Abstract

This document presents achievements of our group project from start to final release. In the first part of the report, we briefly describe what our project is and what we have done. Then we show the teamwork and how we finish our jobs. At the end of system document, we introduce the technical implement we used in our project.

Contents

1 Introduction

In the Hibernia-Sino Insurance Company project, our team are hired to meet the needs of the Hibernia-Sino Travel Insurance Company. Hibernia-Sino is a travel insurance company who specializes in travel between Ireland and China. They are moving their workflow processes from dated desktop software systems to the cloud. This will allow customers to update their details, renew their policies, and register claim online. The solution must be cloud-based, and work on mobile as well as PCs.

As a travel insurance company, Hibernia-Sino mainly concerns about security, reliability, ease of use for the customer and for employees of Hibernia-Sino, the ability for users and employees to interface with the new solution in English or Chinese.

1.1 Requirement Analysis

Before developing the project, we analyzed functional and non-functional requirements.

There are a bunch of functional requirements. In the system, users are able to log in and log out account as a customer or employee. Customers can register and have access to check their own profiles, which includes fields, such as name, date of birth, gender, etc. They are also able to upload order information, including its picture and other description, send claim order with lost date, and check records of their own items and claims. Employees are able to check order information that customer upload, and approve or deny claims. If the information is not clear or has problems, employees and customers should be able to connect to each other. In addition, all interfaces should be in both Chinese and English.

Regarding non-functional requirements, Hibernia-Sino mainly concerns about security, reliability and the professionalism.

1.2 Function Description

Up to now , we have two clients, Android and web, all services are deployed on the server . And the website is http://47.94.214.88:8080/HST1C/, it can be accessed by any device. Customers and employees can operate through the Android and web page. For customers, both clients can register and log in, add items, send claim order, view the item list and order list, check order status and connect to employees. In addition, there is a QR code on the webpage that can download Android client. For employees, both clients can process the claim order sent by the customer and reply to messages intently.

1.3 Plan

First of all, we analyzed the functional requirements of customer, and divided the functional requirements into three parts according to the project time and progress, core functions, basic functions, and additional functions.

- Core functions include real-time information transmission between two clients, account login, submission of claim orders for customer, processing of orders for employee, communication between customers and employees, and internationalization of two clients.
- Basic functions include the connection between customers and employees, item and order list, account registration, personal information filling, taking photos or uploading local pictures.
- Additional functions will develop after finishing core and basic functions, and ensure stability and security of system. It includes intent message, insurance purchases, multiple employee operations on orders, etc.

Then, we develop project as planned.

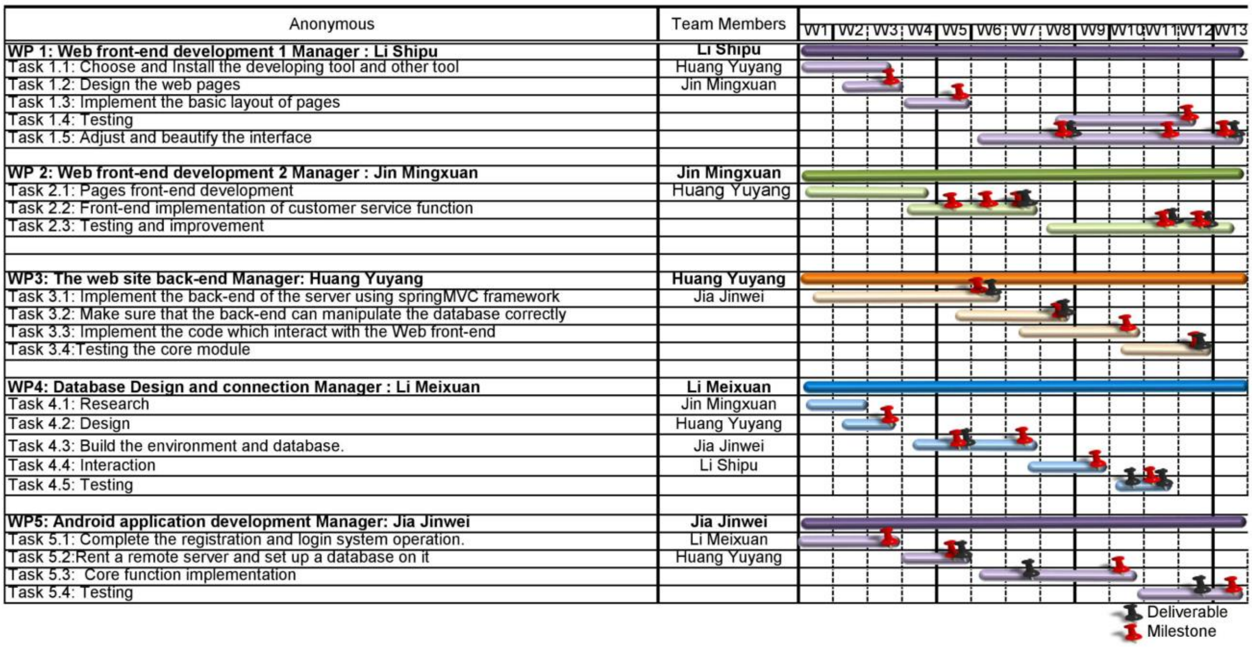
- Up to week 9, we complete core functions and part of basic function;
- Up to week 10, we complete all the basic function;
- Up to week 11, intent message in additional functions is completed.

2 Groupwork

2.1 Division

After consulting the opinions of the teaching assistants and the actual ability of the members in our group, we divided the work into five packages roughly, front-end, back-end, Android, database and other things. Li Shipu and Jin Mingxuan are responsible for the front-end design of customer and employee respectively. Huang Yuyang is in charge of deploying and debugging on the server. Li Meixuan is responsible for database construction and data maintenance. And Jia Jinwei is for the development of android applications.

2.2 Process of development



Anonymous	Team Members
WP 1: Web front-end development 1 Manager : Li Shipu	**Li Shipu**
Task 1.1: Choose and Install the developing tool and other tool	Huang Yuyang
Task 1.2: Design the web pages	Jin Mingxuan
Task 1.3: Implement the basic layout of pages	
Task 1.4: Testing	
Task 1.5: Adjust and beautify the interface	
WP 2: Web front-end development 2 Manager : Jin Mingxuan	**Jin Mingxuan**
Task 2.1: Pages front-end development	Huang Yuyang
Task 2.2: Front-end implementation of customer service function	
Task 2.3: Testing and improvement	
WP3: The web site back-end Manager: Huang Yuyang	**Huang Yuyang**
Task 3.1: Implement the back-end of the server using springMVC framework	Jia Jinwei
Task 3.2: Make sure that the back-end can manipulate the database correctly	
Task 3.3: Implement the code which interact with the Web front-end	
Task 3.4:Testing the core module	
WP4: Database Design and connection Manager : Li Meixuan	**Li Meixuan**
Task 4.1: Research	Jin Mingxuan
Task 4.2: Design	Huang Yuyang
Task 4.3: Build the environment and database.	Jia Jinwei
Task 4.4: Interaction	Li Shipu
Task 4.5: Testing	
WP5: Android application development Manager: Jia Jinwei	**Jia Jinwei**
Task 5.1: Complete the registration and login system operation.	Li Meixuan
Task 5.2:Rent a remote server and set up a database on it	Huang Yuyang
Task 5.3: Core function implementation	
Task 5.4: Testing	

Figure 1.   Gantt Chart

More detailed information is on the 5.2 Work Package in the appendix.

