideToolbar(view, position);
95                 }
96             });
97         registerContextMenu(mRegisteredCoursesListView);
98     }
99 }
100
101 private void populateCourseView(View view, int position) {
102
103     mView = view;
104     // Get current course info.
105     String offering = "";
106     ArrayList<Offering> offs = mRegisteredCoursesList.get(position).mOfferings;
107     ArrayList<OfferingTime> offTimes;
108     ArrayList<Task> tasks = mRegisteredCoursesList.get(position).mTasks;
109     Log.d(TAG, "NUMBER OF TASKS: " + tasks.size());
110
111     // Set Instructor
112     ((TextView) view.findViewById(R.id.tv_prof_name))
113         .setText(mRegisteredCoursesList.get(position).mInstructor);
114     final String e= mRegisteredCoursesList.get(position).mInstructor_email;
115     ((Button) view.findViewById(R.id.prof_email_button))
116         .setOnClickListener(new OnClickListener() {
117             public void onClick(View v) {
118                 sendEmail(e);
119             }
120         });
121
122     // Add Offerings registered for.
123     for (int i = 0; i < offs.size(); i++) {
124         String what = offs.get(i).mType.substring(0, 3).trim();
125         offTimes = offs.get(i).mOfferingTimes;
126         offering = "";
127
128         // Loop through OfferingTimes for each Offering to populate
129         for (int j = 0; j < offTimes.size(); j++) {
130             offering += offTimes.get(j).mDay + " "
131                     + offTimes.get(j).mStartTime + " - "
132                     + offTimes.get(j).mEndTime + " @ "
133                     + offTimes.get(j).mLocation + "\n";
134
135             // Check for type of Offering and add as appropriate
136             if (what.equalsIgnoreCase("lec")) {
137                 TextView tv = ((TextView) view
138                     .findViewById(R.id.tv_lecture));
139                 tv.setText(offering);
140             }
141
142             // ...a lab?

```

```

143         else if (what.equalsIgnoreCase("lab"))
144             ((TextView) view.findViewById(R.id.tv_lab))
145                 .setText(offering);
146
147         // ..a tutorial?
148         else if (what.equalsIgnoreCase("tut"))
149             ((TextView) view.findViewById(R.id.tv_tutorial))
150                 .setText(offering);
151
152         // ...a seminar?
153         else if (what.equalsIgnoreCase("sem"))
154             ((TextView) view.findViewById(R.id.tv_seminar))
155                 .setText(offering);
156     }
157 }
158
159 /*
160  * Add Tasks to view
161  */
162 if (tasks.size() > 0) {
163     mCurTasks = tasks;
164     // Set visibility for Tasks
165     view.findViewById(R.id.sched_tasks_title).setVisibility(VISIBLE);
166     view.findViewById(R.id.sched_tasks).setVisibility(VISIBLE);
167     ((LinearLayout) view.findViewById(R.id.sched_tasks))
168         .removeAllViews();
169     for (int j = 0; j < tasks.size(); j++) {
170         mCurTaskNum = j;
171         // LinearLayout for each task
172         LinearLayout ll = new LinearLayout(this);
173
174         // TextView for task info
175         TextView tv = new TextView(this);
176
177         ll.addView(tv);
178
179         // Each TextView will need an ID when added to the layout
180         mCurTaskTextViewId = 1000+j;
181         tv.setId(mCurTaskTextViewId);
182
183         // Cosmetics
184         tv.setPadding(20, 20, 30, 20);
185         tv.setLayoutParams(new LinearLayout.LayoutParams(425,
186             LinearLayout.LayoutParams.WRAP_CONTENT));
187
188         // Set the TextView
189         tv.setText(tasks.get(j).mName + "\n\tDue: " + tasks.get(j).mDueDate);
190
191         // Checkbox for Task status
192         CheckBox cb = new CheckBox(this);
193
194         // Buttons for Modify/Delete
195         ImageButton ib_mod = new ImageButton(this);
196         ImageButton ib_del = new ImageButton(this);
197         ib_mod.setFocusable(false);
198         ib_mod.setFocusableInTouchMode(false);
199         ib_mod.setBackgroundResource(android.R.drawable.ic_dialog_info);
200         final Task edit = mCurTasks.get(mCurTaskNum);
201         ib_mod.setOnClickListener(new OnClickListener() {
202
203             public void onClick(View v) {
204                 modifyTask(edit);
205             }
206         });
207         ib_del.setFocusable(false);
208         ib_del.setFocusableInTouchMode(false);
209         ib_del.setBackgroundResource(android.R.drawable.ic_delete);
210         ib_del.setOnClickListener(new OnClickListener() {
211
212             public void onClick(View v) {
213                 displayTaskRemoveDialog(edit);

```

