Smart Seat

Details Design V1.0

**Group members：**Hanyu Zhang, Tinghui Zhang, Huiying Han, Yikang Tao

Directory

[Smart Seat 1](#_Toc3925909)

[1 Introduction 4](#_Toc3925910)

[1.1 Purpose of writing 4](#_Toc3925911)

[1.2 Background 4](#_Toc3925912)

[1.3 Definition 4](#_Toc3925913)

[1.4 Reference 5](#_Toc3925914)

[1.5 Standards、conditions and conventions 5](#_Toc3925915)

[2. Overall design 6](#_Toc3925916)

[2.1 Project Objectives 6](#_Toc3925917)

[2.2 Overview of requirements 6](#_Toc3925918)

[2.3 Software structure 7](#_Toc3925919)

[2.4 Project development environment 7](#_Toc3925920)

[2.5 Detailed design methods and tools 8](#_Toc3925921)

[2.5.1 E-R diagram 8](#_Toc3925922)

[2.5.2 Program flow chart 9](#_Toc3925923)

[3 System detailed requirements analysis 10](#_Toc3925924)

[3.1 Detailed functional requirements analysis 10](#_Toc3925925)

[3.2 Detailed function description 10](#_Toc3925926)

[3.2.1 Registration & Login module 10](#_Toc3925927)

[3.2.2 Seat reservation module 10](#_Toc3925928)

[3.2.3 Seat repair module 10](#_Toc3925929)

[3.2.4 Seat information inquiry module 11](#_Toc3925930)

[3.2.5 Personal information inquiry module 11](#_Toc3925931)

[3.2.6 Administrator mailbox module 11](#_Toc3925932)

[3.2.7 PIR sensor remote monitoring seat module 11](#_Toc3925933)

[3.2.8 Socket communication remote access indicator module 11](#_Toc3925934)

[3.3 Interface design 11](#_Toc3925935)

[3.3.1 Internal interface 11](#_Toc3925936)

[3.3.2 External interface 12](#_Toc3925937)

[3.3.3 User interface 12](#_Toc3925938)

[4 Database Design 13](#_Toc3925939)

[4.1 User 13](#_Toc3925940)

[4.2 Admin 13](#_Toc3925941)

[4.3 Seat 13](#_Toc3925942)

[4.4 Reserve 13](#_Toc3925943)

[4.5 Repair 14](#_Toc3925944)

[5 System error handling design 15](#_Toc3925945)

[5.1 System error remedy 15](#_Toc3925946)

[5.2 System maintenance design 15](#_Toc3925947)

[5 System test 16](#_Toc3925948)

[5.1 System test review 16](#_Toc3925949)

[5.2 Login & Register module test 17](#_Toc3925950)

[5.3 Reserve module test 19](#_Toc3925951)

[5.4 Repair module test 19](#_Toc3925952)

[5.5 PIR sensor module test 20](#_Toc3925953)

[5.6 Indicate light module test 21](#_Toc3925954)

# 1 Introduction

## 1.1 Purpose of writing

The purpose is to:

* Provide evidence for the coders；
* Provide conditions for modification and maintenance；
* The project leader will arrange and control the whole process of development work as required by the plan；
* The project quality assurance personnel will perform phased and summarizing quality verification and confirmation according to this plan.

The tone readers of this manual include：

* project developers, especially coding staff；
* Software maintenance personnel；
* technical management personnel；
* Participate in the verification, confirmation and relevant personnel responsible for the final report acceptance and appraisal of the project.

## 1.2 Background

At present, there are many seats in major university libraries, but there are still a large number of seats in short supply during the peak period of use. Occasionally, individual students may experience unpleasantness due to occupying behavior. Many students will also find a place to study and read a lot of time after a long search in the library, which greatly reduces the enthusiasm of the students.

At present, the library equipment is perfect, but the repair is not timely. It often happens that although there is a location, it is not properly used due to the damage of the hardware facilities.

The project's entrusting unit, development unit and competent department:

Entrusted by: Software Project Management Course, School of Information Engineering, Shanghai Maritime University

Development unit: the eighth group

Competent department: School of Information Engineering, Shanghai Maritime University

## 1.3 Terminology and abbreviation

Terminology:

**PIR:** PIR is the abbreviation of Passive Infrared, which is passive infrared technology. The full name of the PIR detector is the Passive Infrared Detector, which is a passive infrared detector or body sensor. Sometimes called Passive Infrared Sensor, the detector in the security industry is called Detector.

**NodeMcu:** An open source IoT platform. It is programmed using the Lua scripting language. The platform is based on the eLua open source project, which uses the ESP8266 sdk 0.9.5 version. The platform uses many open source projects, such as lua-cjson, spiffs. The NodeMCU contains firmware that runs on the esp8266 Wi-Fi SoC chip, as well as hardware based on ESP-12 modules.

**MySQL:** MySQL is a relational database management system developed by MySQL AB of Sweden and currently owned by Oracle. MySQL is one of the most popular relational database management systems. MySQL is the best RDBMS (Relational Database Management System) application for WEB applications.

