

**Higher Diploma in Computing, DT265**

Programming for Mobile and Smart Devices

**Coursework 2 – User Interfaces**

**Release Date**: 28th March 2017 (Wk10)

**Submission Date:** 25th April 2017 (Wk11 / 4wks)

**Student Demo Date:** 25th April 2017 (lab)

|  |  |  |  |
| --- | --- | --- | --- |
| **Name:** | Peter Leonard | **Student Number:** | D16124358 |

Objectives:

Familiarize yourself with Android’s User Interface (UI) classes. Create a simple application that uses a variety of UI elements including: Buttons, TextViews and Checkboxes. You will also reinforce the knowledge you've gained in previous labs by implementing a larger portion of the application from scratch.

Coding:

**1. *ToDoManagerActivity.java***

onCreate()

|  |
| --- |
| @Override **protected void** onCreate(Bundle savedInstanceState) {  RecyclerView mRecyclerView;  RecyclerView.LayoutManager mLayoutManager;   Bundle b = getIntent().getExtras(); *// Get value passed on startActivity() from priority/deadline sort button* **boolean** sortBydeadline = **true**; *// default value* **if**(b != **null**)  sortBydeadline = b.getBoolean(**"sort"**);   **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  **mAdapter** = **new** ToDoListAdapter(getBaseContext(),sortBydeadline);   mRecyclerView = (RecyclerView) findViewById(R.id.***my\_recycler\_view***);   *// use a linear layout manager to style recyclerView* mLayoutManager = **new** LinearLayoutManager(**this**);  mRecyclerView.setLayoutManager(mLayoutManager);   *//****TODO(DONE) - Attach the adapter to this ListActivity's ListView (replaced with recyclerView)*** mRecyclerView.setAdapter(**mAdapter**);   *//* ***TODO 5 (DONE) Add two tab buttons to sort recyclerView -*** *// Buttons restart the activity with an extra boolean value,  //this value is passed to ToDoListAdapter which returns the list in the new order* **final** Button tabDeadline = (Button) findViewById(R.id.***TabDeadline***);  tabDeadline.setOnClickListener(**new** View.OnClickListener()  {  @Override  **public void** onClick(View v)  {  Intent intent = **new** Intent(ToDoManagerActivity.**this**, ToDoManagerActivity.**class**);  Bundle b = **new** Bundle();  b.putBoolean(**"sort"**, **true**); *//Your id* intent.putExtras(b); *//Put your id to your next Intent* startActivity(intent);  finish();  }  });  **final** Button tabPriority = (Button) findViewById(R.id.***TabPriority***);  tabPriority.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  Intent intent = **new** Intent(ToDoManagerActivity.**this**, ToDoManagerActivity.**class**);  Bundle b = **new** Bundle();  b.putBoolean(**"sort"**, **false**); *//Your id* intent.putExtras(b); *//Put your id to your next Intent* startActivity(intent);  finish();  }  });  *//****TODO - Add footerView to ListView(Replaced with recylerView)*** *//****TODO(Done) - Inflate footerView for footer\_view.xml file* final** TextView footerView = (TextView) findViewById(R.id.***footerView***);   *//****TODO(DONE) - Attach Listener to FooterView. Implement onClick()*** footerView.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  log(**"Entered footerView.OnClickListener.onClick()"**);  Intent myIntent = **new** Intent(getBaseContext(), AddToDoActivity.**class**);  startActivityForResult(myIntent, ***ADD\_TODO\_ITEM\_REQUEST***);*//* }  });   } |

onActivityResult()

|  |
| --- |
| @Override  **protected void** onActivityResult(**int** requestCode, **int** resultCode, Intent data) {   log(**"Entered onActivityResult()"**);   *//* ***TODO(DONE) - Check result code and request code* if**(requestCode==***ADD\_TODO\_ITEM\_REQUEST***){   **if**(resultCode==***RESULT\_OK***){  *// create new ToDoItem with data* ToDoItem newItem = **new** ToDoItem(data);  Date date = **new** Date();  **int** count = newItem.getTime() - (**int**)date.getTime();  *//****TODO(DONE) 2 - Schedule Notification with item submitted from AddToDoActivity*** scheduleNotification(getBaseContext(), count, newItem.getUniqueID() ,newItem.getTitle()); *// Context, delay,ID, Title for Text Content   // add this item to the adapter* **mAdapter**.add(newItem);  }   **if**(resultCode==***RESULT\_CANCELED***){  }  }  } |

