

**WKU Easy Request: A Mobile  
Application for Reporting Faulty  
Equipment or Devices**

**Ye Jiada**

**Advisor:**

**Dr. Pinata Winoto**

**Department of Computer Science**

**Wenzhou-Kean University**

**May 2018**

# Abstract

A mobile faulty equipment reporting system called WKU Easy Request is designed to create an online system integrating repair reporting and management. In this project, I develop a Mini Program on WeChat which makes the users no longer need to install an application or register an account before use. Two modules are provided in the program varied with priorities: requester mode and manager mode. In the requester mode, users are allowed to request repair services and send feedbacks; in manager mode, users are allowed to edit repair persons, manage repairs and view feedbacks.

# Acknowledgment

I want to take this chance to thanks to my advisor - Dr. Pinata Winoto, who gave me the opportunity to develop my project topic. Moreover, he actively supported project plan and provided many constructive suggestions, and helped me fix my report errors.

At the meantime, I would like to appreciate my classmates. They offered me confidence and discussed with me about the project and the paper. They, as the testers of my project, helped me find out many bugs. Of course, I do want to thank my father and mother, who carefully grow me up.

# Table of Contents

|  |     |
|--|-----|
| Abstract .....   | ii  |
| Acknowledgment .....                                     | iii |
| Table of Contents .....                                  | iv  |
| Chapter 1 Introduction .....                             | 1   |
| 1.1 Background .....                                     | 1   |
| 1.2 Related Work.....                                    | 3   |
| 1.2.1 Current WKU Facility Service Request Systems ..... | 3   |
| 1.2.2 Micro Repair Platform (微报修平台) .....                | 3   |
| 1.3 Goals and Objectives.....                            | 8   |
| 1.3.1 Product Perspective .....                          | 10  |
| 1.3.2 Product Functions .....                            | 11  |
| 1.3.3 User Characteristics .....                         | 12  |
| Chapter 2 System Design and Specification .....          | 13  |
| 2.1 Requirement Specification .....                      | 13  |
| 2.1.1 External interfaces .....                          | 13  |
| 2.1.2 Functions .....                                    | 15  |
| 2.1.3 Usability Requirements .....                       | 20  |
| 2.1.4 Performance Requirements.....                      | 20  |

|  |    |
|--|----|
| 2.1.5 Logical Database Requirements .....  | 21 |
| 2.1.6 Software System Attributes .....     | 23 |
| 2.1.7 UI Requirement .....                 | 25 |
| 2.2 Design Specification .....             | 27 |
| 2.2.1 Data Design .....                    | 27 |
| 2.2.2 Architecture Design .....            | 29 |
| 2.2.3 User Interface Design .....          | 29 |
| 2.2.4 Interface Design Rules.....          | 33 |
| 2.2.5 UIDS Description .....               | 35 |
| 2.3 Prototyping .....                      | 35 |
| 2.3.1 Requester Mode .....                 | 35 |
| 2.3.2 Manager Mode .....                   | 36 |
| Chapter 3 System Testing .....             | 37 |
| 3.1 Overview of Tests Results.....         | 37 |
| 3.1.1 Tests log.....                       | 37 |
| 3.1.2 Rationale for decision .....         | 38 |
| 3.1.3 Overall assessment of tests .....    | 39 |
| 3.1.4 Impact of the test environment ..... | 39 |
| 3.2 Detailed Tests Results .....           | 39 |
| 3.2.1 Function Test .....                  | 39 |

|   |    |
|---|----|
| 3.2.2 Compatibility Test .....            | 42 |
| 3.2.3 Unit Test .....                     | 46 |
| Chapter 4 Discussion .....                | 51 |
| 4.1 Restrictions .....                    | 51 |
| 4.2 Limitations .....                     | 51 |
| 4.3 Constraints.....                      | 51 |
| Chapter 5 Future Work .....               | 52 |
| 5.1 Dependencies .....                    | 52 |
| 5.2 Assumptions .....                     | 52 |
| References .....                          | 54 |
| Appendix A Sample of Important Code ..... | 55 |
| Code 1 App Launch Code .....              | 55 |
| Code 2 Index Code .....                   | 57 |

# Chapter 1 Introduction

## 1.1 Background

In university, myriad of facilities constitute an integrated environment for students as well as faculty. One recent research indicates that the quality of facilities has significant influences on students' choice of university (Price et al., 2003). Fertile learning atmospheres, advanced laboratories, comfortable accommodations and clean eating environments become increasingly important as the reputation and contribution of institutions.

In addition, according to U.S Department of Education Excellence & Equity Commission, the quality of school facilities affects students' and faculty's basic health and the quality of curriculum (Cheng et al., 2011): public access computers and projectors are mediators between students and teachers; lighting system effectively extends working hours; air conditioners ensure fine indoor air quality and thermal comfort. Furthermore, facility conditions have an apparent connection to learning achievement in the aspect of educational functions (Roberts, 2009).

Apparently, the improvement of facilities is the very concern of university authority, with plenty of funds invested. However, facility maintenance always falls into neglect, especially for electronic facilities. There are many chairs and desks in every room, while the computer, projector and air conditioner are exclusive. When one chair, for example, is broken, it is easy to replace that. However, once a projector breaks down, it is impossible for the teacher to find a substitute in the classroom; and it is also hard to

find another classroom. Therefore, an efficient electronic facility repair request system is indispensable for WKU.

In Wenzhou Kean University (WKU), technical problems about projectors and utility computers frequently emerge, nagging all the teachers; occasional malfunctions of air conditioners and bathroom facilities may irritate some students.

The repair requests are informal and inefficient at present. For example, when a computer breaks down, someone may call IT office and wait for them to repair. Contact information to maintenance personnel is unknown to most of the students yet, and the traditional way of reporting is complicated, so students are generally unwilling to r Furthermore, no repair information would be archived to the office afterward, so it is unlikely for maintenance office to trace and analyze the repair records.

As facility managers, they may confront kinds of facility malfunction every day, like whether the new service is responded, how the repair progress goes, and whether repair persons finish the task in time. For every semester, they are responsible for drafting a regular report to show the facility situation and estimate a purchase plan. However, few records, for IT office, and the manuscript malfunction records, for dormitory office, is tough for managers to gather data. Hence, they require a management tool to record and store service information.

Besides, all malfunction reporting systems on campus are independent each other: technical facilities belong to IT office; air conditioners and lighting system are to Logistic and Asset Management Office, and dorm facilities are under dorm supervisors.

Therefore, it is really hard to gather all facility data and manage overall campus facilities.



In sum, a systematical and integrated digital malfunction reporting system is urgent for students to report malfunctions initiative and for school to manage facilities.

## **1.2 Related Work**

### **1.2.1 Current WKU Facility Service Request Systems**

In WKU, there are several individual maintenance offices: IT office, logistic & asset management, and dorm supervisors, which are unrelated to each other. For IT office, students, and teachers always call them for help and officers will come and solve the problem right away, without any malfunction records kept. If resident students meet some problems with air conditioners and bathroom facilities, they shall fill the paper-and-pencil repair request form at the window of residential supervisors (at the first floor of the dormitory). The form consists of name, room number, malfunction description and signature. Then, the supervisor will distribute a maintainer to repair and come back to feedback the progress.

From the above, it can be found that IT office is advantageous at convenient contact and dorm office is good at record retention. So it is worthwhile to combine the two merits: the service system should be convenient as well as manageable.

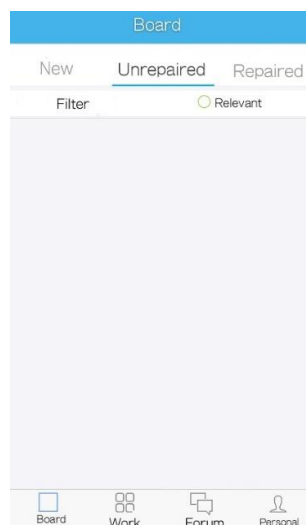
### **1.2.2 Micro Repair Platform (微报修平台)**

Recently, many mobile repair request systems exist in the market. Micro Repair Platform, for instance, is an integrated, multifunctional, charge-free service request system combined with a WeChat platform called Miro Report (微报修) for repair requesters and an individual application, Micro Repair (微检修), for administrators (Micro Repair Platform, 2017).

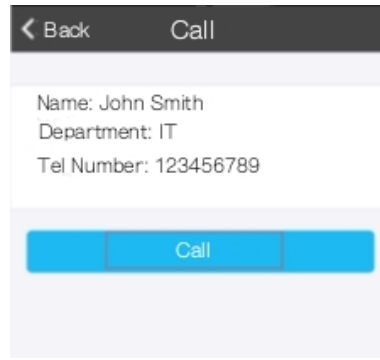
### 1.2.2.1 Micro Repair

Micro Repair is a management Application for information aggregation, service management, and service record trace, which is compatible with both IOS and Android system. Micro Repair is only accessible to Repair persons and administrators (Requester excluded). Administrators can create an individual repair platform for the organization, such as a company or a factory. At the meantime, repair persons can exchange experience in the “Forum” (Figure 1-1) where every user can participate, including external people.

On the board in the application (Figure 1-1), new requests are pending to be allocated by administrators. After allocation, the requests enter "unrepaired" and repair persons can accept them. Because facilities are divided into different departments, repair persons are restricted to accept relevant facility requests. They can phone call requesters for detailed information by the submitted contact information (Figure 1-2). When the repair task is done, the tasks are moved into "Repaired." Apart from request service, the application provides functions of routine inspection, statistical analysis, and personnel management, promoting the facility management and maintenance.