**SQL:** A special purpose programming language, a database query and programming language for accessing data and querying, updating, and managing relational database systems; it is also the extension of database script files.

**Transaction flow:** Entering the system along the incoming path, transforming from the external form to the internal form, and then arriving at the transaction center. The transaction center selects one of several action paths according to the data item evaluation result. Have a transaction like this.

Abbreviation:

**SQL:** Structured Query Language.

**UML:** Unified Modeling Language, a set of standard modeling languages for designing software blueprints, is a standardized modeling language from software analysis and design to programming specification.

## 1.4 Reference

《Software project management》Rajeev T Shandilya Science press

《UML and pattern applications》Craig Larman Machinery industry press

National standards document for software engineering

## 1.5 Standards、conditions and conventions

The project follows the following standards:

**GB/T** computer software classification and code

**GB/T** information technology

**GB/T** software engineering

**GB/T** software engineering standard taxonomy

**GB/T** computer test documentation

**GB/T** computer software requirements specification

**GB/T** computer software testing specifications

**GB/T** information technology programming language

**GB/T** information technology software engineering

**GB/T** computer software documentation specifications

# 2. Overall design

## 2.1 Project Objectives

* Designing an intelligent occupancy system sample based on the Internet of Things.
* Develop a smart occupancy background management system that can be put into use.

