

KeepFeel documentation

Ze Qian, Yang Liu

Contents

1	Introduction	2
2	Process	2
2.1	Agile Development	2
2.2	UserInfo	2
2.3	Refresh	2
2.4	Ground Searching	2
2.5	Match	2
2.6	OthersInfo	2
2.7	Test	2
3	Requirements & Specifications	3
3.1	Flow of Events for the Match Person Use Case	3
3.1.1	Preconditions:	3
3.1.2	Main Flow	3
3.1.3	Subflows	3
3.2	Flow of Events for the Search Site Info Use Case	4
3.2.1	Preconditions	4
3.2.2	Main Flow	4
3.2.3	Subflows	4
3.3	Flow of Events for the Edit Person Info Use Case	4
3.3.1	Preconditions	4
3.3.2	Main Flow	4
4	Architecture & Design	5
4.1	Implementation Details	5
4.1.1	Controller	5
4.1.2	Model	6
4.1.3	View	6
4.1.4	Matching Algorithm	6
4.1.5	Image Storage	6
5	Reflections	6
5.1	Ze Qian	6
5.2	Yang Liu	6

1 Introduction

KeepFeel is a Wechat application that help Wechat users match with people interested in the same sport. Our application provide matching function from several aspects, such as time, site and sport type. After selecting the type of sport they want to play, we can give the nearby playground quickly. After selecting the playground, start time and end time, the sever will back a user list which contain the users can be matched in doing sports. There are also personal information page for helping users showing themselves to others. By giving sports type of situation keyword in ground searching part, we can give the information of playgrounds nearby. After tapping the playground, users can see the people list in that ground and after tapping the person, they can get the information of those people.

Here is our project's github link: <https://github.com/qz701731tby/KeepFeel>.

2 Process

2.1 Agile Development

In that project, we use Agile Development as our basic process frame. Agile development takes the evolution of users' needs as the core, and adopts the iterative and step by step method to develop software. In agile development, software projects are divided into several sub-projects at the beginning of construction, and the results of each sub-project have been tested and have visual, integrated and operational characteristics. In other words, a large project is broken up into smaller projects that are connected but also run independently and completed separately, leaving the software in a usable state all the time.

In this part, we developed the interface and server separately. After finishing each function, we have a unit test together. In the final testing part, we have the ensemble test together. The iteration records are shown in table1.

2.2 UserInfo

In our project, we designed UserInfo firstly, Yang finished the client view, received the data sent from the server and filled the data in the UserInfo. Ze design the *user_info* table in database. The **controller** and **service** logic searched the information for each user and the data back to client. We finished the part of ourselves and make sure the interface render information correctly.

2.3 Refresh

In the next week, we finished the construction of refreshing part, Yang finished the pages in client, got updating data of users, processed different kinds of data and sent it back to the server. Ze finished the **controller** and **service** logic to updated the personal information in *user_info* table and return it back to client.

2.4 Ground Searching

After that, we finished the ground searching part, Yang finished the design of client page and made sure what data should be send to server while Ze design *match_info* table to store the match record, finished the site searching logic, using Baidu map api and check the number of person in the site, sending the data back to client.

2.5 Match

In the following two weeks, we finished the matching page, Yang paid attention to the pages design and getting data from client, saving the data in JavaScript and sent it back to server. Ze paid attention to the the matching model, insert a record into *match_info* and return the person information fitting the match condition back to client.

2.6 OthersInfo

After those part, Ze and yang finished the othersInfo part, Yang took charge of the filling the data from server on the pages while Ze took charge of searching data from database and sending them back to client.

2.7 Test

Finally, we finishing the test part together. Yang paid attention to the data filling part and Ze cared more on the accuracy of searching data.

Time	Goal	Achievement
1st week	Finding mainstream method on design the application	Using Agile Methodology and writing documentation, including use cases, class figure and sequence diagram.
	Design UserInfo Function and OthersInfo Function	Finished UserInfo Function and OthersInfo Function, separated the refreshing part from UserInfo. Rewrite the use case.
2nd week	Design Refreshing Function	Finished Refreshing Function and solving the image sending problem.
3rd week	Design Ground Searching Function	Finished Ground Searching Function, used baidu api to get playground.
4th & 5th week	Design Matching Function	Finished Matching Function
6th & 7th week	Testing and Optimizing	Testing the app in smartphone, designing different test to make sure our app and return right result. Rewriting OthersInfo function.

Figure 1: iteration records

3 Requirements & Specifications

We finished Use Case UML for showing the process of users using KeepFeel.

We have 13 use cases, including View Personal Info, Edit Personal Info, Search Site Info, Enter Keyword, View Site List, Select Site, View Person List, Select Person, View Person Info, Match Person, Set Sport Type, Set Site, Set Time.

We use a few cases as example.

3.1 Flow of Events for the Match Person Use Case

3.1.1 Preconditions:

- A** Users have entered into the app.
- B** His/Her information has been recorded in the database.

