

# **“Visualist” Software Design Document**

# Catalog

<b>“Visualist” Software Design Document .....</b>	<b>1</b>
<b>1. Introduction .....</b>	<b>4</b>
<b>2. Requirements Specification .....</b>	<b>4</b>
2.1 product positioning .....	4
2.2 Target group .....	4
<b>3. Over Design .....</b>	<b>5</b>
3.1 Structure .....	5
3.2 Shooting process and functions .....	6
3.3 Mall function introduction .....	9
3.4 Forum Function Introduction .....	9
3.5 Personal Interface .....	11
<b>4. User Interface Design .....</b>	<b>11</b>
4.1 Concept Outline .....	11
4.2 Design details .....	12
<b>5. Key Technologies .....</b>	<b>14</b>
5.1 Encapsulation of shooting functions .....	14
5.2 Forum Function Encapsulation .....	15
<b>6. Testing and User Experience Analysis .....</b>	<b>17</b>
6.1 Standard compatibility testing .....	17
6.2 Instructions for using research .....	18

<b>7. Problem and Solution .....</b>	<b>24</b>
<b>8. Development planning .....</b>	<b>24</b>
<b>9. Summary .....</b>	<b>25</b>

# 1. Introduction

This article discusses the design framework and different aspects of the "Visualist" system. The requirements for 'Visualist' will also be mentioned in this article, which includes all the functional and performance requirements required to introduce the system. Next, we will elaborate on the overall design and user interface development of 'Visualist'. The article will also mention key technologies, as well as testing and user experience.

The overall design structure of 'Visualist' is based on user needs. It is necessary to fully consider factors such as the ease of use, stability, and performance of the system. When designing different modules of 'Visualist', we should try to be as clear and clear as possible to facilitate future development and maintenance. In addition, the design of the user interface is also a crucial aspect in system development, as it directly affects the user experience and impression of the entire system.

In this article, we will emphasize several key technologies, especially in the design of the "Visualist" architecture and user interface, which can help developers complete their work more efficiently. At the same time, we will also conduct in-depth analysis of the importance of testing and user experience to ensure the stability and user satisfaction of the system in use.

In summary, this article will elaborate on the design framework and different aspects of "Visualist" to help investors and other stakeholders better understand its development process and key ideas.

## 2. Requirements Specification

### 2.1 product positioning

This product is a multifunctional photography mobile application for amateur photographers, creative photographers, and aesthetic enthusiasts. This application not only provides functions such as expert image comparison and shopping mall, but also includes guidance on photography skills, sharing communities, and artistic creation. Our goal is to provide a comprehensive photography ecosystem, providing users with a better photography experience and learning platform.

### 2.2 Target group

This product is aimed at amateur photographers, creative photographers, and aesthetic enthusiasts. Target users can be divided into the following categories:

Amateur photographer: This type of user has basic photography skills but hopes to improve their professional level. They are concerned about the latest developments in the photography industry and need a platform for guidance and sharing in order to gain more inspiration and knowledge.

Creative photographers: These users possess creativity and professional skills, as well as their own artistic style and reputation. They need a mobile application that facilitates their creativity and sharing, as well as gaining more market opportunities.

Aesthetic enthusiasts: This type of user focuses on the art and aesthetic value of photography, and enjoys appreciating and learning photography works. They need a platform for identification, evaluation, and sharing in order to better understand and appreciate photographic works.

In summary, this product aims to provide a comprehensive photography ecosystem for amateur photographers, creative photographers, and aesthetic enthusiasts to meet their various needs and interests, and provide convenience and support for their learning, creation, and communication.

## **3. Over Design**

### **3.1 Structure**

Entering Visualist, there will be a simple boot animation, and clicking on it will enter the official login page. After entering the individual's account and corresponding verification code, you can log in to the application.

After successful login, the default entry is the Camera page. There is a row of navigation bar at the bottom of the application to jump between the main functions. The bottom navigation bar has four options, corresponding to four main functions: Camera, Forum, Store, and Home. The bottom navigation bar is implemented using the BottomNavigationView, and the display of content above the navigation bar is implemented in the fragment. Whenever an option is selected in the navigation bar, it switches to the corresponding fragment to display the corresponding content.

Visualist's photography function: Users upload photos, and by analyzing the uploaded photos, Visualist provides feedback to users on photos taken by professional photographers in the same scene. Users can improve their photography skills and aesthetic level by learning the composition, color, angle, and other aspects of the feedback images. At present, the feedback image function is achieved by connecting to the Baidu AI image recognition API. We have uploaded a large number of works of photography artists in the API's image library for them to provide more accurate and effective feedback. On the Camera page, a visually impactful button is mainly set as the entrance to the secondary page for shooting. After entering the secondary page, users can choose to use the photos in the album or upload ready-made photos using the camera function to receive feedback. After selecting the image, click the confirm button, and Visualist will start analyzing and processing the image. Once completed, it will return to the processed photo. On the feedback page, users can click the share button to share the image, or click the confirm button to save the image to the "My Works" section of their personal page. If they are not satisfied with the effect or want to learn other images, they can also click the return button to return to the main page.