**2. *ToDoListAdapter.java***

getView() NOTE-getView() replaced with viewHolder and onBindViewHolder()

|  |
| --- |
| @Override **public void** onBindViewHolder(**final** ToDoListAdapter.ViewHolder holder, **final int** position) {  sort(**sortByDeadline**);*//****TODO 5 (DONE) Call sort method using constructor parameter* final** ToDoItem current = **mItems**.get(position);*//****TODO(DONE) - Get the current ToDoItem*** *//****TODO(DONE) - Fill in specific ToDoItem data*** holder.**title**.setText(current.getTitle());*//****TODO(DONE) - Display Title in TextView*** holder.**title**.setTextColor(Color.*parseColor*(**"white"**));  holder.**priority**.setText(current.getPriority().toString());*//****TODO(DONE) - Display Priority in a TextView*** holder.**date**.setText((current.***FORMAT***.format(current.getDate())));*//* ***TODO(DONE) - Display Time and Date.*** holder.**checkBox**.setChecked(current.getStatus()== ToDoItem.Status.***DONE***);*//****TODO(DONE) - Set up Status CheckBox*** *//* ***TODO 2(DONE) - Set Warning (TextColour) if Task Deadline Approaching*** checkTime(current,holder);   *//****TODO 1 (DONE) - Set titleView background colour(Seni-Magenta/Cyan according to status.* if**(current.getStatus()== ToDoItem.Status.***NOTDONE***)  {holder.**title**.setBackgroundColor(Color.*parseColor*(**"#80e600e6"**)); *// Magenta NotDone* }  **else** {  holder.**title**.setBackgroundColor(Color.*parseColor*(**"#8000e6e6"**)); *//Cyan Done* }    *//* ***TODO 3 (DONE) - Set Spinner for priority selection*** ArrayAdapter<CharSequence> adapter;  adapter = ArrayAdapter.*createFromResource*(holder.**itemView**.getContext(),R.array.***priority\_Spinner***,android.R.layout.***simple\_spinner\_item***);  adapter.setDropDownViewResource(android.R.layout.***simple\_spinner\_dropdown\_item***);  holder.**spinner**.setAdapter(adapter);  holder.**spinner**.setOnItemSelectedListener(**new** AdapterView.OnItemSelectedListener() {  @Override  **public void** onItemSelected(AdapterView<?> parent, View view, **int** place, **long** id) {  *//Spinner will attempt to reset each priority to position 0(High) every time  //the view refreshes. To avoid this, if the priority passed in is at position 0 the  // spinner will do nothing. To accomodate this postion 0 is replaced with "Options:"  //and high moved to position 1 etc.* **if**(place != 0)  {  holder.**priority**.setText(parent.getItemAtPosition(place).toString());  current.setPriority(ToDoItem.Priority.*valueOf*(parent.getItemAtPosition(place).toString()));*// If it crashes drop this* }   *//Toast.makeText(holder.itemView.getContext(),parent.getItemAtPosition(place)+" selected",Toast.LENGTH\_SHORT).show();  //current.setPriority(ToDoItem.Priority.valueOf(parent.getItemAtPosition(place).toString()));* }  @Override  **public void** onNothingSelected(AdapterView<?> parent) {  }  });   *//****TODO 1 (DONE) - Chenge colour on checkbox status change*** holder.**checkBox**.setOnCheckedChangeListener(**new** CompoundButton.OnCheckedChangeListener() {  @Override  **public void** onCheckedChanged(CompoundButton buttonView, **boolean** isChecked) {  log(**"Entered onCheckedChanged()"**);  **if**(isChecked){  current.setStatus(ToDoItem.Status.***DONE***);  holder.**checkBox**.setChecked(**true**);  holder.**title**.setBackgroundColor(Color.*parseColor*(**"#8000e6e6"**));  }  **else**{  current.setStatus(ToDoItem.Status.***NOTDONE***);  holder.**checkBox**.setChecked(**false**);  holder.**title**.setBackgroundColor(Color.*parseColor*(**"#80e600e6"**));  }  }  });   *//****TODO 4 (DONE) - Delete on longclick, vibrate, alert dialog, delete pending notification*** holder.**itemView**.setOnLongClickListener(**new** View.OnLongClickListener() {  @Override  **public boolean** onLongClick(**final** View v) {  Vibrator vibrator = (Vibrator) v.getRootView().getContext().getSystemService(Context.***VIBRATOR\_SERVICE***);  vibrator.vibrate(100);  **new** AlertDialog.Builder( v.getRootView().getContext())  .setTitle( **"Delete Task"** )  .setMessage( **"Are you sure you want to remove this task?"** )  .setPositiveButton( **"Delete"**, **new** DialogInterface.OnClickListener() {  **public void** onClick(DialogInterface dialog, **int** which) {  **mItems**.remove(position);  checkTime(current,holder);  Intent intent = **new** Intent(**context**, NotificationPublisher.**class**);  PendingIntent pendingIntent = PendingIntent.*getBroadcast*(**context**,  current.getUniqueID(), intent, PendingIntent.***FLAG\_CANCEL\_CURRENT***);  AlarmManager am = (AlarmManager) **context**.getSystemService(Context.***ALARM\_SERVICE***);  am.cancel(pendingIntent);  *// Cancel the `PendingIntent` after alarm cancelled* pendingIntent.cancel();  notifyDataSetChanged();  Toast.*makeText*(**context**, **"Deleted"**,  Toast.***LENGTH\_LONG***).show();  }  })  .setNegativeButton( **"Cancel"**, **new** DialogInterface.OnClickListener() {  **public void** onClick(DialogInterface dialog, **int** which) {  Toast.*makeText*(**context**, **"Cancelled"**,  Toast.***LENGTH\_LONG***).show();  }  } )  .show();  **return true**;  }  }); }  **public static class** ViewHolder **extends** RecyclerView.ViewHolder {  TextView **title**;  TextView **priority**;  TextView **date**;  CheckBox **checkBox**;  Spinner **spinner**;    **public** ViewHolder(View itemView) {  **super**(itemView);  **title** = (TextView) itemView.findViewById(R.id.***titleView***);  **priority** = (TextView) itemView.findViewById(R.id.***priorityView***);  **checkBox** = (CheckBox) itemView.findViewById((R.id.***statusCheckBox***));  **date** = (TextView) itemView.findViewById(R.id.***dateView***);  **spinner** = (Spinner) itemView.findViewById(R.id.***prioritySpinner***);  } } |