3.1.2 Main Flow

When user switch to the interface of matching person, they can set information then click the match button to View Person List with the same thought.

3.1.3 Subflows

User can set sport type, sport site and time.

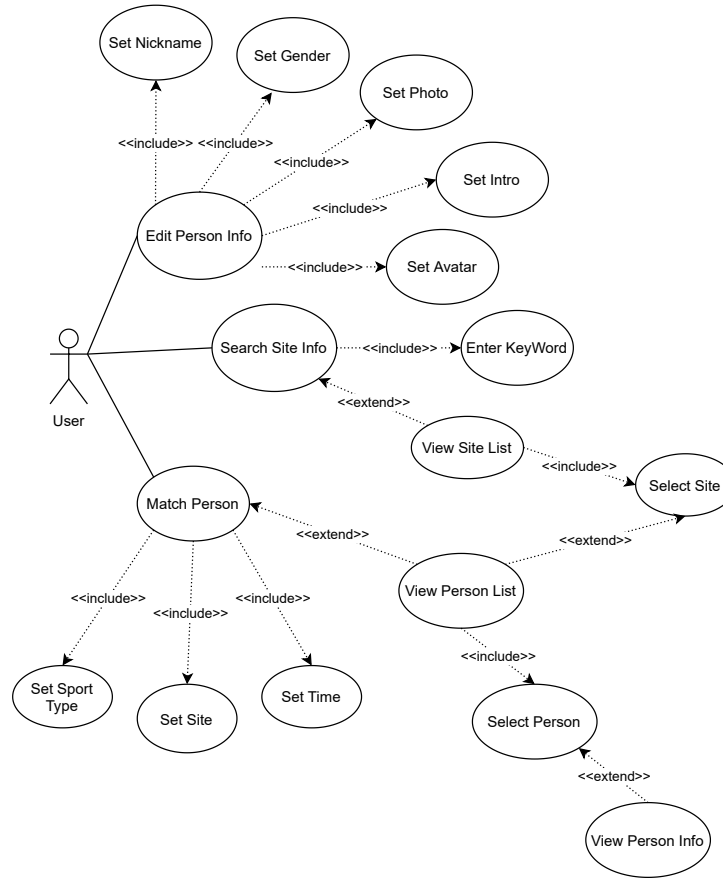


Figure 2: Caption

3.2 Flow of Events for the Search Site Info Use Case

3.2.1 Preconditions

The users have switched to ‘Search Site’ interface.

3.2.2 Main Flow

The user type the key words of their wanted playground in the ‘search’ text box and click the ‘search’ key. After the app returning a site list in the search site page, the user can click the site they interested in and then they can view the Person List. Then, user can select one site and click to view the person list in that site. User can select one person and click to view the detail information of that person.

3.2.3 Subflows

None.

3.3 Flow of Events for the Edit Person Info Use Case

3.3.1 Preconditions

Users are in the Edit Page.

3.3.2 Main Flow

The user input his or her nickname, WechatId and personal introduction in the text frame, select the gender in the picker part and choose the photo list and avatar from client. After clicking the key ‘Make Sure’, the total information will be sent to the server. The app will back to personal info page, in which the data is refreshed.

4 Architecture & Design

We implement the our system with Model-View-Controller (MVC) framework. Here is our class diagram.

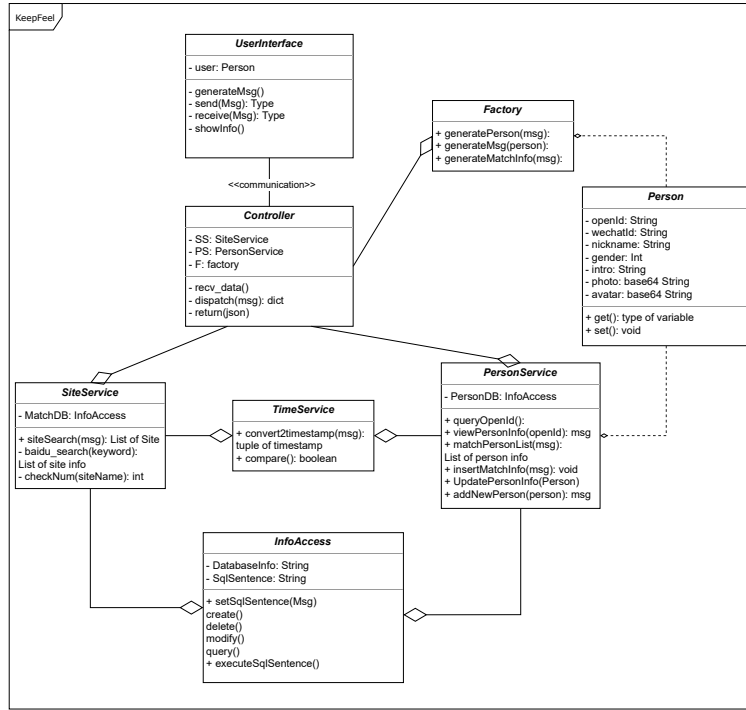


Figure 3: Caption

Here is a typical sequence diagram when a user search sites with keyword and view the person list and person info in the following process.