## 2.2 Overview of requirements

（1）System administrator requirements：

1. Managing seat conditions
2. Check the repair information
3. Manage user information

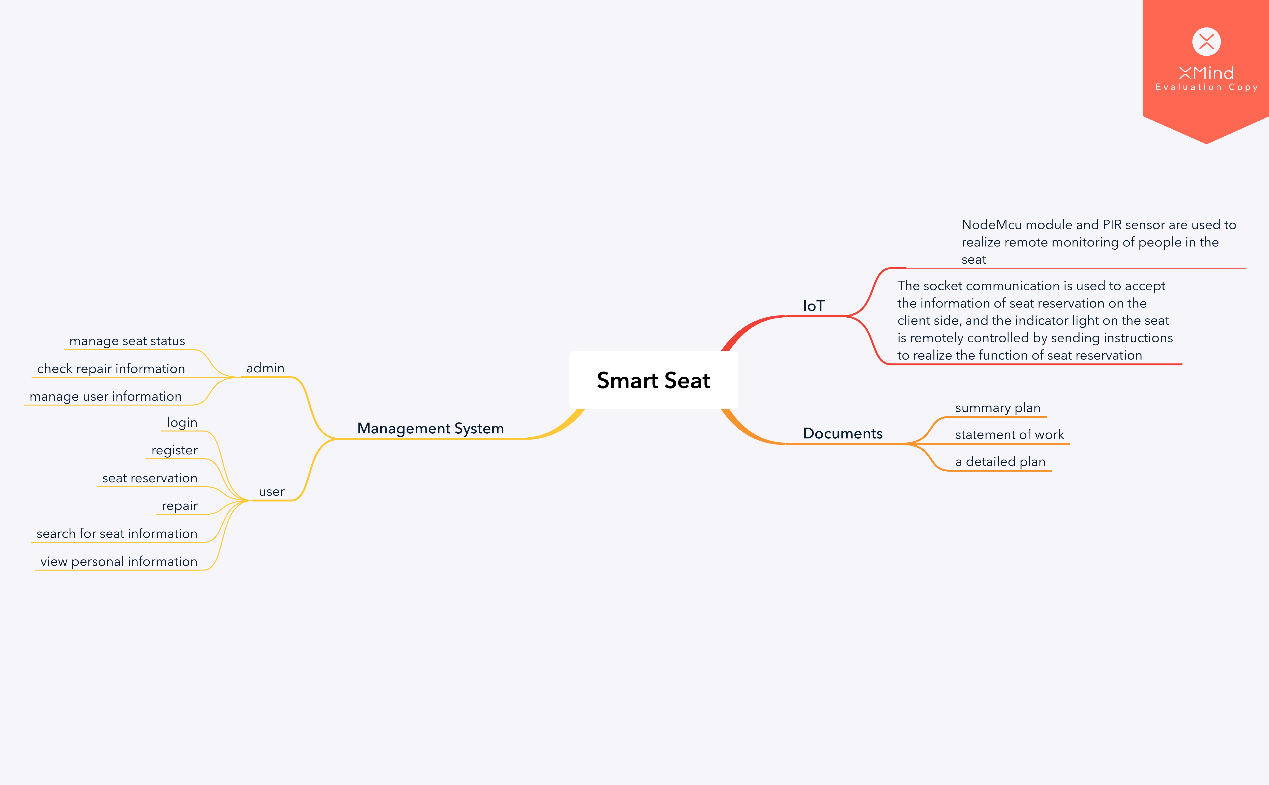
（2）User requirements：

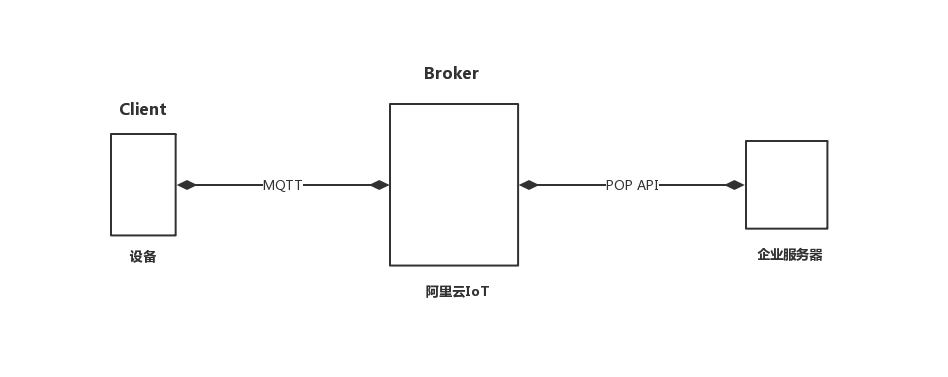
1. Login
2. Register
3. Seat reservation
4. Repair
5. Search for seat information
6. View personal information

（3）Intelligent device networking：

1. PIR sensor detection information
2. NodeMcu platform uploads received data via WiFi
3. The web receives feedback and clicks on the light

## 2.3 Software structure

Figure2-1 Software detailed structure diagram 1



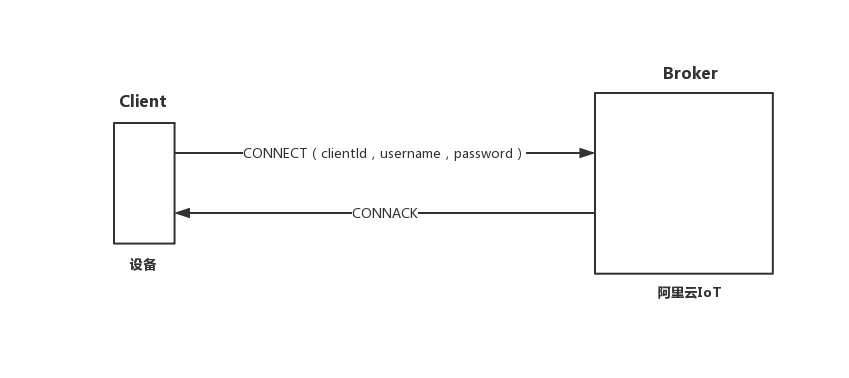
Figure2-2 Software detailed structure diagram 2

Figure2-3 Software detailed structure diagram 3

## 2.4 Project development environment

Support conditions needed during development:

Hardware:

* Server: Pentium III above 500 or higher
* RAM：over 128M
* ROM：at least 10G above

Software:

* The operating system is Windows 8 or above, and the integrated development tools Arduino IDE and PyCharm are used. MySQL is used as the database, and the project running environment is apache2.4.

## 2.5 Detailed design methods and tools

### 2.5.1 E-R diagram

The E-R diagram, also known as the Entity Relationship Diagram, provides methods for representing entity types, attributes, and connections to describe real-world conceptual models.

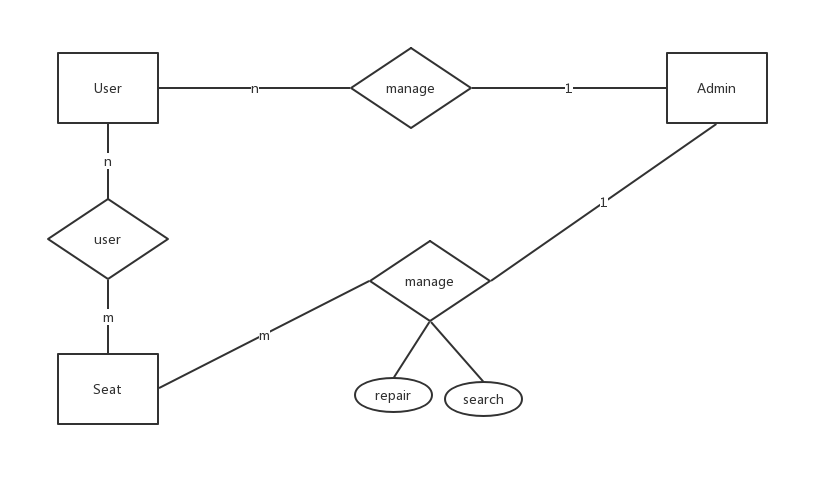
It is an effective way to describe the conceptual model of real world relations. Is a way of representing a conceptual relationship model. Use "rectangular box" to represent the entity type, the rectangle box to indicate the entity name; use "ellipse frame" to represent the attribute of the entity, and use "solid line segment" to connect it with the "physical type" of the corresponding relationship;

