**FaceRecognitionAttendance**

**Project Summary**

Project Name: **FaceRecognitionAttendance**

Project Unit: **Group 1**

Project Time: **2020.3.8 - 2020.5.8**

**Content**

[1. Final product demo 2](#_Toc9017593)

[1.1 Web\_management 2](#_Toc9017594)

[1.2 App 4](#_Toc9017599)

[2 Project execution 8](#_Toc9017607)

[2.1 Schematic 8](#_Toc9017608)

[2.2 Business architecture 9](#_Toc9017609)

[2.3 Technology stack 10](#_Toc9017610)

[2.4 Management System 11](#_Toc9017611)

[2.5 App 13](#_Toc9017614)

[3 Risk plan](#_Toc9017618) 16

[3.1. Software project risk management plan 1](#_Toc9017619)6

[3.2 Risk item 16](#_Toc9017620)

[3.3 Qualitative analysis of risk 18](#_Toc9017628)

[3.4 Risk management list 20](#_Toc9017629)

[3.5 Project risk measures 21](#_Toc9017630)

[4 Test cases 24](#_Toc9017631)

[5 Lessons learnt 26](#_Toc9017632)

[5.1 For the Development 26](#_Toc9017633)

[5.2 For Project management 27](#_Toc9017634)

[6 Future plan for next cycle 28](#_Toc9017635)

[6.1 Web management system 28](#_Toc9017636)

[6.2 Face recognition algorithm 29](#_Toc9017639)

[6.3 APP 29](#_Toc9017642)

[6.4 Project management 30](#_Toc9017645)