```

214     }
215     });
216     ll.addView(ib_del);
217     ll.addView(ib_mod);
218
219
220     // Each TextView will need an ID when added to the layout
221     mCurTaskCheckBoxId = 1000 + j + 1;
222     cb.setId(mCurTaskCheckBoxId);
223     final Task currentTask = tasks.get(j);
224     final View fView = mView;
225     cb.setOnCheckedChangeListener(new OnCheckedChangeListener() {
226
227         public void onCheckedChanged(CompoundButton buttonView,
228             boolean isChecked) {
229             Log.d(TAG, "mCurTaskTextViewId: " + (buttonView.getId()-1));
230             TextView tv = ((TextView) fView.findViewById(buttonView.getId()-1));
231             if (isChecked) {
232                 tv.setPaintFlags(tv.getPaintFlags()
233                     | Paint.STRIKE_THRU_TEXT_FLAG);
234                 currentTask.mIsDone = 1;
235                 mCourseHandle.addTask(currentTask);
236             } else {
237                 tv.setPaintFlags(0);
238                 currentTask.mIsDone = 0;
239                 mCourseHandle.addTask(currentTask);
240             }
241         }
242     });
243     cb.setPadding(20, 0, 0, 0);
244     cb.setFocusable(false);
245     cb.setFocusableInTouchMode(false);
246     ll.addView(cb);
247     ((LinearLayout) view.findViewById(R.id.sched_tasks)).addView(ll);
248     cb.setChecked(currentTask.mIsDone != 0); //HUZZAH!!!
249 }
250
251
252 float mark = mCourseHandle.getMark(mRegisteredCoursesList.get(position));
253 float base = mCourseHandle.getBase(mRegisteredCoursesList.get(position));
254 float total=0;
255 if(base != 0){
256     total= (mark/base)*100;
257 }
258
259 String grade= ""+(int)mark+"/"+(int)base+" = "+ (int)total+"%";
260 ((TextView) view.findViewById(R.id.course_grade_grade))
261 .setText(grade);
262
263 /*
264  * Hide class type if none available
265  */
266 if (((TextView) view.findViewById(R.id.tv_lecture)).getText().
267     toString().equalsIgnoreCase("none"))
268     view.findViewById(R.id.row_lec).setVisibility(GONE);
269 if (((TextView) view.findViewById(R.id.tv_lab)).getText().toString().
270     equalsIgnoreCase("none"))
271     view.findViewById(R.id.row_lab).setVisibility(GONE);
272 if (((TextView) view.findViewById(R.id.tv_tutorial)).getText().
273     toString().equalsIgnoreCase("none"))
274     view.findViewById(R.id.row_tut).setVisibility(GONE);
275 if (((TextView) view.findViewById(R.id.tv_seminar)).getText().
276     toString().equalsIgnoreCase("none"))
277     view.findViewById(R.id.row_sem).setVisibility(GONE);
278 }
279
280 private void displayTaskRemoveDialog(final Task t) {
281     AlertDialog.Builder editalert = new AlertDialog.Builder(this);
282
283     editalert.setTitle("Delete Task?");
284     editalert.setMessage("Are you sure you wish to delete this task?");

```

