

Assignment 4

Task 1

```

@Override
public int describeContents() { return 0; }

@Override
public void writeToParcel(Parcel dest, int flags) {
    dest.writeString(this.name);
    dest.writeString(this.location);
    dest.writeString(this.keyword);
    dest.writeString(this.date);
    dest.writeByte(this.share ? (byte) 1 : (byte) 0);
    dest.writeString(this.email);
    dest.writeInt(this.rating);
}

public Parcelable() {
}

protected Parcelable(Parcel in) {
    this.name = in.readString();
    this.location = in.readString();
    this.keyword = in.readString();
    this.date = in.readString();
    this.share = in.readByte() != 0;
    this.email = in.readString();
    this.rating = in.readInt();
}

public static final Creator<Parcelable> CREATOR = new Creator<Parcelable>() {
    @Override
    public Parcelable createFromParcel(Parcel source) { return new Parcelable(source); }

    @Override
    public Parcelable[] newArray(int size) { return new Parcelable[size]; }
};

```

After setting and getting all the values for the Parcelable. You need to be able to write them to the parcelable. Which is simple enough. But there is no writeBoolean function so you need to use either writeByte or writeInt to setup a boolean value.

So we then create an array of Parcelables and add the data. Then when someone clicks on one of four buttons it will add the Parcelable into the intent via the putParcelableArrayListExtra function and then add it to the next activity.

```

        metaData.add(p1);
        metaData.add(p2);
        metaData.add(p3);
        metaData.add(p4);
    }

    public void onClick(View v){
        switch (v.getId()) {
            case R.id.chocolate:
                Intent intent = new Intent (this, MetaData.class);
                String send = "choc";
                intent.putParcelableArrayListExtra("Parcel", metaData);
                intent.putExtra("id", send);
                startActivity(intent);
                break;
        }
    }

```

Then in the next activity it will retrieve the Parcelable from the intent and then update the view depending on which item was selected

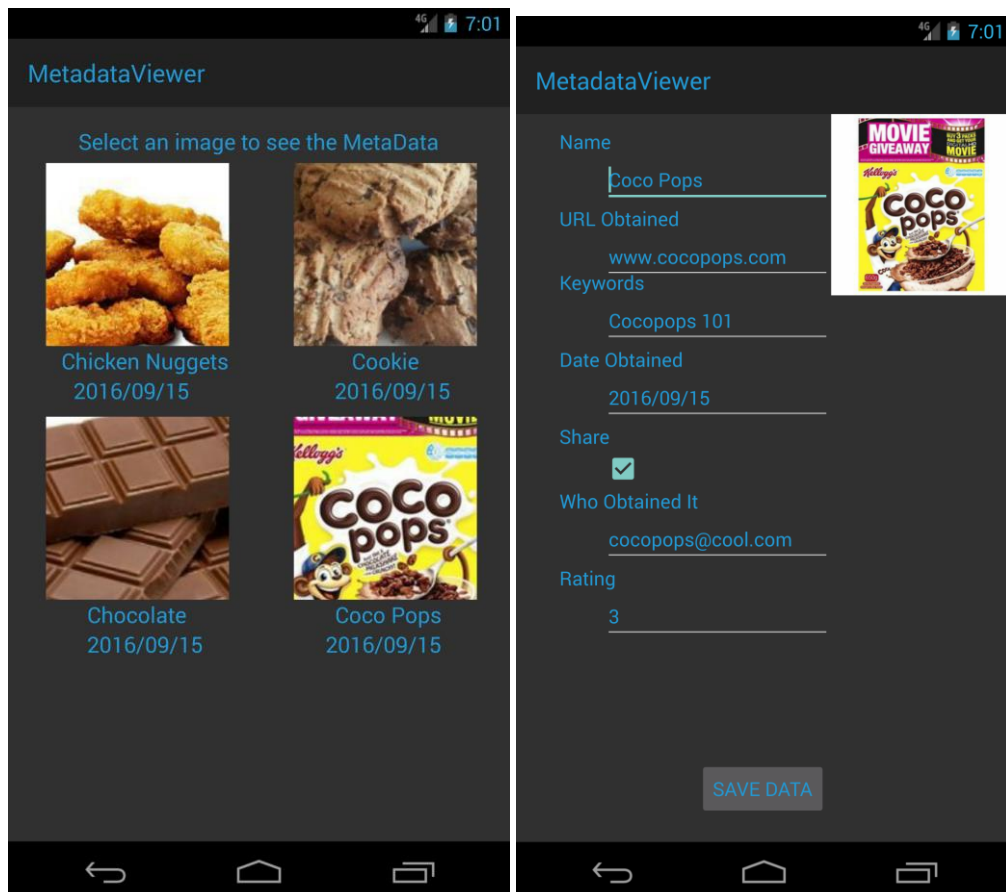
```
bundle = getIntent().getExtras();
metaData = getIntent().getParcelableArrayListExtra("Parcel");
id = bundle.getString("id");

if(id.equals("choc")){
    image.setImageResource(R.drawable.chocolate);
    name.setText(metaData.get(0).getName());
    location.setText(metaData.get(0).getLocation());
    keywords.setText(metaData.get(0).getKeyword());
    date.setText(metaData.get(0).getDate());
    share.setChecked(metaData.get(0).getShare());
    email.setText(metaData.get(0).getEmail());
    rating.setText(String.valueOf(metaData.get(0).getRating()));
    i = 0;
}
```

```
}
```

```
public void onClick(View view) {
    switch (view.getId()) {
        case R.id.button:
            Intent intent = new Intent(this, MainActivity.class);
            metaData.get(i).setName(name.getText().toString());
            metaData.get(i).setLocation(location.getText().toString());
            metaData.get(i).setKeyword(keywords.getText().toString());
            metaData.get(i).setDate(date.getText().toString());
            metaData.get(i).setShare(share.isChecked());
            metaData.get(i).setEmail(email.getText().toString());
            metaData.get(i).setRating(Integer.parseInt(rating.getText().toString()));
            intent.putParcelableArrayListExtra("Parcel", metaData);
            String sent = "sent";
            intent.putExtra("sent", sent);
            startActivity(intent);
            break;
    }
}
```