**Figure 1-1 The Taskboard in Micro Repair**



**Figure 1-2 Contact Information of Requester**

#### 1.2.2.2 Micro Report

Micro Report replaces phone call service, simplifying the request process for a requester. In Micro Report, users can:

- a) Edit the personal information (Figure 1-3)

It contains Priority (including Requester, repair person, and administrator), name and telephone number, which can be changed by the user.

| Personal Information |              |
|----------------------|--------------|
| Priority             | Requester    |
| Name                 | John Smith > |
| Tel Number           | 123456789 >  |

**Figure 1-3 The personal information in English translation in Micro Report**

- b) Request repair services (Figure 1-4)

In a request form, urgency, facility name, facility number, location, and description are required. Audio and photo are optional. The name and telephone number are automatically shown.

**Figure 1-4 The request page in English translation in Micro Report**

c) Invite more people to be requesters (Figure 1-5)

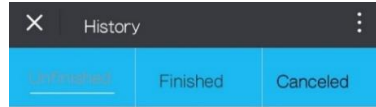
It shows current users and QR code which is used to invite more people into Micro Report as requesters.



**Figure 1-5 The invitation page with QR code in Micro Report**

d) Look at own request history (Figure 1-6)

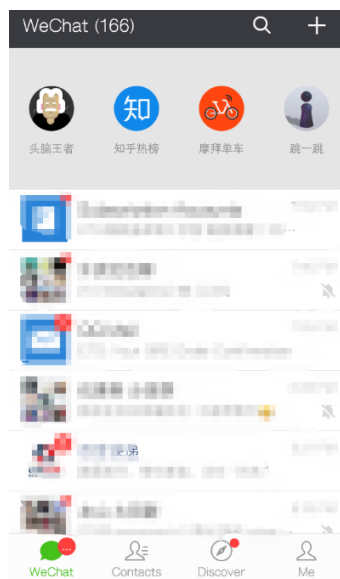
Users can check unfinished, finished and canceled request orders there.



**Figure 1-6 The request history in Micro Report**

### 1.2.2.3 WeChat Mini Programs

WeChat Mini Programs is a kind of programs attached to WeChat which is a social networking mobile application. Users are able to open Mini Programs in WeChat with no need for downloading and deleting it (Wang, 2017). The user can capture and enter the Mini Program by searching or scanning the corresponding QR code. After the application is used, it shall be stored in the user's mini program list. The user can drag down the homepage of WeChat and view it. (Figure 1-7). Mini Programs are always simple applications like hot news notification, little games, and shared bikes. Also, because of the connection to WeChat system of contact books and payment information, payment pattern and contact way in Mini Programs become easier and faster.



**Figure 1-7 The Mini Programs shown on the home page on WeChat**

Mini Program in WeChat has its own API adapting to the WeChat in Android and IOS. The framework of Mini Program provides individual interface expressive languages WXML, WXSS, and an application service language based on JavaScript to call significant functions instantly like user information acquisition and payment service (Framework, 2017).

### **1.3 Goals and Objectives**

WKU Easy Request is developed to replace the paper version of the repair request system for facilities in Wenzhou Kean University. The goals of this software are to

- provide a more orderly and simple fashion of requesting process for users.
- create an integrated and efficient digital campus facility repair system for facility managers.
- make the facility managers conduct facility planning and facility costs effectively through the efficient repair management of facilities.

WKU Easy Request is intended to work as a digital repair request management system for requesters and facility managers. WKU Easy Request will offer a repair request platform for the requester to request a repair and track the repair process; that will also provide a repair management platform for facility managers to process repairs and review the repair history. Hence, the main objective of WKU Easy Request is to improve facility management in Wenzhou Kean University.

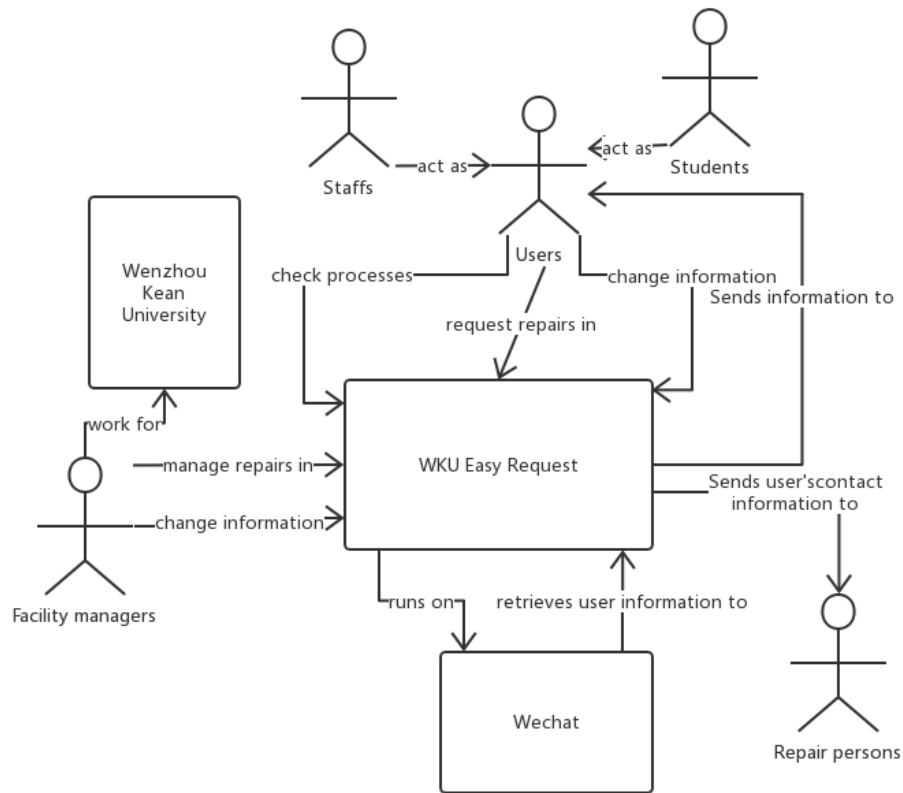
As a role in the repair process, repair persons are not employees in Wenzhou Kean University. There will be no mechanism for repair persons in WKU Easy Request.

Instead, repair persons shall receive contact data of requesters when they are allocated to one repair task by a facility manager.

To provide efficient request process to requesters, there shall be a pairing system between facility and QR code, which shall retrieve facility information from QR code in the repair request to simplify the request process.

As illustrated in the following context diagram (Figure 1-8), WKU Easy Request interfaces with several external actors:

- WKU staffs and small students can act as users. Users can request a repair, check processing repairs and change personal information in WKU Easy Request.
- Facility managers are employees who work for Wenzhou Kean University. They manage repair request using WKU Easy Request. They can change their personal information in WKU Easy Request.
- WKU Easy Request runs on WeChat. When the status of repair requests changes, it shall send information to users. When facility managers designate a repair person, WKU Easy Request shall send the user's contact information to the repair person, and vice versa.
- Repair Persons do not involve in WKU Easy Request. When some tasks are designated, the information shall be sent by WKU Easy Request through WeChat.
- WeChat acts as a platform for WKU Easy Request. It provides communication and data retrieval service for WKU Easy Request.



**Figure 1-8 WKU Easy Request Context Diagram**

### 1.3.1 Product Perspective

WKU Easy Request forms a part of the repair request management system, providing a mobile program on WeChat platform for interactions between requesters and managers. WKU Easy Request is responsible for receiving the requests submitted by the requesters and delegating them to managers for scheduling and placing on appropriate execution hosts. WKU Easy Request communicates its decision regarding repair person allocation to the resource manager, which then dispatches the message to the chosen repair person's phone. Once the assignment to a repair person has finished, WKU Easy Request is responsible for the requester's feedback on the repair.



### **1.3.2 Product Functions**

This subsection provides a high-level overview of major functions of WKU Easy Request. Note that this subsection only provides useful information for understanding the following requirements.

The primary functions of WKU Easy Request include:

- **Request a Repair**

When a facility breaks down, users can request a repair in WKU Easy Request on WeChat.

- **Process a Repair Request**

When a request is generated, facility managers can accept or reject the request.

Once the request is accepted, a repair person shall be decided to repair the facility.

When getting feedback from the repair person, the facility manager shall finish the repair.

- **Check Processing Repairs**

Any users can check processing repairs on Repair Board in WKU Easy Request.

- **Review Repair History**

Facility managers are accessible to review repair history, and the history can be sorted by date or persons in charge.

- **Change Personal Information**

All users and managers can change their personal information in Setting in WKU Easy Request.

### **1.3.3 User Characteristics**

WKU Easy Request provides following capabilities to requesters:

- Apply for a repair request
- View ongoing repairs
- Update personal information
- Send feedback

WKU Easy Request provides following capabilities to managers:

- Accept/reject repair requests
- Arrange repair persons
- Fulfill tasks
- Update personal information
- Review repair history
- View feedbacks
- Edit repair persons' contact information
- View ongoing repairs

# Chapter 2 System Design and Specification

## 2.1 Requirement Specification

### 2.1.1 External interfaces

#### 2.1.1.1 User Interfaces

WKU Easy Request interacts, either directly or indirectly, with the following significant roles, each of which will have its corresponding interface:

- Facility managers, who are any employees related to facilities in WKU:
  - IT officers, who process repair request in WKU Easy Request about electronic utility facilities, such as projectors and computers, in teaching buildings.
  - Dormitory supervisors, who are responsible for repair requests of dormitory facilities in WKU Easy Request.

The manager mode menu shall include:

- Submitted requests, which are submitted by users, pending to accept or reject.
- Ongoing requests, which are accepted and processing requests by the manager.
- Request history, which contains all repaired requests processed by any managers.

- My information, about contact data of the manager, including name and telephone number.
- Repair person's information, which can be edited by all managers.
- Users, who are any members that work or study in WKU:
  - Students, who study and live in WKU.
  - Staffs, including teachers, officers and support crews who work in WKU.