# 1. Final product demo

## 1.1 Web\_management

### 1.1.1 Login module

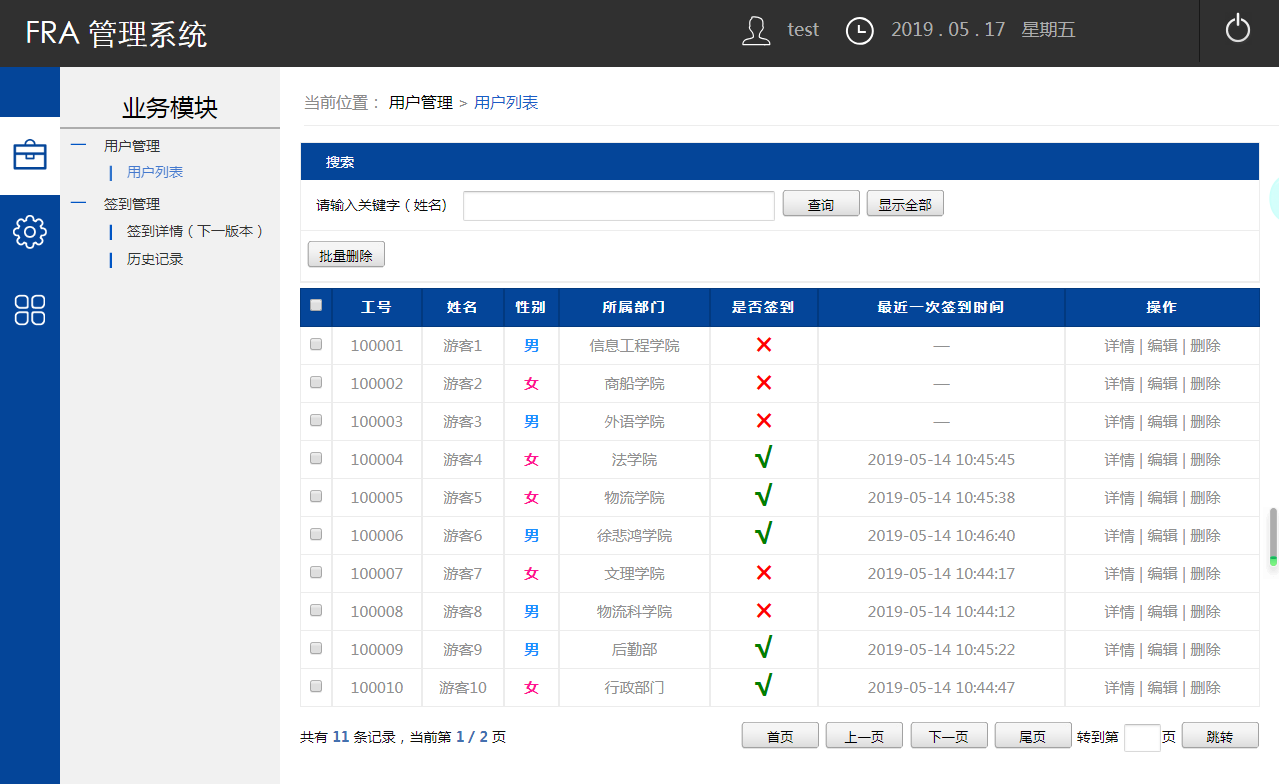


### 1.1.2 Registration module



### 1.1.3 Main interface

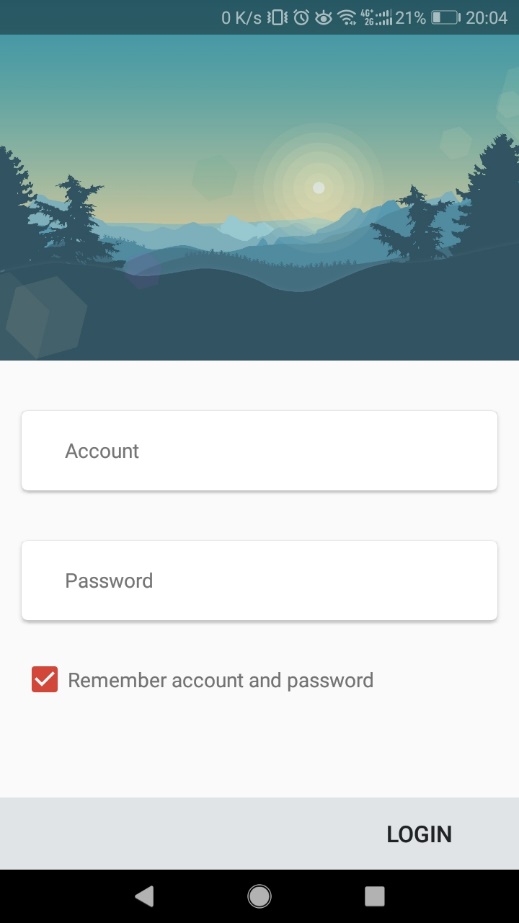
### 1.1.4 User list



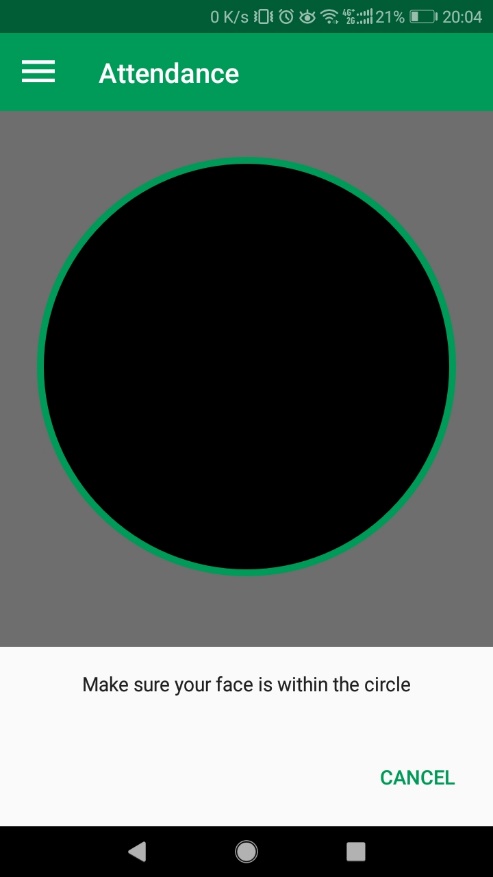
## 1.2 App

### 1.2.1 App icon

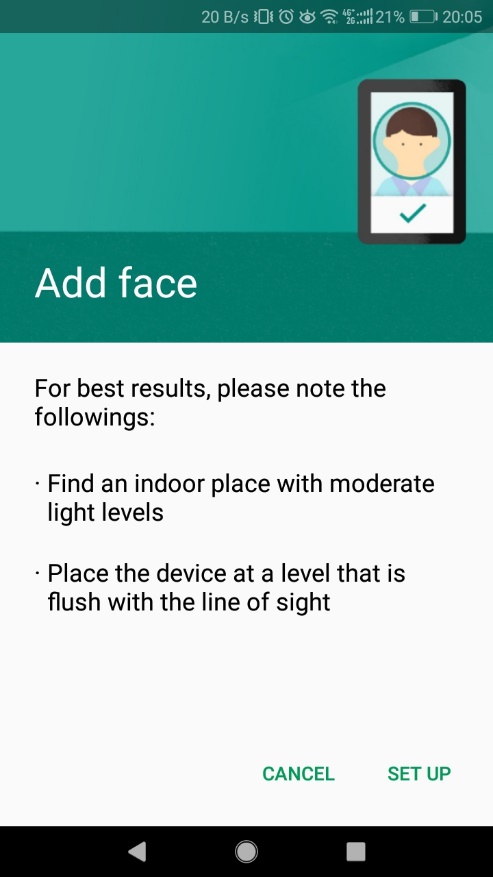
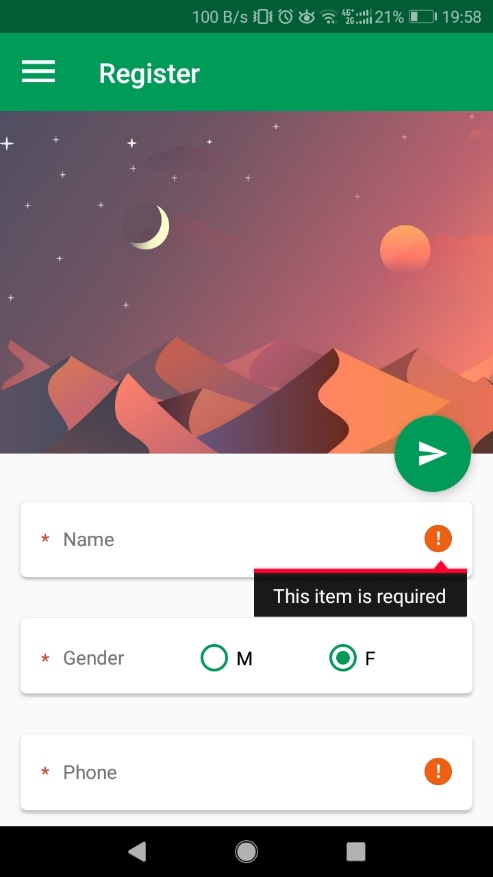
### D:\FRA\FRA_Client-Server\App\Screenshots\icon.png1.2.2 Administrator login