Visualist's forum function: Users can view images posted by others, click to enter the image view details to see the complete image and the author's comments and insights. Visualist also provides users with the option to download as wallpaper. After clicking on download, the image will automatically be saved in the "History

Download" section of their personal page. Users can click on the hover button in the bottom right corner to publish their own images, select the desired image to be posted, and attach a message. After clicking on the post, they can see the latest post on the forum page. The images and text in the forum are stored in the database.

Visualist's shopping mall function: Users can select and purchase products on the shopping mall page. At the top of the mall page is a search box used to search for products. Below the search box is a list of random products, where users can select their desired product and click to enter the product details to learn more about that product. After clicking to enter the product details, larger product images and detailed descriptions will be presented. At the bottom, options for adding to the shopping cart and purchasing directly are provided for selection.

Visualist's personal features: This page stores users' personal information, including nicknames, signatures, works, downloads, etc. Users can view and modify their personal information on this page.

## **3.2 Shooting process and functions**

Shooting is divided into photo shooting and video shooting. Photo shooting process: adjust the camera properties, click the "shoot" button, then the picture is successfully shot, the user can re-shoot, save.

### **3.2.1 Detail description of shooting function**

1. Click the shooting page, press the camera button to enter the shooting page, press the album button to enter the album page. (figure 3.1)

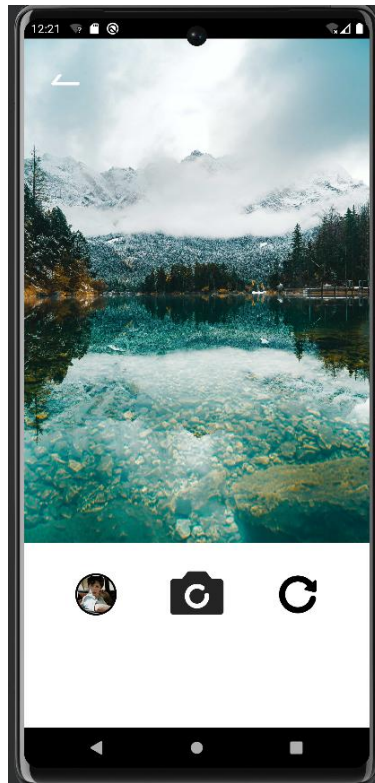


figure 3.1

2. Select the right Angle and composition and press the shoot button. (figure 3.2)



figure 3.2

3. Here is the page to confirm the photo. Press "√" to confirm the photo, press the back sign to take it again, and press "×" to return to the start page. After confirming

the photo, users can send the photo to the platform for expert picture comparison, or send the photo to the forum for sharing with the vast number of netizens. (figure 3.3)

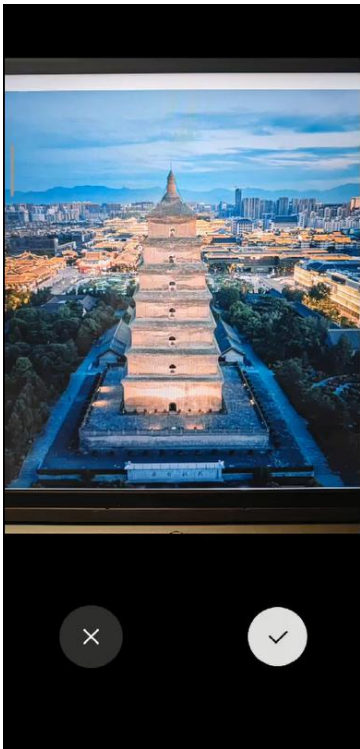


figure 3.3

3.2.2 Detailed introduction of photo album function

If you select the album button, you can select photos from the album of the phone, and then follow the steps similar to shooting. (figure 3.4)



figure 3.4



### 3.3 Mall function introduction

In the mall, you can see a variety of products recommended by "Visualist". If you see something you like, you can click on it to see the details, and then choose to add it to your cart or buy it. (figure 3.5)



figure 4.5

### 3.4 Forum Function Introduction

In the forum, users can view the forum shared by other users, and also share their own photos and thanks on the forum.