The user mode menu shall include:

- Request a repair, through which the user can apply for a request in which any facility information shall be filled by the user.
- Ongoing requests, about the submitted requests by the user, through which the user can view the information of the repair person and the facility manager.
- Repaired requests, in which user can send the feedback to the facility manager.
- My information, about contact data of the user, including name and telephone number.

#### *2.1.1.2 Hardware Interfaces*

WKU Easy Request interacts, either directly or indirectly, with the following significant external hardware:

- Manager Workstations, which are smartphones with IOS or Android system used by facility managers to perform their tasks.

- User Client, which are smartphones with IOS or Android system used by users to request a repair with WKU Easy Request over the Internet.

#### *2.1.1.3 Software Interfaces*

WKU Easy Request interacts, either directly or indirectly, with the following significant client software:

- WeChat, which is the social networking application that runs on manager and user smartphones that allow them to communicate over the networks with the WKU Easy Request.
- MySQL, which is a database management system storing data used in WKU Easy Request.

#### *2.1.1.4 Communications Interfaces*

- Internet, which is the global network used for communication among managers, users, and WKU Easy Request.

### **2.1.2 Functions**

#### *2.1.2.1 Request a Repair*

This functional feature deals with the user and the specific facility and is facilitated by the individual interface of WKU Easy Request. Hence, it is linked to the special page, with the specific user information and facility information.

- Inputs

User Information – relevant user data such as name and authentication id that needs to be used to submit a request to the facility manager. The telephone number is automatically retrieved by WeChat. The name can be edited by the user.

Facility Information – about the facility that the user wants to repair. Some facility details are provided in the initialized facility. By scanning QR Code, the relevant information is connected to the identity number. Photos and descriptions are optional inputs for users to specify the fault details.

- Processing

The system decides whether the facility is repetitively requested, and whether the user that is trying to submit a repair is a valid user or not. If the request information is valid, the request will be generated and forwarded to facility managers.

- Outputs

Request Information – user information, facility information are included. The extra information contains the request time.

#### *2.1.2.2 View Ongoing Requests*

This feature will allow the user/manager to view details about the processing request, and the progress of the execution of the repair.

- Inputs

User/Manager Authentication Information – relevant user/manager information is automatically retrieved by WeChat.

- Processing

Show the user/manager the available options for the requests, and information about the progress of repair execution. Process selected options if choices are provided to the user/manager to view different aspects of request information.

- Outputs

User/manager view of available options about requests, such as information about a user, repair person, manager, facility, and request time.

#### *2.1.2.3 Accept/Reject a Request*

This feature is available to the manager, to accept a request or reject a request under specific circumstances. If a manager accepts the request, parameters such as repair person's information are crucial to the request. If not, the reason shall be specified.

- Inputs

Manager Identity – to ensure that only the appropriate manager can accept/reject the request.

Repair Person Information – about the repair person who to allocated to repair the corresponding facility.

- Processing

Move the request from the queue to ongoing repairs and update request information, or remove it.

- Outputs

Updated request information – The output includes a Boolean value specifying whether the request has been accepted or rejected. If accepted, the state of the request shall be changed, and the manager and the chosen repair person information shall be sent to the user. Otherwise, the request shall be deleted and the reason for rejection shall be sent.

#### *2.1.2.4 Fulfill a Request*

This is the feature of WKU Easy Request that decides whether the repair is successful.

- Inputs

Manager Identity – to ensure that only the appropriate manager is able to accept/reject the request.

Decision – Boolean value specifying whether the repair is successful.

- Processing

Move the request for ongoing request to request history. Send updated request information to the user.

- Outputs

Finished Request Information – includes finish time and information about the facility, user,

#### *2.1.2.5 Edit Personal Information*

This feature allows user/manager to edit contact data.

- Inputs

User/manager Identity – to ensure that only the authenticated user/manager can edit the information.

Information Wanted to Change – can be a name or telephone number.

- Processing

Change the information record of user/manager in the database.



- Outputs

Updated Information – includes a name and telephone number.

#### *2.1.2.6 Edit Repair Person Information*

This feature allows a manager to edit repair persons' contact data.

- Inputs

Manager Identity – to ensure that only the authenticated manager can edit the information.

Repair Person Information Wanted to Update – can be a name or telephone number.

- Processing

Create/update/delete the record of repair person in the database.

- Outputs

Updated Repair Person Information – includes a name and telephone number.

#### *2.1.2.7 Send a Feedback*

This feature offers evaluation for the repair. Once a request is fulfilled, the finished request information shall be sent to the user, and the user is required to give feedback to the request.

- Inputs

User Identity – to ensure that only the authenticated user can give the feedback.

Feedback – includes evaluation scale and comments.

- Processing

Send the evaluation to the manager and update the request information in the request history.

- Outputs

Updated Request – the information would not normally be changeable.

### **2.1.3 Usability Requirements**

#### *2.1.3.1 Efficiency*

This subsection specifies the following requirements associated with the degree to which the system effectively uses its resources:

- WKU Easy Request shall not require users to permanently download software to their phones.
- WKU Easy Request shall allow users to upload pictures and descriptions into repair requests.
- The typical user shall be able to freely, easily, and quickly navigate between relevant pages.

### **2.1.4 Performance Requirements**

#### *2.1.4.1 Capacity*

This subsection specifies the following requirements concerning the minimum number of objects that the software can support:

- The software shall support a minimum of 100 simultaneous request orders.
- The software shall support a minimum of 5,000 repairs per month.

- The software shall support a minimum of 30 managers.
- The software shall support a minimum of 3,000 users.

#### 2.1.4.2 Latency

This subsection specifies the following requirements concerning the maximum time that is permitted for the software to execute specific tasks or use case paths end to end:

- The user shall be able to register a request for repair within 2 minutes.
- The manager shall be able to accept or reject a request within 3 minutes.

#### 2.1.4.3 Response Time

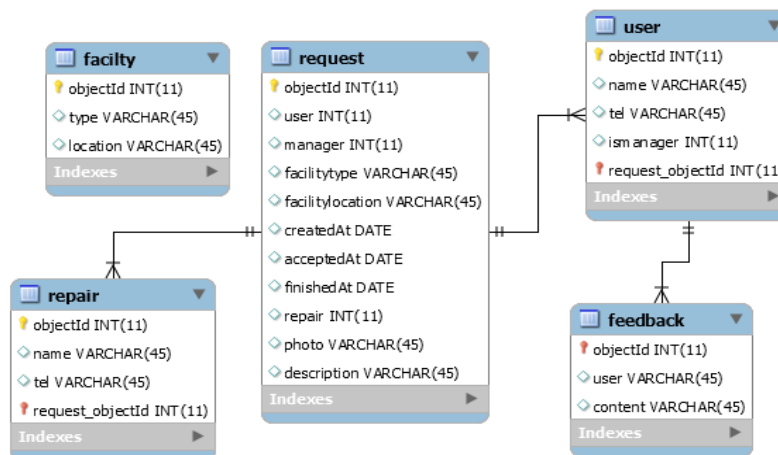
This subsection specifies the following requirements concerning the maximum time that is permitted for the system to respond to requests:

- All system responses shall occur within 5 seconds.

### 2.1.5 Logical Database Requirements

#### 2.1.5.1 Database Tables

As the following database tables show (Figure 2-1), there are four tables: manager, repairperson, request, requester, and facility.



## Figure 2-1 Database Tables

- Manager

The manager is a user stored in user table. Every manager has its identity number, name, telephone number, and priority mark. The integer ismanager (1) describes the position of the manager.

- Requester

The requester is a user stored in user table. Every manager has its identity number, name, telephone number, and priority mark. The integer ismanager (0) describes the position of the manager.

- Repair Person

The repair table stores the contact information of repair persons. It also includes the identity number, name, and telephone number.

- Facility

The facility table records all facilities which may be requested in WKU Easy Request. Every facility includes an identity number, the name, and the location mark.

- Request

The request table stores request id, facility type, facility location, requester pointer, manager pointer, request time, finish time, photo address and malfunction description. The id of the requester, repair person and manager relatively reference to the id column in the corresponding table.

#### *2.1.5.2 Accessing Capabilities*

- Managers can access tables of
  - Facility, including type and location.
  - Repairperson, including name and telephone number.
  - User, including name and telephone number.
  - Request, including manager and repair person.
  - Feedback, including requester contact and feedback content.
- Users can access tables of
  - User, including name and telephone number.
  - Request, including facility, requester, photo and description.
  - Feedback, including own contact information and feedback content.

#### *2.1.5.3 Data Retention Requirements*

The database sever shall store a minimum number of

- 100 facility manager records.
- 100 repair person records.
- 5,000 user records.
- 10,000 facility records.
- 10,000 repair records.

### **2.1.6 Software System Attributes**

#### *2.1.6.1 Human-System Integration Requirements*

This subsection specifies the following requirements for human system integration:

- IT officers are responsible for facilities in teaching buildings; dormitory supervisors can only process repair requests concerned with dormitory facilities.
- WKU Easy Request only covers a majority of facilities in Wenzhou Kean University, which are pasted with QR code.
- Users shall only view own submitted requests in WKU Easy Request, while managers can view all requests divided into different states.
- Repair persons shall have no access to system functions.

#### *2.1.6.2 Maintainability*

This subsection specifies the following requirements associated with which the system can be maintained:

- WKU Easy Request shall permit reaction time within 3 seconds.
- WKU Easy Request shall permit turnaround time within 2 seconds.
- WKU Easy Request shall permit the upgrade of software without downtime.
- The mean time to fix shall not exceed one office day.
- The database maintenance shall be taken once per month.

#### *2.1.6.3 Portability*

This subsection specifies the following requirements associated with the ease with which the system can be moved from one environment to another.

- WKU Easy Request shall enable users to use the following phone environments to interact with the software:
  - iOS: iOS 7.0 or higher.
  - Android: Android 5.0 or higher.