Figure2-4 E-R diagram

### 2.5.2 Program flow chart

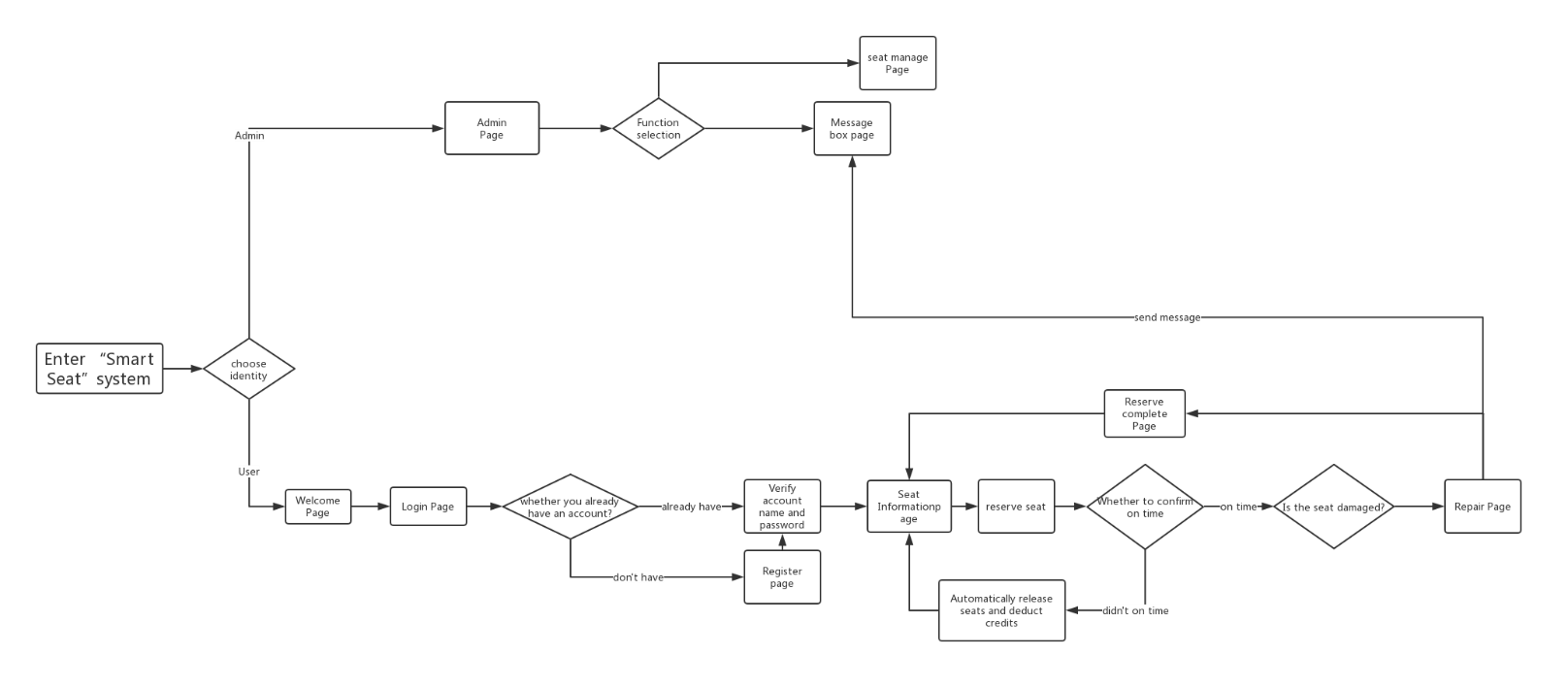
The program flow diagram, also called the block diagram, is a graphical representation of the specific steps of the program running with the standard symbols specified. The block diagram is designed based on the processing flow chart, through the detailed analysis of the input and output data and processing, the main operating steps and content of the computer are identified. The block diagram is the most basic basis for programming, so its quality is directly related to the quality of programming.