#### 3.3.1 View other forums

Users can click on the forum screen to view forums shared by other users. (figure 3.6)



figure 3.6

### 3.4.2 Send function

Click the Development Forum button, select the photo you want and add the text, then click Publish. Then you can create your own post. (figure 3.7)



figure 3.7

## 3.5 Personal Interface

In the personal interface, users can view many basic app functions. For example, my work, history, review progress, help feedback, Settings and other functions. (figure 3.8)

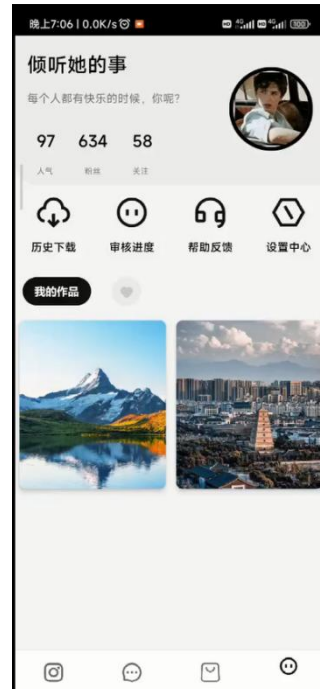


figure 3.8

## 4. User Interface Design

### 4.1 Concept Outline

#### 4.1.1 Style selection

Minimalist black and white style. The overall UI design adopts a black and white style, with a focus on displaying functional goals through simple black and white icons, and conveying software functions through a minimalist style.

#### 4.1.2 Partial content display

1. Upper left: Changing the page display through black and white transformation
2. Upper right: Minimalist style for easy layout design
3. Bottom left: Black and white tones make it easier to unify the style
4. Bottom right: Minimal lines are more prominent in expressing content (figure 4.1.2)



figure 4.1.2

### 4.1.3 Design style

1. User Guide: Collect user feedback and strengthen UI design orientation.
2. Clear guidance: Clarify icon details and guide users correctly.
3. Uniform style: Ensure the consistency of UI design style and ensure that all parts adopt a consistent form.

## 4.2 Design details

### 4.2.1 Rich animation design

Many animation designs are used in UI design, providing timely feedback when users click to optimize the user experience. (figure 4.2.1)



figure 4.2.1

#### 4.2.2 Partial design details

Black and white variation design (figure 4.2.2.1)



figure 4.2.2.1

User click prompt (figure 4.2.2.2)



figure 4.2.2.2

Continuous display of components (figure 4.2.2.3)



figure 4.2.2.3

Button action feedback (figure 4.2.2.4)



figure 4.2.2.4

## 5. Key Technologies

### 5.1 Encapsulation of shooting functions

Determine if you have camera permissions first.(figure 5.1)

```
*/
if (!AndPermission.hasPermissions( activity: this, *Permission.Group.CAMERA)) {
/**
 * AndPermission: 引用权限相关库
 * onGranted: 允许权限
 * onDenied: 拒绝权限
 */
// 动态申请权限
```

figure 5.1

If not, apply for camera permissions. (figure 5.2)

```
// 动态申请权限

AndPermission.with( activity: this).runtime().permission(*Permission.Group.CAMERA)
.onGranted { openCamera() }
.onDenied { denied: List<String?>? ->
    if (denied != null && denied.isNotEmpty()) {
        for (i in denied.indices) {
            if (!shouldShowRequestPermissionRationale(denied[i]!!)) {
                Toast.makeText( context: this, text: "没有拍摄权限", Toast.LENGTH_SHORT)
                    .show()
                break
            }
        }
    }
}.start()
```

figure 5.3

1. Call `launcher.launch("image/*")` to open the image selector and select an image. (figure 5.3)
2. Get the selected image Uri and save it in `Constants.mPictureUri`.
3. Use the `MediaStore.Images.Media.GetBitmap ()` according to the Uri Bitmap, preserved in the mPicture.
4. Use the `uriToFile ()` according to the Uri file, the file path in the `worksPicturePath`.
5. Call `toCameraResultFragment()` to jump to the Fragment that displays the picture.

```

launcher.launch(input: "image/*") { it: Uri?
    it?.let { it: Uri
        Constants.mPictureUrl = it
        val photoBmp: Bitmap = MediaStore.Images.Media.getBitmap(
            view.context.contentResolver,
            it
        )
        Constants.mPicture = photoBmp

        uriToFile(it, view.context).let { file ->
            if (file != null) {
                Constants.worksPicturePath = file.path
            }
        }
        toCameraResultFragment()
    }
}

```

figure 5.4

Initialize the instance of the search image.

```

val APP_ID = "34477461";
val API_KEY = "2Fuj46MGjUhbjIBBGgAU5bm3";
val SECRET_KEY = "XKiGWAekqFmXyWKjRaHfTU8ri0rXBSSG";

// 初始化一个AipImageSearch
client = AipImageSearch(APP_ID, API_KEY, SECRET_KEY)
// 可选: 设置网络连接参数
client.setConnectionTimeoutInMillis(2000);
client.setSocketTimeoutInMillis(60000);

