



SMART CLOUD

Group 7

2017-5-30

DEGREE PROGRAM: SOFTWARE ENGINEERING

GROUP MEMBERS

14207149 HUAXI WANG

14207150 AI ZHANG

14207153 LIN LV

14207199 CONGCONG WANG

CONTENTS

Chapter 1: Project Overview	2
What our project is.....	2
What are the goals	2
Chapter 2: Project Requirements.....	5
Network.....	5
Storage	7
Interface	9
Additional requirements	10
Chapter 3: System Design and Implementation	12
Framework of Android interface	12
Customized components.....	12
Generate tree structure	12
Verification control for user register	13
The uploading and downloading part	14
HttpClient and HttpPost in the Android application	15
Multithreading in the Android application.....	15
Update the progress bar in the Android application.....	15
Communication between the activities	16
Chapter 4: Testing.....	17
White box testing	17
Black box testing	17
User interface testing.....	17
Chapter 5: Individual Contribution.....	18
Wang Congcong	18
Lv Lin.....	19
Wang Huaxi	20
Zhang Ai.....	21
Chapter 6: Conclusion.....	23
Reference.....	24
Appendices	25

CHAPTER 1 - Project Overview

1. What our project is

Nowadays, PC and mobile devices are necessary tools to help people work and study. People often need to share files with others and transfer files from one device to another as well.

Our group is going to develop a cloud storage and files sharing application with both web and Android phone supported, which performs the mainly functions like Baidu Cloud and dropbox.

First, every user will register his individual account and our application support users to upload their files onto the application and download them at any time later. And we have rent a server to build database and store file on it, thus whether users log in using web page or Android devices, the files are the same and the operation and information are synchronized.

Second, in terms of files sharing, users can add friends in our application and tell their friends what files they want to get from them, and friend-relationship users can go to the chat interface and transfer files using our application directly, while for users who are not in the relationship, they can get a sharelink and set password to share their files using social media like WeChat or clipboard.

2. What are the goals:

We decided to make the project support 2 ends: web and Android devices.

2.1 For the website part, it is a browser/server model. The basic communication between the browser and server is achieved by sending post request from webpage to server then get response from it, such as login, register, upload, download, share files

and find password. While for the communication between friends, we use the websocket technique to implement it. For the user interface of website, we also use bootstrap framework and some plugins such as datatable, layui actcompleter and so on to make the webpage seems more beautiful. It can give user a better feeling when they use our website.

2.2 For the Android part, it is a client/server model. The basic communication between the Android device and server is achieved by using HttpClient to build post request and send it to server then get response from it, including login, register, uploading, downloading, sharing files and finding password. Then, for the communication between friends, we use the baidu push technique to implement it. This technique can send message or notification to the specified device. For example, when user A send a message to user B, this information will be sent to our server at first. This information contains the device ID of user B that this message needs to go. Then our server will send this information to the baidu push server, after some processing, this message will arrive to the device of user B. The device ID is unique for each devices, it will be produced and recorded when user use our android application to login. For the android user interface, we use viwepager and localmanagement to create it. It can implement some sub-activity into an integration activity, achieving the page shift from one activity to another activity. It can increase the usability of our application. The figure below shows the principle of our project(Figure 1).