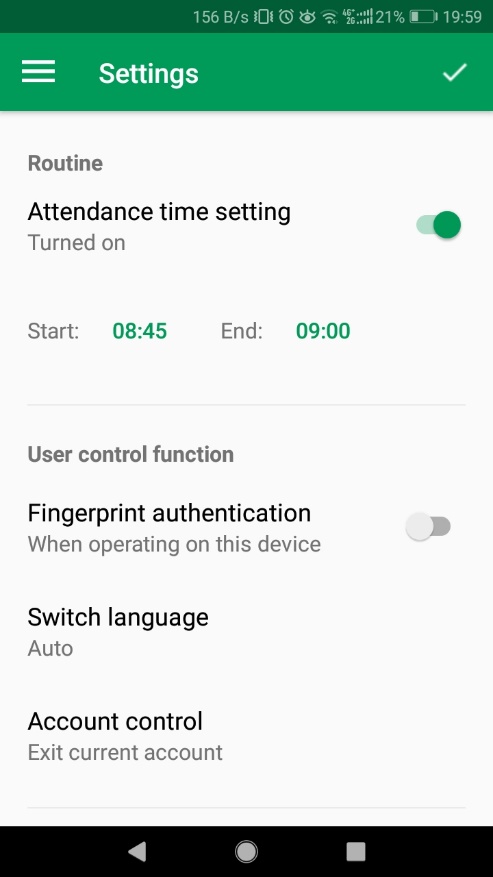
### 1.2.3 Attendance



### 1.2.4 Employee information registration

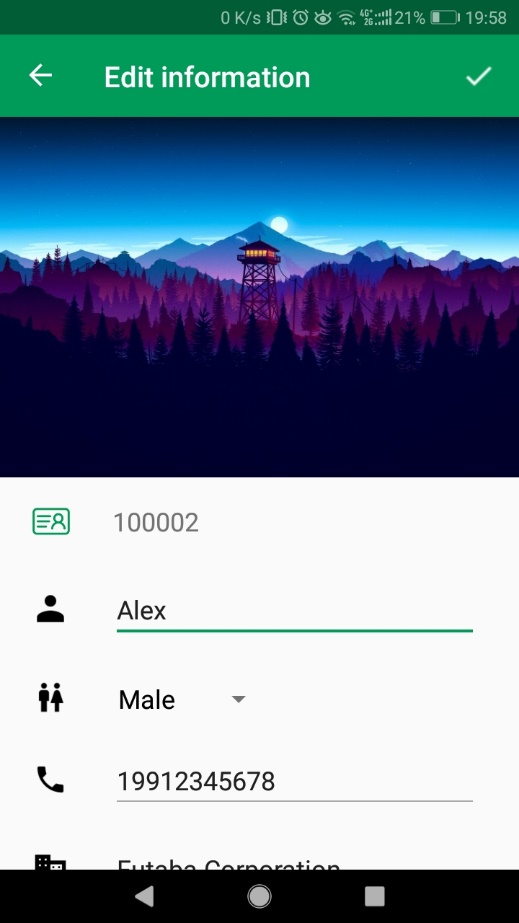


### 1.2.5 Settings



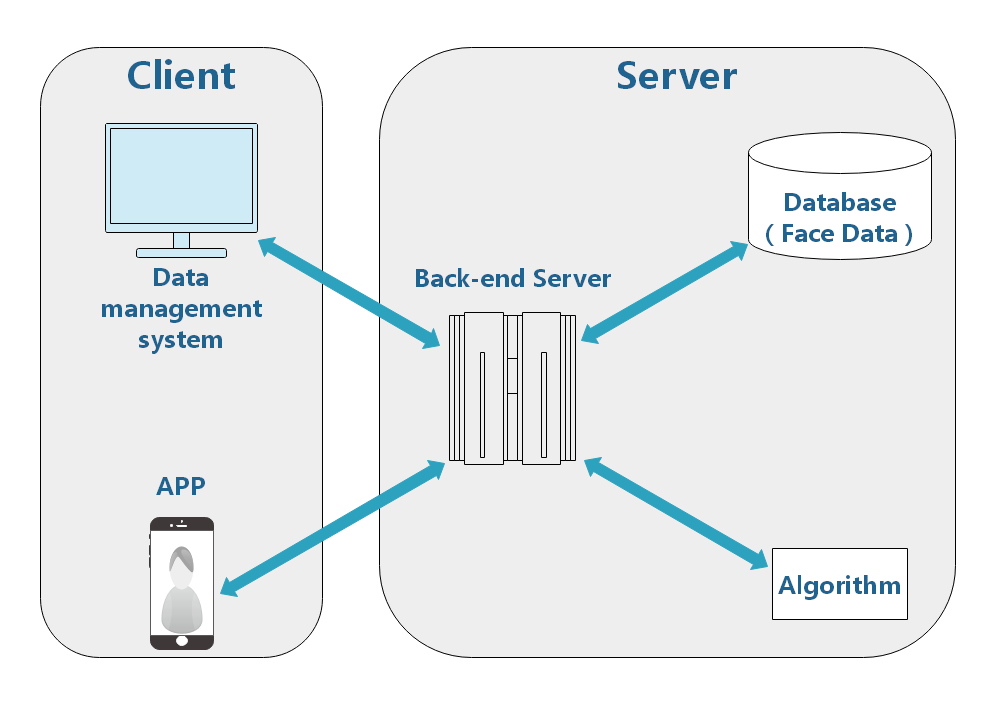
### D:\FRA\FRA_Client-Server\App\Screenshots\Screenshot8.jpg1.2.6 Shortcuts