Figure2-5 Program flow chart

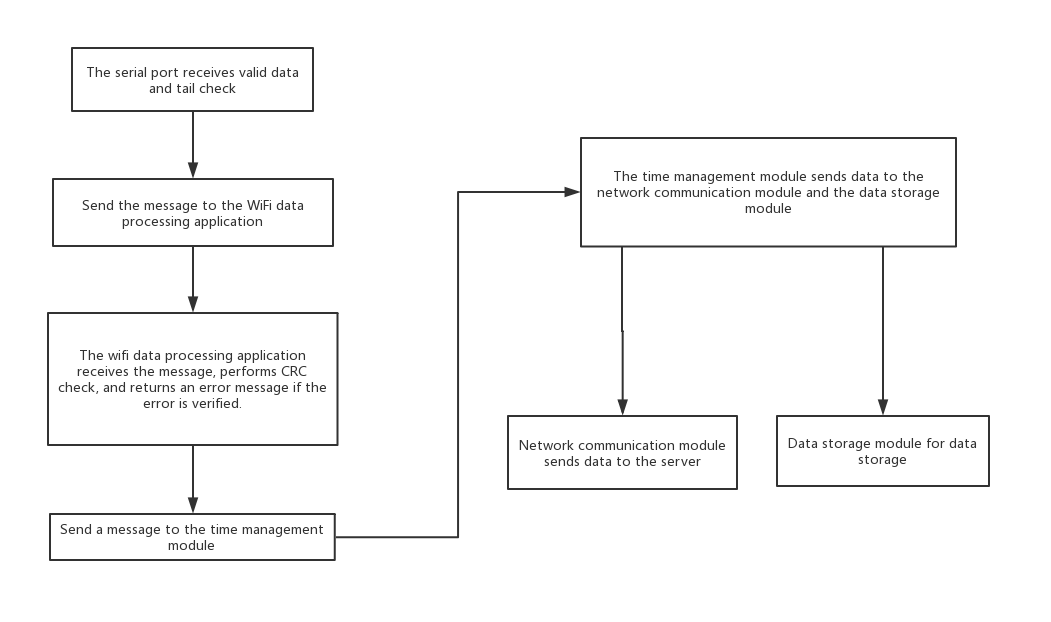


Figure2-6 IoT module flow chart

# 3 System detailed requirements analysis

## 3.1 Detailed functional requirements analysis

This system has the following function modules:

1. Register the login module
2. Seat reservation module
3. Seat repair module
4. Seat information inquiry module
5. Personal Information Query Module
6. Administrator Mailbox Module
7. PIR sensor remote monitoring seat module
8. Socket communication remote access indicator module

## 3.2 Functional block diagram

Figure3-1 Functional block diagram

## 3.3 Overall system requirements

|  |  |  |
| --- | --- | --- |
| The main function | Functional requirements | Quality requirements |
| Login | Enter the account password when logging in to enter the corresponding account. After entering, the system will verify the user's specific identity according to the entered user name and password. If the user name and password are not entered, the user name and password cannot be empty. | Correctness: high  Robustness: high  Performance efficiency: good  Ease of use: good  Security: General  Scalability: good  Compatibility: good |
| Register | When registering, you can select ordinary user registration. When a normal user registers, you need to submit a series of personal information to register successfully. | Correctness: high  Robustness: high  Performance efficiency: good  Ease of use: good  Security: General  Scalability: good  Compatibility: good |
| Seat Reservation | After the user logs in, the seat reservation function can be used normally. After the user successfully reserves the seat, the user must manually confirm the account within the specified time. If the time is not confirmed within the specified time, the seat is automatically released. If the sensor detects that someone is seated within the specified time, If the user does not confirm, a warning will be sent to the administrator mailbox. | Correctness: high  Robustness: high  Performance efficiency: good  Ease of use: good  Security: General  Scalability: good  Compatibility: good |
| Seat Repair | After the user logs in, he can select the seat, click the repair button and fill in the repair information, the system will send the information to the administrator mailbox. | Correctness: high  Robustness: high  Performance efficiency: good  Ease of use: good  Security: General  Scalability: good  Compatibility: good |
| Seat Information Inquiry | After the user logs in, the user can check what status the seat is currently in. | Correctness: high  Robustness: high  Performance efficiency: good  Ease of use: good  Security: General  Scalability: good  Compatibility: good |
| Personal Information Query | After the user logs in, the user can query their related information. | Correctness: high  Robustness: high  Performance efficiency: good  Ease of use: good  Security: General  Scalability: good  Compatibility: good |
| Administrator Mailbox | After the administrator logs in, the administrator can view and receive relevant repair and warning information. | Correctness: high  Robustness: high  Performance efficiency: good  Ease of use: good  Security: General  Scalability: good  Compatibility: good |
| PIR sensor remote monitoring seat module | The PIR sensor can detect the current seat status at all times and send the status information of the current seat to the background. | Correctness: high  Robustness: high  Performance efficiency: good  Ease of use: good  Security: General  Scalability: good  Compatibility: good |
| Socket communication remote access indicator module | If the current seat is reserved, the light is on | Correctness: high  Robustness: high  Performance efficiency: good  Ease of use: good  Security: General  Scalability: good  Compatibility: good |

Table3-1 Quality requirements

## 3.4 Use case diagram analysis

Figure3-1 Use case diagram

|  |  |  |
| --- | --- | --- |
| Number | UC01 | |
| Name | Register | |
| Description | Register as a system user | |
| Primary  participant | User | |
| Precondition | Nothing | |
| Basic event flow | Step | Activities |
| A1 | Click the registration button to register |
| A2 | Fill in a series of personal information |
| A3 | Submit Information |
| Extended event flow | 1a | Basic information input is incorrect |
| 1b | Registration success |

Table3-2 Register use case

|  |  |  |
| --- | --- | --- |
| Number | UC02 | |
| Name | Login | |
| Description | Select user identity to log in to the system | |
| Primary  participant | Administrator、User | |
| Precondition | Already registered as a system user | |
| Basic event flow | Step | Activities |
| A1 | Choose login identity |
| A2 | Enter username and password |
| A3 | Click the confirmation button |
| Extended event flow | 1a | The account or password is empty and the error is displayed. |
| 1b | Account or password does not match, prompt error |