```

figure 5.5

1. In the IO thread, open the input stream inputStream corresponding to Constants.mPictureUrl.
2. Create the byte array output stream byteArrayOutputStream and read and write the data in the inputStream to the byteArrayOutputStream.
3. Convert byteArrayOutputStream to byte array file.
- 4 Upload file for face recognition using similarSearch API of Face ++.
5. If the recognition result res has result, get the brief(image Url) of the first recognition result.
6. Download the picture using brief and save it to the Pictures folder on SD card. The file name is current time in milliseconds +.jpg.
7. If the download is successful, use BitmapFactory to read the local picture file and display it on ImageView in the Main thread and hide the progress bar.

## 5.2 Forum Function Encapsulation

ForumFragment.Kt - Forum home page

Load forum data(figure 5.6)



```
viewModelScope.launch { this: CoroutineScope
    val list = withContext(Dispatchers.IO) { this: CoroutineScope
        val data = mutableListOf<UserAndPostEntity>()
        userDao.getAll().forEach { user ->
            val posts = postDao.getAllByUid(user.uid)
            if (posts.isNotEmpty()){
                data.add(UserAndPostEntity(user, posts))
            }
        }
        data ^withContext
    }
    list.reverse()
    _userAndPost.value = list
}
```

figure 5.6

The forum page is nested using RecyclerView and RecyclerView has a horizontal sliding list for each vertical sliding list.

The adapter logic for RecyclerView is in VerticalAdapter.Kt  
Forum add data EditActivity.kt

Click the button to take a photo with the system camera and save the photo results.  
(figure 5.7)

```
launcher.launch(input:"image/*") { data->
    data?.let { it ->
        uriToFile(data,view.context).let {file ->
            if (file != null) {
                path = file.path
            }
        }
        val photoBmp: Bitmap = MediaStore.Images.Media.getBitmap(
            view.context.contentResolver,
            it
        )
        binding.ivPicture.setImageBitmap(photoBmp)
    }
}
```

figure 5.7

Finally get the input content of the post and save the photo to the database. (figure 5.8)

```
private fun save(){
    GlobalScope.launch(Dispatchers.IO) { this: CoroutineScope
        val content = binding.etContent.text.toString()
        val post = PostEntity(path,content, tag: false, System.currentTimeMillis(), likeCount: 0)
        appDatabase.insertPost(post)
        finish()
    }
}
```



figure 5.8

```
while (inputStream.read(buffer).also { len = it } != -1) {  
    byteArrayOutputStream.write(buffer, off: 0, len)  
}  
val file = byteArrayOutputStream.toByteArray()  
inputStream.close()  
  
val res = client.similarSearch(file, options)  
println(res.toString(indentSpaces: 2))  
if (res.has(name: "result")) {  
    val result = res.getJSONArray(name: "result")  
    val imageUrl = result.getJSONObject(index: 0).getString(name: "brief")  
    val path = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES)  
    val file = File(path, child: "${System.currentTimeMillis()}.jpg")  
    file.createNewFile()  
    val success = loadImage(imageUrl, file)  
}
```

figure 5.9

## 6. Testing and User Experience Analysis

### 6.1 Standard compatibility testing

Select testing platform: WeTest

General testing situation: Number of testing equipment: 20

Equipment pass rate: 90% (Figure 6.1.1)

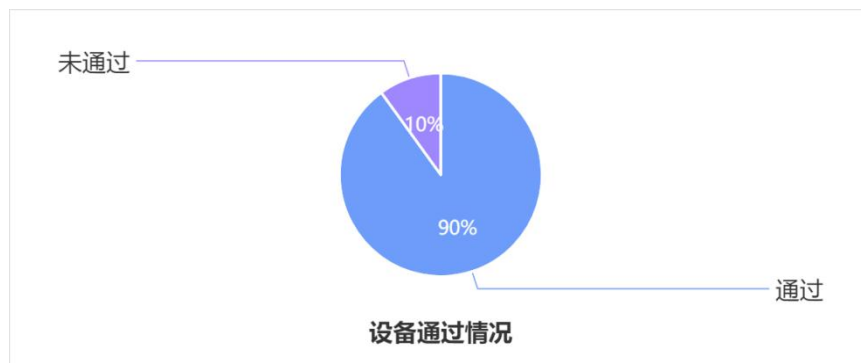


figure 6.1.1

Compatibility Failure Device Distribution: (figure 6.2)

java.lang.OutOfMemoryError(VMRuntime.java:-2)

java.lang.RuntimeException(DisplayListCanvas.java:229)

java.lang.RuntimeException(DisplayListCanvas.java:267)