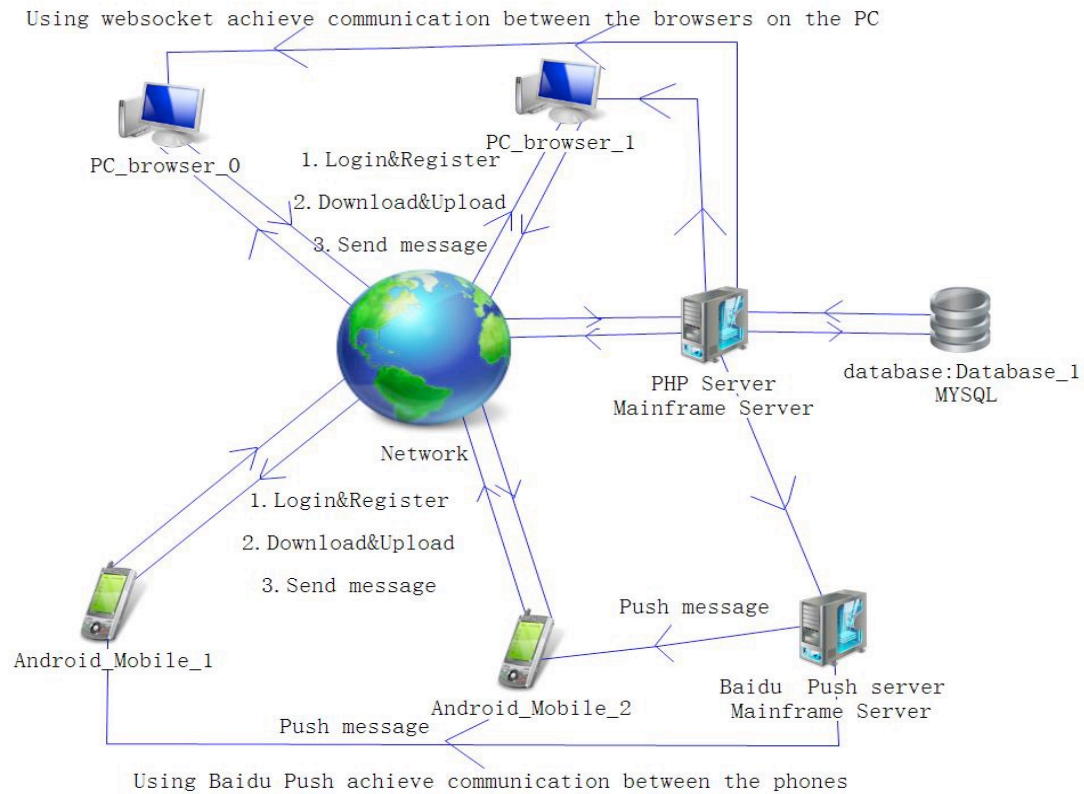


Figure 1. The whole project structure

3. How to achieve these goals:

Since we have 2 supported ends, we realized that the integration and merging are crucial to our project at the beginning. So we start the project with planning structure of the whole project, thinking of how each part will merge at the end and deciding which framework, language and connection ways are we going to use overall.

After plan that, we rent a cloud server to provide stable service, and we also decide to learn PHP framework – Laravel to increase scalability and security of our project.

Chapter 2 – Project Requirements

1. NETWORK

Regarding network requirement, there are several modes used to our project and some concerns to take care.

1) B/S Model

First, A Browser/Server mode was used to build the Web version as a part of our project. Major technologies used in this mode include PHP, HTTP and JS and so on.

To be specific, in order to save work of writing PHP directly, a useful and gentle framework named Laraval was used. In order to achieve real-time communications between users, we also need to rely on a WebSocket protocol.

Laravel is a very popular PHP framework these days. It is based on MVC model and an elegant framework with expressive, beautiful, structured and high efficient syntax^[1]. Hence, applying it into our project in a proper way can save us a lot of time and work, and more importantly, it could bring us much more convenience when refactoring and reconstructing our project due to its property of elegant structure.

The figure 2 below illustrates how the MVC model works as a whole for this framework.

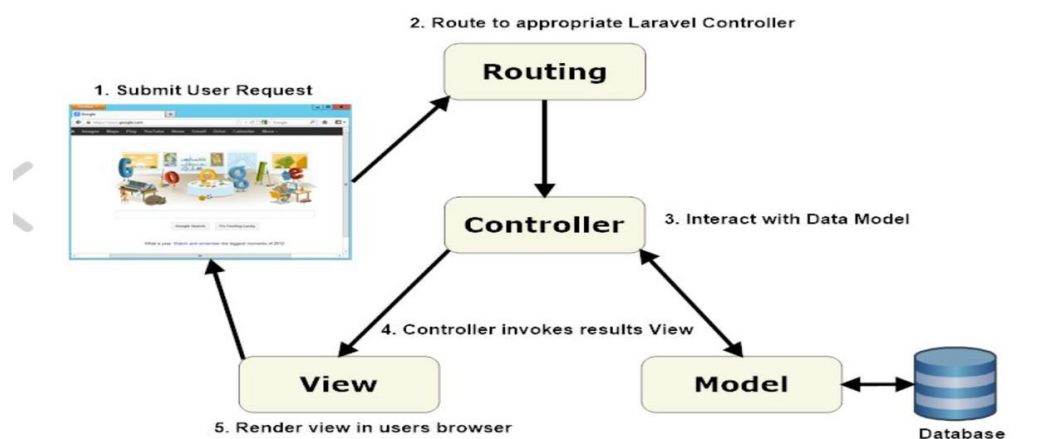


Figure 2.MVC model

In browser, for a request sent by users, GET or POST, first this request will come to the routing part which helps to direct the request in a certain controller. For example, if a GET request named /index is given a method called index in a controller named GlobalController to process and meanwhile we give the route a name 'index', it could be defined easily in way as follows.

```
Route::get('/index','GlobalController@index')->name('index');
```