Table3-3 Login use case

|  |  |  |
| --- | --- | --- |
| Number | UC03 | |
| Name | Seat Reservation | |
| Description | Reserve a library seat | |
| Primary  participant | User | |
| Precondition | Already logged in | |
| Basic event flow | Step | Activities |
| A1 | Choose an idle seat to make a reservation |
| A2 | Confirm arrival time within the specified time |
| Extended event flow | 1a | The system automatically releases the seat if it is not confirmed within the specified time. |
| 1b | If the sensor detects that someone is seated within the specified time, If the user does not confirm, a warning will be sent to the administrator mailbox. |

Table3-4 Seat Reservation use case

|  |  |  |
| --- | --- | --- |
| Number | UC04 | |
| Name | Seat Repair | |
| Description | Report library seat damage | |
| Primary  participant | User | |
| Precondition | Already logged in | |
| Basic event flow | Step | Activities |
| A1 | Choose an idle seat to report |
| A2 | Fill in the damage information |
| A3 | Confirm report information |

Table3-5 Seat Repair use case

|  |  |  |
| --- | --- | --- |
| Number | UC05 | |
| Name | Seat Information Inquiry | |
| Description | View current seat status | |
| Primary  participant | User | |
| Precondition | Already logged in | |
| Basic event flow | Step | Activities |
| A1 | Choose a seat |
| A2 | View current seat status |

Table3-6 Seat Information Inquiry use case

|  |  |  |
| --- | --- | --- |
| Number | UC06 | |
| Name | Personal Information Query | |
| Description | View personal information | |
| Primary  participant | User | |
| Precondition | Already logged in | |
| Basic event flow | Step | Activities |
| A1 | Enter the personal center |
| A2 | View personal information |

Table3-7 Personal Information Query use case

|  |  |  |
| --- | --- | --- |
| Number | UC07 | |
| Name | Administrator Mailbox | |
| Description | Check for repairs and warnings | |
| Primary  participant | Administrator | |
| Precondition | Already logged in as administrator | |
| Basic event flow | Step | Activities |
| A1 | Enter the administrator mailbox |
| A2 | View message |

Table3-8 Administrator Mailbox use case

## 3.5 Detailed function description

### 3.5.1 Registration & Login module

It is divided into normal user login and administrator login. The administrator account is unique and does not need to be registered. Ordinary users can only use all the functions of this system only after they have registered an account and logged in.

### 3.5.2 Seat reservation module

It is only restricted to ordinary users. When the credit value of the ordinary user is higher than the critical value, the seat reservation function can be used normally. The seat can be reserved by inquiry. After the user selects the reservation, the user is prompted to successfully reserve the seat. The seat is reached within the specified time and manually confirmed. Successfully booked once. If the predetermined seat is not confirmed manually within the specified time, the system will automatically release the seat and deduct the user credit value.

### 3.5.3 Seat repair module

Only for ordinary user operations, after the ordinary user successfully makes a reservation, the user can choose to enter the repair page, and after inputting relevant repair information, the repair information will be automatically sent to the administrator mailbox, and the administrator will receive the repair information for related processing.

### 3.5.4 Seat information inquiry module

Only for ordinary user operations, ordinary users can view all seat statuses in the current library and make seat reservations by finding free positions.

### 3.5.5 Personal information inquiry module

Both normal users and administrators can operate. Ordinary users can view their current information and current credit values. Administrators can view and modify user related information.

### 3.5.6 Administrator mailbox module

For administrator operations only, the administrator can open the mailbox to view the current seat repair information or suggestions for the system.

### 3.5.7 PIR sensor remote monitoring seat module

Remotely monitor the seat with the NodeMcu module and the PIR sensor. If the seat is currently unoccupied and the seat is not reserved, it will be released as idle.

### 3.5.8 Socket communication remote access indicator module

The socket communication is used to receive the client seat reservation information, and the indicator light on the seat is remotely controlled by sending an instruction. If the user successfully booked the seat but has not confirmed it, the indicator light on the desk lights up, indicating that the person who has not used the system has been occupied.