设备品牌	设备型号	问题描述	推测原因
vivo	V1818A	java.lang.RuntimeException	图片加载超时
oppo	A83t	java.lang.OutOfMemoryError	图片内存溢出

figure 6.1.2

## 6.2 Instructions for using research

Select questionnaire platform: Questionnaire Star

Set number of questions: 12

Effective sample collection: 25 copies

### 6.2.1 Distribution of mobile phone brands

The usage of mobile phones is relatively average, and there are all types of Android phones available. Uniform distribution of mobile phone brands (figure 6.2.1)

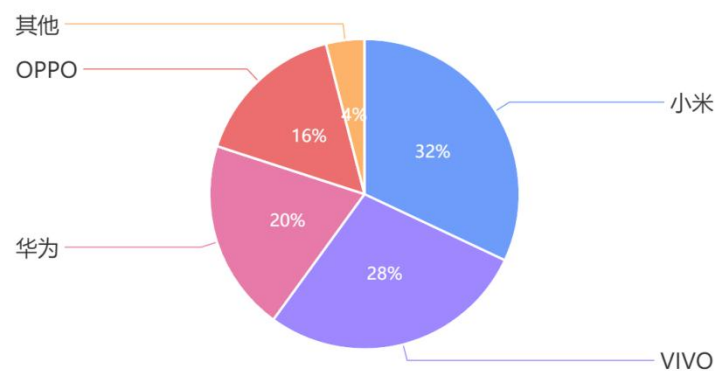


figure 6.2.1

### 6.2.2 APP experience duration

Many people spend time on our app and experience its features for reasons. 84% of users have a complete experience process. (figure 6.2.2)

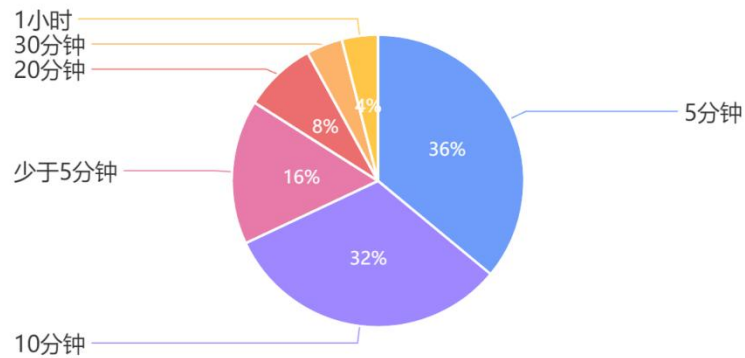


figure 6.2.2

### 6.2.3 Photography function

By analyzing user feedback data from photography applications, we extracted key vocabulary from four aspects - functionality, UI design, convenience, and practicality, and quantitatively evaluated these keywords. Our data results are very exciting, with high scores in all aspects. Users are very satisfied with the functionality and UI design of the application, and also believe that the application is very convenient and practical. This indicates that our product has met the expectations of users and received their recognition and support. This is also our confidence and motivation to continue working hard to improve product quality and user experience.

Function: 71% of users give a general or above rating.(figure 6.2.3.1)

UI design: 95% of users give a general or above rating.(figure 6.2.3.2)

Convenience: 81% of users gave a general or above rating.(figure 6.2.3.3)

Practicality: 62% of users gave a general or above evaluation.(figure 6.2.3.4)

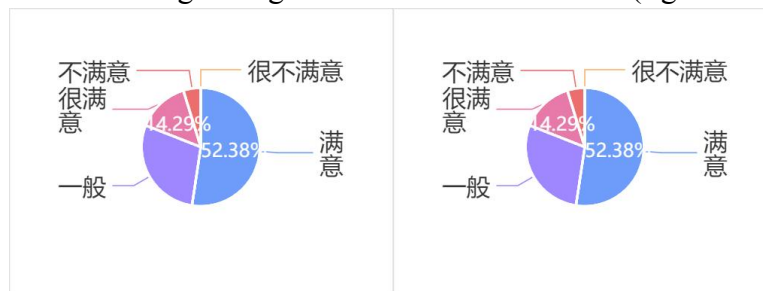


figure 6.2.3.1

figure 6.2.3.2

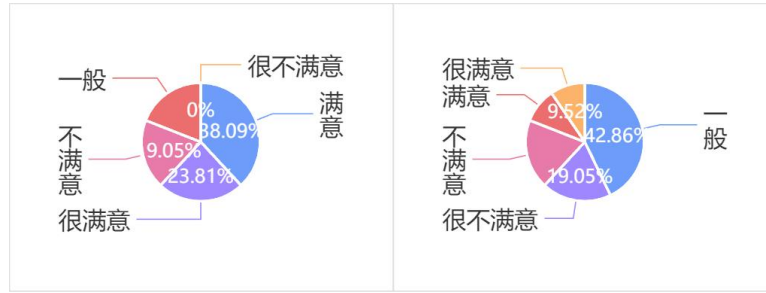


figure 6.2.3.3

figure 6.2.3.4

## 6.2.4 Forum Functions

We analyzed user feedback data on forum functions from four aspects - functionality, UI design, convenience, and practicality of mobile data. We are very pleased to find that our forum application scores very high in these aspects, and users are very satisfied with the functionality and UI design of the application. At the same time, we also believe that the application is very convenient and practical. This indicates that our application provides a good community platform for users and has met their expectations. This is also our confidence and motivation to continue to strive to improve product quality and user experience.