Next, the pivotal part would come down to controller, which is the central and core module in the framework and play an extremely important role in processing service logic. To finish a certain task, controller can cooperate with model which is used to put data into the database and view which is front-end web pages used to present the data from the server. For example, for the register service, here what the controller needs to do is passing the submitted data into model to store into the database persistently and then return a view to tell users whether the service is finished successfully. As seen from the MVC model, each part takes their own responsibility which leads to be high-cohesion and loosely-coupling which is beneficial to extendibility of our system.

Another important problem in B/S mode is instant-messaging. To deal with this problem, we applied a crucial technology into our project. The technology is called WebSocket which is a computer communication protocol, providing full-duplex communication channels over a single TCP connection ^[2]. Fortunately, because it has already been provided by many browsers and server programming language like PHP, it is a proper way for us to achieve real-time data transfer between different terminals.

2) C/S Model

Second, a Client/Server mode was applied to build the Mobile version, namely

android application version. Major technologies used in this version including HTTP, JAVA, BaiDu Push etc. BaiDuPush which is a platform of BaiDu company providing messages push service for developers. It has a very good set of APIs supporting android Application Development and available for PHP language on the server side. For the android side, we rely on such a tool to achieve real-time communication between friends.

On top of that, we also resolved some other problems about network requirement. Such as, controlling the storage capacity for each user by calculating the size of the file every time users uploaded, setting a private folder for each user to manage each their uploaded files on the server side, applying some digest encryption algorithms to promise security and file integrity when files are transmitting across the internet and so on.

One last point to emphasize is that the server side was employed onto a cloud server so that this system is not just restricted to work in a LAN, because we rented a cloud server to achieve this.

2. STORAGE

For data persistence, the database we used in our project is MYSQL. We divided our system into three parts, user system, file system and communication system.

1) User System

For user system(Figure 3), the major information that we need to store is users' basic information, so the main table created in this system is the table named USERS as follows.

id	name	email	password	remember_token	created_at	updated_at
----	------	-------	----------	----------------	------------	------------

Figure 3. User information table

The information it stores mainly include user name, unique email and password, timestamp of creation and updating.

2) File System

The second part is file system(Figure 3), where table named FILES was created to store basic information of files. It is like the following.

file_id	user_account	filename	filepath	desc	filetype	filesize	parent_id	status
---------	--------------	----------	----------	------	----------	----------	-----------	--------

Figure 4. file information table

Attributes containing in this table are primary key file id, foreign key user account referencing to email attribute of table users, and file path, size in a unit of bytes, type, parent id which is used to generate a tree structure of files finally on the front-end, and status to label different status of files like where it is on file list or trash list or shared by users and so on. Of course, in order to achieve other functionalities of file system, like generating share link and recording storage capacity, other tables like SHARELIST and STORAGE were created as well.

3) Communication System

The final part is communication system. In order to persistent data very well, in this part, a table in database named CONTACT was created(Figure 5). Major pieces of information are shown as follows. This table is primarily used to present the relationship between friends. To emphasize one attribute in the table, which is channel, it is used to locate a mobile device -like a unique ID for a single device produced by BaiDu PUSH - so that the message could be sent from one device to another one correctly as real-time communication is happening between users on mobile devices

contact_id	user_account	face	signature	channel	friend_account	created_at	updated_at
------------	--------------	------	-----------	---------	----------------	------------	------------

Figure 5. Contact information table

Because of cross-platform supports of our system, all tables created above are shared between web side and android side so that synchronization can be guaranteed.

3. INTERFACE

Concerning interfaces, there are major two kind of interfaces required in our project.

One is web pages and the other one is android application interfaces.

