**EmotiCan**

1. **Abstract**

Nowadays more and more emoticons piled up our life, only few of them, however, are created by their users, since making an emoticon is still not simple enough. Our project aims to solve this puzzle by separating an emoticon into different kinds of elements and provide them to our users to combine new emoticons with their mind. The project provides three functions. The first one is recognizing and separating the facial expression automatically, which can be helpful when users want to use the photo they take to create new emoticons. The second function is an online library that provides users oceans of elements of the emoticon. The third key feature is a recommending system, which can recognize the meaning of the facial expression provided by users, and automatically recommend the appropriate elements for user to choose. This mechanism can greatly promote the convenience as well as the quality of making new emoticons.

1. **Team**

Our team consists of five members. They are:

李汉青、孙克斌、杨天宇、郑钿彬、周辉

1. **Project Descriptions**
   1. **Motivation**
      1. Background

The self-made emoticons have below up the internet in recent years. According to a report provided by Baidu, the use rate of the self-made emoticons is three times more than the built-in emoticons in people less than 20-year-olds. But these well spread emoticons are made by only a few of people. Compared with the people who use the emoticon, the creators number is considerably small. The most significant reason of this phenomenon is that the existing image processing software is still too complicated and too heavy for a beginner to use them making a simple emoticon easily. Our project aims to provide a lightweight approach for the creation of emoticons.

* + 1. Project’s overview

Nearly all of the self-made emoticons can be divided into several elements such as the background, text and faces or cartoons, as shown in Figure 3.1.2. Even a series of emoticons can be made using a same image with different text. Thus, this work usually requests swift, light weighted tools.



Figure 3.1.2 Components of ordinary emoticons

The existing powerful software like Photoshop, is too complicated and too heavy for such kind of work. So if most of the emoticons are made up of several elements, why don’t the software just provides users choices of elements, and let the users to combine them up? This turns out to be a much easier task for the beginners who want to try emoticon creation. At the same time, it still provides enough possibilities for users to create. This approach can solve the situation effectively.

* 1. **Feature Description**
     1. Overview

To achieve the project’s aim, this project will mainly consist three parts: detection, online-library and recommendation. These three parts are not completely isolated, some of them also depends on others. We will talk about them in detail in following paragraphs.

* + 1. Detection

As what the title means, the detection part aims to detect something from an image. Its target can be divided into two kinds, human faces and text. These two kinds of detections are the basis of the following works.

The human face detection can detect and cut out human faces from an existing image automatically. This feature can help user cutting out the human faces from the images the user provided. Then, the faces can be provided to users to create any new emoticons they want to build. This detection feature can greatly assist the dull cutting work, which can promote the experience.

Text detection aims to recognize the text on an existing image if there is one. There is usually the case when users want to edit or delete the text from an emoticon they received or found. In this case the text recognition is necessary for supporting a quick editing. After recognizing the text on an image, the software can separate it from the background of the image, then users can edit these text, delete it or put new text on anywhere of the image.

* + 1. Online Library

As we have discussed above, the project aims provide users plenty of elements for users to combine new emoticons, no matter the element comes from the users themselves or the app provided. The detection function can provide users a convenient tool to add their own elements, and the online library is another choice of adding the existing elements. Our online library will store different kinds of elements as many as possible, including not only the background, the foreground such as popular facial expressions or interesting cartoons as well. This feature can make users get the popular choices swiftly and conveniently. If a user want to combine his newly captured picture with a funny cartoon figure, he will never need to find a picture of that cartoon independently and then cut it out to achieve his new idea. All these works can be done in this only app just in a while. Meanwhile, users can also upload their own products to our library if they want to share their great ideas. Users’ idea will be selected and would be available to all the users. These are the meaning of the online library.

* + 1. Recommendation

From the creation process described above, we can find that if users want to create use their own materials, they need to search the library for an appropriate element or the element they keep in their mind. Although the elements can be labeled according many of their features, it is still a difficult and time consuming process, since the amount of the emoticons are extraordinarily large, so does the resulting elements. There can be hundreds of elements share the similar meanings. For instance, if a user want to use a “laugh” facial expression, he can use a filter to search for “laugh”, and get hundreds of choices back. You can imagine hundreds of choices waiting for you to choose from. You will get pretty tired, or, more likely, unable to memorize and find out what you are looking for. This process is pretty interrupting, which has great chance making users lose their idea. This annoying process can be dealt with by introducing the recommendation system. Contributed by dee learning technique, the deep neural networks nowadays are able to recognize the meaning of a human expression, such like happy, sorrow, or astonishing, then the system can use these key words to search in the database, and return the searching result automatically. Moreover, DL can do much further by trained with thousands of existed emoticons, find the miraculous relationship between the expressions and its backgrounds. This relationship is even hard to describe in human language, which is a sign of human creativity. But, at least, the computer can imitate this creative process now. In some aspects, this recommending system is creating the emoticon itself, and just return the users some of its great artworks. By this way, users are able to not only find the most adopted choice others are using, but even some more extraordinary element they even didn’t occurred as well. In all, the recommendation will be the most brightening part of this project, and turns to be the most difficult part as well.