## 3.6 Interface design

### 3.6.1 Internal interface

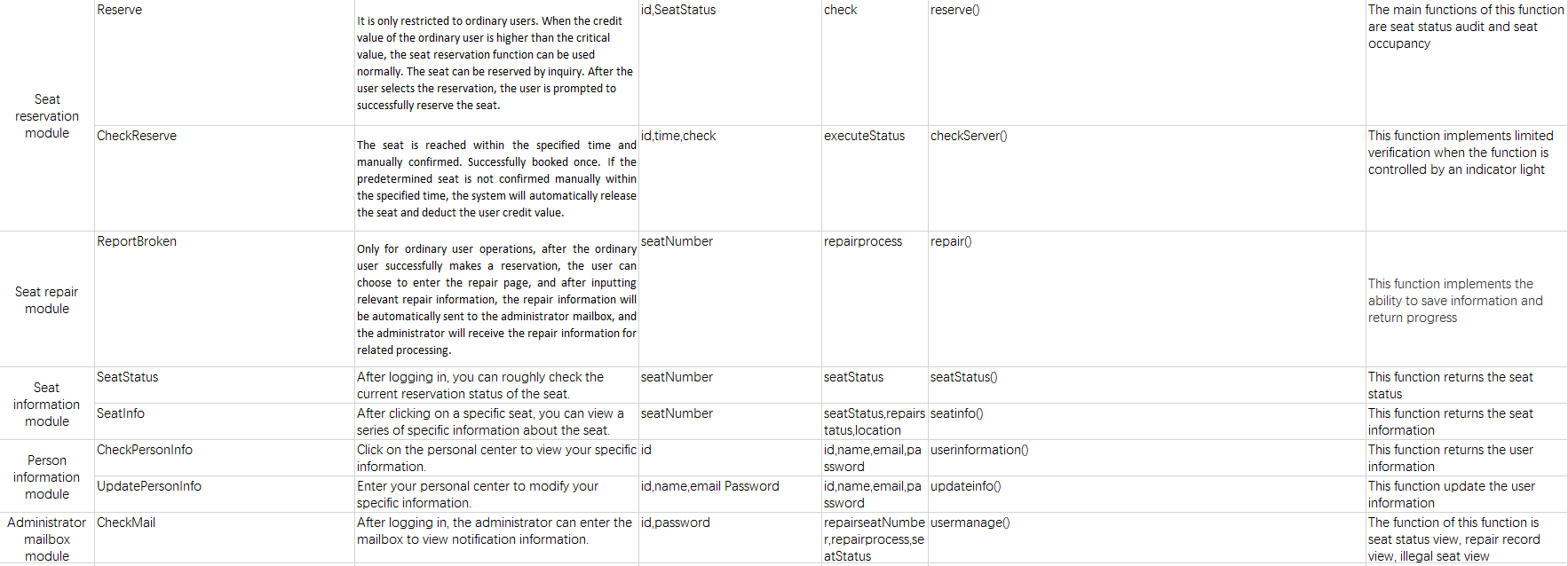
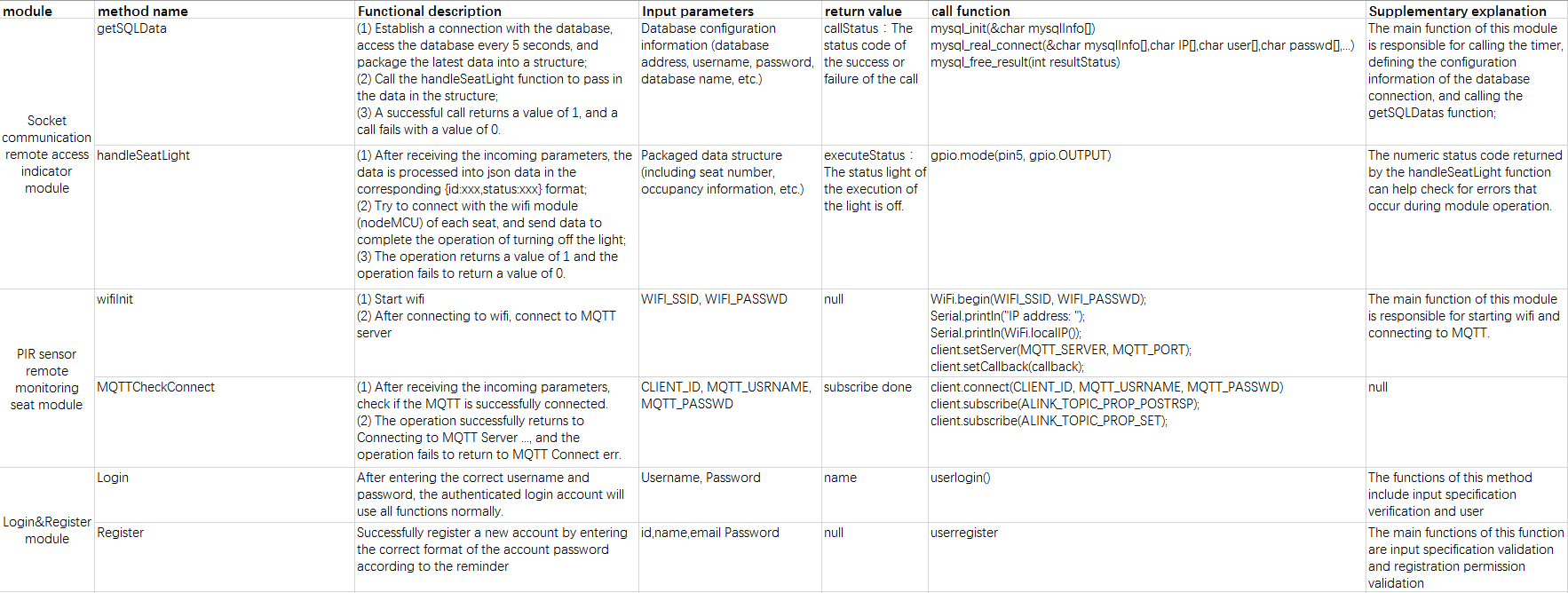


Figure3-2 Internal interface design

### 3.6.2 External interface

Use components and technologies based on the right open standards to ensure maximum collaboration and ease of integration with third-party systems and components. Such standards include but are not limited to the following：

* Network protocols and standards（TCP/IP、HTTP、SSL、etc）
* Language（Python、C、SQL、etc）
* Database connectivity（ADO.net）

### 3.6.3 User interface

Provide users and administrators with an easy-to-use UI and help documentation.