1) Web Interface

For web pages, in order to improve the interaction with users, some fancy front-end plugins and framework were used. JavaScript framework was used to replace the work of writing original JS codes so that more friendly-interactive pages could be generated highly efficiently. another framework is HMLT/CSS framework which is Bootstrap which is used for designing websites and web applications containing HTML- and CSS-based design templates for typography, forms, buttons and other interface components ^[3]. By using this framework, our pages could be much more fancy, attractive and beautiful. As well as these, we also used some plugins, such as web layers plugin LAYER, table plugin DataTable, progress label of file uploading AjaxFrom and so on. All of them were embedded into building web pages very well. As far as interfaces of android side are concerned, we mainly rely on android software development kit to design and build all interface. Of course, some tool software like Photoshop, Illustrator supported.

2) Android Interface

For the Android application, the main interface is built using ViewPager and LocalActivityManager, which allows users change page only by sliding left or right. And we also override many classes like Dialog, list view and so on to customize them

and make the page look more organized and nicer.

4. ADDITIONAL REQUIREMENT

1) Security

For security purpose, some security requirements need to be considered. Here is a mechanism named csrf provided by Laravel as you can see in Figure 6. It deals with security threat from outside system and protect our system from being attacked when users submit form programmatically instead of artificially. Fortunately, the Laravel framework helps us resolve this problem so as to save us a lot of work.

```
<form id="myupload" enctype="multipart/form-data" method="post"
  action="{{route('files.store')}}">
  {{ csrf_field() }}
  <div class="form-group">
    <label><span class="glyphicon glyphicon-file"></span> <font
      color="red">*</font>Pick a file</label> <input type="file"
      class="form-control" name="file">
  </div>
  <button type="submit" id="uploadfile"
    class=" btn btn-success btn-block">
    <span class="glyphicon glyphicon-upload"></span> upload
  </button>
</form>
```

Figure 6. Protection for injection problem when submitting form

The second security problem is information encryption. To protect users' sensitive information like password, we applied an irreversible algorithm named md5 to the password storing in MYSQL database (Figure 7). One point to highlight is that the only way for users to find their passwords is resetting their passwords due to the irreversibility.

id	name	email	password	remember_token	created_at	updated_at
9	abc	admin3	202cb962ac59075b964b07152d234b70	NULL	NULL	NULL

Figure 7 .MD5 encryption for user sensitive information

The next solution to security problems is email verification system. If a user forgets his password, our system will send an email to him help him reset his password. Only in this way can the user's information be kept secure. Furthermore, to improve the

reliability of the email system, the URL for the user to reset password brings a parameter whose name is token. The following line shows how the token is generated (Figure 8). Obviously, it was the value computed by md5 jointing the email, a private key “mimi” and the timestamp together so that only the person knowing the private key is possible to produce a fake URL for resetting password. Meanwhile, to further enhance the security, every email was given a valid time 5 minutes which means users have to get access to their email to reset their password within 5 minutes.

```
$token = md5($email . "mimi" . $now);
```

Figure 8. Generating token

Of course, there are still many other security problems need to be considered. Such as, asking users to set their password with both digits and letters between 8 and 16 for the sake of brute force crack, and file integrity protection, and digest verification for cross-platform communication and so on.

2) Reliability and Stability and Scalability

Regarding reliability, the main aspect is that our system is cross-platform to some extent, available for both android mobile device and web browser. For the final release, the Smart Cloud enables users between different PC or mobile devices to manage their files (uploading, downloading and sharing, etc.). Also, the system will have to guarantee synchronous updating or refreshing on both PC end and Mobile end.

Apart from this, a cloud server was rented to guarantee reliability and stability. Additionally, an administration system will be generated later on in our project which plays an important role in filtering some illegal files and detecting malignant users so that all users could use our system in a relatively friendly environment.

In terms of scalability, on the server side, we improve the scalability of the application thanks to the framework Laravel again. As we discussed in the report, it is an elegant

PHP framework based-on MVC, which means our codes written on the sever side are organized in a very logic way, resulting in a high scalability for our system. For example, if we want to apply a language internationalization function to our web version, we could achieve it by simply changing a little few codes.

Chapter 3 – System Design and Implementation

1. Framework of Android interface.

Many kinds of implement structures of this part were once in our choice list, including fragment+tabhost, fragment+radio groups and so on. Many of them are very useful but contain the deprecated functions. In other word, each of them has pros and cons. We tried many methods and finally decided to use localactivitymanager + viewpager structure to perform the divagation, because it allows users to slide to each page and allow us to easily implements each page independently

2. Customized components.

We override lots of components in order to enable the default components to performs our own function. For instance, the default dialog can not include multiple components and cannot pass information of what user choose and enter into the activity. These problems are overcome after I override the default dialog. I override the Dialog class and create an interface to enable the information translation between the dialog and the activity.