**3. *AddToDoActivity.java***

onClick() for cancelButton

|  |
| --- |
| **final** Button cancelButton = (Button) findViewById(R.id.***cancelButton***); cancelButton.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  log(**"Entered cancelButton.OnClickListener.onClick()"**);   *//****TODO (DONE) - Implement onClick()*** Intent intent = **new** Intent();  setResult(***RESULT\_CANCELED***, intent);  finish();  } }); |

onClick() for resetButton

|  |
| --- |
| **final** Button resetButton = (Button) findViewById(R.id.***resetButton***); resetButton.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  log(**"Entered resetButton.OnClickListener.onClick()"**);   *//****TODO (DONE) - Reset data fields to default values* mTitleText**.setText(**""**);  **mDefaultStatusButton**.setChecked(**true**);  **mDefaultPriorityButton**.setChecked(**true**);  setDefaultDateTime();  } }); |

onClick() for submitButton

|  |
| --- |
| *// OnClickListener for the Submit Button* **final** Button submitButton = (Button) findViewById(R.id.***submitButton***);  submitButton.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  log(**"Entered submitButton.OnClickListener.onClick()"**);   *//****TODO (DONE) - Get Priority*** Priority priority = getPriority();   *//****TODO (DONE) - Get Status*** Status status = getStatus();   *//****TODO (DONE) - Title*** String titleString = **mTitleText**.getText().toString();   *//****TODO (DONE) - Date*** String fullDate = *dateString* + **" "** + *timeString*;   *// Package ToDoItem data into an Intent* Intent data = **new** Intent();  ToDoItem.*packageIntent*(data, titleString, priority, status, fullDate);    *//****TODO (DONE) - return data Intent and finish*** setResult(***RESULT\_OK***, data);   finish();  }  }); } |

**Testing:**

TestSubmit LogCat Output:

04-25 00:29:29.574 31317-31317/com.example.peter.mytodomanager I/Lab-UserInterface:

**Entered footerView.OnClickListener.onClick()**

04-25 00:29:48.422 31317-31317/com.example.peter.mytodomanager I/Lab-UserInterface:

**Entered submitButton.OnClickListener.onClick()**

04-25 00:29:48.933 31317-31317/com.example.peter.mytodomanager I/Lab-UserInterface:

**Entered onActivityResult()**

TestReset LogCat Output:

04-25 00:39:39.370 8307-8307/com.example.peter.mytodomanager I/Lab-UserInterface:

**Entered footerView.OnClickListener.onClick()**

04-25 00:40:11.802 8307-8307/com.example.peter.mytodomanager I/Lab-UserInterface:

**Entered resetButton.OnClickListener.onClick()**

04-25 00:40:25.996 8307-8307/com.example.peter.mytodomanager I/Lab-UserInterface:

**Entered submitButton.OnClickListener.onClick()**

04-25 00:40:26.526 8307-8307/com.example.peter.mytodomanager I/Lab-UserInterface:

**Entered onActivityResult()**

04-25 00:40:29.719 8307-8307/com.example.peter.mytodomanager I/Lab-UserInterface:

**Entered onCheckedChanged()**

TestCancel LogCat Output:

04-25 00:35:47.603 31317-31317/com.example.peter.mytodomanager I/Lab-UserInterface:

**Entered footerView.OnClickListener.onClick()**

04-25 00:36:07.763 31317-31317/com.example.peter.mytodomanager I/Lab-UserInterface:

**Entered cancelButton.OnClickListener.onClick()**

04-25 00:36:08.293 31317-31317/com.example.peter.mytodomanager I/Lab-UserInterface:

**Entered onActivityResult()**

04-25 00:36:12.837 31317-31317/com.example.peter.mytodomanager I/Lab-UserInterface:

**Entered footerView.OnClickListener.onClick()**

04-25 00:36:21.176 31317-31317/com.example.peter.mytodomanager I/Lab-UserInterface:

**Entered submitButton.OnClickListener.onClick()**

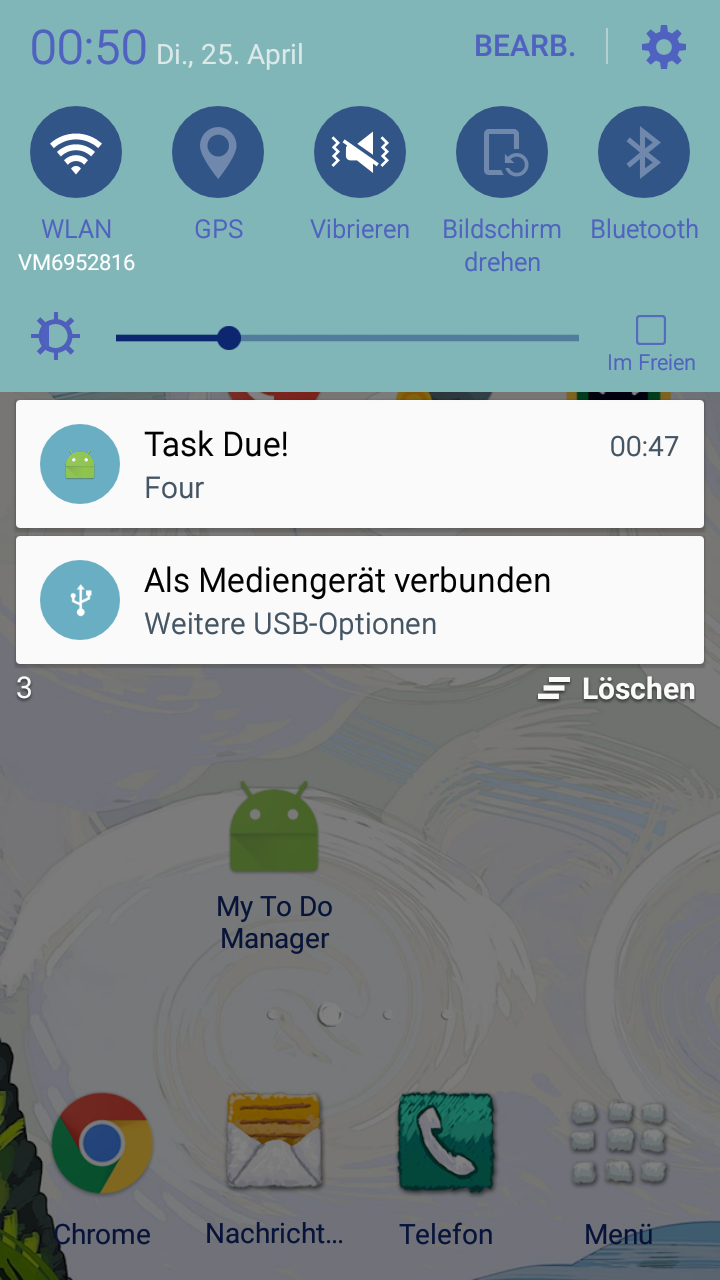
04-25 00:36:21.716 31317-31317/com.example.peter.mytodomanager I/Lab-UserInterface:

**Screenshots:**

**Sort By Date Due**

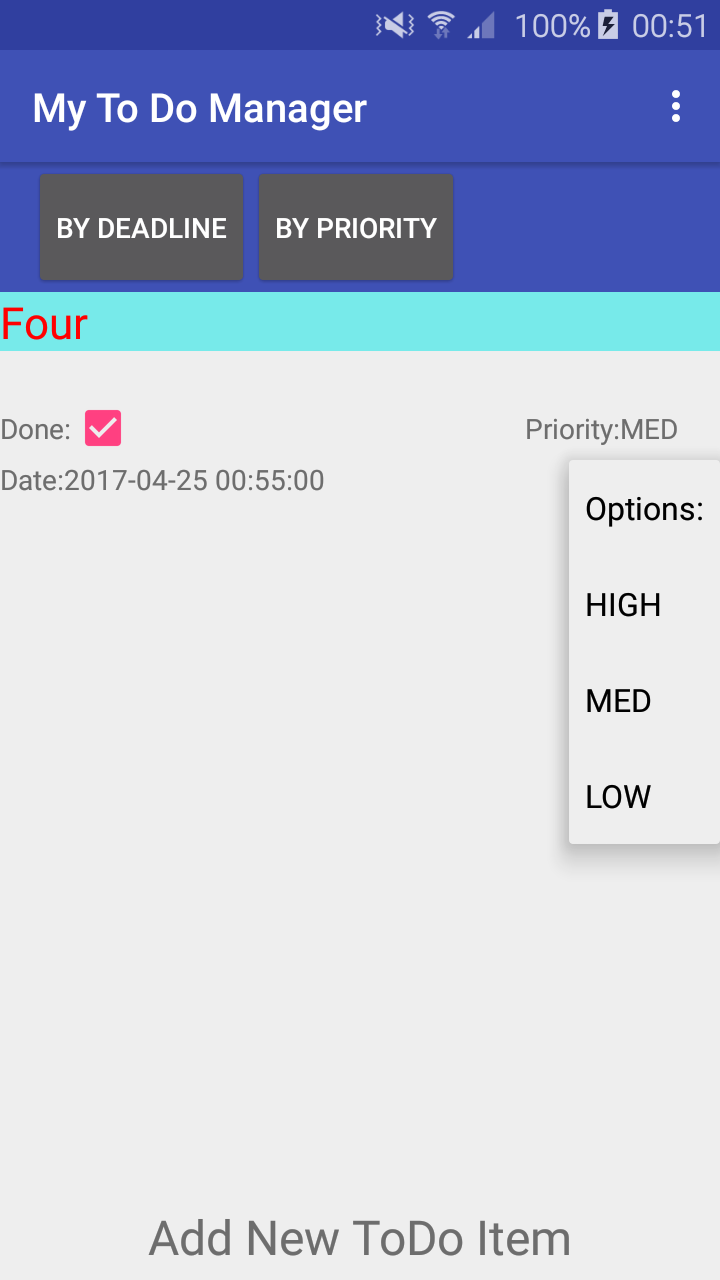


Sort By Priority



Task four is set for 00:55, notification received 5 minutes beforehand.