* 1. **Requirements**

Some functions of the project are built on the basis of some other basic functions. At the same time, limited by the time and coding ability, the completion of the project needs to be divided into three levels. The most primitive level will achieve the functions listed below:

* Basic image processing functions
* Build the online library
* Rough facial recognition and separation
* Text detection and separation, the ability of re-editing the text.
* Can operate on general devices
* No performance restricts.

The second level of completion will include the listing features:

* Much more accurate face recognition, including the eyes, nose and mouth recognition etc.
* Build the primitive recommendation system, recommend according to the expression recognized.
* Optimize the UI design
* Optimize the performance
* Optimize the filtering strategy of the server
* Optimize the storage request of the client application

The final level of the project will include these features:

* Implement the recommendation system based on big data analysis
* Add user management system, provide online storage of users’ data
* Connect with the general used IM software like QQ and WeChat to provide a much more convenient experience in editing and sharing
* Transplant the app onto iOS platform
* Further optimization on UI and performance
  1. **Architecture**
     1. Use Case Diagram

As Figure 3.4.1.1 has shown, the use case diagram of the whole project looks like this.

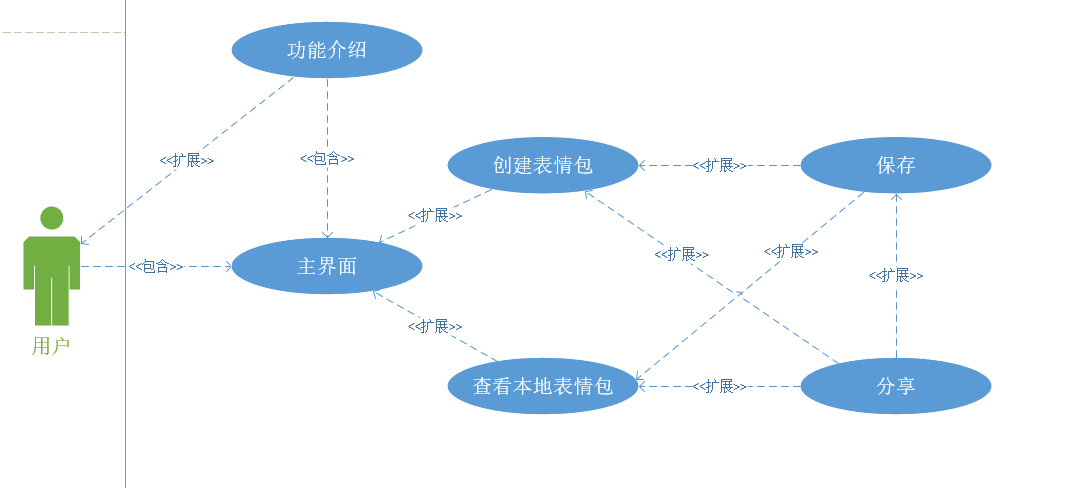


Figure 3.4.1.1 Use case diagram of the system

The app will begin with an entering the main interface, where users can choose the function they need, such as create new emoticons or just look through the local emoticons saved before. After looking through the local emoticons or creation, users can choose to save or share the image with others.

The details of “create emoticons” use case will be shown below, like the Figure 3.4.1.2 shows:

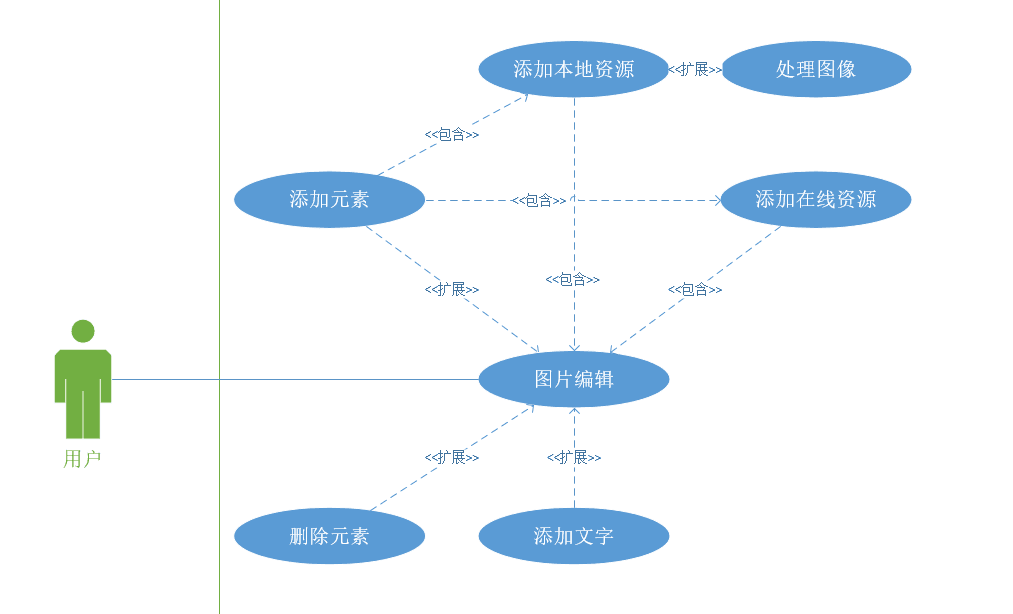


Figure 3.4.1.2 Use case diagram of create emoticons

In create emoticons use case, users will enter an edit interface, where users can choose to add elements. When users choose to add elements, they can choose add from local files or online library. Both local images and online library will give the app an image the user choose. After users choose an image, the app can enter another edit interface, where the detection and some simple editing are made. After that, the new added element will be returned to the main edit interface, where users can combine elements together, add text and do some processing. Also the edit interface has the ability to delete an element. After all these works are done, the create emoticon use case is end.

* + 1. Database Schema

The project will keep an online database to store the different kinds of elements. The element consists of the image and the label of that. The label can be of different amount, as a key word to describe the image content. The database schema is shown like Figure 3.4.2 has shown.