3. Generate tree structure.

In the file system, we need to add a function to mange files layer by layer. This function requires us to turn pieces of information in database into a tree structure. Because rows of files in database is discrete and there is an attribute named parent_id to label the parent-child relationship in the file tree structure, we applied

a recursion algorithm named `getNode`s to convert into the suitable format as follows(Figure 9).

```
var nodes = getNode(0, allfiles, allfolders);  
var nodes=[  
  {  
    id:0,  
    name:'root',  
    alias:'folder',  
    spread:true,  
    children:nodes  
  }  
];
```




Figure 9. Applying a recursion algorithm named `getNode`s

4. Verification control for user register

When registering, we need to guarantee that users cannot sign up an existed account and should set their passwords with both digits and letters. Except the verification on the server side, we also need to verify validity of data on the front-end. To deal with the problem of email uniqueness in our system, Ajax technology was used when registering. The process is just like this. First sending to the content of email input box to the server as soon the input box was blurred, and then the server side use the content to check database to tell if this account has already existed and finally return the checking result to the front-end(Figure 10). To control the validity of password, on the front side, we used regular expression to verify.

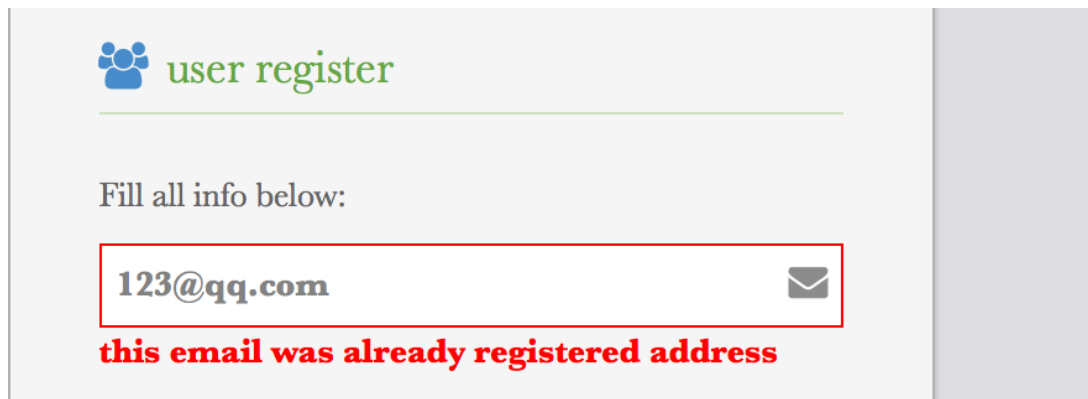


Figure.10 Verification user register

5. The uploading and downloading part.

This picture below shows that each person has a specific file and the user's upload files are contained in each user-file(Figure 11).

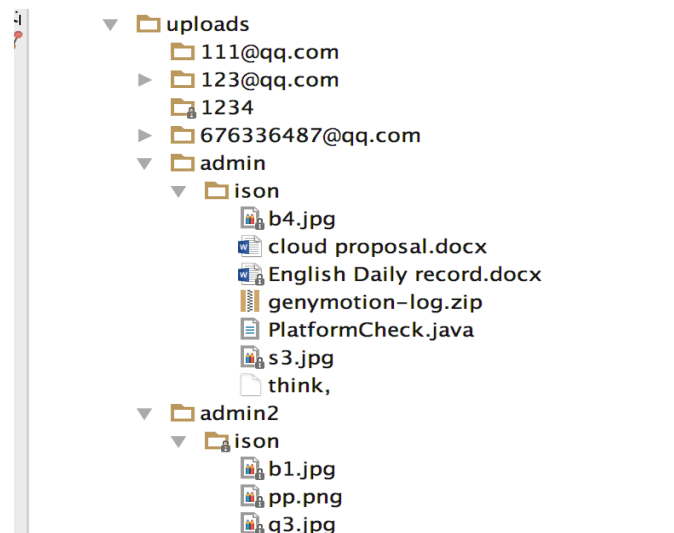


Figure 11.Users files

At the uploading stage, the major problem is the size of the file which the user wants to upload. We change the configuration file to limit a specific size (for example,200 mb). After finishing this part, We build document files for each user and save the user's upload files to this document file.

