The Roster - Android Application Report

1.) View the app title label

TextView control was used to display app title on the screen.
Text was formatted to medium, 25sp font-size and purple color using following properties:

```
android:textAppearance="?android:attr/textAppearanceMedium"
android:textColor="#CC00FF"
android:textSize="25sp"
```

TextView was centered horizontally using:

```
android:layout_centerHorizontal="true"
```

2.) Enter text in the textbox

EditText control was used to allow user to provide name as an input.
Its text value is stored in a string key named "Name" in the shared preferences on the click event of "Save" button.

```
//save the textbox value
EditText txt1 = (EditText)findViewById(R.id.txtName);

String name = txt1.getText().toString();
prefsEditor.putString("Name", name);
```

This value is then retrieved onCreate() of main activity as:

```
//get the textbox value
 String name=prefs.getString("Name", "");
 EditText txt1=(EditText)findViewById(R.id.txtName);
 txt1.setText(name);
```

3.) Check a checkbox

CheckBox control was used.
Its Boolean value is stored in shared preferences on the click event of "Save" button.

```
//save the checkbox value
CheckBox Chk1=(CheckBox) findViewById(R.id.chkbox);
 Boolean checkBoxValue = Chk1.isChecked();
prefsEditor.putBoolean("isChecked", checkBoxValue);
```

This value is then retrieved onCreate() of main activity as:

```
 //get the checkbox value
  Boolean checkBoxValue=prefs.getBoolean("isChecked",false);
  CheckBox Chk1=(CheckBox) findViewById(R.id.chkbox);

  if(checkBoxValue)
  {
       Chk1.setChecked(true);
  }
  else
  {
       Chk1.setChecked(false);
    }
```

4.) Manipulate spinner control

Spinner control was used.
First the spinner control is populated with the list of items using *ArrayAdapter*.

```
//populate the dropdown list
private String[] eyeColor = {"Brown", "Black", "Green","Blue" };
Spinner spinnerEyeColor = (Spinner)findViewById(R.id.ddlEyeColor);
ArrayAdapter<String> adapter_color = new ArrayAdapter<String>(this,
                  android.R.layout.simple_spinner_item, eyeColor);
adapter_color.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
spinnerEyeColor.setAdapter(adapter_color);
spinnerEyeColor.setOnItemSelectedListener(this);
```

Its selected item position (integer value) is stored in shared preferences on the click event of "Save" button.

```
//save the dropdown value
 Spinner spinnerEyeColor = (Spinner)findViewById(R.id.ddlEyeColor);
prefsEditor.putInt("Position",spinnerEyeColor.getSelectedItemPosition());
```

This value is then retrieved onCreate() of main activity as:

```
//set the dropdown to selected value
int position=prefs.getInt("Position", 0);
spinnerEyeColor.setSelection(position);
```

5.) Manipulate calendar control UI

DatePicker control was used to allow user to select their date of birth.
The selected year, month and day are stored as integer values in shared preferences on the click event of "Save" button.

```
//save the date value
DatePicker date = (DatePicker) findViewById(R.id.date_picker);
prefsEditor.putInt("Year", date.getYear());
prefsEditor.putInt("Month", date.getMonth());
prefsEditor.putInt("Day", date.getDayOfMonth());
```

These values are then retrieved onCreate() of main activity as:

```
// set current date into Date Picker
DatePicker date = (DatePicker) findViewById(R.id.date_picker);
int year=prefs.getInt("Year", 0);
int month=prefs.getInt("Month", 0);
int day=prefs.getInt("Day", 0);
date.init(year, month, day, null);
```

6.) Manipulate radio button group

RadioGroup control was used to allow user to select one shirt size from given options.It was aligned horizontally using following attribute:

```
android:orientation="horizontal”
```

The selected radio button id is stored as integer value in shared preferences on the click event of “Save” button.

```
//save the selected radio button value
RadioGroup SizeGroup=(RadioGroup)findViewById(R.id.SizeGroup);
int selectedId = SizeGroup.getCheckedRadioButtonId();
prefsEditor.putInt("Size", selectedId);
```