### 2.1.7 UI Requirement

This subsection documents the verification operations that qualify the system.

- A user scans QR code:

First time:

- When requesters scan QR code for the first time, the QR code shall transfer to identity number of the facility by the system, and My page shall be displayed. The Name and Tel column shall relatively show "Please input real name" and "Please input telephone number." Requesters can only enter New Request page after they edit their name and telephone number. Otherwise the page shall compulsively turn to My page.

Normal use:

- My page shall display the user information including name and telephone number.
- A user/manager enters the WKU Easy Request in Mini Program list
  - The Mini Program list shall be displayed when the user/manager drags down the homepage of WeChat,
  - The homepage of WKU Easy Request shall be displayed after the user/manager click the icon of the application.
- A user/manager edits personal information:
  - The information page shall be displayed after the user/manager clicks “my”.
  - The information edit page shall be displayed after the user/manager clicks “edit”.

- The information shall be updated and sent to the database after the user/manager clicks “ok”.
- A manager updates a repair person’s information
  - The repair person page shall be displayed after the manager clicks “repair person”.
  - The information edit page shall be displayed after the manager clicks “edit”.
  - The information shall be updated and sent to the database after the manager clicks “ok”.
- A manager processes a request
  - The submitted page shall be displayed after the manager clicks “submitted repairs”.

Accept:

- The choice box shall be displayed to choose a repair person after the manager accepts a request.
- The repair person’s information shall be saved into the request in the database and the repair person’s information shall be sent to the user after the manager clicks “ok”. The choice box shall disappear.

Reject:

- The reason edit box shall be displayed after the manager clicks “reject”.
- The request shall be deleted on the page and the database, and the reason shall be sent to the user after the manager clicks "ok". The reason edit box shall disappear.



Finish:

- The ongoing request page shall be displayed after the manager clicks “ongoing”.
- The ongoing request shall turn from the ongoing request page into request history after the manager clicks “finish”. The message of finish request shall be sent to the user.
- A user gives feedback
  - The evaluation page shall be displayed with input area after the user clicks “feedback”.
  - The page redirects back to My page after the user clicks “ok”.
- A user/manager view own request
  - The system displays processing request page after the user clicks “my request”.
  - All submitted, processing or historical requests relevant to the user/manager shall be displayed with requester's name, and request accept time.
  - The specific information page shall be displayed when the user/manager click the column of a request.

## **2.2 Design Specification**

### **2.2.1 Data Design**

#### *2.2.1.1 Internal Software Data Structure*

- System Files

Each page in WKU Easy Request has backend files, which store all of the interface functions, stored procedures, configurations, and variables.

- Pictures

All user interface basic pictures including button pictures and background pictures are stored in the image folder on the client server.

#### *2.2.1.2 Global Data Structure*

The user information is divided into two parts: WeChat user information and the application user information.

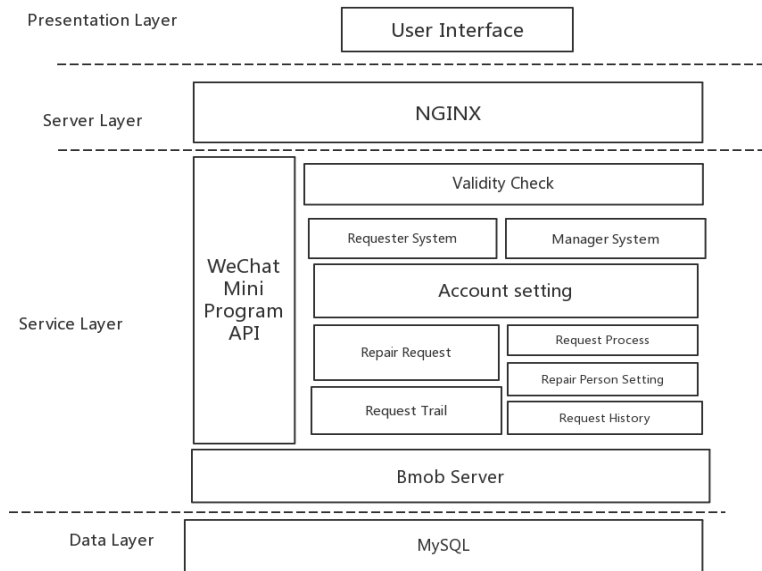
- WeChat User Information

WeChat user avatar, WeChat nick name, and openId which is the identifier of a WeChat user.

- Application User Information

That includes the telephone number, real name, priority, and objectId that is the identifier of an application user.

## 2.2.2 Architecture Design

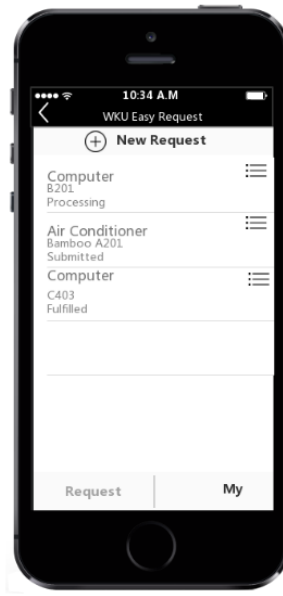


**Figure 2-2 WKU Easy Request Architecture Diagram**

## 2.2.3 User Interface Design

### 2.2.3.1 Homepage

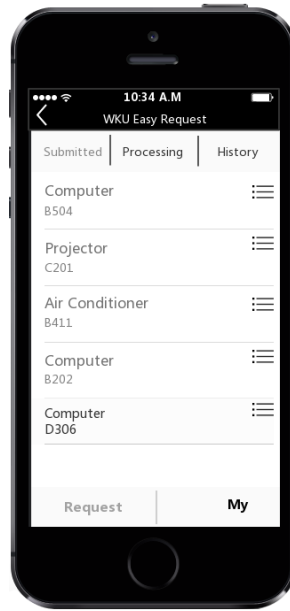
After the user finishes a request, or enter WKU Easy Request, the following page (Figure 2-3) shall be displayed. The user can press the top button of New Request to fill a request form manually. There is a list of repair information that the user is requesting. The request summary includes the facility name, the location, and the repair status. The setting button right to the summary allows the user to check the specific data including information of the facility, manager (if the request is accepted) and repair person (if the repair person is allocated). The grey button Request means disabled because Request page is the current page. The My button is linked to the personal information page.



**Figure 2-3 Index of User Mode**

#### *2.2.3.2 New Request*

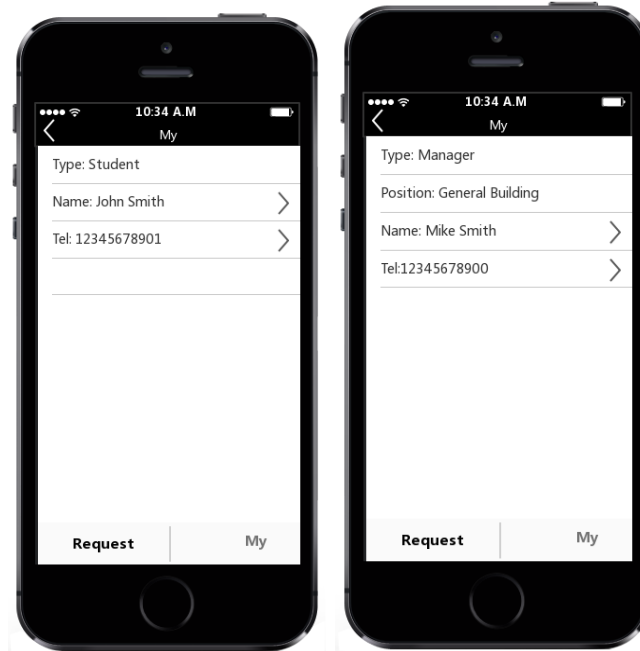
The request form is used to provide the specific facility information, user contact data, and failure cause. The user name and telephone number are already initialized and editable. If the user scans the QR code to enter the page, the basic facility data including that name, location, and identity number are completed by the system. The user can alternatively input the description and upload photos to specify the facility situation.



**Figure 2-4 The Request Form Page**

#### *2.2.3.3 My Information*

The personal information page differs from the user types. There are basic data like the user type, name, and telephone number. In the manager mode, the position is specified for the manager to take charge of facility repairs in different locations. Name and telephone number are editable by press the right arrow.



**Figure 2-5 The Personal Information Pages of User Mode and Manager Mode**

#### *2.2.3.4 Request Management*

The request page of manager mode is similar to the user mode. However, there are many modules for managers: submitted, processing and history.

In submitted page, managers are allowed to view the requests within their jurisdiction and accept or reject the submitted request.

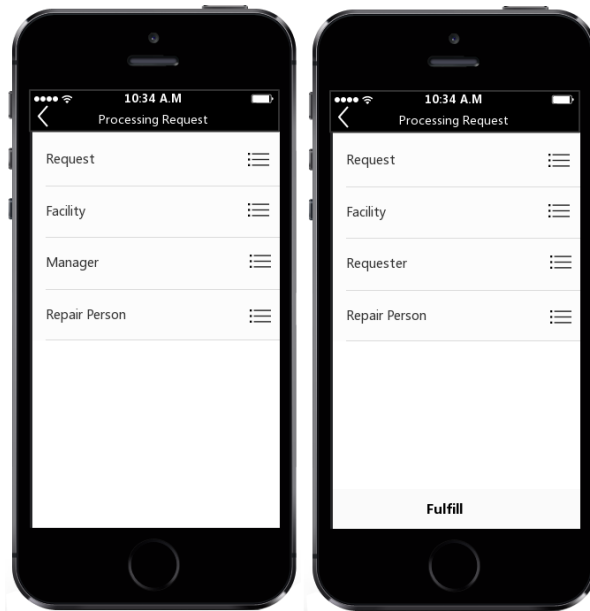
In processing page, the manager is allowed to check their requests and fulfill the requests.