2.3 Individual contribution

2.3.1 Huang Yuyang

I am responsible for the website back-end and the combination work between web front-end and back-end. I also made some contribution to other module. E.g. I fixed some bugs in the web front-end code, gave some advice to the structure of the database and done some internationalization work. Web front-end developers(Li Shipu and Jin Mingxuan) have made several web pages correspond to the back-end server logic. The android application developer(Jia Jinwei) and I also discussed frequently to make sure that the way we manipulate the database is consistent. E.g. the encoding and decoding format of the user icon and the item picture.

2.3.2 Jia Jinwei

I was responsible for android app development and android server responsiveness. The first part is to develop a login system for users and employees, and users can add insured items and initiate claims. The employee can check the customer's claim order and claim item information; Users and employees can also communicate via instant messaging or comment feedback. The android server's response is to receive the request and give the response. In the development process, I received the help of our team members Li Meixuan and Huang Yuyang. Li provides stable database support for my part, and gives the method of adding, deleting, changing and checking about the operation of database related tables. Huang provided me with a method of Base64 encoding and decoding images, and assisted me in using the BCrypt encryption algorithm to encrypt data.

2.3.3 Li Meixuan

I am in charge of the database construction and data maintenance, meanwhile, I am the project manager of our team. I designed the database in week 4 and updated it in next several weeks according to the changes of requirement. In week 5, database is deployed in server with the help of Jia Jinwei. After beta version release, I helped Huang Yuyang and Jia Jinwei develop internationalization of web pages and Android. As a project manager, I am responsible for checking the progress of the team members and planning the work of next week, preparing the presentation (including making PowerPoint, recording demo, arranging the presentation content), writing documentation (user document and system document).

2.3.4 Li Shipu

I am in charge of designing and making the logo of the company, designing the web front-end pages, implementing the pages of customer part, and the adjustment to unify styles of all pages.

I addition, I have assisted Jin Mingxuan to design the staff interfaces, optimized the pattern and layout of the claim list, chat box, buttons and other parts in staff pages; adjusted selection of the components and arrangement of the interfaces according to the requirements in the process of back-end function implementation worked by Huang Yuyang; formatted the code for the subsequent bilingual switching function while writing HTML file (e.g. always need a <span></span> to wrap the content directly); assisted Jia Jinwei in the layout arrangement, color assortment and icon selection of the android interface.

2.3.5 Jin Mingxuan

I was responsible for the front-end development work, and my main work was to write the customer's claim application page and the employee's work page. Huang Yuyang suggested that customer service functions should be added in the front-end webpage, so I made a chat box in the table of the manage-claims-list in the employee page. Huang Yuyang and I designed the structure and elements of the chat box. Li Shipu was also involved in the development of the front-end. She helped me unify the style of the header and footer parts of the two pages I wrote with other pages. Li Shipu also helped me do some work about website art design.

3 Technical Implement

3.1 Web

3.1.1 Front-End

- Bootstrap

We use Bootstrap as the framework for front-end development. It includes HTML, CSS and JavaScript frameworks, offering font, icons, buttons and other elements and JavaScript plug-in. Bootstrap has its own textured styles of the elements and standards for HTML and CSS, adopting modular design, and use LESS preprocessing language to implement various components and tools. It makes the development of dynamic Web pages and Web applications easier and more efficiently.[1]

The class provided by bootstrap make the developer define various elements more quickly and efficiently. We used approximately 30 bootstrap classes when developing the front-end pages to define the styles or other display effects of elements. For example, “input-group” class is to unity the style of the input elements in the form; “navbar-fixed-top” class is to fix the navbar at the top.

a. Grid System

It is a streaming system. As the size of screen or viewport increases, the system will automatically divide the screen or viewport into up to 12 columns. [2] The grid system contains predefined classes that are easy to use, creates the layout by a series of combinations of rows and columns, so the text content can then be placed into these created layouts. [3] In our web pages, the grid system is used for all forms and tables. We specified the number of columns that an element occupies by setting its class（e.g. class="col-sm-4"）, in order to make each row of the table or form have the same format, look neat and beautiful.

b. Glyphicons

At the same time, we used various Glyphicons to distinguish between different text parts of a page and beautify it: “Glyphicons Halflings” provide by Bootstrap; icons provide by FontAwsome; Google font ; also several fonts comes with the browser.

- jQuery

We also used jQuery, which is a framework of JavaScript. It encapsulates common functional code, optimizes HTML document manipulation, event handling, animation design, and Ajax interaction. It encapsulates the node as a JQ object in the framework, and carries out development operations by calling JQ object events, event-triggering functions and function operation elements. jQuery separate the JS code from the HTML code, simplifying the code, and makes development process clearer and faster. [4]

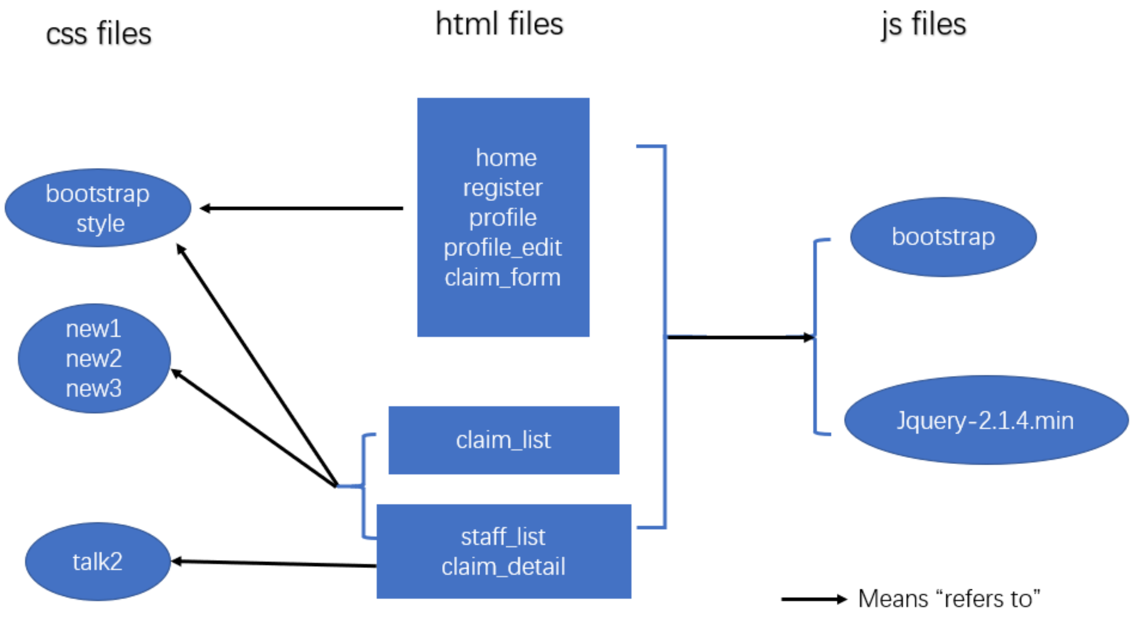


Figure 2. Reference associations between files

- Chat box

To meet the demand of the communication between staff and customers, we designed a ”Chat box” in both customer’s claim_list page and staff’s staff_list page.