Besides, we store the links of the user's uploaded files to the database. When the user wants to download the file, we just give him the link. The major problem of

the downloading part is the picture. In this part the stream-path technique is really helpful.

6. HttpClient and HttpPost in the android application

The most information that be sent to the server is implemented by using `httpClient` and `HttpPost` to simulate a post request. The reason for we use it is that our server is consisted of PHP files and it needs to provide services to website and android application at the same time. The android application need to give the same form of data as the website to the server. We never done this before, so it takes us some time to learn and implement it.

7. Multithreading in the android application

In the android application, there are plenty of interaction between android device and server to exchange data and update UI. But after Android 4.0, the main thread can't access the network. The only way to access the network is starting a new thread. After getting the information from server, it is hard to update UI in this new thread. In order to solve this problem, we create a handler in main thread then start a new thread to access the server. After the new thread get the information from server, sending message to the handler to update the UI.

8. Update the progress bar in the android application

Initially, we combine `HttpClient` and `HttpPost` to simulate post request to upload files before we do progress bar. When we begin to do the progress bar, we find that it is hard to check the uploading progress when simulate post request. But using the post request seems is the only way to implement our uploading function. We do search on the Internet and find a solution. We can use the `AsyncTask` to achieve it. It works like the thread but provide method to check the progress. Finally, we use `AsyncTask` to achieve the uploading function and update progress

bar at the same time.

9. Communication between the activities

Due to the requirement of our application, we need to pass information from one activity to another activity but not shift page, such as after choose files that we want to download in the file activity, the information should be showed in transfer activity. In order to achieve this, we decide to use broadcast to implement the communication between activities. We create the broadcast receiver in activities that need to receive information. Then creating intent, put data into it and broadcast it in activities that need to send information.

CHAPTER 4 – TESTING

In our project, three different types of software testing have been used, including white box testing, black box testing, and user interface testing.

1. White box testing:

In this part, our group members mainly focus on the logic of the project. To be more specific, after finishing a function, we need to check the internal logic of this function. We need to find and delete those statements which are useless or redundant. We should make sure our code is clear and efficient.

2. Black box testing:

In this part, we just test the function of our program and do not care about the internal logic. If the program reacts to the operations incorrectly, we have to read the code again carefully and find all the bugs. For our group, each member do the black box testing independently at his or her laptop and after we integrate all the code written by different people, we do the final project's black box testing.

3. User interface testing:

In this part, our group focus on the user interfaces part. It is common knowledge that UI is quite important to the website and android app. What we have to do is to create beautiful, cool, and user-friendly interfaces. At the beginning, we make basic interfaces to let us put more emphasis on the function. It's time to beautify our interfaces when the basic functions of the project have already finished. Thanks to the web search engine and friends' suggestions, fantastic and cool interfaces of both the website and android have been created.

Chapter 5 – Individual Contributions

Member1: Wang Congcong:

Specific tasks I contribute to:

1. Know and learn JQuery and BOOTSTRAP
2. Design and build all static webpages
 - 1) Get login and register two Interfaces done
 - 2) Main Interfaces realization, content including, buttons for uploading/downloading, icons for sharing, creating folder and deleting file and so on, a table for file list, menu of user information management, extending folder dynamically, progress bar for files when uploading, etc.
 - 3) 2.3Achieve windows for file transferring and real-time communication between friends.
 - 4) Achieve Administration Interface, which helps to manage files or information of all users
3. Try and learn many plugins used in our project
4. Send Ajax request to the server and interpret JSON data responded by the server
5. Optimization and beautification for all interfaces to make them more friendly-user interactive

Integration:

1. With Lv Lin:

To unify interfaces on android side and web pages to some degree through discussion, such as, try to use similar font family and font size, color and same layout in some places on both sides and so on. As well as this, we also work together to think how to design our interfaces as properly as possible
2. With Zhang Ai:

Lots of works were done with Zhang Ai because he is responsible for server side.

To make pages dynamic, with no doubt, we need to interact with the server, including sending requests to the server, such as login, register, upload files, communicating with another user, and accepting response from the server, like Ajax response. All these things are what I need to cooperate with him to complete.