```
285
286     editalert.setPositiveButton("Yes",
287         new DialogInterface.OnClickListener() {
288             public void onClick(DialogInterface dialog, int whichButton) {
289                 mCourseHandle.removeTask(t);
290                 populateCoursesLayout();
291             }
292         });
293     editalert.setNegativeButton("No",
294         new DialogInterface.OnClickListener() {
295             public void onClick(DialogInterface dialog, int whichButton) {
296                 // do nothing
297             }
298         });
299
300     editalert.show();
301 }
302
303 private void showHideToolbar(View view, int position) {
304     View toolbar = view.findViewById(R.id.sched_toolbar);
305     populateCourseView(view, position);
306
307     // Creating the expand animation for the item
308     ExpandAnimation expandAni = new ExpandAnimation(toolbar,
309         ExpandAnimation.ANIMATE_SHORT);
310
311     // Start the animation on the toolbar
312     toolbar.startAnimation(expandAni);
313
314 }
315
316 public void showHelp(MenuItem item) {
317     Intent intent = new Intent(SchedulerActivity.this, HelpActivity.class);
318     Bundle bundle = new Bundle();
319     bundle.putString("activity", "scheduler");
320     intent.putExtras(bundle);
321     startActivity(intent);
322 }
323
324 public void modifyTask(Task task){
325     Intent intent = new Intent(SchedulerActivity.this, ModifyTaskActivity.class);
326     Bundle bundle = new Bundle();
327     bundle.putInt("assign", task.mAssign);
328     bundle.putString("title",task.mName);
329     bundle.putFloat("weight",task.mWeight);
330     bundle.putFloat("base",task.mBase);
331     bundle.putFloat("mark",task.mMark);
332     bundle.putString("course",task.mSubj + " " + task.mCode);
333     bundle.putInt("priority",task.mPriority);
334     bundle.putInt("is_done",task.mIsDone);
335     if (task.mDueDate.equals(getResources().getString(R.string.activity_add_task_date
336     ))) {
337         final Calendar cal = Calendar.getInstance();
338         bundle.putInt("year",cal.get(Calendar.YEAR));
339         bundle.putInt("month",cal.get(Calendar.MONTH));
340         bundle.putInt("day",cal.get(Calendar.DAY_OF_MONTH));
341     }
342     else {
343         int year = Integer.parseInt(task.mDueDate.substring(0,4));
344         int month = Integer.parseInt(task.mDueDate.substring(5,7));
345         int day = Integer.parseInt(task.mDueDate.substring(8));
346         bundle.putInt("year",year);
347         bundle.putInt("month",month);
348         bundle.putInt("day",day);
349     }
350     intent.putExtras(bundle);
351     startActivity(intent);
352 }
353
354 public void removeTask(Task task){
355     //REMOVE THE TASK
```

```
355     }
356
357     @Override
358     public boolean onCreateOptionsMenu(Menu menu) {
359         // Inflate the menu; this adds items to the action bar if it is present.
360         getMenuInflater().inflate(R.menu.activity_scheduler, menu);
361         return true;
362     }
363
364     private void sendEmail(String instr_email) {
365         Intent i = new Intent(Intent.ACTION_SENDTO);
366         i.setType("message/rfc822");
367         i.setData(Uri.parse("mailto:"+instr_email));
368         try {
369             startActivity(Intent.createChooser(i, "Send mail..."));
370         } catch (android.content.ActivityNotFoundException ex) {
371             Toast.makeText(SchedulerActivity.this,
372                 "There are no email clients installed.", Toast.LENGTH_SHORT)
373                 .show();
374         }
375     }
376
377     public void addTask(MenuItem item) {
378         Intent i = new Intent(SchedulerActivity.this, AddTaskActivity.class);
379         startActivity(i);
380     }
381
382     /**
383      * A simple implementation of list adapter.
384      */
385     class CustomListAdapter extends ArrayAdapter<String> {
386
387         public CustomListAdapter(Context context, int textViewResourceId) {
388             super(context, textViewResourceId);
389         }
390
391         @Override
392         public View getView(int position, View convertView, ViewGroup parent) {
393
394             if (convertView == null) {
395                 convertView = getLayoutInflater().inflate(
396                     R.layout.sched_list_item, null);
397             }
398
399             ((TextView) convertView.findViewById(R.id.sched_list_item_title))
400                 .setText(getItem(position));
401
402             // Resets the toolbar to be closed
403             View toolbar = convertView.findViewById(R.id.sched_toolbar);
404             ((LinearLayout.LayoutParams) toolbar.getLayoutParams()).bottomMargin = -(toolbar
405                 .getHeight());
406             toolbar.setVisibility(View.GONE);
407
408             return convertView;
409         }
410     }
411 }
412
413 }
```