Take the staff_list as example, we add a hidden chat box in the table of the staff_list page that connect to the talk2.css and talk3.css file. In the CSS code of the button, define aria-expanded= "false" to keep the hidden state of the chat box when there is not triggered, and control the pop out of the chat box by connecting the id of the chat box with the href of the button.

The chat box consists of three parts, header, message display area and message input area. Write the mytime() function in the js file to get the current time and display the current time in the header of the chat box. The input section contains a text box that defines the placeholder= "say something" of the text. When the user enters information in the text box, "say something" will be automatically cleared. The message display area contains a ul tag, and each message sending or receiving can trigger the addition of a li element to the ul tag. In talk2.css and talk3.css files, pradding is used to define outer borders with fixed spacing between the message and wrap the message inside the borders. In js file, make ul call scrollTop () method[5], so that when there is too much message， the scroll bar will automatically slide to the bottom of the content.

3.1.2 Server

- Overall server structure

On the server side, we developed two different server application to process request from android client and web browser. We chose this structure because of the reasons bellow: on one hand, this structure allows us to choose the best solution for the corresponding environment, on the other hand, by communicating with the database using the same interface, we maintained the user experience across different platforms. Furthermore, this architecture reduces the load on each of the server program, which improves the stability, also give us the ability to serve more customers compared to the single server program solution.

- SpringMVC[6]

According to the official document of SpringMVC, "Spring Web MVC is the original web framework built on the Servlet API and has been included in the Spring Framework from the very beginning." The whole framework, is designed around a controller called DispatcherServlet, while the functional work is delegated to a branch of other highly configurable components. Built with philosophy of the well-known spring framework in mind, springMVC is a really powerful framework for web application development.

- DI

One of the most important concepts in the spring framework is dependency injection. In legacy java programming every object is responsible for managing reference it depends on, which usually result in closely coupled code. While in dependency injection, those references are managed by the third party, when a new object is created, the dependency is injected automatically, this creates loose coupled code, which is superior in terms of maintainability.

- Package division

  The web back-end server is divided into four packages, the root package, the data package, the web package and the config package.

  1. The root package contains all the fundamental classes in this project, such as userinfo, login, insurance order and the item. A utility called HashGen, which is used to generate cipher-text from clear-text password and makes it easier for the developers to set up test cases and test accounts, is also included in this package.

  2. The data package contains all the repository interfaces which is used to communicate with the database. Each table in the database has its corresponding interface. All the default implementation, i.e. the implementation based on JDBC, is also in this package.

     In the JDBC implementation of the repository, we used a row mapper to convert the result set provided by the database into Java objects, this is also the place where we read the pictures form blob(binary large object) into byte array, since blobs are no longer readable once the connection to the database is closed. When inserting data, the spring framework gives us an easy to use tool called KeyHolder to deal with the data generated during the database accessing process, for example, the object ID row where auto increment is used.

  3. The web packages contain two categories of code and a web configuration class, the first one is the forms. Bound to the thymeleaf templates, these classes are responsible for the validation of user input. The other category is the controllers, this is where the logic take place.

     The web configuration class defines how the view is resolved, the parameters passed to the thymeleaf template resolver, and some encoding configuration for the web pages, which is essential for the internationalization to work correctly.

  4. The config package is the place where we put our non-web configuration classes. In these classes we tell the spring frame work where to find all the components and where to find the data source.

- Message implementation

I would like to highlight our implementation of the messaging sub-system. In order to maximize the code re-usability, all the messages related request are sent to an internal controller called FetchMessageController. To make such a request, a hidden form contains the user's role and the order which the message linked to is sent, along with the optional message text. As soon as the controller receives the request, it first save the message text to the database if it exists, then retrieve all the messages linked to the specified order. After that, the control is forwarded to another controller based on the role the customer belongs to, in order to do further processing.

- Thymeleaf [7]

According to the official website of Thymeleaf, Thymeleaf is a "modern server-side Java template engine for both web and standalone environments", aiming "to provide an elegant and highly-maintainable way of creating templates". Compared to the JSP, thymeleaf looks more similar to the normal HTML file. Benefit from its spring integrated design, it is extremely convenient to bind variables from spring code. Its conditional rendering feature makes it possible for us to combine several variations of the same page into one HTML file, thus enhance the code re-usability and maintainability. Most of the calculation done in server-side, the requirement of computational power at browser is reduced. The externalizing text feature simplifies the procedure of internationalization into writing a branch of properties file.

- Security

As we are developing the HST1C Platform, we understand that the security of sensitive user information is critical, so we used an algorithm which is called BCrypt to encrypt the user password. Developed by Niels Provos and David Mazieres in 1999, BCrypt is a cryptography hash function that is proven to be strong enough. By incorporating a random generated string, known as the salt in cryptography, BCrypt is immune from rainbow table attacks.[8] Because of the existence of this salt, even if the password of two users is identical by accident, their encrypted password, which is the one that actually being stored in the database, will not be the same.

3.1.3 Deployment

We use a light-weight Aliyun server and choose the micro-service architecture to deploy over web application and corresponding services. The micro-service architecture is a technique to deploy application and service on the cloud. By using servlet as a container, and communicate with HTTP

API via light-weight devices, we are able to deploy the war package to the tomcat program on the server. Then the end user is able to access the application and service by visiting the location of tomcat.