Member2: Lv Lin:

Specific tasks I contribute to:

1. Android interface and framework
 - 1) I am responsible for the Android interface and framework, which means anything that is presented to users on Android devices is implemented by me.
 - 2) I use viewpager + localmanagement to establish the framework of Android interface which allows us to implement and manage page individually, avoids the complex life cycle of fragment and becomes user-friendly because users can switch pages by slide left and right.
 - 3) Components like list view, dialog or title are customized which means that they are overridden, not only resulting in nice appearance, but also allowing us to implement our own functions.
2. Files sharing.
 - 1) For the two ways of sharing among the total three, after I register as a WeChat developer and applying for the valid code for our project, I learn how to use WeChat API to share files to WeChat. And I also learn how to copy contents onto the clipboard, so that users can share files using more flexible ways.
 - 2) I make post request to the server to ensure that the password for the file can match that on the web end.
3. Some additional functions
 - 1) The major functions like file uploading, downloading and chat are implemented by Wang Huaxi. So I implement some small but important

functions like exploring files on SD card, remembering and retrieving password and so on.

Integration:

1. With Wang Huaxi:

- 1) When I implement interface, I present mock data and leave the interface to provide the actual data and user information. When Wang Huaxi finished his part, our work can integrate by just using the data he processed to replace the mock data.
- 2) There are some situations that the point 1 does not work, we will change his or my implementation to ensure it works well.

2. With Wang Congcong;

I make post request to the server implemented by him and we unify the name and the type of every parameter and returned value, so that we can send and get right information for every interaction. And this also guarantees the functions on the webpage are comparable to those on the Android devices.

Member3: Wang Huaxi:

Specific tasks I contribute to:

1. Implementing login and register by combining the HttpClient and HttpPost to simulate post request and send it to the server.
2. Implementing uploading and downloading functions in the android application by using AsyncTask and update progress bar.
3. Achieve the communication between activities by using broadcast.
4. Creating user's personal folder after they done the download operation
5. Sending post request to server to get the file's information of user, then create a file list in the user interface and provide it to user to check.
6. Sending post request to server to delete the specified file provided by user

7. Refresh the activity to get new information by using SwipeRefreshLayout and OnRefreshListener.
8. Using Baidu push technique to achieve the communication between friends on the android application. I learn the SDK document of android and PHP. Applying this technique into our android application. By using it, we can achieve adding new friend, communicating between friends and sharing files between friends.

Integration:

1. With Lv Lin:

After I confirm those functions works well, I cooperate with Lv lin to integrate those functions into the interface. Such as login, register, uploading, downloading, communication between friends.

2. With Zhang Ai:

Due to our web version and Android version use the same database on the server. I also cooperate with Wang congcong and Zhang Ai to unify the parameter's form when sending and receiving data between server and Android terminal. Such as the parameter's name of username, password, file path and so on

Member4: Zhang Ai:

Specific tasks I contribute to:

1. the login and register part(website):

According to the basic database operations, I check the user's input information. If the user has already registered and fill in the correct way(email address, password), he or she can be allowed to enter in our smart cloud system. Otherwise, she or he failed.

2. the uploading and downloading part(website):

Creating each file for a specific user, and when the user uploads the file, his or her files can be stored in the specific file. I set a limit size to the user's uploaded file. I

store the links of the user's uploaded file and when he or she wants to download it, just gives the link back. The major problem of the downloading part is the picture and this issue can be handled by using the stream-path technique.

3. the communication with friends part:

With the help of our group members, and based on the websocket protocol, finally, the communication between friends achieved.

Integration:

1. With Wang Congcong:

Wang Congcong does so much work at the front-end, including html, css, js... As the server part, I need to handle the requests made by the front-end and return the results. We get together to discuss the internal logic of the website and how to displays to the user

.

2. With Wang Huaxi:

Because of the connection between website and android devices and using the same mysql database, we need to talk about the form of the parameters.

CHAPTER 6 – CONCLUSION:

We learn a lot from last year's project. Therefore, we have excellent plan this year and we benefit a lot from it. During the three months of development, we cooperate very well with each other all the time. Every member has roughly equal individual work package and tasks and everyone makes great efforts to the project. Thanks to the plan, our integration is completed well at last.

References:

- [1] Laravel. 2017: The Official Website. Retrieved from <https://www.laravel.com>.
- [2] Wikipedia. 2017. WikipediA: The Free Encyclopedia. Retrieved from <https://en.wikipedia.org/wiki/WebSocket>.
- [3] Wikipedia. 2017. WikipediA: The Free Encyclopedia. Retrieved from [https://en.wikipedia.org/wiki/Bootstrap_\(front-end_framework\)](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework))

Appendices:

Appendix 1: Previous Gantt Chart (Timeline)

