

NN4M challenge

APP GUIDE

RHUARHRI CORDON

Contents

1, App idea	2
Why use colour theory.....	3
How the app interprets colour theory	3
2, How it meets the requirements.....	4
3, Design decision	5
4, Testing.....	5
6, User Manual.....	6
7, JSON files.....	14
References	Error! Bookmark not defined.

How to get the app

Copy of the code available at <https://github.com/rhuarhri/NN4M.git>

Purpose of the document

This document is a guide to the companion app, how it works and the research that went into making the app.

1, App idea

The app is design to pair clothes that the user owns with clothes that are available to buy thus helping the user decide what clothes to buy.

The reason pairing clothes will help some decide is based on the idea that someone would be more likely to buy something if there is a reason for it (for example purchased because it looks good with that).

The app can pair clothes together through a combination of simply matching data and the use of colour theory.

Colour theory is a selection of rules using in design to determine how certain colours should be used. If colour theory is used correctly the colours used will look like they belong together.

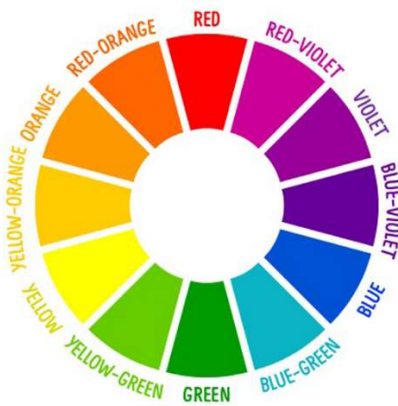


Fig 1.1 colour wheel used in colour theory

Out of the various rules related to colour theory the app uses two of them. This is because not all rules are applicable, for example the use of some colour can suggest things (like red meaning danger) which does not help this app.

Colour theory rules used

- Adjacent colours
Adjacent colours are colours that appear next to each other on a colour wheel (like green and blue-green in fig 1.1). Two colours that are adjacent to each other on a colour wheel will match. This can be expanded into clothing as two pieces of clothing that have colours that are adjacent to each other might look good together (see fig 1.2)



Fig 1.2 clothing with adjacent colours

- **Contrasting colours**

Contrasting colours are colours that appear opposite each other on a colour wheel (like red and green in fig 1.1). According to colour theory contrasting colours look good together. This also can be used in pairing two items of clothing together as can be seen in fig 1.3.



Fig 1.3 clothing with contrasting colours

Why use colour theory

As seen in fig 1.2 and fig 1.3 colour theory can be used with clothes, and with colour theory you can determine what clothes look good together. Since the main functionality of the app is to pair clothes together, using colour theory is one way the app could achieve that.

How the app interprets colour theory

The code that interprets colour theory is in a file called colour matcher code. It follows these steps.

1. The code gets the red, green and blue colour values of an item of clothing. These colour values are round to 255 or 0, this ensures that the code produces reliable set of colours.

2. This colour value is then copied and modified to create it's adjacent colours. Which is done like this. One of the zero values is selected and changed to 255 keeping the other values the same. For example, the start colour is (255,0,0) i.e. red which is converted to (255,255,0) i.e. yellow and (255,0,255) i.e. pink which will be the adjacent colours.
3. The colour value is then modified again to create the contrasting colours. Which is done by flipping the RGB colour values, for example (0,255,0) i.e. green becomes (255,0,255) i.e. pink.
4. Finally, all the adjacent colours and contrasting colours are added to a list. The colour of a second item of clothing will be checked against this list. If a match is found, then the clothes match.

Why having a reliable set of colours is important?

This is important as it means that a match can be easily found. For example, take the clothes in fig 1.3, these clothes will have a specific RGB colour value which if not modified will create equally specific adjacent and contrasting colours. This will result in the purple top will not matching the orange skirt as it's not the right shade of orange.

2, How it meets the requirements

The Development task

Start-up Screen

This is the equivalent to the welcome activity in the app which is the very first screen the user sees when they open the app.