In history page, there are all finished repairs. They can sort the repairs by setting different conditions.

#### *2.2.3.5 Specific Information*

After a user/manager presses the button right to the request in processing request page, the following pictures (Figure 2-7) shall be displayed with the specific request

information. Users are allowed to view all information; managers are allowed to edit repair person information and fulfill the request.

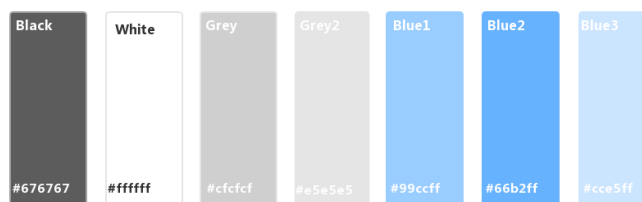


**Figure 2-7 Processing Request Specific Information Page**

## **2.2.4 Interface Design Rules**

### *2.2.4.1 Colors Scheme*

There are following colors (Figure 2-8) which are used in WKU Easy Request.



**Figure 2-8 Color Card of Interface**

#### *2.2.4.2 Interface Layout*

There are three areas in the basic interface: Blue top column, white main area, and grey navigation bar. In the main area, different information is divided by the grey lines. The interface must be designed with the minimum operation and easy navigation.

#### *2.2.4.3 GUI Components*

- Text Box

The text box is a basic component that allows the user to enter alphanumeric text on the form.

- Label

The label is a basic component that allows alphanumeric text to be displayed. That is similar to a text box, but it does not allow the user to edit.

- Data Grid

The data grid is used to display data from the database. In addition, it also can contain other components using columns and cells. This capacity allows users to create, update and delete data from the database.

- Button

Buttons allow users to perform actions upon pressing them. Different buttons may have different styles and sizes, meaning different functions.

- Dropdown List



The dropdown list is a component that allows a list of items such as repair persons. The dropdown list also allows data binding from the database.

- Image

In the request form page, there is a camera button which allows the user to upload images.

### **2.2.5 UIDS Description**

JSON — the file app.json is the global configuration file of the Mini Program as WKU Easy Request, including all page paths, user interface presentation, tab bars and so on.

WXML (Weixin Markup Language) —WXML plays a role like HTML in WeChat Mini Program, describing the structure of the page, such as labels, data grids, dropdown menus, text box, list boxes and several other objects.

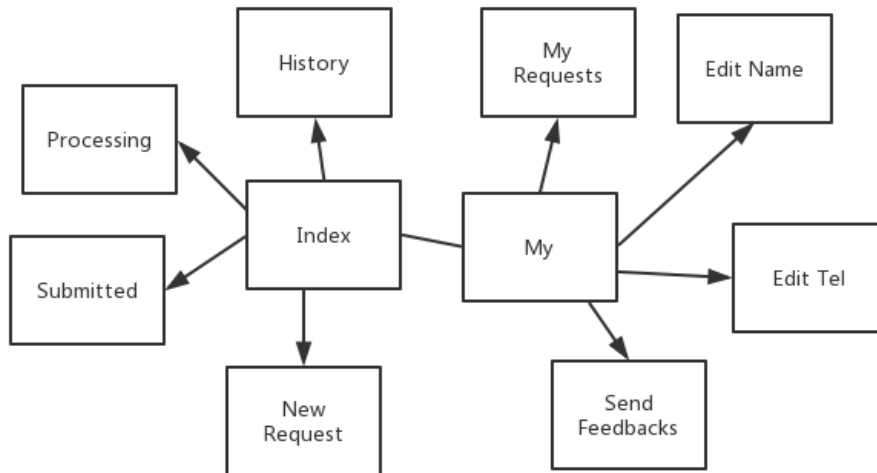
WXSS (Weixin Style Sheets) — WXSS possesses a majority of features of CSS, deciding how the components in WXML present in the user interface.

JavaScript — JavaScript files define user events in the WKU Easy Request. WeChat Developer Tool provides rich kinds of API to call some functions from WeChat conveniently, like user data retrieval and image selection and voice play.

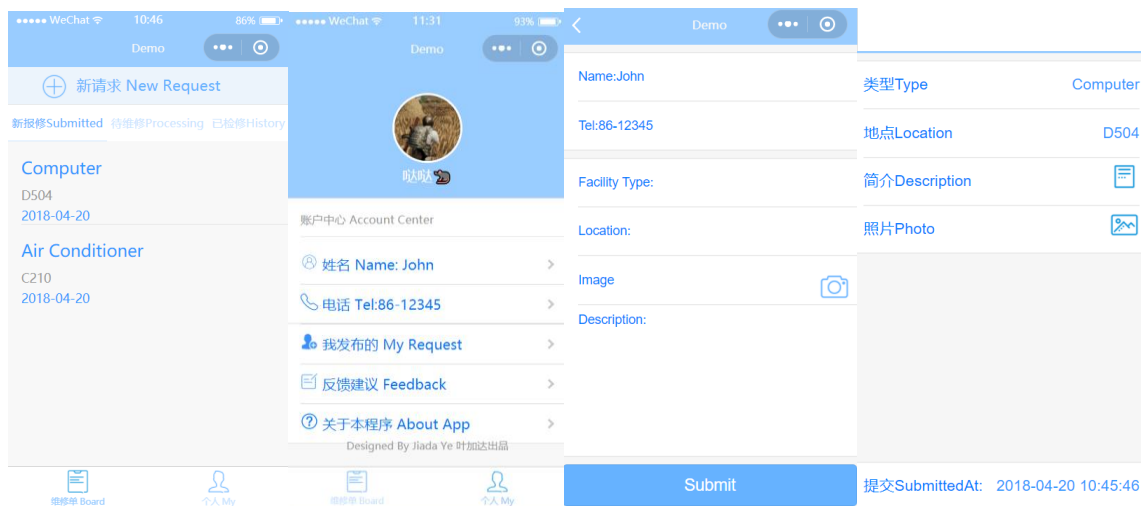
## **2.3 Prototyping**

### **2.3.1 Requester Mode**

The requester mode covers ten pages in the requester mode: My, My Requests, Edit Name, Edit Tel, Send Feedbacks, Index, History, Processing, Submitted, and New Request (Figure 2-9).



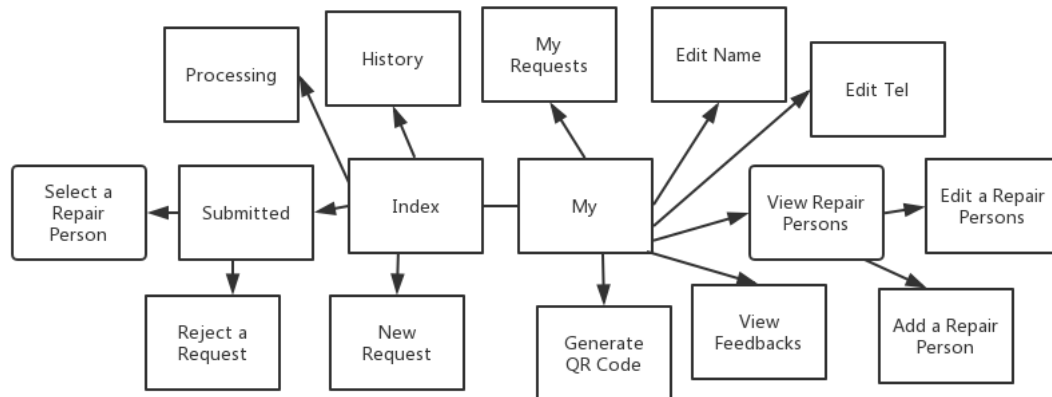
**Figure 2-9 The Page Paths of Requester Mode**



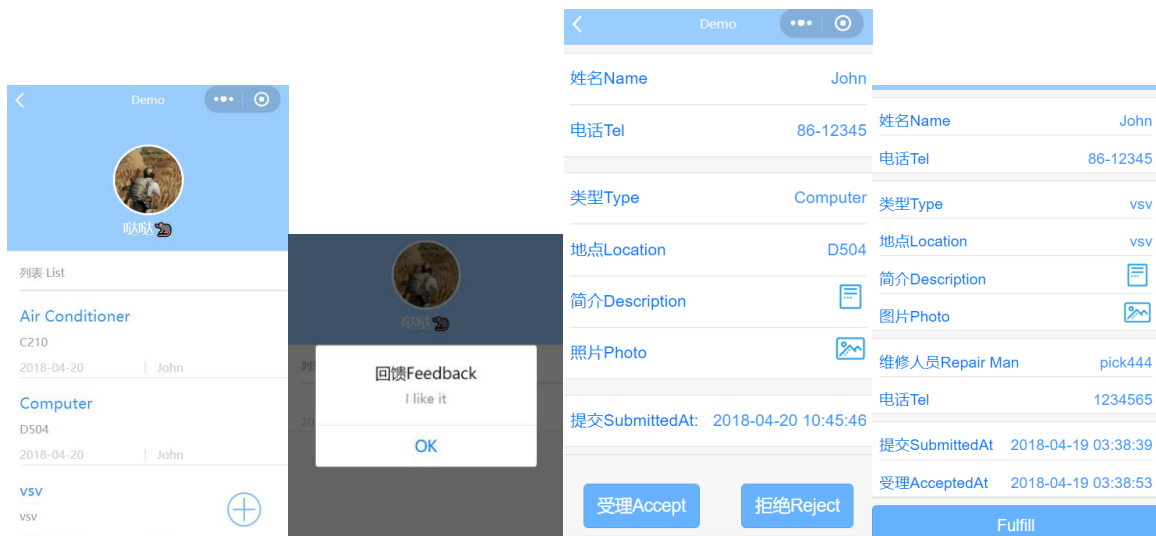
**Figure 2-10 User Interfaces of Requester Mode**

### 2.3.2 Manager Mode

The manager mode covers 16 pages in the requester mode: My, My Requests, Edit Name, Edit Tel, View Feedbacks, View Repair Persons, Edit a Repair Person, Add a Repair Person, Generate QR Code, Index, History, Processing, Submitted, Select Repair Person, Reject Request and New Request (Figure 2-11).