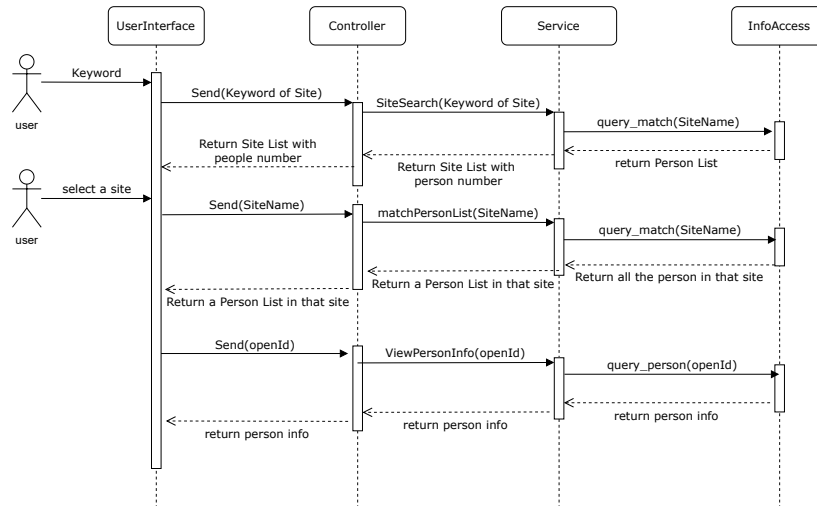


Figure 4: Caption

4.1 Implementation Details

4.1.1 Controller

The main functionality of controller is a dispatcher which calls the service function according to the message mark.

4.1.2 Model

Service class and database `infoAccess` class in the class diagram belongs to model part.

- **service:** We have site service and person service to deal with the corresponding logic. Time service maintains the whole system's time efficiency. Time service will be called when it comes to matching problems.
- **database:** The SQL operation is complicated and the same SQL function like query match will be called by many different service functions. Therefore, if we just realize it in the service part, the code will be highly redundant. To make code structure more clear, we realize the `infoAccess` class and wrap database operations in it.

4.1.3 View

Using wxml on setting blocks in pages, wxss on setting showing style and JavaScript on writing functions, saving data and receiving data from pages and sending data to server.

4.1.4 Matching Algorithm

Matching algorithm realized in our project is very simple.

- `sportsType` field is equal.
- `site` field is equal.
- period of time has overlap.

We plan to use some recommendation algorithms based on Collaborative Filtering to optimize our match results.

4.1.5 Image Storage

We use Node.js to make the server path accessible by the Wechat app with `src` path. So in database we can just store the `src` path in the "photo" and "avatar" field instead of base64 String. This can reduce the database cost and communication cost under the flask framework, since we do not have to transmit the base64 String, when a user view person information.

5 Reflections

5.1 Ze Qian

For development method, we choose the Agile Methodology, which concentrates on the customers' requirements. Each time we finish one functionality of our project, mostly based on the user interface. Once we have realized a simple version of our system and established the whole framework, we can implement iteration on our project and optimize the system step by step. For example, optimization can be better matching algorithm, faster database access, more image storage for one person, etc.

For technology realization, I gain these several views:

- I am more familiar with the MVC framework. This framework is quite convenient for app with the user interface and large amount of access database. It will reduce the code redundancy when our project becomes larger and larger.
- When designing a time system for projects, timestamp always tends to be a good choice. Since the timestamp is int type, it is easy to be compared and suitable for most condition.
- Database access with Python and operate MySQL database.

5.2 Yang Liu

For progress part, from my point of view, Agile Methodology is a perfect method to design a software for a small team. After me and Ze Qian set a plan, we work together. We are not only developer, but also consumer. After a part we have already designed, we can test it immediately, which can give a feed back quickly. We also finished a comprehensive documentation, in which we list the use cases, class figure and sequence diagram. Of course, in the progress we also find some difficulties and something that we haven't recognized before. But, after our discussion,

we solve those problems quickly.

For technology part, I learned a lot on solving different problems. For example, we solve the problem of getting users personal identities by getting openID and during the progress, I learned how apply for the login information and cutting openID from that. We also solve the problem that sending a image from client to server. I transform the image from png/jpg to base64 which is a string list and sent the data in base64 to server. In server part the data in base64 can be transformed back to image and send a src to client. In that way, the user can visit the image by using src. Except that, we also find that if the src of image does not change, even if the image is already changed in the database, the image on the client will not change. It is because the image is saved in client, if the src does not change, the client will believe that image is not changed, it will keep using the data saved locally instead of the image in server. After adding a timestamp at the image name, that problem is solved finally. We also have some problem that cannot solve now, including the time that transforming a data from png into base64 and sending base64 data will be wasted a lot. We will keeping finding good ways to improve the efficiency of our program.

Totally, I have to say that SE class and this program really helps me a lot on building a clearly structure of design and developing a program.