```
1  /**
2   * Tasks.java
3   * Brock Butler
4   * A wrapper class for tasks information
5   * Created by James Grisdale on 2013-02-24
6   * Copyright (c) 2013 Sea Addicts. All rights reserved.
7  **/
8
9  package edu.seaaddicts.brockbutler.scheduler;
10
11  import java.util.ArrayList;
12
13  import edu.seaaddicts.brockbutler.contacts.Contact;
14
15  public class Task {
16      public String mSubj; //course name
17      public String mCode; //course code
18      public boolean mIsPastDue; //is the task late
19      public float mMark; //mark on task
20      public float mBase; //base mark of task
21      public float mWeight; //weight of the mark
22      public int mAssign; //assignment number
23      public String mName; //name of task
24      public String mCreationDate; //creation date
25      public String mDueDate; //due date
26      public int mPriority; //priority of task
27      public int mIsDone; //is the task complete
28      public ArrayList<Contact> mContacts; //task contacts
29
30      //constructor - initializes the contacts arraylist
31      public Task(){
32          mContacts = new ArrayList<Contact>();
33      }
34
35      //isPastDue - calculates if the task is past the due date
36      public boolean isPastDueDate(){
37          return true;
38      }
39  }
40 }
41
```

```
1  <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
2      xmlns:tools="http://schemas.android.com/tools"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent"
5      android:layout_margin="10sp"
6      android:orientation="vertical"
7      tools:context=".AddTaskActivity" >
8
9      <LinearLayout
10         android:layout_width="fill_parent"
11         android:layout_height="wrap_content"
12         android:orientation="vertical" >
13
14         <LinearLayout
15             android:layout_width="fill_parent"
16             android:layout_height="wrap_content"
17             android:orientation="horizontal" >
18
19             <TextView
20                 android:layout_width="85sp"
21                 android:layout_height="wrap_content"
22                 android:layout_gravity="center_vertical"
23                 android:text="@string/activity_add_task_course" />
24
25             <Spinner
26                 android:id="@+id/add_task_course_spinner"
27                 android:layout_width="fill_parent"
28                 android:layout_height="wrap_content"
29                 android:layout_gravity="center_vertical"
30                 android:layout_marginLeft="25sp"
31                 android:prompt="@string/task_priority_prompt" />
32         </LinearLayout>
33
34         <LinearLayout
35             android:layout_width="fill_parent"
36             android:layout_height="wrap_content"
37             android:orientation="horizontal" >
38
39             <TextView
40                 android:layout_width="85sp"
41                 android:layout_height="wrap_content"
42                 android:text="@string/activity_add_task_task_title" />
43
44             <EditText
45                 android:id="@+id/add_task_title"
46                 android:layout_width="fill_parent"
47                 android:layout_height="wrap_content"
48                 android:layout_marginLeft="25sp"
49                 android:hint="@string/activity_add_task_hint_task_title"
50                 android:inputType="text" />
51         </LinearLayout>
52
53         <LinearLayout
54             android:layout_width="fill_parent"
55             android:layout_height="wrap_content"
56             android:orientation="horizontal" >
57
58             <TextView
59                 android:layout_width="85sp"
60                 android:layout_height="wrap_content"
61                 android:layout_gravity="center_vertical"
62                 android:text="@string/activity_add_task_priority" />
63
64             <Spinner
65                 android:id="@+id/add_task_priority_spinner"
66                 android:layout_width="fill_parent"
67                 android:layout_height="wrap_content"
68                 android:layout_gravity="center_vertical"
69                 android:layout_marginLeft="25sp"
70                 android:entries="@array/task_priority_array"
71                 android:prompt="@string/task_priority_prompt" />
```