**Figure 2-11 The Page Paths of Manager Mode**



**Figure 2-12 User Interfaces of Manager Mode**

## Chapter 3 System Testing

### 3.1 Overview of Tests Results

#### 3.1.1 Tests log

In the development process, I found the structures of WeChat Mini Program were very different from the conventional web code style. In WXML, `<view></view>` is the

major division tag. No JavaScript module is allowed to appear in WXML. In JavaScript, some execution codes are alternately allocated into different fixed action modules instead of one major module, to adapt various page display mode and user actions. The value shown in the WXML can be mapped by `{{ value }}` from JavaScript, in which the values are passed by `that.setData({ })` function.

Due to distinctive code framework, the unit test tools testing JavaScript like JsUnit and Macha are incompatible with JavaScript files. Fortunately, WeChat Web Developer Tool provides integrated long-distance debugging tools and testing reporter.

The WKU Easy Request software (version 0.4) was tested on the WeChat Web Developer Tool located in ASUS laptop, from the April 10<sup>th</sup>, 2018 to the April 14<sup>th</sup>, 2018. The tests of the test phase (ref. software test plan) were executed.

Testers:

- Ye Jiada

### 3.1.2 Rationale for decision

After executing a test, the decision is defined according to the following rules:

- **OK:** The test sheet is set to "OK" state when all steps are in "OK" state. The real result is compliant with the expected result.
- **NOK:** The test sheet is set to "NOK" state when all steps of the test are set to "NOK" state or when the result of a step differs from the expected result.
- **NOT RUN:** Default state of a test sheet not yet executed.
- **NOT COMPLETED:** The test sheet is set to "Not Completed" state when at least one step of the test is set "Not Run" state.

### 3.1.3 Overall assessment of tests

Function test, compatibility test and unit test are executed to respond correct and proper operation of WKU Easy Request.

In function test, the table and data configurations are checked.

In compatibility test, four Android phones are tested, showing that all four models could run the application correctly, although some loaded some pages slowly.

In unit test, Entering application, creating a new request and accepting a request are thought important part to be tested. Three units smoothly ran and outputted correct data.

### 3.1.4 Impact of the test environment

All tests are performed in WeChat Web Developer Tool in the tester's ASUS laptop linked with WKU Wi-Fi.

## 3. 2 Detailed Tests Results

### 3.2.1 Function Test

**Table 3-1 Role Configurations**

| ID | Data Structure   | Comments                                 | Results |
|----|--|--|---------|
| D1 | Manager - stores information about facility managers. Every manager has its identity number, name, telephone number. | It is stored as objects in "user" table. | OK      |

|    |  |  |    |
|----|--|--|----|
| D2 | Requester - table stores the contact information of every registered requester. It includes the unique identity number, name, and telephone number.        | It is stored in “user” table, set as int (0) in column “ismanager” to be verified as a requester.                          | OK |
| D3 | Repairperson - stores the contact information of repair persons. It also includes the identity number, name, and telephone number.                         | The data are stored in “repair” table.   | OK |
| D4 | Facility - records all facilities which may be requested in WKU Easy Request. Every facility includes an identity number, the name, and the location mark. | The facility information is recorded in "facility" data table although they are not in use.                                | OK |
| D5 | Request - table stores request id, facility id, requester id, manager id, request time, finish time and malfunction description.                           | The user, manager, and repair person are set as pointers in "request" table to map the data into the corresponding tables. | OK |
| D6 | Feedback- table stores feedback contents and feed backer’s information.  | “User” column in “feedback” table is pointer using “objectId” to map data in “user” table.                                 | OK |

**Table 3-2 User functions**

| ID | General User Requirements  | Comments  | Decisions |
|----|--|---|-----------|
| U1 | General users must have the ability to request a repair                          | The contact information of the requester is fixed on the request form.  | OK        |
| U2 | General users must have the ability to set personal information                  | The name and telephone number shall be allowed to alter by the user.  | OK        |
| U3 | General user must have the ability to view part of information of all requests.  | Facility information, submit time, finish time, and accept time, manager information and repair person's information shall be shown to users. | OK        |
| U4 | General users must have the ability to view complete information of own requests | Facility information, manager information, and repair information, submit time, accept time and finish time shall be shown to the user.       | OK        |
| U5 | General users must have the ability to send feedbacks                            | The feedback shall be only shown to managers.   | OK        |

**Table 3-3 Manager functions**

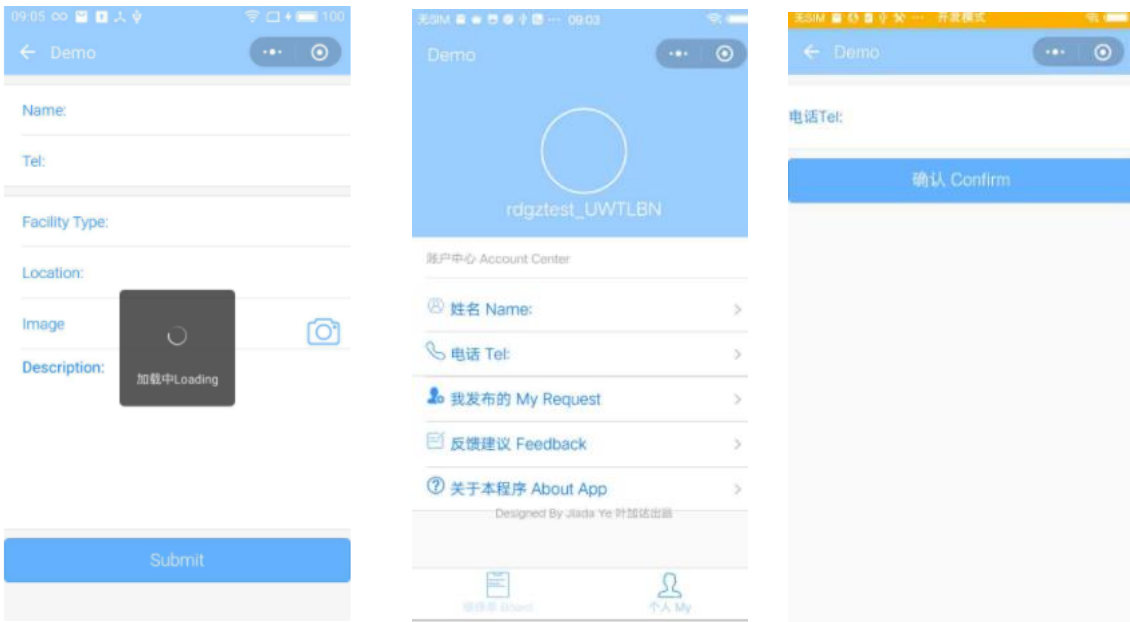
| ID | Manager Requirements   | Comments   | Results |
|----|--|--|---------|
| M1 | Managers must have the ability to accept/reject a request.       | The repair person shall be immediately assigned to the request when accepting the request. | OK      |
| M2 | Managers must have the ability to set personal information       |  | OK      |
| M3 | Managers must have the ability to fulfill a request              | The request shall be moved into history status once managers fulfill the request.          | OK      |
| M4 | Managers must have the ability to edit repair person information | The repair person information shall be allowed to create and edit by managers.             | OK      |
| M5 | Managers must have the ability to view request history           | The requester information shall be allowed to shown to all managers.                       | OK      |

### 3.2.2 Compatibility Test

WeChat Web Developer Tool offers an integrated testing tools for developers to test Mini Program performance running on different phone systems. In this test, WKU Easy Request was tested on totally 4 Android Phone models. No parameter and account setting were set in these test, so some pages were unavailable and the arguments were not



shown in the screen. Page/my/my is the home page of the Mini Program, so the load time was relatively longer.



Meizu M3, New Page

vivo Y55A, My Page

vivo Y51, Tel Page

Figure 3-1 The Testing Page on different Phone Models

3.2.2.1 Meizu M3

Resolution: 720 x 1280 pixels

System: Android 5.1

WeChat Version: 6.6.5.1280

Table 3-4 Test on Meizu M3

| Page              | Average   | CPU %   |       | Memory  |         | Results |
|-------------------|-----------|---------|-------|---------|---------|---------|
|                   | Load Time | Average | Peak  | Average | Peak    |         |
| pages/my/my       | 2.68s     | 13.18   | 39.04 | 100.47M | 133.00M | OK      |
| pages/index/index | 0.28s     | 18.44   | 39.23 | 126.46M | 138.00M | OK      |

|                            |       |       |       |         |         |    |
|----------------------------|-------|-------|-------|---------|---------|----|
| pages/my/tel/tel           | 0.46s | 18.85 | 32.04 | 109.19M | 136.00M | OK |
| pages/my/release/release   | 0.29s | 12.90 | 40.54 | 62.00M  | 132.00M | OK |
| pages/new/new              | 0.31s | 17.47 | 41.07 | 126.13M | 131.00M | OK |
| pages/my/feedback/feedback | 0.28s | 15.12 | 42.74 | 98.82M  | 132.00M | OK |