Demonstrating colour changes on status check as well as due date less than 24 hours. Spinner dialogue shows priority options:



Additional Functionality:

**1.**

**Background colours are set using the following if statement in the ToDoListadapter. A similar statement is made within the onChecked method for the check box to ensure immediate colour change.**

*//****TODO 1 (DONE) - Set titleView background colour(Seni-Magenta/Cyan according to status.*if**(current.getStatus()== ToDoItem.Status.***NOTDONE***)  
{holder.**title**.setBackgroundColor(Color.*parseColor*(**"#80e600e6"**)); *// Magenta NotDone*}  
**else**{  
 holder.**title**.setBackgroundColor(Color.*parseColor*(**"#8000e6e6"**)); *//Cyan Done*}

**2.**

**Title colour change is made using the follow method called within the ToDoListAdapter:**

**private void** checkTime(ToDoItem item, ViewHolder holder)  
{  
 Date date = **new** Date();  
 **int** count = item.getTime() - (**int**)date.getTime();  
 **if**(count < (24 \* 60 \* 60 \* 1000) ) *//if Deadline is less than one day* {  
 holder.**title**.setTextColor(Color.*parseColor*(**"red"**));  
 }**else** {  
 holder.**title**.setTextColor(Color.*parseColor*(**"white"**));  
 }  
}

**The items date is compared to the current date. A similar comparison is made for the scheduled notification:**

*//****TODO schedule notification with alarm five minutes before due*public void** scheduleNotification(Context context, **long** delay, **int** notificationId, String title){  
 NotificationCompat.Builder builder = **new** NotificationCompat.Builder(context)  
 .setContentTitle(**"Task Due!"**)  
 .setContentText(title)  
 .setAutoCancel(**true**)  
 .setSmallIcon(R.mipmap.***ic\_launcher***)  
 .setSound(RingtoneManager.*getDefaultUri*(RingtoneManager.***TYPE\_NOTIFICATION***));  
 Intent intent = **new** Intent(context, ToDoListAdapter.**class**);  
 PendingIntent activity = PendingIntent.*getActivity*(context, notificationId, intent, PendingIntent.***FLAG\_CANCEL\_CURRENT***);  
 builder.setContentIntent(activity);  
  
 Notification notification = builder.build();  
  
 Intent notificationIntent = **new** Intent(context, NotificationPublisher.**class**);  
 notificationIntent.putExtra(NotificationPublisher.*NOTIFICATION\_ID*, notificationId);  
 notificationIntent.putExtra(NotificationPublisher.*NOTIFICATION*, notification);  
 PendingIntent pendingIntent = PendingIntent.*getBroadcast*(context, notificationId, notificationIntent, PendingIntent.***FLAG\_CANCEL\_CURRENT***);  
  
 **long** futureInMillis = SystemClock.*elapsedRealtime*() + (delay - (5\*60\*1000));  
 AlarmManager alarmManager = (AlarmManager) context.getSystemService(Context.***ALARM\_SERVICE***);  
 alarmManager.set(AlarmManager.***ELAPSED\_REALTIME\_WAKEUP***, futureInMillis, pendingIntent);  
}

**The tasks duration in milliseconds is passed to the alarm manager(minus five minutes) which creates an alarm and uses the additional NotificationPublisher class to create a notification. The schedule notification method is called within onActivityResult() in the main activity ToDoListManagerActivity.**

**3.**

**The spinner is created within the ToDoListAdapter as follows:**

*//* ***TODO 3 (DONE) - Set Spinner for priority selection***ArrayAdapter<CharSequence> adapter;  
adapter = ArrayAdapter.*createFromResource*(holder.**itemView**.getContext(),R.array.***priority\_Spinner***,android.R.layout.***simple\_spinner\_item***);  
adapter.setDropDownViewResource(android.R.layout.***simple\_spinner\_dropdown\_item***);  
holder.**spinner**.setAdapter(adapter);  
holder.**spinner**.setOnItemSelectedListener(**new** AdapterView.OnItemSelectedListener() {  
 @Override  
 **public void** onItemSelected(AdapterView<?> parent, View view, **int** place, **long** id) {  
 *//Spinner will attempt to reset each priority to position 0(High) every time  
 //the view refreshes. To avoid this, if the priority passed in is at position 0 the  
 // spinner will do nothing. To accomodate this position 0 is replaced with "Options:"  
 //and high moved to position 1 etc.* **if**(place != 0)  
 {  
 holder.**priority**.setText(parent.getItemAtPosition(place).toString());  
 current.setPriority(ToDoItem.Priority.*valueOf*(parent.getItemAtPosition(place).toString()));*// If it crashes drop this* }  
 }  
 @Override  
 **public void** onNothingSelected(AdapterView<?> parent) {  
 }  
});

**As stated in the code, an if statement is used to prevent the spinner firing when initializing. Without this, the spinner would attempt to set each views priority to the position 0 priority every time it refreshes the view.**

4.

