FINAL REPORT

TEAM 2

CONTENTS

1 ORIGINAL IDEA

When we gathered together to discuss what app to accomplish, we decided to do something new, something that nobody did before. We thought about making an app that can help us manage our account, but many companies did it before. At last, we came up with the idea that we could make an app that enables people to match their clothes. Nowadays, a lot of young men are worried about how to dress themselves, and we believe we can figure out an algorithm to realize our thoughts.

2 FUNCTIONS

GENERAL STEPS The user can input the pictures of his clothes followed by the statistics such as type, thickness, material and color. After pressing the button "get a suit", the app will pop up a window and the use can choose the style he wants for that day, and the styles are casual, sport, business and pajama, and he will also choose a favorite match from the library. After receiving the command from the user, the app begin to do traversal work, using the algorithm designed by ourselves to judge if the result match the temperature. Simultaneously, the app will calculate the integrating degree between every traversal result and the match user chooses from the library, and give the best result according to the user's favorite match. And if the user doesn't like it, he can get the second best result and so on.

ASSUMED STEPS

1. Input all the clothes with parameters
   Two choices of pressing buttons
   Input a picture of user's clothes
   Entering all the characteristics
2. Passing user's favor to the app
   Pressing the "Get a suit" button
   Choosing the style the user will wear
   Letting the app know user's favorite
3. Match clothes according to temperature and user's favor using algorithm
4. Present recommended schemes in the order of integrating degree
   The best match will appear on the screen
   If satisfied, just press the "yes" button
   If not satisfied, just press the "next" button

3 ALGORITHM

3.1 traversal

First part is to choose clothes according to temperature
According to the temperature, we can get all the suitable results when the style of the match is determined. This part is relatively easy. We just need to get every match for top clothes and pants, and check if the thickness of clothes fit the temperature. And all the combination which are OK will be stored in a list.

3.2 matching

And the second part are as follows.

We use some knowledge of algorithm of fuzzy clustering. But now include clustering part. We implement the part before clustering.

Steps of this algorithm are below:

1. Choose attributes
   It is very crucial to choose proper attribute which can directly influence the result of algorithm. After careful consideration, we choose these attributes for our clothes: color, length and thickness.
2. Regularize data
   When solving practice problem, due to different metrics and magnitude of number which vary from attributes to attributes, we need to regularize our attribute data in order to avoid situation that bigger data has bigger influence. Through proper regularization, we change the data into same scale. For this program, we use average regularization method.
   The formula is below:

$$X_{ij}{}' = \frac{X_{ij} - X_{min}}{\bar{X_i}}$$

3. Calculate matching degree
   We first define that

   Each cloth is presented as X_i (i is for cloth number).

   Each attribute is presented as X_{ij} (j is for attribute number)

   And matching degree between two clothes is denoted as r_{ij}

   Denote: r is between 0 and 1.1 is for perfectly match and 0 is for definitely not match.

   To calculate this matching degree, we use length of Euclid space. Formula is below:

$$r_{ij} = 1 - b\sqrt{\sum_{k=1}^{m}(X_{ik} - X_{jk})^2}$$

   b is a proper number. In this program we choose b as 1/4 Then we choose the largest matching degree to be the best match cloth.

4 UI

Since our project is a practical functional tool, and it is designed to cater for not only developers or specialists but ordinary people, we consider beauty and accessibility in the first place. In this case, material design is a good choice for our UI style. It's an offical UI designing standard released by Google.

Therefore, we use a hug amount of "design widget", such as "Navigation Drawer" "Floating Action Button", and "Cardview". Here are several screen shots of our app.

However, a problem arise when we are using "Intent" to make transitions between every to sections. We found that a drawer will be destroyed before it is completely closed, and the target activity will take the place. It causes a visible pause affecting user experience

To solve this problem, we use Fragment to make trasitions between two sections. Then we can overwrite the coClick() method to make the drawer close and the fragment changed. With this method, the pause doesn't exist anymore.