Main menu page that has a button that progress to a list of products

The app does not have a specific menu page as the app has been designed to not need one. The reason it does not need one is because app automatically searches for clothes, as a result the user won't need a menu page to search for and find clothes.

The list of products will be generated from the JSON content which should do the following.

- **Show the name, image and price**
- **The image should be loaded from a URL and should only load as they become visible**
- **You should be able to tap on a product and go to another page showing a larger version of the image**

In the view cart activity and user clothing display activity the app does display a list of clothing items, but when the user is searching for new clothes the clothes are displayed one at a time.

The user can select a product which will take them to the item description activity where they will see the price, name, description and various images of the product. This information comes exclusively from a JSON file.

All images that are not icons come from the URLs in the JSON files and from the smart phone's internal storage and are processed on their own thread.

Go To user manual for information on how the app works.

3, Design decision

The design of the app's UI is based on two apps called Stylebook and Stitch fix. Both apps are clothing related. Stylebook is a clothing organiser that allows the user to pair clothes together and save them for future reference. Stitch fix is an automated clothing picker which finds a selection of clothes based on your personal style.

The main part of the app is the functionality to match clothes together, which is something that I want the user to be aware of. As a result, the app must be designed to fit that functionality. Stylebook and Stitch fix like my app have a main piece of functionality that they are designed around and how they did that is what I want to emulate in my app.

The main things that I took from these apps was the use of large images for displaying clothing (which comes from Stylebook), and white background and coloured buttons (which is in both but prominent in Stitch fix).

Custom JSON files

This app requires some colour value, which the json file provided did not have. As a result, new json files were created. They exist on a firestore database and are accessed by downloading a copy from a URL. Examples of which can be seen in the json section of this report.

As can be seen in this report there are two distinct json files. The one called clothing catalogue contains a list of objects that represents every clothing item available, and these objects contain only the data necessary for the search functionality of the app. The other file contains a description for a clothing item. Each clothing item has its own description file. This description file is accessed only when the user wants to see an item's description.

These json files could be combined into one large json file, however the app will have to store a lot of information that it does not need. For example, out of all the products available the user may only see a handful of them, as a result most products will not be seen by the user this includes their descriptions. These two json files help as it ensures that the app only has access to the information that it needs.

The information in the json files comes from the json file provided and was chosen at random.

4, Testing

If you are wondering if the app works go to `java/ExampleUnitTest` where you will see a selection of unit tests, which should all pass. However, what you will have noticed is that there are no instrumentation tests, because I can assume that an empty thread will work, and the simple database queries will return a result. All that is left is the app logic which is tested.

6, User Manual

The app has a limited number of clothes available, as a result you will most likely get no result. In order to get a result, complete the steps marked with **Do**. In the future the app would have hundreds of different clothes available so not getting a result will not be a problem.

Welcome screen

When you first open the app, you will be shown the welcome screen. The purpose of this screen is to welcome the user to the app as well doing checks in background.



Fig 6.1 welcome scree with set up required



Fig 6.2 welcome screen with setup done

Possibly the first thing you will notice is the large image on the left. This image is meant to show the user any special offers that are available. The image is not chosen at random in fact the image is chosen based on how they have set up the app. In this case the user has told the app that they are a woman so got an appropriate image. There are also images for men, girls and boys all of which come from a URL from the river island website.

In the background the welcome screen will check to see if app has been setup by checking if the internal databases are empty.

If setup completed. Pressing the shop button will bring you to the main screen. Pressing the search button will bring you to the quick search screen.

If setup not completed. Pressing any of the buttons will take you to the setup screen.

Quick search screen

The quick search screen is where the user can check if an item of clothing matches the clothes the user already owns from within the app.



Fig 6.3 quick search screen

This screen may not seem strictly necessary as it is not really linked to the main part of the app which is selling clothes found through colour theory. However, it exists in order to provide some functionality that the user might find useful, and it is an alternative way to use the app's existing functionality. Take this scenario for example. The user finds a nice jacket in a shop with this screen they can check if the jacket matches any of the clothes they already have, thus aiding in their decision to purchase.

The top half of this screen is a horizontal recycler view that displays all the images of the clothes that the user owns which match the input. All these images are stored on the smartphone and processed by a thread.