### 3.2.2.2 vivo Y55A

Resolution: 720 x 1280 pixels

System: Android 6.0

WeChat Version: 6.6.5.1280

**Table 3-5 Test on vivo Y55A**

| Page                       | Average<br>Load Time | CPU %   |       | Memory  |         | Results |
|----------------------------|----------------------|---------|-------|---------|---------|---------|
|                            |                      | Average | Peak  | Average | Peak    |         |
| pages/my/my                | 3.85s                | 14.45   | 39.04 | 104.60M | 149.00M | OK      |
| pages/index/index          | 0.28s                | 18.44   | 39.23 | 126.46M | 138.00M | OK      |
| pages/my/tel/tel           | 0.37s                | 1.48    | 6.43  | 109.09M | 132.00M | OK      |
| pages/my/release/release   | 0.20s                | 6.70    | 20.18 | 85.29M  | 139.00M | OK      |
| pages/new/new              | 0.25s                | 4.73    | 13.42 | 134.57M | 141.00M | OK      |
| pages/my/feedback/feedback | 0.118s               | 2.29    | 4.39  | 140.13M | 146.00M | OK      |

### 3.2.2.3 vivo Y51

Resolution: 540 x 960 pixels

System: Android 5.1

WeChat Version: 6.6.5.1280

**Table 3-6 Test on vivo Y51**

| Page                           | Average Load<br>Time | CPU %   |       | Memory  |         | Results |
|--------------------------------|----------------------|---------|-------|---------|---------|---------|
|                                |                      | Average | Peak  | Average | Peak    |         |
| pages/my/my                    | 8.68s                | 7.81    | 17.31 | 101.61M | 137.00M | OK      |
| pages/index/index              | 1.10s                | 13.90   | 48.48 | 141.69M | 166.00M | OK      |
| pages/my/tel/tel               | 0.27s                | 13.28   | 24.86 | 121.77M | 158.00M | OK      |
| pages/my/release/<br>release   | 0.73s                | 11.04   | 28.19 | 75.99M  | 138.00M | OK      |
| pages/new/new                  | 0.61s                | 12.76   | 38.26 | 128.72M | 146.00M | OK      |
| pages/my/feedbac<br>k/feedback | 0.57s                | 9.30    | 31.80 | 111.69M | 145.00M | OK      |

### 3.2.2.4 vivo X7

Resolution: 1080 x 1920 pixels

System: Android 5.1

WeChat Version: 6.6.5.1280

Table 3-7 Test on vivo X7

| Page                           | Average Load Time | CPU %   |       | Memory  |          | Results |
|--------------------------------|-------------------|---------|-------|---------|----------|---------|
|                                |                   | Average | Peak  | Average | Peak     |         |
| pages/my/my                    | 1.78s             | 3.25    | 14.78 | 96.62m  | 133.00M  | OK      |
| pages/index/index              | 0.32s             | 5.55    | 12.63 | 122.15M | 135.00M  | OK      |
| pages/my/tel/tel               | 0.25s             | 3.45    | 12.74 | 96.30M  | 131.000M | OK      |
| pages/my/release/<br>release   | 0.23s             | 4.19    | 7.95  | 106.15M | 125.00M  | OK      |
| pages/new/new                  | 0.35s             | 4.80    | 13.85 | 119.31M | 131.00M  | OK      |
| pages/my/feedbac<br>k/feedback | 0.27s             | 3.61    | 10.87 | 87.82M  | 117.00M  | OK      |

### 3.2.3 Unit Test

Because no unit test tool can be used to unit test WeChat Mini Program, I just run the application in WeChat Web Developer Tool to view application data used in the current page in AppData window.

```
pages/my/my {9}
  ▼ userInfo {7}
    nickName : 啦啦
    gender : 1
    language : zh_CN
    city : Wenzhou
    province : Zhejiang
    country : China
    avatarUrl : https://wx.glogo.cn/mmope
n/v1_32/0014TwGfTLpnUF16
DaydofTX0791X001Yc0lx5IX2
ng0B15ulexj07xib5BYoI98dK2
27c9BCLnKcokozwdtw/0
    logged : false
    takeSession : false
    requestResult : value
    name : John
    tel : 86-12345
    ismanager : 1
    __webViewId__ : 136
    objectid : CZ00444N
```

App Data in My Page

```
▼ pages/new/new {6}
  imgUrl : value
  openid : value
  name : John
  tel : 86-12345
  facilitytype : value
  __webViewId__ : 149
```

App Data in New Page

```
pages/index/submitted/submitted {7}
  ► array [0]
  ► objectArray [0]
    showaccept : false
    showreject : false
    ismanager : 1
    __webViewId__ : 143
  ► result {8}
  ► Tree
  pages/index/submitted/repair/repair {4}
    ► array [7]
    ► objectArray [7]
      index : 0
      __webViewId__ : 144
```

App Data in Submitted Page

Figure 3-2 Data shown in AppData Windows from pages

**Table 3-8 Unit Test of Entering App**

| Test ID                       | Comments   | Results |
|-------------------------------|--|---------|
| Test description              | Users enter App and My page shows.   | OK      |
| Verified Requirement          | User identification and data retrieval.  | OK      |
| Initial conditions            | None   | OK      |
| Tests inputs                  | None   | OK      |
| Data collection actions       | Gets openid from storage/from WeChat (New user).   | OK      |
| Tests outputs                 | WeChat user info, log state, App user info   | OK      |
| Expected results and criteria | Shows WeChat user avatar, nickname, user name, tel number; stores objectId into local storage.   | OK      |
| <b>Test procedure</b>         |  |         |
| Step number                   | Descriptions   |         |
| 1                             | Initializes app.js, login bmob database, checks whether openid exists in storage, if so, goes to next step; if not, logins WeChat to get objected. | OK      |

|   |   |    |
|---|---|----|
| 2 | Loads and shows My page, retrieves user avatar, nick name, name, and tel number; stores object id; shows page elements. | OK |
|---|---|----|

**Table 3-9 Unit Test of Requesting a New Repair**

| Test ID                       | Comments  | Results |
|-------------------------------|---|---------|
| Test description              | Enters New page and inputs all data including photo and description, then submits the request.  | OK      |
| Verified Requirement          | User information shall be shown and a new request shall be stored in request database with correct information.                           | OK      |
| Initial conditions            | User login  | OK      |
| Tests inputs                  | Facility type, facility location, a photo and a text description.   | OK      |
| Data collection actions       | Gets openid from storage/from WeChat (New user)   | OK      |
| Tests outputs                 | WeChat user info, log state, App user info  | OK      |
| Expected results and criteria | Request information including: name, tel, facility type, facility location, a photo, text description, state (as int(1)) and create time. | OK      |

|                       |   |    |
|-----------------------|---|----|
| <b>Test procedure</b> |   |    |
| Step number           | Descriptions  |    |
| 1                     | Retrieves openid from local storage and searches user data from user table; then shows username and tel.                | OK |
| 2                     | Clicks photo button, chooses a photo from album or taking from camera to upload; then gets temporary picture file path. | OK |
| 4                     | Receives data, creates a new request in request table, and sets column value.   | OK |
| 5                     | Redirects to the index page.  | OK |

**Table 3-10 Unit Test of Accepting a Request**

|                      |  |                |
|----------------------|--|----------------|
| <b>Test ID</b>       | <b>Comments</b>  | <b>Results</b> |
| Test description     | Views the specific information of a request, accepts it and then allocates a repair person.  | OK             |
| Verified Requirement | All request information are shown in appropriate places; after accepting, the request status shall be changed to 2 and a repair person and the manager is set in that request row. | OK             |

|                               |   |    |
|-------------------------------|---|----|
| Initial conditions            | User login and ismanager equals.  | OK |
| Tests inputs                  | Accept button event, and a repair person objectId.  | OK |
| Data collection actions       | Manager and repair person object, and accept time.  | OK |
| Tests outputs                 | Back to Index Page. A reminder message is sent to the requester in WeChat.                        | OK |
| Expected results and criteria | The request is updated and the user receives the reminder.  | OK |
| <b>Test procedure</b>         |   |    |
| Step number                   | Descriptions  |    |
| 1                             | Receives objectId of the request from Index page, and searches request information and show them. | OK |
| 2                             | Receives ismanager and checks if it is 1, after verifying, shows accept/reject button.            | OK |
| 4                             | Clicks Accept button and navigates to Repair Page.  | OK |
| 5                             | Select a repair person from the picker and click Confirm Button.                                  | OK |



# Chapter 4 Discussion

## 4.1 Restrictions

- Due to the unfixed employment of repair persons, the application does not provide functions to them.
- Because WeChat does not grant extensive function of QR Code (Scan a QR Code to the specific page in App) to individual developers, the QR code Scanning function is only accessible to the homepage of the application.
- No restriction in App is set to managers (e.g., how long time shall the manager fulfill a request), so WKU Easy Request may not help speed up the repair process.

## 4.2 Limitations

- In the primary development stage, the application will be applicable in Wenzhou Kean University.
- Because a large number of facilities, some kinds of facilities are not available in the current development plan, such as desks and chairs.
- When a manager is employed or retired, the administrator must go into the database management system and manually set the status of the manager.
- Anyone using the software must have their phone connected to the Internet.
- Every time users want to enter the application, they must log in WeChat.

## 4.3 Constraints

The operation of software shall confirm to Student Code of Conduct in Wenzhou Kean University

- <http://www.wku.edu.cn/org/student-affairs/code-of-conduct/>

The software shall conform to ISO/IEC/IEEE 29148 International Standard for system requirement specification and software requirement specification.

- [www.ieee.org](http://www.ieee.org)

The development of WKU Easy Request shall confirm to WeChat Open Platform Developer Service Agreement

- [https://open.weixin.qq.com/cgi-bin/frame?t=news/protocol\\_developer\\_tmpl](https://open.weixin.qq.com/cgi-bin/frame?t=news/protocol_developer_tmpl)

## Chapter 5 Future Work

### 5.1 Dependencies

The development and maintenance of WKU Easy Request require the assistance of IT office and dorm office. The future improvement and maintenance will be incorporated into IT office of Wenzhou Kean University.