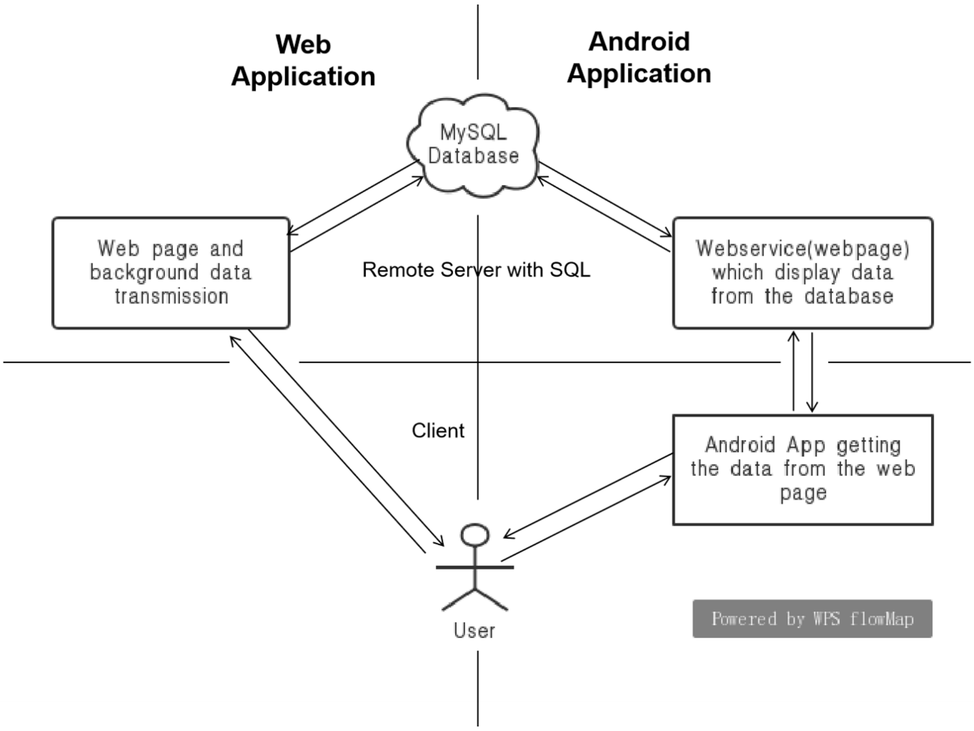

Web Application
Android Application
MySQL Database
Web page and background data transmission
Remote Server with SQL
Webservice(webpage) which display data from the database
Client
Android App getting the data from the web page
User
Powered by WPS flowMap

Figure 3. Macroscope Architecture

3.2 Android

The mobile terminal of our project is an application built with android framework that can be shared by employees and users. It has an independent server background processor and integrates instant messaging module. Adapted to Android 7.0 or above.

The core function of the application is to upload or download data to the server through the HTTP network protocol. The implementation of this function is to send POST requests to the server by creating a new OkHttpClient object (which is derived from class OkHttp3[9]). Since any time-consuming operation in android cannot be performed in the main thread, we chose AsynTask, an asynchronous thread to perform network data transmission in the sub-thread. Meanwhile, we also need to generate the header for the POST request through the FormBody before transmission, and add the data we want to send to the header through the Request object; after sending the request, the JSON format data returned from the server is received through the Response object. Finally, Google's Gson[10] class is used to parse the returned data. And Other functions will be analyzed according to the android activity page.

3.2.1 Common Page

- The first interface MainActivity.java is an animation interface, showing the animation effects for trickling in and out, and the techniques involved in this page are:

1. Implementing delayed UI operations in sub-thread using Handler's postDelayed() method, which corresponds to the thread's sleep() method, with a delay of 1500ms (that is 1.5s). The purpose of this design is to complete the preloading of the application. Within 1.5s, it is necessary to complete the check of local memory and cache, determine whether there is a login of the existing user and check the identity of the logged-in user, whether it is employee or client, and then decide whether to jump to the user interface or the customer service interface, or go to the login interface.

2. More importantly, preloading also completes the loading and deployment of the third-party instant messaging SDK. This includes the detection of network, camera, memory card storage and access permissions and some mobile phones required by the push configuration, to ensure that we can still receive instant messages and feedback from users or employees when we close the application (not log out).

- The second page LoginActivity.java contains the following functions:

3. Input field uses a listener with focus change (setOnFocusChangeListener()), When the focus of the input field changes, both the image above the input field and the icon in front of it will change to highlight the field.

4. After successful login, the user name and user type will be cached by internal storage, which will be provided to the application for uploading and judgment.

3.2.2 Client Page

- Registration

The third page RegisterActivity.java completes registration, only user accounts can be registered here, and employee accounts need to be assigned by the company. The methods and techniques included in this page are:

5. All input fields on the registration page use the TextInputLayout and addTextChangedListener to preprocess the data entered by the user. Also listen for text changes, including listening for the length of the current input string (password length

must be greater than 6 characters, less than 16 characters), whether the two password inputs are consistent, and whether the captcha input matches the content on the graphic captcha. And any item typed as null cannot be submitted to the server.

6. Verification codes are generated by Verification.java. The process is:

    a. First declare an array of random numbers, removing confusing numbers and letters, such as: 1 and I, or 6 and b, etc, and store the remaining 45 elements {'2', '3', '4', '5', '7', '8', 'a', 'd', 'e', 'f', 'g', 'h', 'j', 'k', 'm', 'n', 'p', 'r', 's', 'u', 'v', 'w', 'x', 'y', 'z', 'A', 'B', 'D', 'E', 'F', 'H', 'J', 'K', 'M', 'N', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z'} into the array, and pick any four of them.
    b. Draw interference lines, generate random colors, randomly generate text styles, colors, thickets and gradients, etc.
    c. Set four random characters and elements according to the former steps in the bitmap and return the generated bitmap to be rendered in the interface and the captcha string ready for matching. A new captcha image is switched, when the captcha picture is clicked.

- Fill personal information

The fourth page Perfect_Info.java is for filling in personal information, including the following functions：

The user's phone number and email address are still monitored by a text listener. It also uses regular expressions to verify that the phone number or E-mail address is in the correct format. The regular expressions used are as follows:

Phone:"^((17[0-9])|(14[0-9])|(13[0-9])|(15[^4,\\D])|(18[0,5-9]))\\d{8}$"

Email:"^([a-z0-9A-Z]+[-|\\.]?)+[a-z0-9A-Z]@([a-z0-9A-Z]+(-[a-z0-9A-Z]+)?\\.)+[a-zA-Z]{2,}$"

- IM authentication[11][12]

The fifth page is the authentication page, only the first time into the mobile terminal is needed. This page will complete the registration of new users for instant messaging. The technique involved is calculating the value of CheckSum (CheckSumBuilder.java). In detail, CheckSum is

SHA1(AppSecret + Nonce + CurTime), a string concatenated by three parameters, taking SHA1 hash and converting it to a hexadecimal character (string, lowercase), and upload it to the cloud database and kept in sync with our project database to ensure the consistency of data.

- Home page

The sixth interface is the HomePage.java. The interface layout uses techniques that include displaying the sidebar through DrawerLayout and NavigationView, a ViewPager that can swipe through pages, and a BottomNavigationView[13]. Meanwhile we divided three tabs, the first tab is the claim order, the second tab is the item list, the last is instant messaging.

1. Claim order information

In the claim order interface (FragmentHome.java), we added the ability to determine the current status of each claim order and to set the color for the different state by setTextColor(). It also enlarges the currently displayed claim order by determining where the current claim item is in the list.

2. Item list

In the item list interface(FragmentThree.java), RecyclerView, CardView, swipeRefreshLayout (drop-down refresh) and FloatingActionButton are used for layouts techniques. And this page also contains the following functions:

a. The first function is to add photos of items. There are two optional ways, one is to call the phone's camera to take a photo, and the other is to open the phone's album to select photos. According to Google's android developer documentation[14], the key to taking photos and selecting photos from albums is to dynamically check the read and write permissions for camera and storage card data[15], and to use content-provider instead of clear text to save the data storage path[16]. When we pass the check of the dangerous permission, open the camera or album, select the photo and finish clipping with the system tool[17], it will be stored in the album and return a parameter to the application, namely the absolute path of the picture. In order to facilitate the insertion of image information into the database, and to better share data with the web side, we choose to turn the absolute path into a bitmap firstly. And then, the image is compressed with methods ByteArrayOutputStream(). After that, converting byte array output stream to byte array (byte[]). Finally, using Base64's default encoding (base64.encodetostring())[18][19] to transfer the byte array into a String, and uploading to the server.

b. When we get the Base64 encoding of the image from the server side, we turn it into a byte array using the base64.decode() method, and present the byte array as an image through the Glide library.

c. When we click on the picture of each item in the list, the picture will be enlarged for picture preview, and can be enlarged double and dragged for preview. To achieve this, we mainly used photoView technology (ImageZoom.java).