Function: 83% of users give a general or above rating.(figure 6.2.4.1)

UI design: 96% of users give a general or above rating.(figure 6.2.4.2)

Convenience: 100% of users give a general or above rating.(figure 6.2.4.3)

Practicality: 86% of users gave a general or above evaluation.(figure 6.2.4.4)

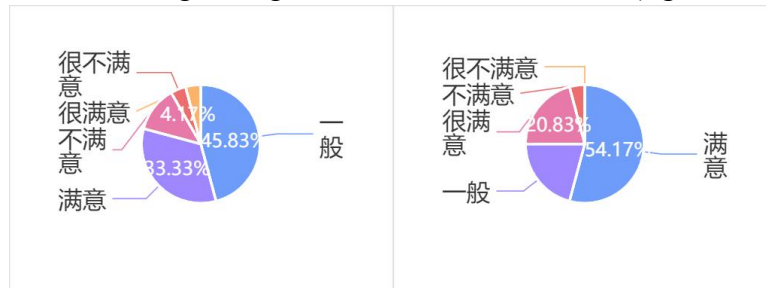


figure 6.2.4.1

figure 6.2.4.2

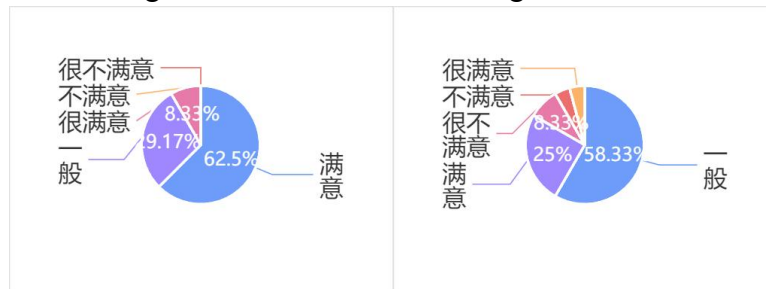


figure 6.2.4.3

figure 6.2.4.4

## 6.2.5 Mall functions

We analyzed user feedback data on mall functions and collected data from four aspects: functionality, UI design, convenience, and practicality. We are pleased to find that our e-commerce application scores very high in these aspects, and users are very satisfied with the functionality and UI design of the application. At the same

time, compared to other mall functions, our application is more convenient and practical. This fully demonstrates that our application provides users with a good mall functions platform and has met their needs and expectations. At the same time, we will continue to work hard to improve product quality and user experience, continuously improve and perfect application functions, and provide users with a better shopping experience.

Function: 95% of users give a general or above rating.(figure 6.2.5.1)

UI design: 100% of users give a general or above rating.(figure 6.2.5.2)

Convenience: 95% of users give a general or above rating.(figure 6.2.5.3)

Practicality: 100% of users give a general or above rating.(figure 6.2.5.4)