This value is then retrieved onCreate() of main activity as:

```
//get the selected radio button identifier
int selectedId=prefs.getInt("Size", 0);
if(selectedId==0)
{
        //by default, the first radio button in the group should be checked.
        RadioGroup SizeGroup=(RadioGroup)findViewById(R.id.SizeGroup);
        int selId = SizeGroup.getCheckedRadioButtonId();
        RadioButton SizeButton = (RadioButton) findViewById(selId);
        SizeButton.setChecked(true);
 }
 else
 {
        RadioButton SizeButton = (RadioButton) findViewById(selectedId);
        SizeButton.setChecked(true);
 }
```

7.) Manipulate sliders

SeekBar control was used to allow user to select their pant size(0-16), shirt size(0-12) and shoe size(4-12).
The minimum value of SeekBar is 0 always while the maximum value was set to 8 for pant size, 12 for shirt size and 8 for shoe size.
The current value of seekBar is identified by progress property which can be set and retrieved by setProgress(int) and getProgress() methods.
The pant size varies from 0-16 in the step of 2. So the range of pantsize seekbar is set from 0 to 8.
So if user selects pantsize 6 on the seekbar, its value is stored as 12 i.e (6*2).This is done considering that the pant sizes are generally multiples of 2(even number).
The shirt size varies from 0-12 in the step of 1. So the range of shirtsize seekbar is set from 0 to 12.
So if user selects shirtsize 6 on the seekbar, its value is stored as 6.
The shoe size varies from 4-12 in the step of 1. So the range of shoesize seekbar is set from 0 to 8.
So if user selects shoesize 6 on the seekbar, its value is stored as 10 i.e (6+4).This is done considering the seekbar’s default minimum value of 0 to correspond to minimum available shoe size option of 4.

The selected values of all 3 slides is stored as integer value in shared preferences on the click event of “Save” button.

```
//save the pant size
SeekBar seekpant=(SeekBar)findViewById(R.id.seekPant);
int pantsize=seekpant.getProgress();
prefsEditor.putInt("PantSize", pantsize*2);

//save the shirt size
SeekBar seekshirt=(SeekBar)findViewById(R.id.seekShirt);
int shirtsize=seekshirt.getProgress();
prefsEditor.putInt("ShirtSize", shirtsize);

//save the shoe size
SeekBar seekshoe=(SeekBar)findViewById(R.id.seekShoe);
int shoesize=seekshoe.getProgress();
prefsEditor.putInt("ShirtSize", shoesize+4);
```

The selected values are also displayed in the textview associated with each slider.

This value is then retrieved onCreate() of main activity as:

```
//get pant size
int pantsize=prefs.getInt("PantSize",0);
TextView DisplayPantSize = (TextView)findViewById(R.id.displayPantSize);
DisplayPantSize.setText(Integer.toString(pantsize));

SeekBar seekPant = (SeekBar)findViewById(R.id.seekPant);
seekPant.setProgress(pantsize/2);

//get shirt size
int shirtsize=prefs.getInt("ShirtSize",0);
TextView DisplayShirtSize = (TextView)findViewById(R.id.displayShirtSize);
DisplayShirtSize.setText(Integer.toString(shirtsize));

SeekBar seekShirt = (SeekBar)findViewById(R.id.seekShirt);
seekShirt.setProgress(shirtsize);

//get shoe size
int shoesize=prefs.getInt("ShoeSize", 4);
TextView DisplayShoeSize = (TextView)findViewById(R.id.displayShoeSize);
DisplayShoeSize.setText(Integer.toString(shoesize));

SeekBar seekShoe = (SeekBar)findViewById(R.id.seekShoe);
seekShoe.setProgress(shoesize-4);

seekPant.setOnSeekBarChangeListener(this);
seekShirt.setOnSeekBarChangeListener(this);
seekShoe.setOnSeekBarChangeListener(this);
```

8.) Click on command button to save data to shared preferences

Button control is used to allow user to save his preferences.

Onclick event of this button, an instance of SharedPreferences class is initiatilized and user data is stored in key value pairs.

```
SharedPreferences prefs = getSharedPreferences("RosterPreferences", MODE_PRIVATE);
SharedPreferences.Editor prefsEditor = prefs.edit();
```

```
prefsEditor.commit();
```

A message pop up to display that data is saved.

```
//show user preferences stored successfully pop up message
Toast.makeText(getBaseContext(), "Saved your preferences.", Toast.LENGTH_SHORT).show();
```

This stored data can be retrieved from shared preferences onCreate method and the controls are populated with these values.

```
//initialize sharedpreferences lib
SharedPreferences prefs = getSharedPreferences("RosterPreferences", MODE_PRIVATE);
```