3. IM contact and chat[12]

The IM interface(FragmentTwo.java) page integrates with the third-party SDK so that users or administrators can communicate with each other instantly, and we use the following functions:

a. Send message: create the message object through the interface provided by MessageBuilder, and then call the sendMessage interface of MsgService to send it out. The messages are divided into text messages and picture messages, and the chat type is P2P single chat mode.

b. Receiving messages: by adding a message receiving observer, MsgServiceObserve#observeReceiveMessage, our project can receive notifications when new messages arrive. A typical scenario of this code is the message dialog interface, which registers a message-receiving observer in the onCreate interface and logs out the observer in onDestroy. Upon receiving the message, determine if it is the message of the current chat object, and if so, add it to the list for display.

- Sidebar

The final user interface is the drawer layout, in which we add the user's personal information to the local cache to speed up data processing. At the same time, it also adds the function of exit, users can log out from the current interface and instant messaging system.

3.2.3 Employee Page

- The employee interface refers to the file E_WelcomeActivity.java and E_ListorderActivity.java and the functions that are included are:

1. Determine which of the three status an employee has chosen (processing, approving or denying). If the employee selects the “processing”, the claim form will have the "approving" or "denying" buttons, others will not.

2. The employee interface will also greet employees according to the current system time of the device. For example, at 9:30 AM, the page will show "Good Morning". This is done by getting the Calendar.TIME.

3. The employee interface also integrates instant messaging and chat. The principle is the same as above. And the user name list of all the current users will be obtained through the POST request for the convenience of communication.

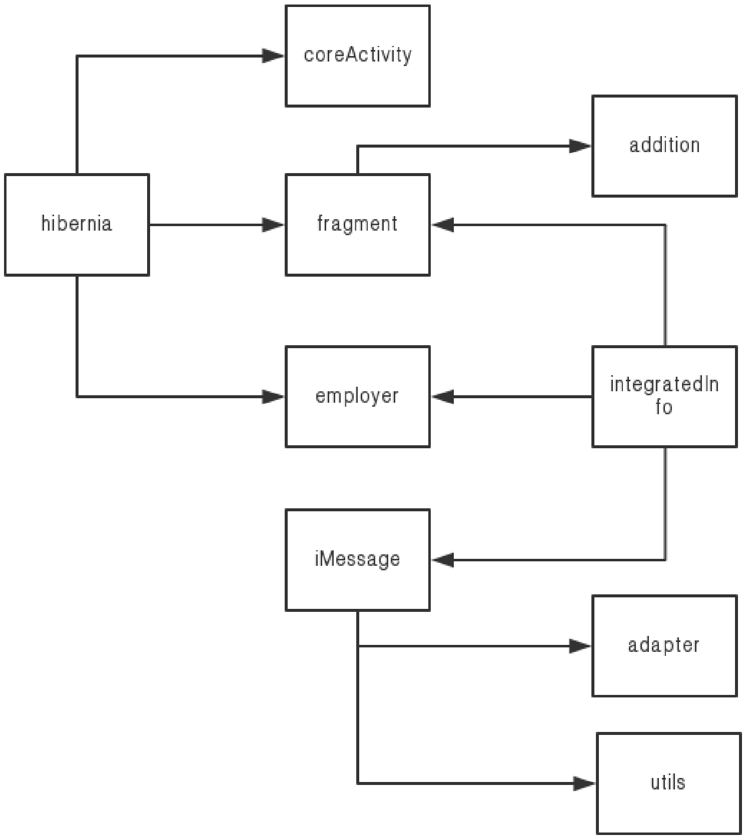


Figure 4. Android Project Package Structure

3.3 Database

3.3.1 E-R diagram

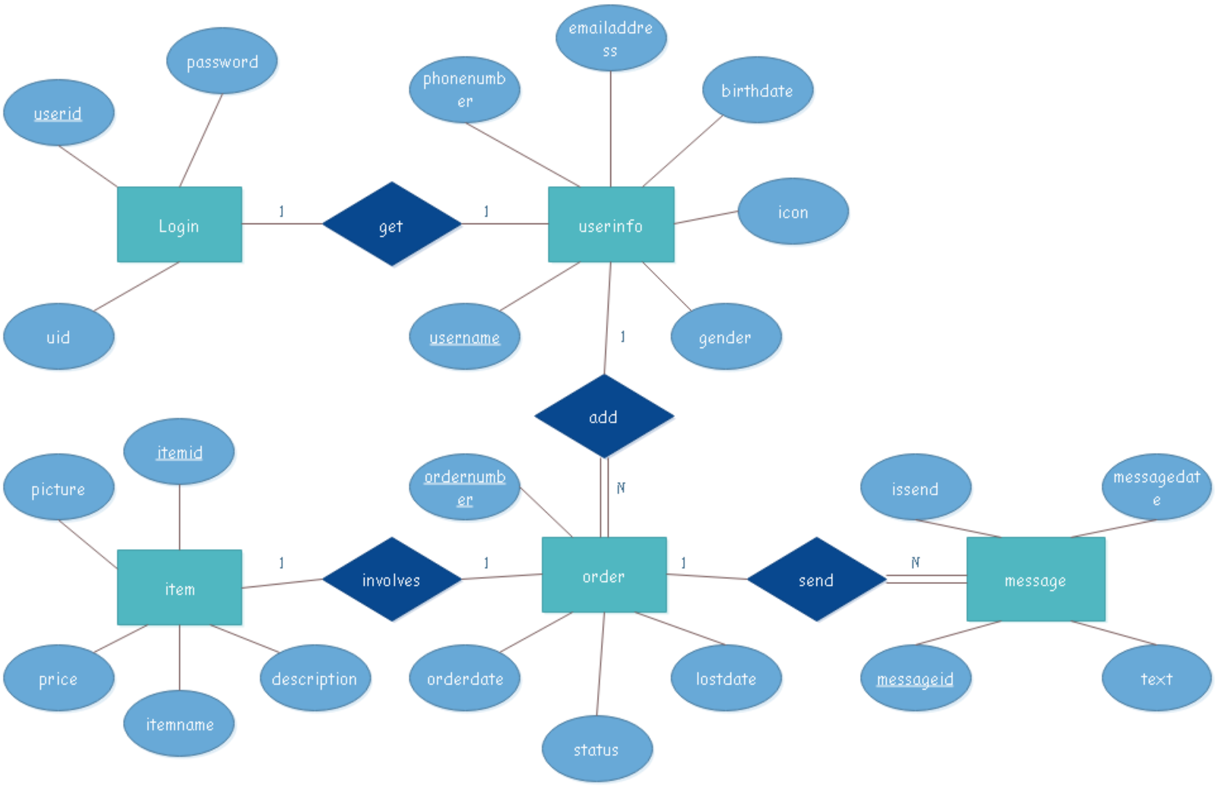


Figure 5. E-R diagram

According to the user requirement, we can easily find user information table and order table are necessary. The login table only has user name and password, and a uid to determine whether the user who logged in is an employee or a customer. Separating passwords from user information forms can keep information security better. One account has one kind of information.