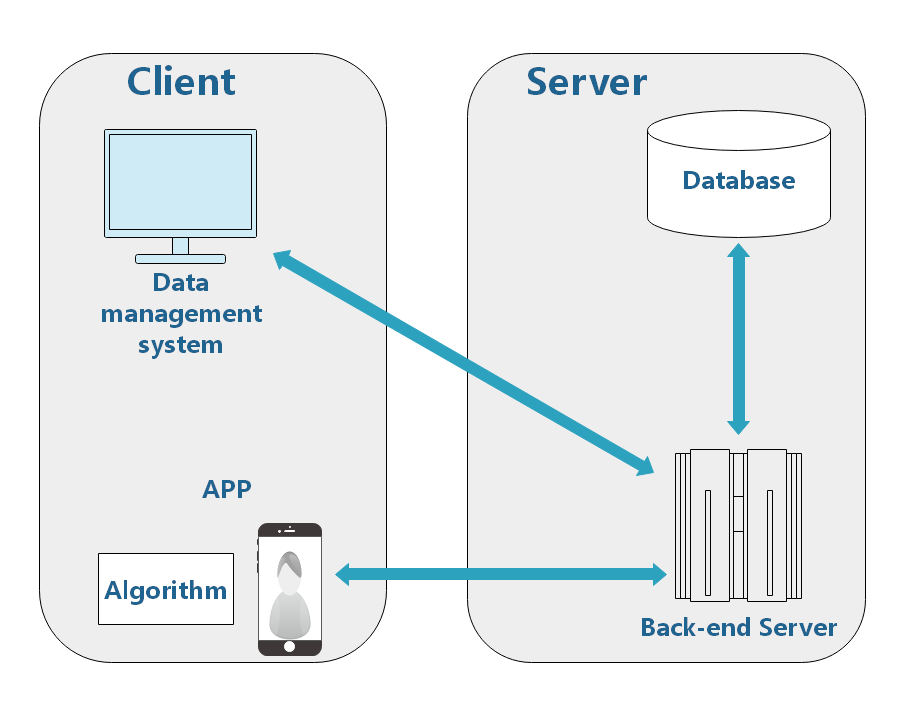
### D:\FRA\FRA_Client-Server\App\Screenshots\Screenshot4.jpgD:\FRA\FRA_Client-Server\App\Screenshots\Screenshot5.jpg1.2.7 Management



# 2 Project execution

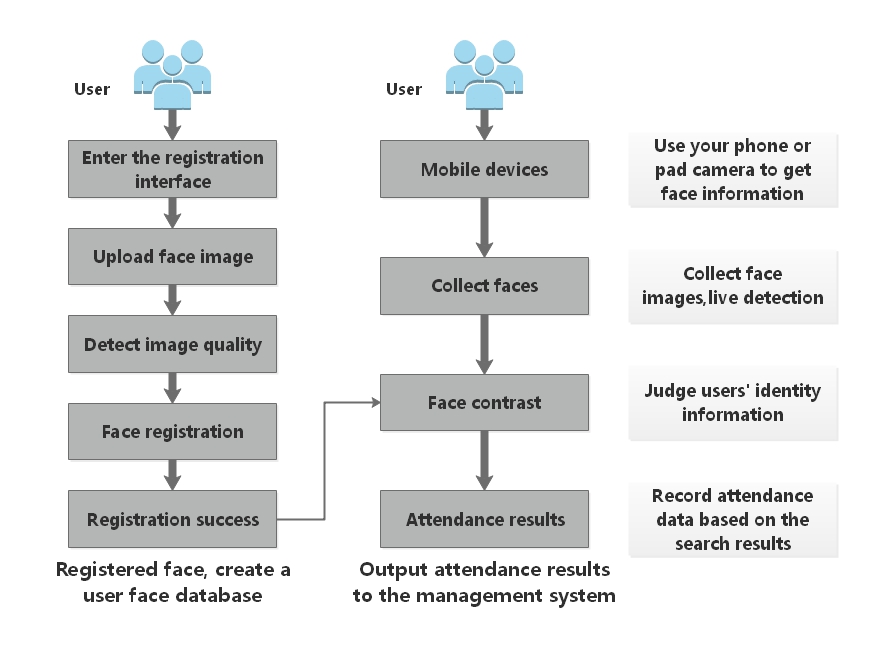
## 2.1 Schematic

**Design:**

**Achievement:**

Compared with the original system architecture, we have some changes in the implementation. The original design was to put the algorithm on the server side as an API interface for the App to use. However, in the actual development, we found that it was difficult to run the algorithm on the server side. In order not to delay the development progress, we put the algorithm in the App and directly called it. Of course, the effect that brings also has advantage and disadvantage each. The advantage is that the App will be more flexible and faster when using the algorithm. The disadvantage is that all kinds of resource library files will be encapsulated in the App, resulting in the large size of the App.