```
72         </LinearLayout>
73
74         <LinearLayout
75             android:layout_width="fill_parent"
76             android:layout_height="wrap_content"
77             android:orientation="horizontal" >
78
79             <TextView
80                 android:layout_width="85sp"
81                 android:layout_height="wrap_content"
82                 android:text="@string/activity_add_task_weight" />
83
84             <EditText
85                 android:id="@+id/task_weight"
86                 android:layout_width="fill_parent"
87                 android:layout_height="wrap_content"
88                 android:layout_marginLeft="25sp"
89                 android:hint="@string/activity_add_task_hint_weight"
90                 android:inputType="numberDecimal" />
91         </LinearLayout>
92
93         <LinearLayout
94             android:layout_width="fill_parent"
95             android:layout_height="wrap_content"
96             android:orientation="horizontal" >
97
98             <TextView
99                 android:layout_width="85sp"
100                 android:layout_height="wrap_content"
101                 android:text="@string/activity_add_task_base" />
102
103             <EditText
104                 android:id="@+id/task_base"
105                 android:layout_width="fill_parent"
106                 android:layout_height="wrap_content"
107                 android:layout_marginLeft="25sp"
108                 android:hint="@string/activity_add_task_hint_base"
109                 android:inputType="numberDecimal" />
110         </LinearLayout>
111
112         <LinearLayout
113             android:layout_width="fill_parent"
114             android:layout_height="wrap_content"
115             android:orientation="horizontal" >
116
117             <TextView
118                 android:layout_width="85sp"
119                 android:layout_height="wrap_content"
120                 android:text="@string/activity_add_task_mark" />
121
122             <EditText
123                 android:id="@+id/task_mark"
124                 android:layout_width="fill_parent"
125                 android:layout_height="wrap_content"
126                 android:layout_marginLeft="25sp"
127                 android:hint="@string/activity_add_task_hint_mark"
128                 android:inputType="numberDecimal" />
129         </LinearLayout>
130
131         <LinearLayout
132             android:layout_width="fill_parent"
133             android:layout_height="wrap_content"
134             android:layout_marginTop="10sp"
135             android:gravity="center_horizontal"
136             android:orientation="vertical" >
137
138             <Button
139                 android:id="@+id/add_task_due_date_button"
140                 android:layout_width="150sp"
141                 android:layout_height="wrap_content"
142                 android:gravity="center"
```

```
143         android:text="@string/activity_add_task_due_date" />
144
145         <TextView
146             android:id="@+id/add_task_date_textview"
147             android:layout_width="wrap_content"
148             android:layout_height="wrap_content"
149             android:layout_marginTop="10sp"
150             android:text="@string/activity_add_task_date"
151             android:textSize="20sp" />
152     </LinearLayout>
153
154     <LinearLayout
155         android:layout_width="fill_parent"
156         android:layout_height="wrap_content"
157         android:layout_margin="10sp"
158         android:gravity="center_horizontal"
159         android:orientation="horizontal" >
160
161         <Button
162             android:id="@+id/add_task_save_button"
163             android:layout_width="100sp"
164             android:layout_height="wrap_content"
165             android:text="@string/button_save" />
166
167         <Button
168             android:id="@+id/add_task_cancel_button"
169             android:layout_width="100sp"
170             android:layout_height="wrap_content"
171             android:layout_marginLeft="20sp"
172             android:text="@string/button_cancel" />
173     </LinearLayout>
174 </LinearLayout>
175
176 </ScrollView>
```

```
1  <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
2      xmlns:tools="http://schemas.android.com/tools"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent"
5      android:layout_margin="10sp"
6      android:orientation="vertical"
7      tools:context=".ModifyTaskActivity" >
8
9      <LinearLayout
10         android:layout_width="fill_parent"
11         android:layout_height="wrap_content"
12         android:orientation="vertical" >
13
14         <LinearLayout
15             android:layout_width="fill_parent"
16             android:layout_height="wrap_content"
17             android:orientation="horizontal" >
18
19             <TextView
20                 android:layout_width="85sp"
21                 android:layout_height="wrap_content"
22                 android:layout_gravity="center_vertical"
23                 android:text="@string/activity_add_task_course" />
24
25             <Spinner
26                 android:id="@+id/add_task_course_spinner"
27                 android:layout_width="fill_parent"
28                 android:layout_height="wrap_content"
29                 android:layout_gravity="center_vertical"
30                 android:layout_marginLeft="25sp"
31                 android:prompt="@string/task_priority_prompt" />
32         </LinearLayout>
33
34         <LinearLayout
35             android:layout_width="fill_parent"
36             android:layout_height="wrap_content"
37             android:orientation="horizontal" >
38
39             <TextView
40                 android:layout_width="85sp"
41                 android:layout_height="wrap_content"
42                 android:text="@string/activity_add_task_task_title" />
43
44             <EditText
45                 android:id="@+id/add_task_title"
46                 android:layout_width="fill_parent"
47                 android:layout_height="wrap_content"
48                 android:layout_marginLeft="25sp"
49                 android:hint="@string/activity_add_task_hint_task_title"
50                 android:inputType="text" />
51         </LinearLayout>
52
53         <LinearLayout
54             android:layout_width="fill_parent"
55             android:layout_height="wrap_content"
56             android:orientation="horizontal" >
57
58             <TextView
59                 android:layout_width="85sp"
60                 android:layout_height="wrap_content"
61                 android:layout_gravity="center_vertical"
62                 android:text="@string/activity_add_task_priority" />
63
64             <Spinner
65                 android:id="@+id/add_task_priority_spinner"
66                 android:layout_width="fill_parent"
67                 android:layout_height="wrap_content"
68                 android:layout_gravity="center_vertical"
69                 android:layout_marginLeft="25sp"
70                 android:entries="@array/task_priority_array"
71                 android:prompt="@string/task_priority_prompt" />
```