Users can complete lost information to add claim order, but not all the users need to have orders. One order corresponds to one item. The item is described separately for the development of other functions, like item list.

Sometimes an order requires more information to determine whether it need to be approved or denied. So, it needs an optional table to store the message between customer and employee.

More details can be found in the appendix.

3.3.2 Database in Back-End

All repository interfaces for database communication and all the default implementation, i.e. the implementation based on JDBC, are stored in data package of back-end. Benefit from the dependency injection mechanism provided by the Spring Framework, these classes are auto-wired into the Controllers wherever they are used. This means if the development team later decide to use a different approach to connect the server program and the database, the only thing they need to do is to implement those interfaces. No modifications, not even a single line, is needed in the controllers. Also benefit from the framework, our implementations of the repositories are free from boilerplate code which contains endless boring SQLException catching code blocks.

3.4 Testing

3.4.1 User Testing

According to the feedback received by the third group after user testing, we made the following modifications to our project:

1. Since there is too little history of non-instant messaging in android app, we get and display all of the user's historical chats about an item through a list from the database.
2. With regard to the breakdown of Glide third-party library in android app, we could not change the source code of third-party developers, so we chose to use Picasso library to deal with it.
3. For redundancy in Ease of Use, we believe that order_number, item_name, and so on are not redundant information, just like Tmall or Amazon orders.
4. For when you click on the notification bar, it might jump to the error page, but it has been fixed and will jump to the main interface for preloading.
5. For the feedback from no back button on the staff side. In fact, our session returns by clicking on the space, so we add a notice to prompt the user.

3.4.2 System Testing

According to the feedback received by the third group after user testing, we made the following modifications to our project:

1. We are aware of the insecurity cased by using the http instead of https, but due to the complexity of applying for a ssl certificate and the policy of our server provider, we are not able to solve this problem. The using of https also impact over server performance.

2. We have added the verification process when new user tries to register.
3. The system tester seems to believe that the using of BLOB is the cause of slow picture loading, however, after detailed investigation we find that this is caused by the limited bandwidth of our server, which we are not able to decide. In spite of that, we do change some picture on our website to accelerate loading speed, also, solution with SQL BLOB makes it easier for us to do crash recover.
4. We decide not to manually cache the image as required by the testers, since this will introduce cache-inconsistency problems.
5. Switching of the language now return to the correct pages.
6. After discussing, we do not think policy list is a part of the basic functions, since they does not appear in the requirement.
7. For the documentation, we have add some macro-scope architecture explanations

4 Conclusion

In this version, we have implemented the basic functions, claim and confirmation of the baggage, and add some additional features like instant messaging and non-instant messaging. Features such as renew policy can be added in subsequent version updates.

With the help of teachers and teaching assistants, we successfully completed the project for this semester. In addition, thanks to the feedback from the testers, we can better improve our project.

[1] https://baike.baidu.com/item/Bootstrap/8301528?fr=aladdin

[2] https://segmentfault.com/a/1190000012889793?utm_source=tag-newest

[3] https://v3.bootcss.com/css/#grid

[4] https://blog.csdn.net/weixin_43470419/article/details/89295180

[5] https://m.jb51.net/html5/441754.html

[6] https://docs.spring.io/spring/docs/current/spring-framework-reference/web.html

[7] https://www.thymeleaf.org/

[8] https://www.usenix.org/legacy/events/usenix99/provos/provos_html/node1.html

[9] https://square.github.io/okhttp/

[10] https://blog.csdn.net/aa6385422/article/details/52511547

[11] https://wenku.baidu.com/view/d669ef93cd22bcd126fff705cc17552707225ece.html

[12]

https://dev.yunxin.163.com/docs/product/IM%E5%8D%B3%E6%97%B6%E9%80%9A%E8%AE%AF/SDK%E5%BC%80%E5%8F%91%E9%9B%86%E6%88%90/Android%E5%BC%80%E5%8F%91%E9%9B%86%E6%88%90/%E6%A6%82%E8%A6%81%E4%BB%8B%E7%BB%8D

[13] https://blog.csdn.net/fly_li_sir/article/details/79704021

[14] https://developer.android.com/training/camera/photobasics

[15] https://stackoverflow.com/questions/39787129/permission-denial-writing-android-support-v4-content-fileprovider-uri

[16] https://www.jianshu.com/p/7be437d75768

[17] https://blog.csdn.net/alex01550/article/details/82115074

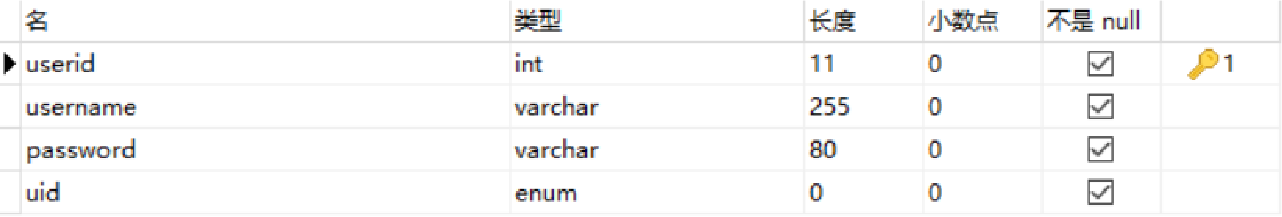
[18] https://blog.csdn.net/merbn/article/details/80410053

[19] https://blog.csdn.net/qq_35372900/article/details/69950867

5 Appendix

5.1 Detail database description

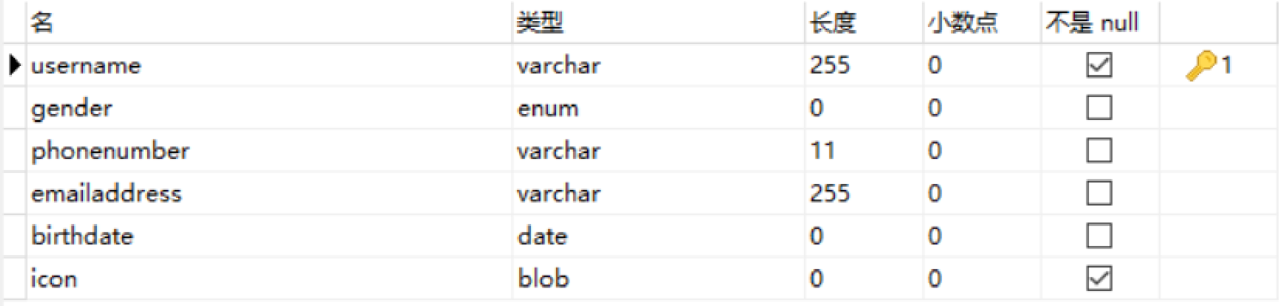
- Login



	名	类型	长度	小数点	不是 null	
▶	userid	int	11	0	☑	🔑1
	username	varchar	255	0	☑	
	password	varchar	80	0	☑	
	uid	enum	0	0	☑	