## 2.2 Business architecture

In terms of business logic, we basically implemented the business process as required.

## 2.3 Technology stack

|  |  |  |
| --- | --- | --- |
|  | **Design** | **Achieve** |
| **Algorithm** | C++ | Java |
| **APP** | MUI、5+Runtime | Android |
| **Management System** | HTML、CSS、JS | HTML、CSS、JS、jQuery |
| **Server** | Node.js | Node.js |
| **Database** | MongoDB | MySQL |

As can be seen from the above table, our technology stack has also undergone some changes, mainly algorithms, apps and databases.

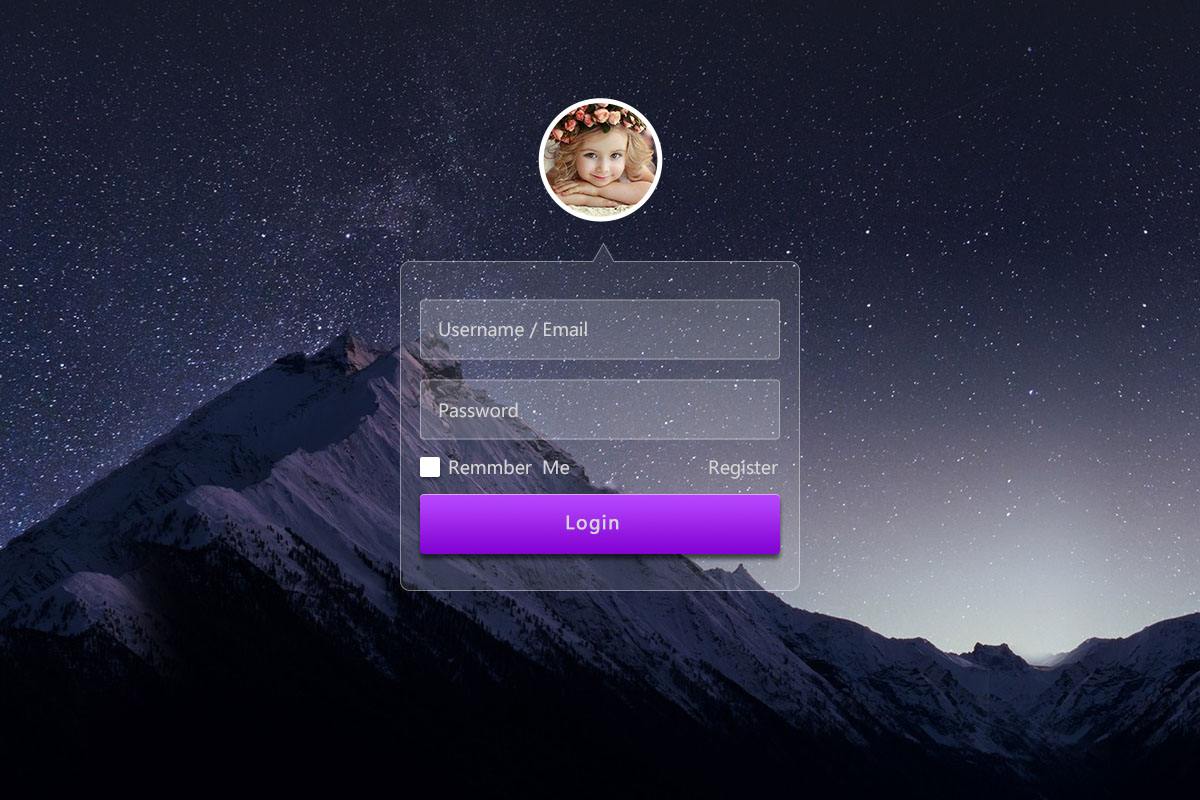
The algorithm changed from C++ to Java because the algorithm should be placed on the App side, so to be called by Android, it can only be rewritten into the Java language.

The development language of App was changed to Android, because the front-end framework we planned at the beginning could not realize the functions we had before, so we had to change to the native Android language.

The reason why I changed the database to MySQL was that MongoDB was not familiar with MongoDB, so I could not solve many problems and errors I encountered, so I had to change the database.

## 2.4 Management System

### 2.4.1 Login module

**Design:**



**Achievement:**

According to the prototype design, we also basically realized the functional requirements of the design. The late design of IU can be modified slowly.