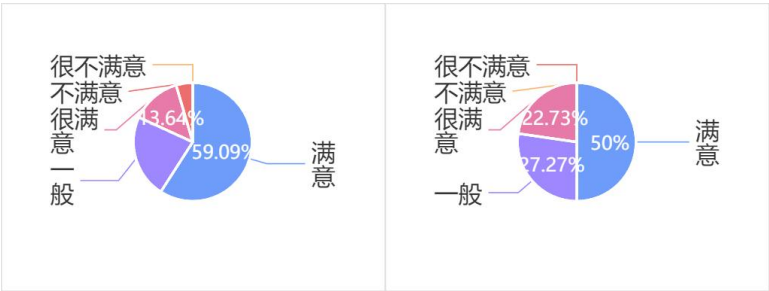


figure 6.2.5.1

figure 6.2.5.2

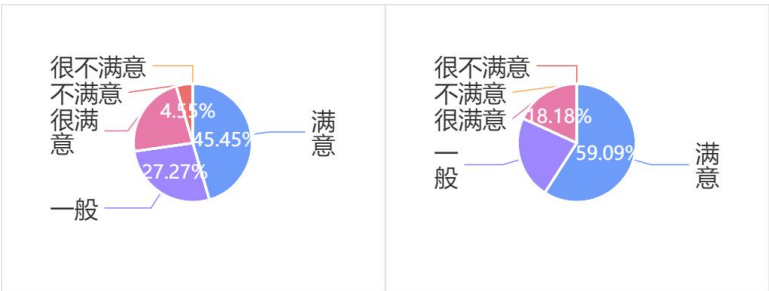


figure 6.2.5.3

figure 6.2.5.4

### 6.2.6 Personal interface functions

We analyzed user feedback data on personal interface functions and collected data from four aspects: functionality, UI design, usability, and practicality. We are pleased to find that our personal interface functions scores very high in these aspects, and users are very satisfied with the functionality and UI design provided by the function. At the same time, we also believe that the function is very easy to use and practical. This indicates that our personal interface function provides a good platform for users, meeting their needs and expectations. And we will continue to strive to improve product quality and user satisfaction, while continuously optimizing the functionality and user experience of the application. We hope that our future development can better serve users and provide them with the best personal interface functions.

Function: 95% of users give a general or above rating.(figure 6.2.6.1)

UI design: 95% of users give a general or above rating.(figure 6.2.6.2)

Convenience: 100% of users give a general or above rating.(figure 6.2.6.3)

Practicality: 100% of users give a general or above rating.(figure 6.2.6.4)

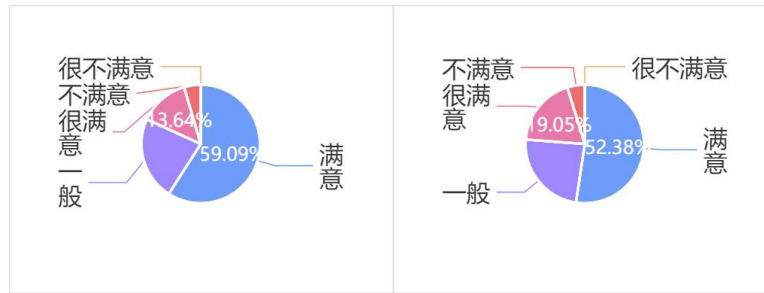


figure 6.2.6.1

figure 6.2.6.2p-



figure 6.2.6.3

figure 6.2.6.4

### 6.2.7 Overall functionality

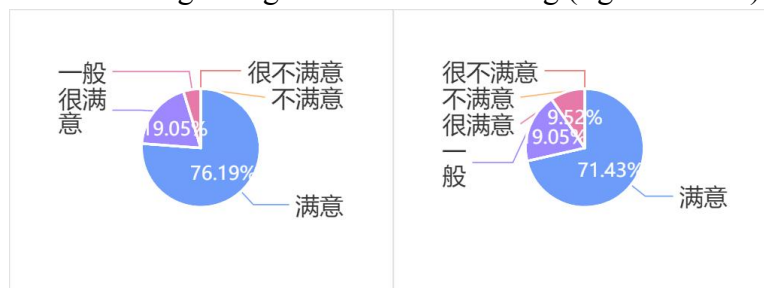
We conducted user feedback data analysis on our overall functional application from multiple perspectives and conducted a comprehensive evaluation of functionality, UI design, usability, and practicality. We are very pleased to find that our application scores very high in these aspects, and users are very satisfied with the functionality and UI design of our application. At the same time, we also believe that the application is very easy to use and practical. This indicates that our application provides users with comprehensive functionality and experience, meeting their requirements and expectations. We will continue to strive to improve product quality and user experience, and continuously improve and optimize application functions to provide users with a better user experience and services. We hope to better serve users and become a better application in the future.

Function: 100% of users give a general or above rating.(figure 6.2.7.1)

UI design: 100% of users give a general or above rating.(figure 6.2.7.2)

Convenience: 100% of users give a general or above rating.(figure 6.2.7.3)

Practicality: 95% of users give a general or above rating.(figure 6.2.7.4)



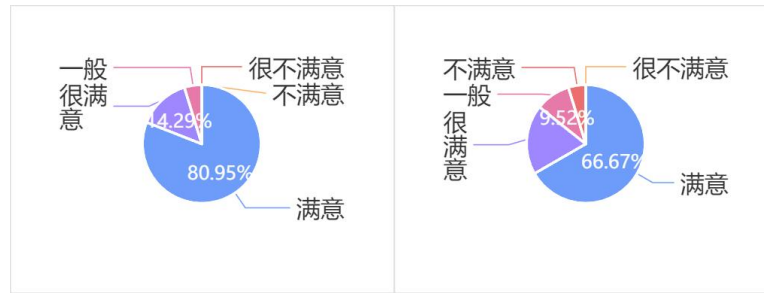


figure 6.2.7.4

## 6.2.8 Availability questionnaire

A usability survey was conducted on the app to determine its usability. The final SUS availability score is 68.9.(figure 6.2.8)