Figure 6. Login Table

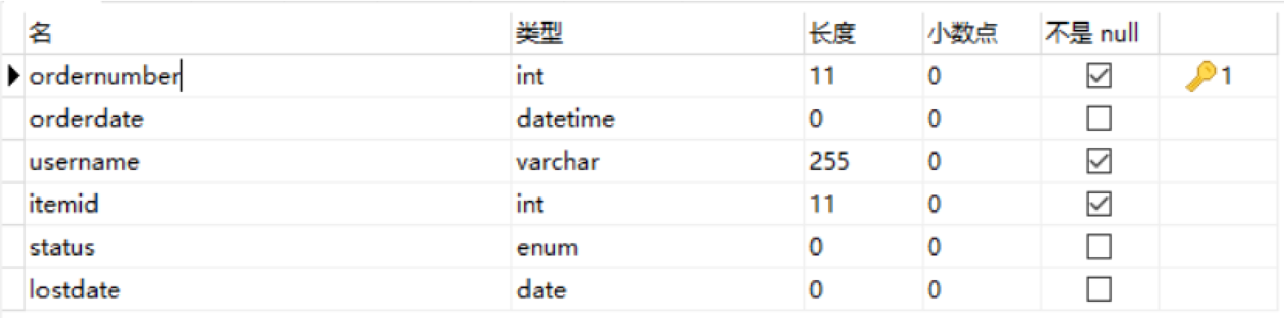
- Userinfo



	名	类型	长度	小数点	不是 null	
▶	username	varchar	255	0	☑	🔑1
	gender	enum	0	0	□	
	phonenumber	varchar	11	0	□	
	emailaddress	varchar	255	0	□	
	birthdate	date	0	0	□	
	icon	blob	0	0	☑	

Figure 7. Userinfo Table

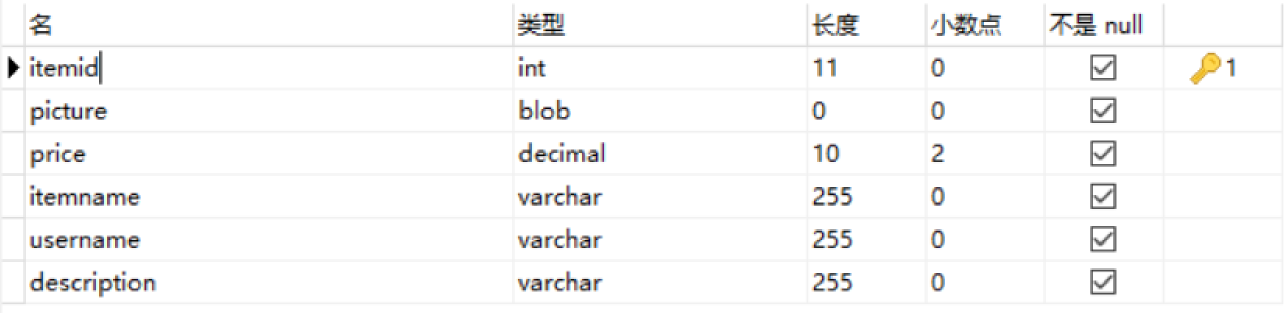
- Order



	名	类型	长度	小数点	不是 null	
▶	ordernumber	int	11	0	☑	🔑1
	orderdate	datetime	0	0	□	
	username	varchar	255	0	☑	
	itemid	int	11	0	☑	
	status	enum	0	0	□	
	lostdate	date	0	0	□	

Figure 8. Order Table

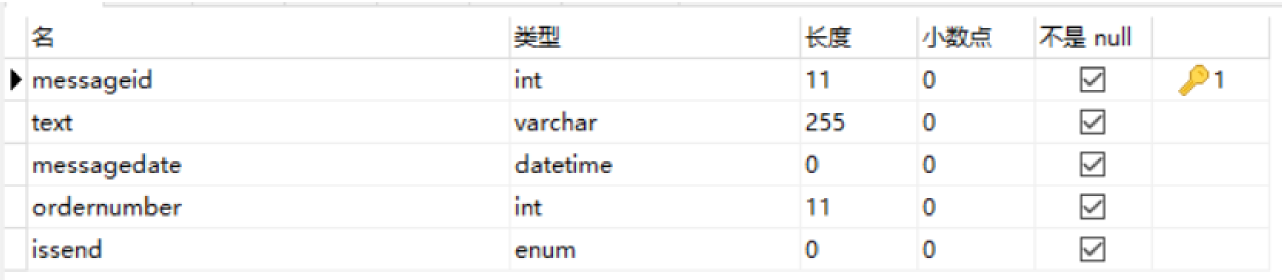
- Item



名	类型	长度	小数点	不是 null	
itemid	int	11	0	☑	🔑1
picture	blob	0	0	☑	
price	decimal	10	2	☑	
itemname	varchar	255	0	☑	
username	varchar	255	0	☑	
description	varchar	255	0	☑	

Figure 9. Item Table

- Message



名	类型	长度	小数点	不是 null	
messageid	int	11	0	☑	🔑1
text	varchar	255	0	☑	
messagedate	datetime	0	0	☑	
ordernumber	int	11	0	☑	
issend	enum	0	0	☑	

Figure 10. Message Table

5.2 Work Package

PROJECT / GROUP NAME	**Group_5_Anonymous**		
Start Date	2019.02.18	**Finish Date**	2019.05.13
Aim / Objective	Web front end development 1 (mainly for customer)		
Work package Manager	LI SHIPU 16206786		
Contributors to this package	HUANG YUYANG 16206474 JIN MINGXUAN 16206792		
Description / Activities	Task 1.1 Choose and Install the developing tool and other tool (IDEA & GitKraken). Search for information and choose a frame. Task 1.2 Design the web pages • Customer side a. main page b. register page c. claim list page d. profile view & edit page e. claim adding page f. claim detail page • Employee side: g. claim manage page Task 1.3 Implement the basic layout of page a to e Task 1.4 Testing the pages when actual apply 1.4.1 Check whether all pages display normally 1.4.2 Whether the application of components is normal 1.4.3 Test load speed Task 1.5 Adjust and beautify the interface		
Milestones		Week	
	M 1.1 a. Install all the tools needed b. Determine the front-end frame to use c. Design all the web pages	**3**	
	M 1.2 a. Implement the basic layout of page a to e	**5**	
	M 1.3 The integration of frontend and backend	**6**	
	M 1.4 Beta version report	**8**	
	M 1.5 Adjust and beautify the interface	**11**	
	M 1.6 Testing： a. Self-testing: load speed test, stability test, b. Peer-testing: User testing & system testing	**12**	
	M 1.7 Final release. Adjust according to the testing & finally optimize the interface & prepare for final report.	**15**	
Deliverables		Week	
	D 1.1 An beta version web that implements the basic functions (customer submit the claim & employee process the claims)	**8**	
	D 1.2 Implement all functions, release a final beautified, mature web	**13**	