### 2.4.2 Management module

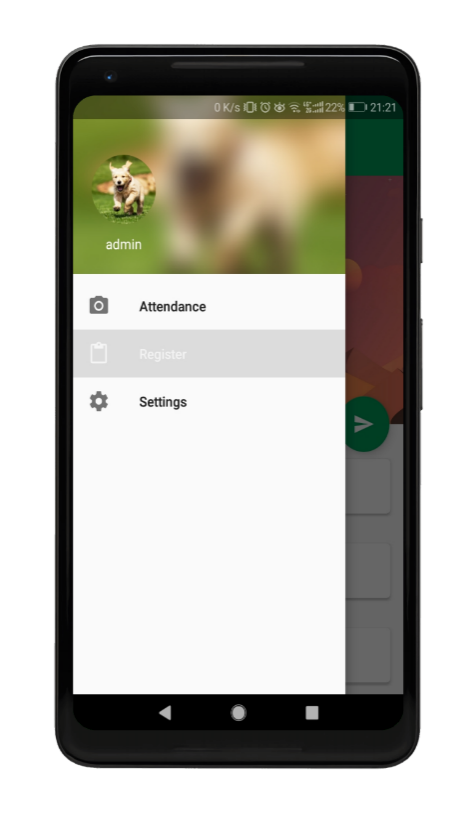
**Design****:**

**Achievement:**

According to the design requirements, we designed the management interface to meet the basic needs, but also according to the future may be added to the function of some foreshadowing design, including the design of the search box function, check-in information display.

## 2.5 App

### 2.5.1 Menu



**Design Achievement**

### 2.5.2 Registration interface

**Design Achievement**

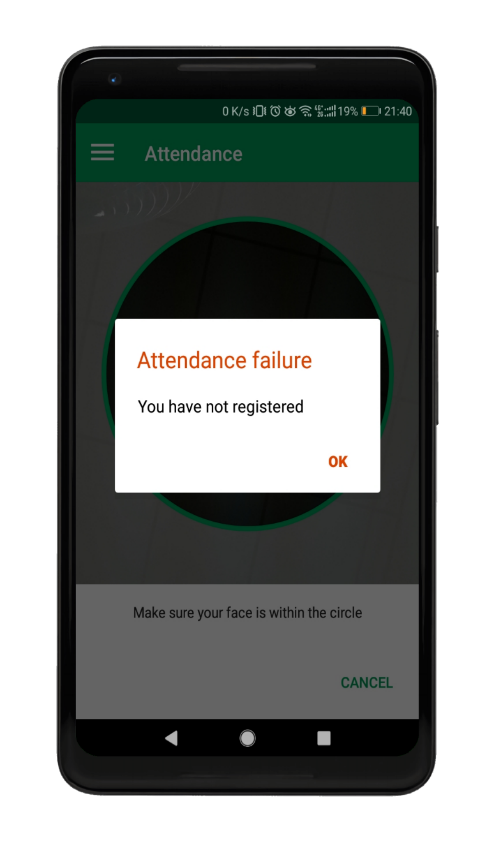
### 2.5.3 Attendance

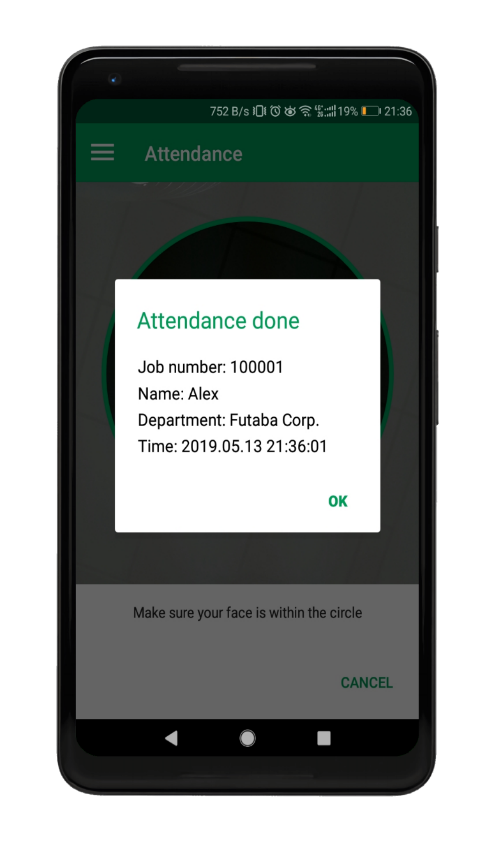
Since the original plan was implemented in the front-end language and later changed to Android, there is a big difference in the overall interface style, mainly because the two languages differ greatly in essence. At the time of later development, only relevant language design specifications can be followed. Therefore, it is found that the contrast before and after is more obvious.





**Design**





**Achievement**

# 3 Risk plan

## 3.1. Software project risk management plan

The plan focuses on the risks involved in project development, including the risks that may arise during the project development cycle and the risks that may arise from changes in the external environment during project implementation. In the article, the risks mentioned are analyzed one by one, and the corresponding risk avoidance measures are proposed. Since the risk starts to have a negative impact on the development of the project after the project starts, the lack of risk analysis or the risk avoidance measures are not effective, and it is likely to cause the failure of project development. Risk analysis is an estimate beforehand. With certain technical means and rich experience, it can basically make a relatively accurate estimation of the risk of the project. After careful consideration, it proposes feasible risk avoidance measures, which is an important link to avoid losses.

## 3.2 Risk item

### 3.2.1 Product size risk

* Develop additional features that are not needed.
* To meet software product size and speed requirements.
* Strict requirements for compatibility with existing systems.
* Relying on the technology under development.

### 3.2.2 Demand risk

* Lack of customer involvement in doing requirements.
* Poor definition of demand.
* Lack of effective demand change management processes.

### 3.2.3 Business impact risk

* No budget or manpower guarantee.

### 3.2.4 Relevant risk

* The project manager lacks management experience.