The bottom half of the screen is where the user inputs the required information by selecting one of the various options for each category, before pressing the search button. The search itself takes place on it's own thread where in which the images of the clothes that match the input are found and displayed on the top half of the screen.

Do

1. After doing all the **Do** tasks come back to this screen
2. For type select top, for colour select pink and for season select summer
3. Press the search button and an image of a red skirt (or the image you used instead of) should appear

Setup screen

The setup screen is where the user adds information about themselves like the size and age. This is used later to aid in the searching of clothing.



Fig 6.4 setup screen

In the background the setup screen will record what the user has selected. Then when the user presses the save button all the collected information is stored on the app's internal database, and the add screen will appear. In order to prevent the app from slowing down process of saving the data is handled by its own thread.

On the left-hand side of the screen there are two icons. These are meant to show the user where to measure and what measurement the app is referring to. This adding of information is also the reason why when you select a measurement it shows different versions of the same measurement.

Do

On this screen select the following

1. Gender as female
2. Age as adult
3. Chest as medium
4. Waist as medium
5. Shoe size as 6

Add screen

The add screen allows the user to input information about an item of clothing they have.

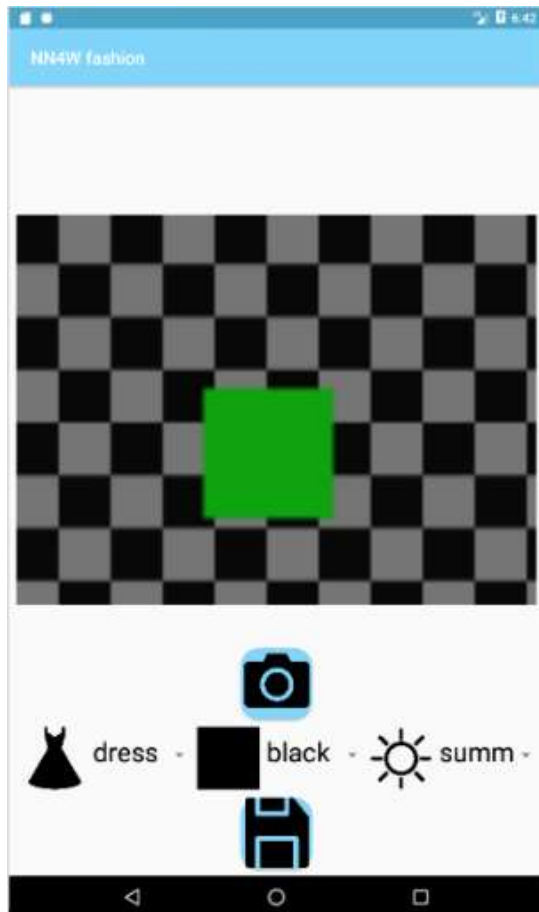


Fig 6.5 add screen

On this screen the user must add the type, colour and season (and or function) of the item of clothing being added. They also must take a picture of this item with the device's built in camera.

The image is used as an easy way for the user to identify the item of clothing. The rest of the information is used by the app to aid in searching. Like the setup screen all the information gathered is saved to an internal database which is carried out in it's own thread. Once saved the main screen will appear.

Do

On this screen do the following

1. Select dress as type
2. Select black as colour
3. Select summer as season
4. Take a picture

Main screen

The main screen is where the user see what products are available to them.



Fig 6.6 main screen

The two main parts of this screen are the two images. The top image is an image of a product supplied by image URL in the clothing catalogue json file and processed in it's own thread. The bottom image is a picture of one of the user's clothes supplied by the app's internal database and also processed in it's own thread. These two images are paired together through the app's understanding of colour theory and matching of the information gathered in the setup screen and add screen.

The screen only has four buttons the two top ones are for accessing the cart screen (which is the one on the left) and accessing a user clothing display screen (which is the one on the right). The two buttons at the bottom is how the user navigates few the various products. If they press the like button this displays the item description screen. If they press the dislike button, then the screen will move onto the next pair.