PROJECT / GROUP NAME	**Group_5_Anonymous**		
Start Date	2019.02.18	**Finish Date**	2019.05.13
Aim / Objective	Web front-end development 2		
Work package Manager	Jin Mingxuan 16206792		
Contributors to this package	Huang Yuyang 16206474		
Description / Activities	Task 2.1 Pages front-end development 2.1.1 Customer claim-order page a. Customer page header b. A "Add a policy" button c. The main body is a table that shows all claim order records of individual customers. Each order includes information about customers and lost or damaged items, state, as well as a button to view the details of the claim order. d. Footer 2.1.2 Add claim application page a. Customer page header b. A form that needs to be filled in and submitted by the customer. The information that needs to be filled in includes customer and item information, photos, and the reasons for the claim, etc. c. Footer 2.1.2 Staff claim-order-management page a. Staff page header b. A table that shows all claim application, employ can manage all applications. c. Another table shows applications that have been processed. d. Footer Task 2.2 Front-end implementation of customer service function 2.2.1 Chat box----All claim orders in claim-order-management page have a chat box to help the staff communicate with the customer. Task 2.3 Testing and improvement 2.3.1 Adjust pages structure and style----Adjust the selection and layout of components to unify the style of all pages.		
Milestones			Week
	M 2.1 Develop customer claim-order page		**5**
	M 2.2 Develop adding claim application page		**6**
	M 2.3 Develop staff claim-order-management page		**8**
	M 2.4 Add customer service to complete the Beta version		**11**
	M 2.5 Adjust pages and components structure and style		**12**
Deliverables			Week
	D 2.1 Three completed pages without customer service.		**8**
	D 2.2 The front end of the development of customer service.		**11**
	D 2.3 Unified style for all pages, and complete code.		**12**

PROJECT / GROUP NAME	**Group_5_Anonymous**		
Start Date	2019.02.18	**Finish Date**	2019.05.13
Aim / Objective	The web site back-end of the application.		
Work package Manager	Huang Yuyang 16206474		
Contributors to this package	Jia Jinwei 16206773		
Description / Activities	Task 2.1 Implement the back-end of the server using springMVC framework • 2.1.1 Basically these functions include but not limit to registration and login functions, the password should be hashed, in order to meet the security requirements. • 2.1.2 Messaging system should be implemented. Task 2.2 Make sure that the back-end can manipulate the database correctly, main part of API between back-end and the database freezes. Task 2.3 Implement the code which interact with the Web front-end • 2.3.1 Write the web page template which binds the input/elements to variables in the server code. • 2.3.2 Merge the template with the front-end. Task 2.4 Testing the core module. • 2.4.1 web site should work normally when there is a single mobile/PC device. • 2.4.2 web site should work normally when there are more than one mobile/PC device.		
-Milestones			Week
	M 2.1 Implement the basic functions (login, register, etc..)		**4**
	M 2.2 Implement the message system		**6**
	M 2.3 Data base manipulating code implemented		**8**
	M 2.4 Code merged with front-end		**10**
	M 2.5 Final Release with testing.		**12**
Deliverables			Week
	D 2.1 Back-end codes.		**6**
	D 2.2 Data base manipulating codes.		**8**
	D 2.3 Front-end interacting codes.		**10**

<table>
<tr><td><b>PROJECT / GROUP NAME</b></td><td colspan="3"><b>Group_5_Anonymous</b></td></tr>
<tr><td><b>Start Date</b></td><td>2019.02.18</td><td><b>Finish Date</b></td><td>2019.05.13</td></tr>
<tr><td><b>Aim / Objective</b></td><td colspan="3">The design and construction of Database and connect the database to the server.</td></tr>
<tr><td><b>Work package Manager</b></td><td colspan="3">Li Meixuan 15205911</td></tr>
<tr><td><b>Contributors to this package</b></td><td colspan="3">Jia Jinwei 16206773
Jin Mingxuan 16206792
Huang Yuyang 16206474
Li Shipu 16206786</td></tr>
<tr><td><b>Description / Activities</b></td><td colspan="3">Task 5.1 Research
Conduct a demand analysis to the user to determine the basic requirement.
Task 5.2 Design
According to the research in the task 5.1, design the whole databases.
Task 5.3 Build the environment and database.
• 5.3.1 Build the local database for testing.
• 5.3.2 Build the environment in the server.
• 5.3.3 Build the database in the server.
Task 5.4 Interaction (insert, modify, delete, etc.)
Task 5.5 Testing the connection of each part, make sure they can connect and interact the database.</td></tr>
<tr><td rowspan="2"><b>Milestones</b></td><td colspan="2"></td><td>Week</td></tr>
<tr><td colspan="2">M 5.1 The determination of demand.
M 5.2 Build the local database for test
M 5.3 Implement the database in server
M 5.4 Adding the interaction
M 5.5 Test the function of the PC and Android about database
M 5.6 Final Release</td><td><b>3</b>
<b>5</b>
<b>7</b>
<b>9</b>
<b>11</b>
<b>12</b></td></tr>
<tr><td rowspan="2"><b>Deliverables</b></td><td colspan="2"></td><td>Week</td></tr>
<tr><td colspan="2">D 5.1 The basic part of database and the local database established.
D 5.2 The implement of database in server and the function of interaction.
D 5.3 Testing</td><td><b>5</b>
<b>10</b>
<b>12</b></td></tr>
</table>

PROJECT / GROUP NAME	**Group_5_Anonymous**		
Start Date	2019. 02. 18	**Finish Date**	2019. 05. 13
Aim / Objective	Android application development		
Work package Manager	Jia Jinwei 16206773		
Contributors to this package	Huang Yuyang 16206474 Lei Meixuan 15205911		
Description / Activities	Task 3.1 For the first task, we need to complete the registration and login system operation. Task 3.2 Rent a remote server through AliYun, and set up a database on the server. Task 3.3 Core function implementation Task 3.4 The last section is testing.		
Milestones		Week	
	M 3.1 Week 2-3, complete the registration and login system development. It enables users to fill in graphic captchas or even SMS captchas when registering, analyse password strength through regular expressions, encrypt passwords in three ways (any one), and store all registration information in a remote database.	**3**	
	M 3.2 Week 4 will test the login and registration system against the local database. If successful, the system can be optimized and improved, and additional functions can be added as appropriate, if there is enough time. Then I quickly started the construction of the remote server and the creation and testing of the database.	**5**	
	M 3.3 From week 9 to week 10, all the core functions, including the user layer and the employee layer, will come to an end and proceed to the test stage.	**10**	
	M 3.6 Final Release. The program on a single device runs steadily and at an acceptable speed, and online operation can achieve the desired effect (the program runs quickly, gets information accurately and won't crash).	**13**	
Deliverables		Week	
	D 3.2 In week 5, the deliverable is the login and registration system	**5**	
	D 3.2 Week 10, all the android code for the deliverable core functionality (week 5 through week 10, the core code for the user or employee layer is delivered in the middle of this phase).	**7**	
	D 3.3 The final deliverable is complete android code (Java code and corresponding layout code Xml and corresponding configuration file), database building code and query statement summary and test code.	**12**	