```
72         </LinearLayout>
73
74         <LinearLayout
75             android:layout_width="fill_parent"
76             android:layout_height="wrap_content"
77             android:orientation="horizontal" >
78
79             <TextView
80                 android:layout_width="85sp"
81                 android:layout_height="wrap_content"
82                 android:text="@string/activity_add_task_weight" />
83
84             <EditText
85                 android:id="@+id/task_weight"
86                 android:layout_width="fill_parent"
87                 android:layout_height="wrap_content"
88                 android:layout_marginLeft="25sp"
89                 android:hint="@string/activity_add_task_hint_weight"
90                 android:inputType="numberDecimal" />
91         </LinearLayout>
92
93         <LinearLayout
94             android:layout_width="fill_parent"
95             android:layout_height="wrap_content"
96             android:orientation="horizontal" >
97
98             <TextView
99                 android:layout_width="85sp"
100                 android:layout_height="wrap_content"
101                 android:text="@string/activity_add_task_base" />
102
103             <EditText
104                 android:id="@+id/task_base"
105                 android:layout_width="fill_parent"
106                 android:layout_height="wrap_content"
107                 android:layout_marginLeft="25sp"
108                 android:hint="@string/activity_add_task_hint_base"
109                 android:inputType="numberDecimal" />
110         </LinearLayout>
111
112         <LinearLayout
113             android:layout_width="fill_parent"
114             android:layout_height="wrap_content"
115             android:orientation="horizontal" >
116
117             <TextView
118                 android:layout_width="85sp"
119                 android:layout_height="wrap_content"
120                 android:text="@string/activity_add_task_mark" />
121
122             <EditText
123                 android:id="@+id/task_mark"
124                 android:layout_width="fill_parent"
125                 android:layout_height="wrap_content"
126                 android:layout_marginLeft="25sp"
127                 android:hint="@string/activity_add_task_hint_mark"
128                 android:inputType="numberDecimal" />
129         </LinearLayout>
130
131         <LinearLayout
132             android:layout_width="fill_parent"
133             android:layout_height="wrap_content"
134             android:layout_marginTop="10sp"
135             android:gravity="center_horizontal"
136             android:orientation="vertical" >
137
138             <Button
139                 android:id="@+id/add_task_due_date_button"
140                 android:layout_width="150sp"
141                 android:layout_height="wrap_content"
142                 android:gravity="center"
```

```
143         android:text="@string/activity_add_task_due_date" />
144
145         <TextView
146             android:id="@+id/add_task_date_textview"
147             android:layout_width="wrap_content"
148             android:layout_height="wrap_content"
149             android:layout_marginTop="10sp"
150             android:text="@string/activity_add_task_date"
151             android:textSize="20sp" />
152     </LinearLayout>
153
154     <LinearLayout
155         android:layout_width="fill_parent"
156         android:layout_height="wrap_content"
157         android:layout_margin="10sp"
158         android:gravity="center_horizontal"
159         android:orientation="horizontal" >
160
161         <Button
162             android:id="@+id/add_task_save_button"
163             android:layout_width="100sp"
164             android:layout_height="wrap_content"
165             android:text="@string/button_save" />
166
167         <Button
168             android:id="@+id/add_task_cancel_button"
169             android:layout_width="100sp"
170             android:layout_height="wrap_content"
171             android:layout_marginLeft="20sp"
172             android:text="@string/button_cancel" />
173     </LinearLayout>
174 </LinearLayout>
175
176 </ScrollView>
177
```