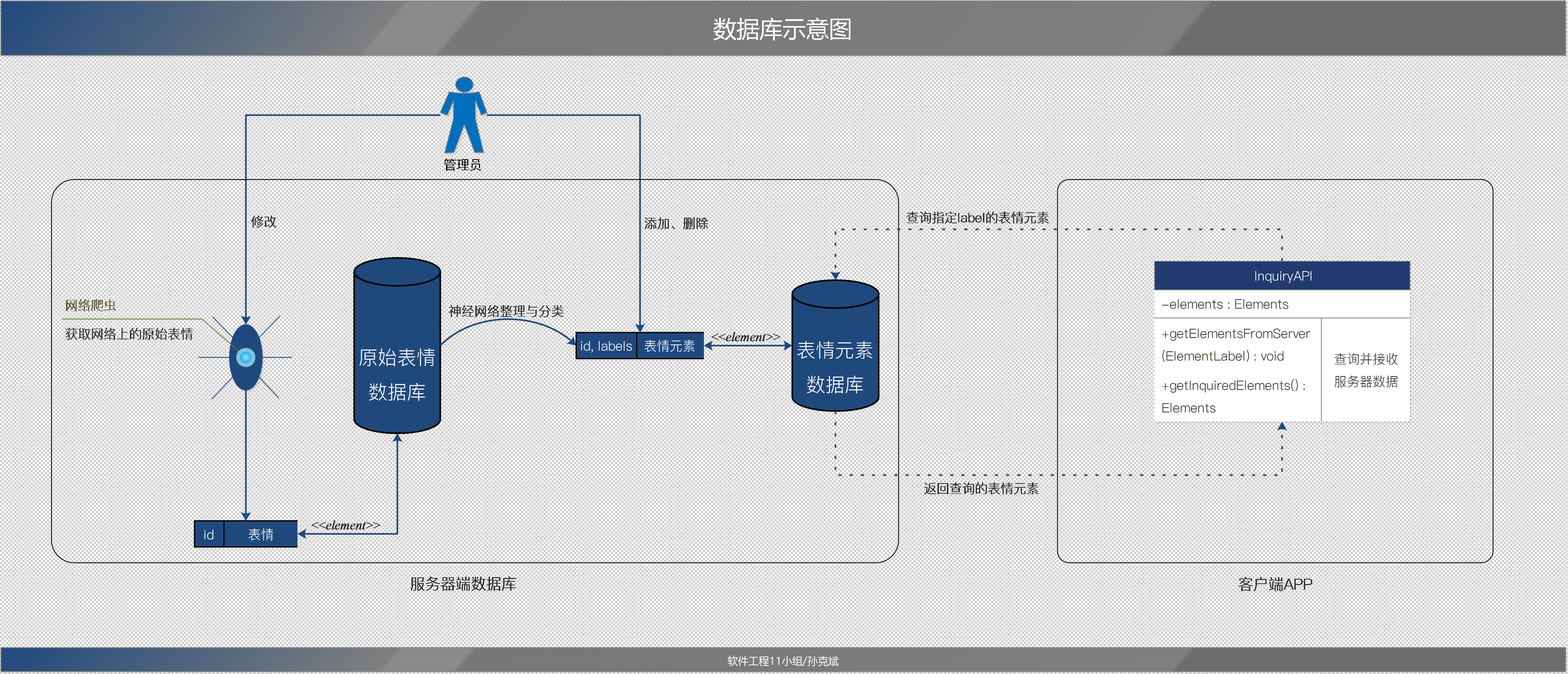


Figure 3.4.2 Database Schema

* 1. **Timeline**

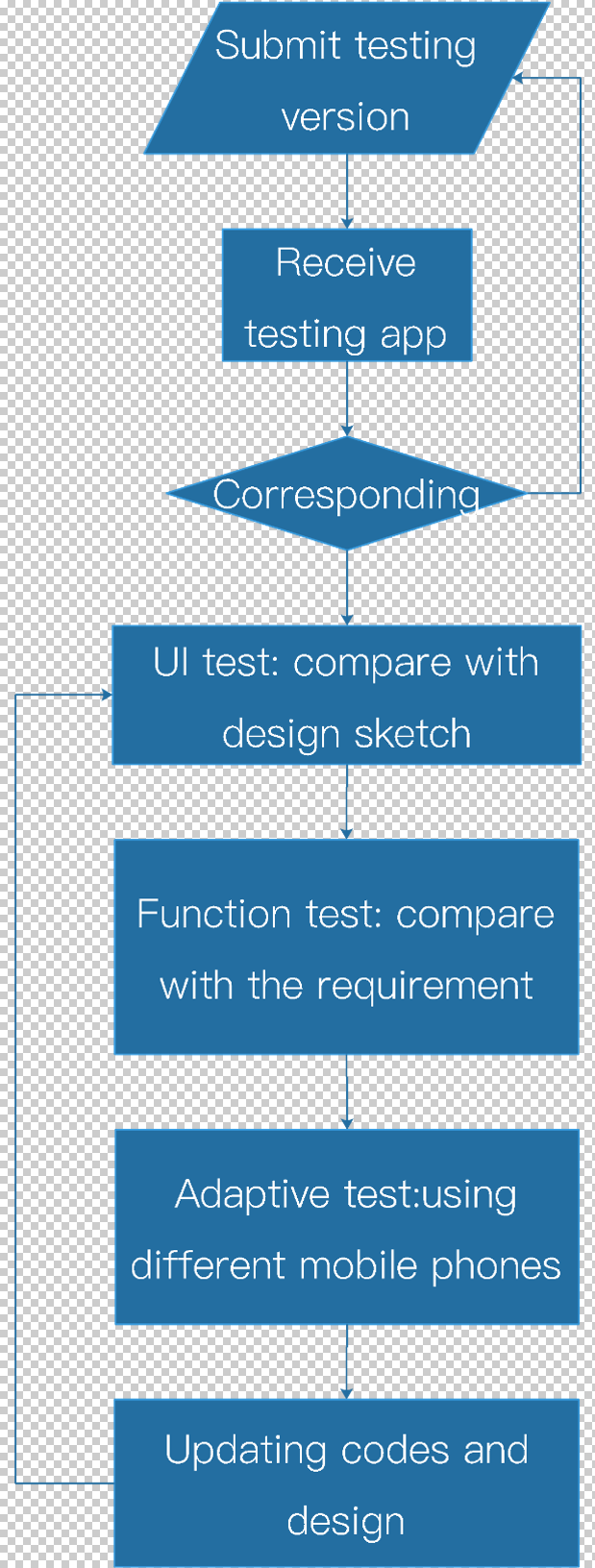
At the end of the week 5, the project is still under designing. Some technique details still need to be specified to finish the whole design of the application. But this work will be done by the end of week 6. So, begin from the week 7, the project will enter the programming stage. We estimate that it will need about 4 weeks to finish the whole programming works. The testing plan will begin from week 10, which will examine the work entirely. The timeline is shown in Figure 3.5.



Figure 3.5 Timeline

* 1. **Testing Plan**

Our testing plan will begin at the week 10, which aims to test the project thoroughly. Thus, we will proceed testing from a lot of aspects. The details of our testing plan will be shown in Figure 3.6.



* 1. **Feasibilities**

As the 3.3 has discussed above, the basic functions of the project are not a tough task. These are also the basis of the further advanced features. Thus, this part of work will be done as soon as possible. The second level of the developing needs some specific techniques, such as deep learning, which we still need to improve our abilities. But the stage 2 only requires the basic implementation of these techniques, so by the end of the semester, this part of work will be done anyway. The final stage is indeed a challenge. Despite of the optimization works, the new recommend system requires tremendous resources to train the neural network. If these large, labeled data is not accessible, this new kind of recommend system can never be achieved in this way. So, the stage 3 is not strictly required by the end of this semester, but we will try our best.