### 3.2.5 Management risk

* Project scope definition is unclear.
* Procrastination.
* Poor communication.

### 3.2.6 Technical risk

* Lack of test plan.
* Lack of quality tracking.

### 3.2.7 Number of personnel and experience risk

* Limited human resources.
* Developers have no formal training.
* Some developers in the project can work only part of the time.

## 3.3 Qualitative analysis of risk

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Category** | **Potential risk event** | **Qualitative rating of risk occurrence probability** | **Qualitative level of risk consequences** | **Comprehensive risk index** |
| **Product size risk** | Develop additional features that are not needed. | D | Ⅲ | 14 |
| To meet software product size and speed requirements. | B | Ⅱ | 5 |
| Strict requirements for compatibility with existing systems. | A | Ⅱ | 3 |
| Relying on the technology under development. | B | Ⅰ | 2 |
| **Demand risk** | Lack of customer involvement in doing requirements. | D | Ⅱ | 10 |
| Poor definition of demand. | C | Ⅱ | 6 |
| Lack of effective demand change management processes. | B | Ⅱ | 5 |
| **Business impact risk** | No budget or manpower guarantee. | C | Ⅲ | 11 |
| **Relevant risk** | The project manager lacks management experience. | B | Ⅱ | 5 |
| **Management risk** | Project scope definition is unclear | C | Ⅰ | 4 |
| Procrastination | C | Ⅲ | 11 |
| Poor communication | D | Ⅲ | 14 |
| **Technical risk** | Lack of test plan | B | Ⅱ | 5 |
| Lack of quality tracking | C | Ⅱ | 6 |
| **Number of personnel and experience risk** | Limited human resources | B | Ⅲ | 9 |
| Developers have no formal training | A | Ⅲ | 7 |
| Some developers in the project can work only part of the time | A | Ⅲ | 7 |
| Developers lack of experience | B | Ⅱ | 5 |

Risk assessment matrix example:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Impact**  **level**  **Probability**  **level** | **Ⅰ(Disastrous)** | **Ⅱ(Serious)** | **Ⅲ(Mild)** | **Ⅳ(Slight)** |
| **A (Extremely high)** | 1 | 3 | 7 | 13 |
| **B (High)** | 2 | 5 | 9 | 16 |
| **C (Medium)** | 4 | 6 | 11 | 18 |
| **D (Low)** | 8 | 10 | 14 | 19 |
| **E (Extremely low)** | 12 | 15 | 17 | 20 |

## 3.4 Risk management list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk** | **Category** | **Probability** | **Influence** | **Sequence** |
| Relying on the technology under development. | Product size risk | 90% | 3 | 1 |
| Strict requirements for compatibility with existing systems. | Product size risk | 95% | 4 | 2 |
| Project scope definition is unclear. | Management risk | 75% | 5 | 3 |
| The project manager lacks management experience. | Relevant risk | 85% | 4 | 4 |
| Developers lack of experience. | Number of personnel and experience risk | 88% | 4 | 5 |
| Lack of test plan. | Technical risk | 85% | 4 | 6 |
| To meet software product size and speed requirements. | Product size risk | 80% | 5 | 7 |
| Lack of effective demand change management processes. | Demand risk | 80% | 5 | 8 |
| Lack of quality tracking. | Technical risk | 65% | 5 | 9 |
| Poor definition of demand. | Demand risk | 70% | 5 | 10 |

## 3.5 Project risk measures

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Risk awareness** | | **Risk response measures** | |
| **Project management process** | **Potential risk event** | **Risk consequences** | **Emergency measures** | **Preventive measures** |
| **Product size risk** | Develop additional features that are not needed. | Project delay | Stop extra features, return plan | Develop and manage according to detailed design documents |
| To meet software product size and speed requirements. | Project delay | Complete a lower standard size | Product design should be considered early |
| Strict requirements for compatibility with existing systems. | Project delay | Specify a reasonable compatibility ratio | System occupancy survey in the early |
| Relying on the technology under development. | Project delay | Additional resources | Work overtime |
| **Demand risk** | Lack of customer involvement in doing requirements. | Software cannot meet the business needs | Communicate directly with the department | Designated communication management plan |
| Poor definition of demand. | Further definitions will expand the scope of the project | Modified according to department manager requirements | Let the user confirm the demand report |
| Lack of effective demand change management processes. | The project has become endless | Submit a discussion, decide | Establish a demand change management program |
| **Business impact risk** | No budget or manpower guarantee. | Project delay | Additional resources | Increase budget or manpower guarantees |
| **Relevant risk** | The project manager lacks management experience. | Project delay, Impede the ability of employees | Training or substitution | Equipped with experienced managers |
| **Management risk** | Project scope definition is unclear | The project is endless | Change according to user requirements | Clearly defined in advance and confirmed by the user |
| Procrastination | Project delay | Work overtime | Develop a detailed work plan |
| Poor communication | Project delay | Timely communication | Develop a detailed communication plan |
| **Technical risk** | Lack of test plan | Project delay, Quality problems can't be found | Additional test plan | Pre-assessment test plan |
| Lack of quality tracking | Quality issues | Solve problems in time | Develop a quality tracking plan |
| **Number of personnel and experience risk** | Limited human resources | Project delay | Additional resources | Prepare sufficient staff and arrangements in advance |
| Developers have no formal training | System function cannot be fully realized | One-on-one training | Develop a training plan |
| Some developers in the project can work only part of the time | Project delay | Additional resources | Arrange staff with sufficient working hours |
| Developers lack of experience | Project cannot be completed | Training or substitution | Equipped with experienced developers |

# 4 Test cases

Junit for App—Login module



# 5 Lessons learnt

## 5.1 For the Development

**(1) Don't use a database that you are not familiar with.**

The first big mistake we made in our development was making the mistake of using MongoDB as the database for our data persistence. As a result, I encountered many error reporting problems and could not solve them. The data collected on the Internet was also very small, so that the project development had to delay the time.