Here we update the data and send it back to the previous activity.



As you can see. Using Parcelables makes it easier to send junks of data to different activities because the built-in Java system can't handle the speed requirements of the Android user. So, for a small data size like this there wouldn't be any noticeable difference. But once you start adding more data, the average Android user would probably uninstall the app and find a better that is much faster. So logically use Parcelables whenever you can.

Task 2

Usability Test

Executive Summary

For the usability testing I have gathered 3 people to see what size font would best suite labels and information displays. I have discovered that people prefer the standard size text compared to the others. This means that there shouldn't be any changes to the current standards

Introduction

The test we are conducting is about learning what people think about what size font people think is better for labels and information displays. What happened was that I tested 3 people individually and presented them 3 font size and they judge which one is better.

Task Overview

Task 1 Font Size for Labels

Goal:	To find the best size font for a label
Inputs:	Button to change font size
Assumptions:	They know how to use a smart phone
Steps:	<ol style="list-style-type: none"> 1. Look at the font and discuss what they like/ don't like about it 2. Press the button to change the font 3. Repeat until all fonts are looked at
Time for Expert:	3-4 minutes
Instructions for user:	You have to look at the different font sizes and address why which one should/ shouldn't be used for a label.
Notes:	Ask what they think about the font size Sizes: <ul style="list-style-type: none"> • 12dp • 14dp • 16dp

Task 2 Font Size for Information Display

Goal:	To find the best size font for an information display
Inputs:	Button to change font size
Assumptions:	They know how to use a smart phone
Steps:	<ol style="list-style-type: none"> 1. Look at the font and discuss what they like/ don't like about it 2. Press the button to change the font 3. Repeat until all fonts are looked at
Time for Expert:	3-4 minutes
Instructions for user:	You have to look at the different font sizes and address why which one should/ shouldn't be used for an information display.
Notes:	Ask what they think about the font size Sizes: <ul style="list-style-type: none"> • 12dp • 14dp • 16dp

Task Details

Font Size for Labels

The participant must cycle through each of the texts and then talk about what they think of each text. Once completed they must discuss why they choose that specific font to be the perfect size for a label.

The purpose of the test is determined what is the best font size for a label.

Font Size for Information Displays

The participant must cycle through each of the texts and then talk about what they think of each text. Once completed they must discuss why they choose that specific font to be the perfect size for an information display.

The purpose of the test is determined what is the best font size for an information display.

Results

Participants

1. 54, M – Has low experience with Smart Phone
2. 17, F – Has High experience with Smart Phone
3. 51, F – Has Med experience with Smart Phone

Participant	Label			Information Display		
	12	14	16	12	14	16
George	×	✓	✓	×	✓	×
Nicola	×	✓	×	×	✓	×
Nancy	×	✓	✓	×	✓	×

As you can see from the above table. It's noticeable that the standard font 14 is loved by all participants. Testing some older people provide me information that they prefer either 14 or 16 for labels as their eye sight is slowly getting worst. So either size will be good for a Label. They all agreed that size 14 was the best for an information display because if the text is too big or small it will make the information harder to read. Even when the size is bigger they would have to scroll more and some people don't have the patients to read a whole page of text if the font size is too big.

Recommendations

After speaking to each participant I have discovered that the standard size of 14dp is best for both a label and an information display. Also depending on the size of the page you can even bump up the size of the label to 16dp because some people with poor eye sight will find it easier to see.

Reflection

This usability test went too smoothly. I should have made the test a little harder for the participants or at least gotten someone with no experience and see what they thought about the test to get a better result.