5 CONCLUSION & ACQUISITION

Firstly, if we want to develop Android apps, we must use Java, which is a new language to most of us, so we need to learn some basic knowledge of it. Having learnt two languages: C++ and Python, it's easy to master the similar parts. But Java has some new functions we haven't got in touch with, and in the process, we gain some core ideas. After developing the app, we've already lay a solid foundation for future study on Java.

Secondly, since it's a big project, the teacher can't teach us everything, he invited some of his senior students to give some lectures for us. These lectures didn't cover many contents, but they actually inspired us, especially some of them talked about the projects they were working on, and they indeed broadened our horizons.

6 FUTURE WORK

1. Algorithm
   When thinking about what topic to choose, we take for granted that we could find highly-developed algorithm (the first part) to select proper clothes according to the temperature. But to our disappointment, there is only an empirical formula, because it's a new area that nobody conducted research on it. So if there are some researches carried out later, our algorithm can be improved, and the schemes can be more accurate.
2. Picture Recognition API
   Similarly, we also thought that we can find proper APIs to process the pictures of clothes in the library and obtain all the information of pictures, so the user doesn't need to input the parameters of clothes. But what a pity that free APIs don't have expected influences, and other possible APIs charge a lot. So we have to compromise and let users input the characteristics.
3. More Complicated Matches
   At first, when we were designing the algorithm, the first part of which covered matches containing 2 to 7 pieces of clothes corresponding to different temperatures. But due to the limit time, the UI part can't finish that complex work. So we simplify the algorithm and recommend matches containing a piece of top clothes and a pair of pants.

7 SCREEN SHOTS

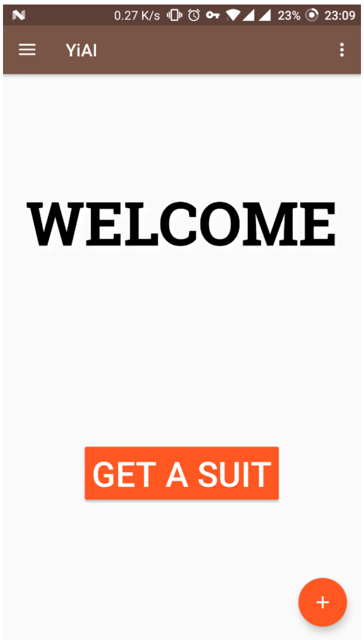


Figure 1: homepage
"GET A SUIT" button: press the button to start the program the button with a "+":
input clothes to the wardrobe

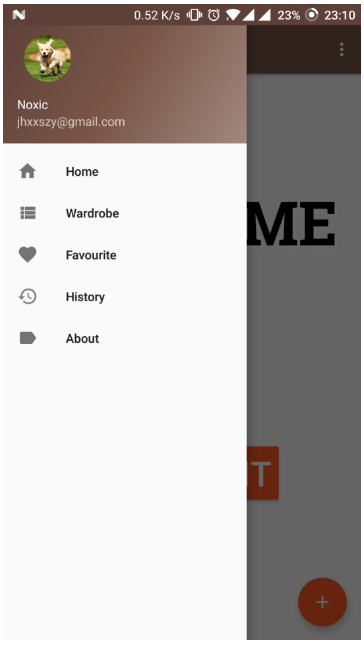


Figure 2: sidebar: all the functions

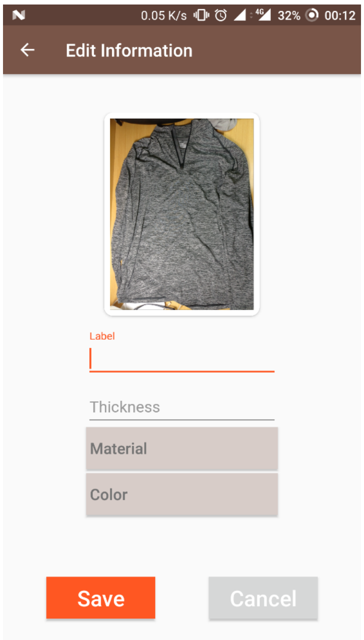
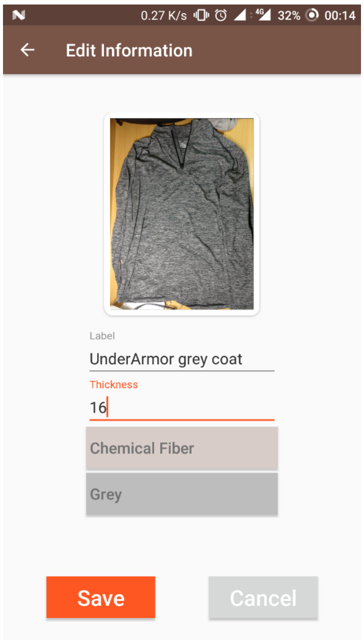