```
1  <menu xmlns:android="http://schemas.android.com/apk/res/android" >
2
3      <item
4          android:id="@+id/menu_add_task"
5          android:orderInCategory="100"
6          android:showAsAction="never"
7          android:onClick="addTask"
8          android:title="@string/menu_add_task"/>
9  <!--
10     <item
11         android:id="@+id/menu_expand_all"
12         android:orderInCategory="100"
13         android:showAsAction="never"
14         android:title="@string/scheduler_menu_showhide_completed"/> -->
15     <!--
16         <item -->
17             android:id="@+id/menu_collapse_all" -->
18             android:orderInCategory="100" -->
19             android:showAsAction="never" -->
20             android:title="@string/scheduler_menu_show_completed"/> -->
21     <item
22         android:id="@+id/menu_show_scheduler_help"
23         android:orderInCategory="100"
24         android:showAsAction="never"
25         android:onClick="showHelp"
26         android:title="@string/menu_show_scheduler_help"/>
27 </menu>
```

```
1  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2      android:id="@+id/activity_scheduler"
3      xmlns:tools="http://schemas.android.com/tools"
4      android:layout_width="match_parent"
5      android:layout_height="match_parent"
6      tools:context=".SchedulerActivity" >
7
8      <ListView
9          android:id="@+id/sched_list"
10         android:layout_width="fill_parent"
11         android:layout_height="fill_parent"
12         android:divider="@drawable/grad_course_title" >
13  </ListView>
14
15  <TextView
16      android:id="@+id/tv_no_courses"
17      android:layout_width="fill_parent"
18      android:layout_height="fill_parent"
19      android:text="@string/sched_no_courses"
20      android:textColor="#ddd"
21      android:textStyle="italic"
22      android:textSize="20sp"
23      android:gravity="center"
24      android:visibility="gone" />
25
26  </LinearLayout>
```

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:id="@+id/sched_list_item"
4      android:layout_width="fill_parent"
5      android:layout_height="fill_parent"
6      android:layout_marginBottom="10dip"
7      android:orientation="vertical" >
8
9      <TextView
10         android:id="@+id/sched_list_item_title"
11         android:layout_width="fill_parent"
12         android:layout_height="60sp"
13         android:layout_marginBottom="5dip"
14         android:background="@drawable/grad_course_title"
15         android:focusable="false"
16         android:focusableInTouchMode="false"
17         android:padding="10dip"
18         android:textSize="30sp"
19         android:textStyle="bold" />
20
21      <!-- Toolbar -->
22
23      <LinearLayout
24         android:id="@+id/sched_toolbar"
25         android:layout_width="fill_parent"
26         android:layout_height="wrap_content"
27         android:orientation="vertical"
28         android:padding="10dip"
29         android:visibility="gone" >
30
31         <include
32             android:id="@+id/row_prof"
33             layout="@layout/row_prof_simple" />
34
35         <include
36             android:id="@+id/row_lec"
37             layout="@layout/row_task_lecture" />
38
39         <include
40             android:id="@+id/row_lab"
41             layout="@layout/row_task_lab" />
42
43         <include
44             android:id="@+id/row_tut"
45             layout="@layout/row_task_tutorial" />
46
47         <include
48             android:id="@+id/row_sem"
49             layout="@layout/row_task_seminar" />
50
51         <include
52             android:id="@+id/row_grade"
53             layout="@layout/row_grade" />
54
55         <TextView
56             android:id="@+id/sched_tasks_title"
57             android:layout_width="fill_parent"
58             android:layout_height="wrap_content"
59             android:layout_gravity="center_vertical"
60             android:layout_marginLeft="20dip"
61             android:background="@drawable/grad_task_title"
62             android:paddingLeft="5sp"
63             android:paddingRight="10sp"
64             android:paddingTop="2sp"
65             android:text="@string/title_tasks"
66             android:textSize="16sp"
67             android:textStyle="bold"
68             android:typeface="sans"
69             android:visibility="gone" />
70
71         <LinearLayout

```



```
72         android:id="@+id/sched_tasks"
73         android:layout_width="fill_parent"
74         android:layout_height="wrap_content"
75         android:layout_marginLeft="10dip"
76         android:orientation="vertical"
77         android:paddingLeft="10sp"
78         android:paddingRight="10sp"
79         android:visibility="gone" >
80     </LinearLayout>
81 </LinearLayout>
82
83 </LinearLayout>
```