Finally, we had to replace the familiar database -- MySQL, and refactor all the database operations that had been used, which took a lot of extra time. Therefore, in the choice of database must be careful. Unfamiliar databases can cause development problems ranging from project delays to project incompletion.

**(2) Don't use technology by a subjective judgment.**

When we did the architectural design, we made a big mistake. When we chose the App function implementation, we chose the wrong programming technology, which led to all the design mistakes later. Because of the wrong design, we didn't discover the problem until we started developing. The final solution is to start from the architectural design, change all the document design.

At the beginning, our design was to solve the problem of camera call with the front-end language framework. Later, after consulting the document, we found that the API to call the camera could not meet the requirements of our check-in. Therefore, after a meeting, we changed the programming language of this part into android development. The following design is also updated according to this.

After this incident, I have learned a lesson that I cannot trust my subjective judgment of technology and language. I need to look at the description of the document and see what the technology can do before making a decision.

## 5.2 For Project management

**(1) Demand analysis and software design are very important.**

Requirement analysis is a bridge between system analysis and software design. On the one hand, requirements analysis takes system specification and project planning as the basic starting point of analysis activities, and checks and adjusts them from the perspective of software. On the other hand, requirements specifications are the primary basis for software design, implementation, testing, and maintenance. Good analytical activities help to avoid or eliminate early errors, thereby improving software productivity, reducing development costs, and improving software quality.

**(2) Execution is important. —Plan can't always be a plan.**

Executive power refers to the ability and efficiency of leading cadres to implement, and the practical ability of cadres to implement the spirit of superior and company policy instructions to achieve the set goals, including the four basic abilities of organization and coordination, leadership, policy marketing and inspection and supervision. Executive ability is an important factor that determines whether an enterprise can develop steadily and the key to the formation of its core competitiveness. Many projects fail because of poor execution.

# 6 Future plan for next cycle

## 6.1 Web management system

### 6.1.1 New features

**(1) Add application scenarios. E.g.: Campus, Company and Meeting.**

The current system function is too simple, only simple check-in function, cannot be applied to the current market, so in the next version of development, we will customize the use of user needs, such as: campus, company, conference.

**(2) Data Display, generate scenario report.**

A good management system must have strong ability in data processing, so our next version will support more data presentation and report generation to meet the needs of different users.

**(3) Support data import and export.**

Similarly, users should be able to freely use the data collected by the management system, so it is necessary to develop data import and export.

### 6.1.2 Optimization

**(1) Login and registration module details.**

The login and registration interface only does simple data processing without data validation and restriction. For the next version, you need to add the corresponding processing.

**(2) Introducing a new front-end framework.**

This version only uses simple front-end language development, no front-end framework is involved. In the next version, we will introduce a new front-end framework to make the system architecture more orderly.

**(3) Redesign UI.**

The UI style of the system needs to be reconsidered to design a more elegant interface style.

## 6.2 Face recognition algorithm

### 6.2.1 New features

**(1) Bio-assay.**

In vivo detection is now an integral part of the face recognition system, we did not add this function at the beginning due to time reasons. So in the next version of development, this will catch up.

**(2) Transfer algorithms to cloud computing services.**

Because we could not deploy the face recognition algorithm to the server, this offline algorithm App appeared. So, the next version is also something we need to work on.

### 6.2.2 Optimization

**(1) Improve the accuracy of the algorithm.**

**(2) Increase the speed of the algorithm.**

## 6.3 APP

### 6.3.1 New features

**(1) Add application scenarios (Synchronize with the web).**

The function of the app is also followed up synchronously.

**(2) Simple data display.**

The function of the app is also followed up synchronously.

**(3) Launched iOS version (Very difficult)**

At present, we only develop the android version of the app, but in the following cycle, the iOS version will also follow.

### 6.3.2 Optimization

**(1) Adapt to more phone models.**

At present, not all phones are compatible, so in the next cycle, we will look for more phones to adapt to our App.

## 6.4 Project management

**(1) Need more manpower.**

As our projects get bigger, we need more people to keep up with the pace of development. Therefore, increasing manpower is a problem that must be solved.

**(2) Use project management tools. (Likes: Huawei Software Development Cloud)**

Using project management tools is a current development trend. Management tools enable us to complete the whole process of project development more clearly, and can supervise and ensure the development of our projects. Moreover, at present, we do not use project management tools, mainly because the project scale is not large enough and the project manager's experience is not enough.

**(3) Staff training (Project manager、developer)**

Staff training is now imperative arrangement, the whole team is lack of project development experience. Therefore, if employees can participate in the training, it will improve the overall quality of the project in the later development.