Figure 3: take photos of clothes and input the parameters



Label
UnderArmor grey coat

Thickness
16

Chemical Fiber

Grey

Figure 4: an example of the clothes

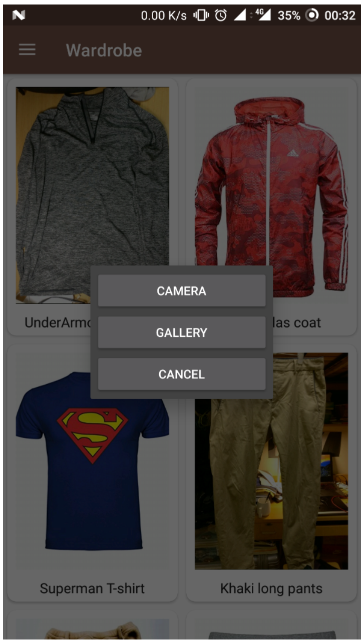


Figure 5: Operation to the wardrobe

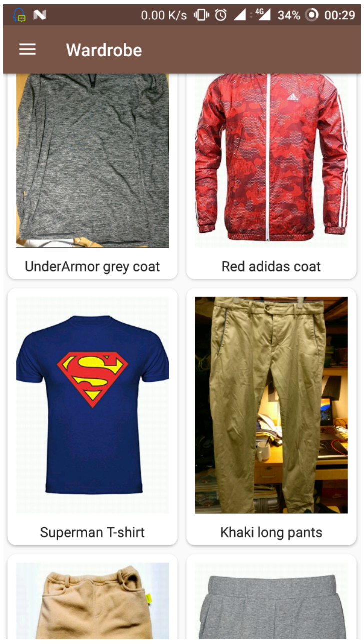


Figure 6: an example of the wardrobe

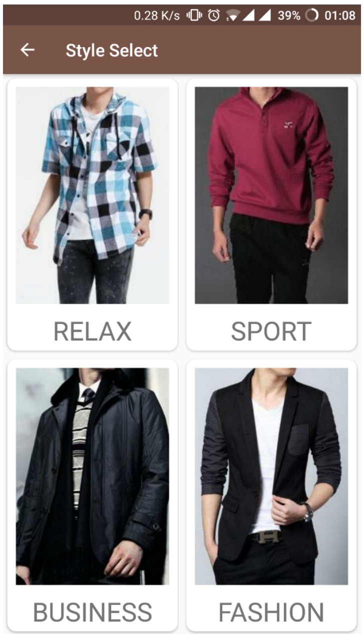


Figure 7: Choose a style for a particular day

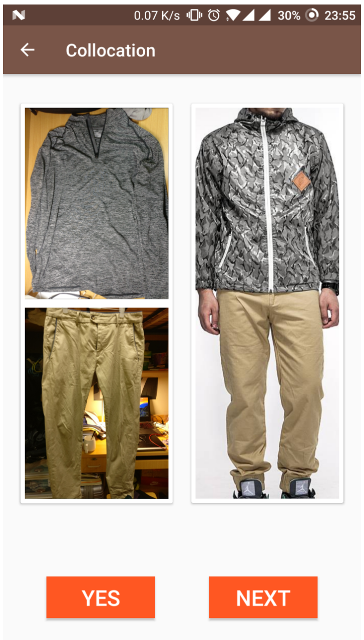


Figure 8: Final result