Do

Pick whatever pair looks good to you. There should be only two options available.

Item description screen

The item description screen displays information about a clothing item and allows them to save the clothing item to the app's cart.

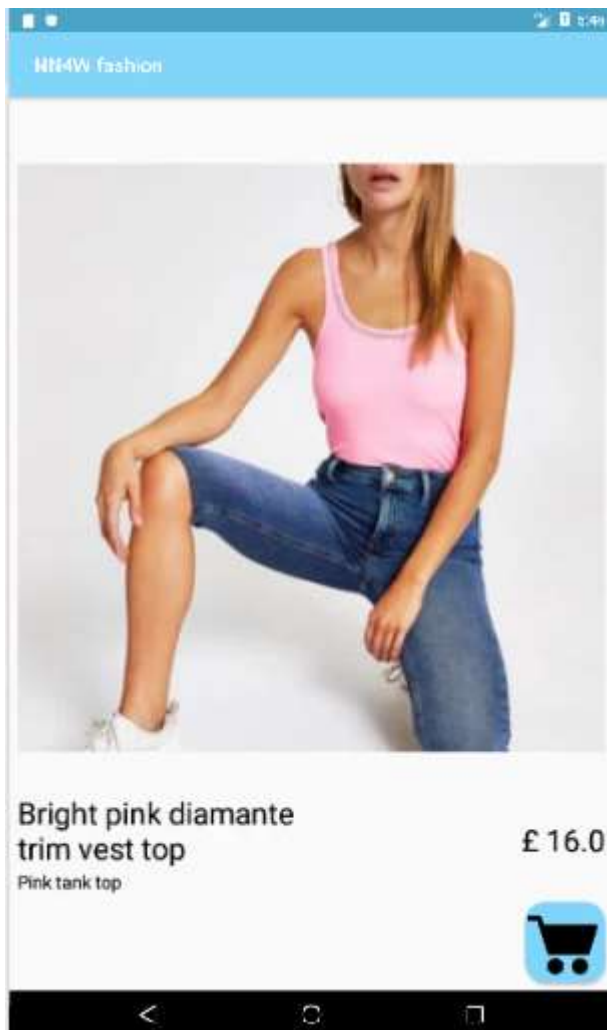


Fig 6.7 item description screen

The top half of the is dedicated to pictures of the selected clothing item. These pictures are displayed in a horizontal recycler view so the user can swipe left and right to view the images. All these images are processed and displayed with the help of multi-threading.

The bottom half of the screen contains the name of the item and it's description (on the bottom left) and the price and with the reduced amount (on the bottom right). The reduced amount is not a price value but instead the percentage difference between the old and new price (i.e. 50% off) it is not visible as no reduction was found.

Cart screen and check out screen

The cart screen can be accessed from the main screen and displays a list of items the user may want to buy.