**Delete on long click is implemented as follows within the onBindData mehtod within the ToDoListAdapter class:**

*//****TODO 4 (DONE) - Delete on longclick, vibrate, alert dialog, delete pending notification***holder.**itemView**.setOnLongClickListener(**new** View.OnLongClickListener() {  
 @Override  
 **public boolean** onLongClick(**final** View v) {  
 Vibrator vibrator = (Vibrator) v.getRootView().getContext().getSystemService(Context.***VIBRATOR\_SERVICE***);  
 vibrator.vibrate(100);  
 **new** AlertDialog.Builder( v.getRootView().getContext())  
 .setTitle( **"Delete Task"** )  
 .setMessage( **"Are you sure you want to remove this task?"** )  
 .setPositiveButton( **"Delete"**, **new** DialogInterface.OnClickListener() {  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 **mItems**.remove(position);  
 checkTime(current,holder);  
 Intent intent = **new** Intent(**context**, NotificationPublisher.**class**);  
 PendingIntent pendingIntent = PendingIntent.*getBroadcast*(**context**,  
 current.getUniqueID(), intent, PendingIntent.***FLAG\_CANCEL\_CURRENT***);  
 AlarmManager am = (AlarmManager) **context**.getSystemService(Context.***ALARM\_SERVICE***);  
 am.cancel(pendingIntent);  
 *// Cancel the `PendingIntent` after alarm cancelled* pendingIntent.cancel();  
 notifyDataSetChanged();  
 Toast.*makeText*(**context**, **"Deleted"**,  
 Toast.***LENGTH\_LONG***).show();  
 }  
 })  
 .setNegativeButton( **"Cancel"**, **new** DialogInterface.OnClickListener() {  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 Toast.*makeText*(**context**, **"Cancelled"**,  
 Toast.***LENGTH\_LONG***).show();  
 }  
 } )  
 .show();  
 **return true**;  
 }  
});

**The pending notification is deleted by creating a new intent which takes the tasks unique ID (Created in the ToDoItem constructor by default) and passing this intent to the alarm managers cancel method.**

5.

**The Tabs in this application are created as buttons within the app's toolbar. On selecting a button, the MainActivity is restarted with an extra boolean value added to it:**

Bundle b = getIntent().getExtras(); *// Get value from priority/deadline sort button***boolean** sortBydeadline = **true**; *// default value***if**(b != **null**)  
 sortBydeadline = b.getBoolean(**"sort"**);

**This value is then passed to the adapters constructor.If no extra is found the boolean defaults to true(i.e. sorting by deadline)**

mAdapter = **new** ToDoListAdapter(getBaseContext(),sortBydeadline);

**The adapter will then call the sort() method:**

**private void** sort(**boolean** sortByDeadline)  
{  
 Comparator<ToDoItem> rankDeadline = **new** Comparator<ToDoItem>()  
 {  
 **public int** compare(ToDoItem item1, ToDoItem item2) {  
 **return** item1.getTime() - item2.getTime();  
 }  
 };  
 Comparator<ToDoItem> rankPriority = **new** Comparator<ToDoItem>(){  
  
 **public int** compare(ToDoItem item1, ToDoItem item2) {  
 *//return item1.getPriority().compareTo(item2.getPriority());* **int** priority = item1.getPriority().compareTo(item2.getPriority());  
 **if** (priority != 0) {  
 **return** priority;  
 }  
 **int** date = item1.getTime()- item2.getTime();  
 **if** (date!= 0) {  
 **return** date;  
 }  
 **return** item1.**title**.compareTo(item2.**title**);  
 }  
 };  
 **if**(sortByDeadline)  
 {  
 Collections.*sort*(**mItems**,rankDeadline);  
 }  
 **else** {  
 Collections.*sort*(**mItems**,rankPriority);  
 }  
}

**Which uses comparators to sort the recyclerView as needed. As the main activity opens, this newly sorted list is added.**