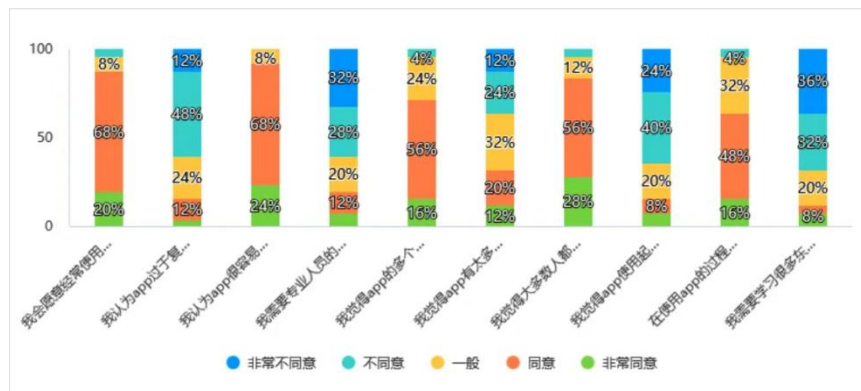


figure 6.2.8

## 6.2.9 Targeted adjustments

We have made improvements based on the suggestions put forward by users(figure 6.2.9)

Unable to share photos: A sharing button has also been added to the results, allowing you to share captured images with third-party apps.

Personal information cannot be edited: Added the ability to change personal information such as avatars.

The forum cannot find the downloaded file: Download details have been added to the personal interface, and photos downloaded in the forum can be edited in detail here.

There is no animation on the bottom navigation bar: Animation highlighting has been added to the navigation bar at the bottom to provide a more intuitive view of the selection interface.

题：请问对我们的app有什么改进意见吗？		
分析	观点分析	搜索答案文本
搜索		
号		答案文本
		建议论坛中单栏加一个滑动效果
		底部栏框没有动画看不出来我在用哪一个页面
		没有换头像功能，建议加一个！！
		建议增加换头像功能
		拍照最好能做到分享给其他人
2		拍照后没有办法直接跳到个人主页，感觉不好
3		论坛里的图片不能下载到本地相册里
7		找不到从论坛里下载的图片
8		拍照不能分享

figure 6.2.9

## 7. Problem and Solution

Our team encountered the following issues while collaborating on the development of the "Visualist":

Poor communication leads to difficulties in cooperation: During the collaborative development process, such as poor communication or lack of coordination among team members, it affects the development progress and quality.

Lack of clear planning and task allocation: At the beginning, there was no clear task allocation and planning, and team members experienced repetitive work or uneven workload, which affected development efficiency and quality.

Difficulty in technical implementation: encountered difficulties or challenges in technical implementation during the development of the app. In terms of image recognition function, we originally planned to use Python to complete it, but after development, we found it too difficult, so we switched to Baidu's API to complete it.

The methods to solve the above problems include:

Establish clear communication and collaboration mechanisms, such as weekly group meetings.

Determine the responsibilities and tasks of each member, refine the tasks, and develop progress and assessment standards.

Strengthen technical research and learning, continuously update technology and implementation methods.

Strengthen control and adjustment of cycle arrangements, reasonably arrange tasks for each stage, and ensure the overall progress and quality of the project.

## 8. Development planning

The development of "Visualist" will have three stages, which are to realize basic functions, realize personalized functions, and realize the combined development of online stores and offline stores.



1. The first stage is to realize basic functions, we will improve the expert picture comparison function, mall function and forum function. Fix App bugs through testing and user feedback. The duration of this phase is about three to five months

2. The second stage is the implementation of personalized functions, we will refine the users and recommend personalized services. The duration of this phase is about five to seven months.

3. The third stage is the combination of online and offline stores. After we complete our online functions, we will target offline and expand our revenue programs. That phase will happen in about a year.

## **9. Summary**

After a comprehensive analysis and consideration of our "Visualist" design, we believe that the "Visualist" design solution can achieve high quality and user experience levels, thereby meeting the various needs and expectations of users. In our "Visualist" design solution, we attach importance to several key aspects such as requirements analysis, process design, user interface design, key technologies, testing, and user experience, which enable our design to fully meet the needs of users. Although there are still some potential improvements and room for improvement, we will continue to pursue excellence to further improve user experience and satisfaction.

This has taken a solid step towards the success of our software design solution, and we believe it will be an overwhelming factor for future success. We look forward to observing the positive impact of our design and are willing to do whatever it takes to ensure that the services we provide consistently meet the expectations and needs of our users. Therefore, we firmly believe that in the future, our "Visualist" design solution will be further improved and developed in various aspects.