### 5.2 Assumptions

If the school supports the program and grant us the official authority to develop and maintain WKU Easy Request, the expanded QR code scanning function will be open to the application. Then, the facility recognition system will be available to users, making the requesting process simpler. However, Putting up QR Code stickers is time-consuming. Once the infrastructures need a massive shift from one place to another, it is hard to adjust the facility data.

There is an assumption that WKU Easy Request will reflect the facility use in Wenzhou Kean University. The facility management will be based on WKU Easy Request. The school will employ fixed repair persons and they will be one group of members in WKU Easy Request. In addition, WKU Easy Request will extend to a universal application in schools and other possible industries.

# References

Cheng, Gracye, English, Steve & Filardo, Mary (2011). Facilities: fairness and effects.

U.S Department of Education Excellence & Equity Commission. Retrieved from:

<http://www.21csf.org/csf->

[home/publications/ImpactSchoolFacilitiesCivilRightsAug2011.pdf](http://www.21csf.org/csf-home/publications/ImpactSchoolFacilitiesCivilRightsAug2011.pdf)

Framework (2017). Retrieved on January 31, 2018, from WeChat Public Platform:

<https://mp.weixin.qq.com/debug/wxadoc/dev/framework/MINA.html>

Price, I. F., Matzdorf, F., Smith, L., & Agahi, H. (2003). "The Impact of Facilities on

Student Choice of University", *Facilities*, Vol. 21 Issue: 10, pp.212-222,

Retrieved from <https://doi.org/10.1108/02632770310493580>

Roberts, L. W. (2009). Measuring School Facility Conditions: An Illustration of The

Importance of Purpose. *Journal of Educational Administration*, 47(3), 368-380.

Retrieved from: <http://dx.doi.org/10.1108/09578230910955791>

Micro Repair Platform (2017). "Micro Repair Platform User Manual", Version 0.9.4.

Retrieved from: <http://cdn.51jianxiu.com/app/微报修平台用户操作手册>

[v0.9.4.pdf](http://cdn.51jianxiu.com/app/微报修平台用户操作手册)

Wang, Siqi (2017). WeChat Mini Program is launched officially - Can It Make WeChat a

"Super Application"? Retrieved from: <http://www.yicai.com/news/5200804.html>

# Appendix A Sample of Important Code

## Code 1 App Launch Code

```
//Get Bmob API app.js
var Bmob = require('utils/bmob.js')
var BmobSocketIo = require('utils/bmobSocketIo.js').BmobSocketIo;
Bmob.initialize("ac78b2a4ba1f5e0eb18e971037cbc555",
"e2cb6f72850eb7954f2ad0defb250126"); //API code
BmobSocketIo.initialize("ac78b2a4ba1f5e0eb18e971037cbc555");

var objectid;
App({
  //Launch APP
  onLaunch: function () {
    //show loading window
    wx.showLoading({
      title: '加载中 Loading',
    })
    console.log("welcome to wku er")
    //initialize a bmob user
    var user = new Bmob.User();

    //login wechat
    wx.login({
      success: function (res) {
        //login bmob with wechat
        user.loginWithWeapp(res.code).then(function (user) {
          objectid=user.id;

          var openid = user.get("authData").weapp.openid;
          //set storage of openid&objectid
          wx.setStorage({
            key: "openid",
            data: openid
          })
          wx.setStorage({
            key: "objectid",
            data: objectid
          })

          if (user.get("nickName")) {
            // login after 1st time
            console.log(user.get("nickName"), 'res.get("nickName")');
            wx.setStorageSync('openid', openid)
```

```

wx.hideLoading()
} else {
    //store other info of the user
    wx.getUserInfo({
        success: function (result) {
            var userInfo = result.userInfo;
            var nickName = userInfo.nickName;
            var avatarUrl = userInfo.avatarUrl;

            var u = Bmob.Object.extend("_User");
            var query = new Bmob.Query(u);
            // when store successfully, it will get userid
            query.get(user.id, {
                success: function (result) {
                    console.log("register successfully")
                    // bound the account
                    result.set('manager', 0);
                    result.set('nickName', nickName);
                    result.set("userPic", avatarUrl);
                    result.set("openid", openid);
                    result.save();
                },
                //store app user info into another table
                success: function (result) {
                    console.log("begin register")
                    var user = Bmob.Object.extend("user");
                    var user = new user();
                    user.set("openid", openid);
                    user.set("manager", 0);
                    user.set("name", "init name")
                    user.set("tel", "init tel")
                    wx.hideLoading()
                    //add data
                    user.save(null, {
                        success: function (result) {
                            // set storage objectid of the user
                            wx.setStorage({
                                key: "objectid",
                                data: result.id
                            })
                            console.log("save, objectId:" + result.id);
                        },
                        error: function (result, error) {
                            // failure
                            console.log('创建日记失败 failure');
                        }
                    });
                }
            });
        }
    });
}

```

```

        }
    });

    },
    function (err) {
        console.log(err, 'errr');
    });
}
});

wx.hideLoading() //hide loading
},
getUserInfo: function (cb) { //get user info function
    var that = this
    if (this.globalData.userInfo) {
        typeof cb == "function" && cb(this.globalData.userInfo)
    } else {
        //call login interface
        wx.login({
            success: function () {

                wx.getUserInfo({
                    success: function (res) {
                        that.globalData.userInfo = res.userInfo
                        typeof cb == "function" && cb(that.globalData.userInfo)
                    }
                })
            }
        })
    }
}
},
globalData: {
    userInfo: null,
    objectid: objectid
}
})

```

## Code 2 Index Code

```

//index.js
//get bmob api
var Bmob = require('../utils/bmob.js');
var common = require('../utils/common.js');
var app = getApp();
var that;
var ismanager, objectid;

```

```

Page({

  data: {
    //initialize data of page
    loading: false,
    windowHeight: 0,
    windowWidth: 0,
    limit: 10,
    submit: [],
    process: [],
    finish: [],
    navbar: ['新报修 Submitted', '待维修 Processing', '已检修 History'],
    currentTab: 0
  },
  onPullDownRefresh: function () {
    wx.showNavigationBarLoading() //refresh when pulling down the top of
    page

    setTimeout(function () {
      // complete
      wx.hideNavigationBarLoading() //stop loading when finished
      wx.stopPullDownRefresh() //stop pull downrefresh
    }, 1500);

  }
,
  navbarTap: function (e) {
    this.setData({
      //set different view (submitted,processing&history)
      currentTab: e.currentTarget.dataset.idx
    })
  }
,
  //load page
  onLoad: function () {
    that = this;
    wx.showLoading({
      title: '加载中 Loading',
    })
    wx.showShareMenu({
      withShareTicket: true
    })
    wx.getStorage({ //get storage of objectid
      key: 'objectid',
      success: function (res) {
        objectid = res.data
        console.log(res.data)
      }
    })
  }
})

```



```

    })
    //search user info in user table using objectid
    var user = Bmob.Object.extend("user");
    var query = new Bmob.Query(user);
    query.get(objectid, {
      success: function (result) {
        // The object was retrieved successfully.
        ismanager=result.get("manager");
        console.log(objectid + "ismanager" + ismanager)
        wx.setStorageSync("ismanager", ismanager)
        wx.hideLoading()
      },
      error: function (result, error) {
        console.log("查询失败");
      }
    });
  },
  //click button to turn to add page
  add: function () {
    var objectid = wx.getStorageSync("objectid");
    if (wx.getStorageSync("name") == "init name"){
      wx.redirectTo({
        url: '/pages/my/name/name',
      })
    }
    else if (wx.getStorageSync("tel") == "init tel"){
      wx.redirectTo({
        url: '/pages/my/tel/tel',
      })
    } else{
      wx.navigateTo({
        url: '../new/new'
      })
    }
  },
  //show the different views, set window height
  onShow: function () {
    getsubmit(this);
    getprocess(this);
    getfinish(this);
    wx.getSystemInfo({
      success: (res) => {
        that.setData({
          windowHeight: res.windowHeight,
          windowWidth: res.windowWidth
        })
      }
    })
  }
}

```

```

        }
    })
},
pullUpload: function (e) {
    var limit = that.data.limit + 2
    this.setData({
        limit: limit
    })
    this.onShow()
},
showInput: function () {
    this.setData({
        inputShowed: true
    });
},
hideInput: function () {
    this.setData({
        inputVal: "",
        inputShowed: false
    });
    getsubmit(this);
    getprocess(this);
    getfinish(this);
},
clearInput: function () {
    this.setData({
        inputVal: ""
    });
    getsubmit(this);
    getprocess(this);
    getfinish(this);
},
inputTyping: function (e) {
    //search data
    getsubmit(this,e.detail.value);
    getprocess(this,e.detail.value);
    getfinish(this,e.detail.value);
    console.log(e.detail.value)
    this.setData({
        inputVal: e.detail.value
    });
}
})

//submitted view function
function getsubmit(t, k) {

```

```

that = t;
var request = Bmob.Object.extend("request");
var query = new Bmob.Query(request);
var query1 = new Bmob.Query(request);

//search box
if (k) {
    query.equalTo("facilitytype", { "$regex": "" + k + ".*" } );
    query1.equalTo("facilitylocation", { "$regex": "" + k + ".*" });
}
query.equalTo("state", 0);
query1.equalTo("state", 0);
//match data
query.ascending('createdAt');
query.include("user")
query.include("manager")
query.include("repair")
// all data
query.limit(that.data.limit);

var mainQuery = Bmob.Query.or(query, query1);
mainQuery.find({
    success: function (results) {
        // find data from table
        console.log(results);
        that.setData({
            submit: results
        })
    },
    error: function (error) {
        console.log("查询失败 failure: " + error.code + " " +
error.message);
    }
});
}

```