Fig 6.8 cart screen

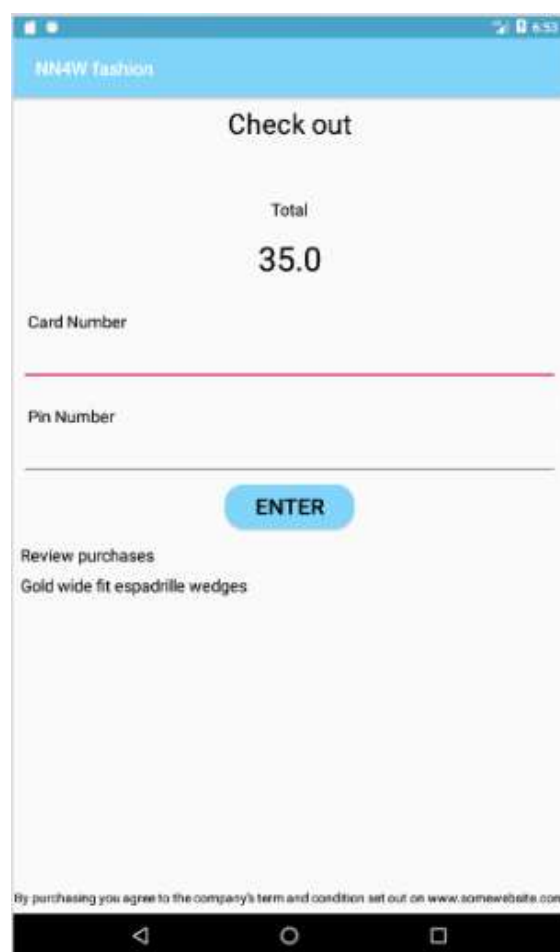


Fig 6.9 check out screen

The main part of cart screen is the list of clothing items. Each item in the list has an image (which comes from a URL and processed by it's own thread), name, price, delete and buy buttons.

If the user presses either the buy all button or the buy button on the item, then they will be shown the check out screen which will show the amount they have to pay, a way to pay it and a list of the items they are going to buy. This screen only exists for demonstration purposes so does not do a lot. The small line of text at the bottom informs the user of the app's terms and condition. When the user presses the enter button a pop up will appear informing the user of the terms and conditions and checks if they agree to them.

User clothing display screen

The user clothing display screen can be accessed from the main screen and displays a list of all the clothes the user has saved.



Fig 6.10 User clothing display screen

The main part of this screen is the item list. Each item in the list contains the type of the item, its related season, a delete button, and an image which comes from the app's internal database and like the cart screen this image is processed by its own thread.

The two top buttons are the settings button which does not do anything and the account button which when pressed displays the setup screen and allows the user to update their information. The button at the bottom is the add button which allows the user to add more clothes as time goes on.

Do

1. Press the plus button at the bottom
2. This will display the add screen
3. Select skirt as type
4. Select red as colour
5. Select summer as season
6. Take a picture

This will add a new item to the user clothing display screen and allow the app to find more clothing matches which you can see in the main screen.

7, JSON files

Clothing catalogue.json

```
{
  "clothing":[
    {
      "colour":"0x000000",
      "gender":"female",
      "season":"summer",
      "type":"dress",
      "maxSize":"16",
      "minSize":"6",
      "image":"https://images.riverisland.com//is//image//RiverIsland//739288_rollover",
      "age":"adult",
      "description":"https://firebasestorage.googleapis.com/v0/b/nn4mfashion.appspot.com/o/black%20print%20playsuit.json?alt=media&token=e341d428-e109-471c-bc81-a8d28af1d335"
    }
  ]
}
```

Companion description json file

```
{
  "name":"Caroline Flack black print playsuit",
  "cost":"46",
  "wascost":"",
  "costEUR":"60",
  "wascostEUR":"",
  "costWER":"60",
  "wascostWER":"",
  "costUSD":"92",
  "wascostUSD":"",
  "costAUD":"92",
  "wascostAUD":"",
  "costSEK":"649",
```

```
"wascostSEK":"","  
"costWEK":"649",  
"wascostWEK":"","  
  
"allImages":["https:\\\\images.riverisland.com\\is\\image\\RiverIsland\\739288_main","https:\\\\i  
images.riverisland.com\\is\\image\\RiverIsland\\739288_rollover","https:\\\\images.riverisland.com  
\\is\\image\\RiverIsland\\739288_back","https:\\\\images.riverisland.com\\is\\image\\RiverIsland\\  
/739288_alt1","https:\\\\images.riverisland.com\\is\\image\\RiverIsland\\739288_alt2"],  
  
"description":"Black with white bots dress"  
}
```

References

- Aiezza, M. (2019, July 13). *Color And Fashion. Color Matching 101: Make Great Clothing Matches by Understanding Color Theory*. Retrieved from myclothinghelper.com:
<https://myclothinghelper.com/color-matching-101-make-great-clothing-matches-by-understanding-color-theory/>
- Stitch Fix, I. (2019, July 14). *Stitch Fix: Fashion Stylist*. Retrieved from apps.apple.com:
<https://apps.apple.com/us/app/stitch-fix-fashion-stylist/id1022579925>
- Stylebook. (2019, July 14). *stylebook*. Retrieved from stylebookapp.com:
<http://www.stylebookapp.com/index.html>