The login page is popped up first, and the user identity is selected for the user to enter a username and password. After the login is successful, you can choose to enter the seat reservation interface. After the reservation is successful, you can select the repair page.

After the administrator logs in successfully, you can choose to enter the seat management page and the administrator mailbox page.

# 4 Database Design

## 4.1 User

|  |  |  |
| --- | --- | --- |
| **User** | | |
| **Name** | **Type** | **Note** |
| UserId | Int | Key |
| Username | Varchar255 |  |
| Password | Varchar255 |  |
| Name | Varchar255 |  |
| Sex | Varchar255 |  |
| Image | Varchar255 |  |
| Birthday | Varchar255 |  |
| Credit value | Int |  |

## 4.2 Admin

|  |  |  |
| --- | --- | --- |
| **Admin** | | |
| **Name** | **Type** | **Note** |
| AdminId | Int | Key |
| Username | Varchar255 |  |
| Password | Varchar255 |  |

## 4.3 Seat

|  |  |  |
| --- | --- | --- |
| **Seat** | | |
| **Name** | **Type** | **Note** |
| SeatId | Int | Key |
| Area | Varchar255 |  |
| Status | Varchar255 | 0-Free，1-Reserving，2-Using |
| Repair | Boolean | 0-intact,1-broken |

## 4.4 Reserve

|  |  |  |
| --- | --- | --- |
| **Reserve** | | |
| **Name** | **Type** | **Note** |
| ReserveId | Int | Key |
| SeatId | Int |  |
| UserId | Int |  |
| Start | Varchar255 |  |
| End | Varchar255 |  |
| Information | Varchar255 |  |

## 4.5 Repair

|  |  |  |
| --- | --- | --- |
| **Repair** | | |
| **Name** | **Type** | **Note** |
| RepairtId | Int | Key |
| SeatId | Int |  |
| UserId | Int |  |
| ReportDate | Varchar255 |  |
| CompleteDate | Varchar255 |  |
| Information | Varchar255 |  |

# 5 System error handling design

## 5.1 System error remedy

Possible workarounds after a failure, including:

a. Backup technology: The backup technology used, when the original system data is lost, the establishment and startup of the enabled copy, such as periodically recording the disk information to the tape is a backup technology for the disk media;

b. Fallback technology: The backup technology used, using another less efficient system or method to obtain some parts of the desired results, such as the automatic system's fallback technology can be manual operation and manual recording of data;

c. Recovery and Restart Technology: The recovery restart technology that will be used to resume software from a point of failure or to re-run the software from the beginning.

## 5.2 System maintenance design

Arrangements made in the internal design of the program for the convenience of system maintenance, including inspection points and dedicated modules specially arranged for inspection and maintenance of the system in the program.

System error handling:

1 Error message:

A. Unable to receive sensor information

B. Login failed, please try again

C. The button is invalid, please try again

2 Remedy information:

A. Hardware restart

B. Software restart

3 system maintenance design:

A. Add or remove existing features based on new user needs

B. System optimization based on user feedback

# 6 System test

## 6.1 System test review

A series of functions of the system have been described in detail above, and this chapter will carry out a crucial test session of the system. In the production process of this project, it is actually the process of repeated testing and running and continuous debugging. We also found many deficiencies in the whole system production process. The following is my specific test plan for each part of the system.

## 6.2 Login & Register module test

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Function test | | | | | | |
| Overview | | | | | | |
| Test number | | | GN001 | | | |
| Functional description | | | User use login and register functions | | | |
| Module manager | | | Yikang Tao | | | |
| tester | | | Huiying Han | | | |
| Use purpose | | | Whether the user can enter the system normally when logging in or registering | | | |
| Precondition | | | User login through the login interface | | | |
| Test operation | | | | | | |
| Number | Input/action | Expected output response | | The actual situation | Correct or not? | Error number |
| 1 | Enter the correct username and password | The system prompts that login/register succeed. | |  |  |  |
| 2 | Enter a username that does not exist, do not fill in the password | The system prompts that login/register failed. | |  |  |  |
| 3 | Enter the correct username and enter the wrong password | The system prompts that login/register failed. | |  |  |  |
| 4 | Enter the wrong username and enter the correct password. | The system prompts that login/register failed. | |  |  |  |
| 5 | Do not fill in the username and password | The system prompts that login/register failed. | |  |  |  |

## 6.3 Reserve module test

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Function test | | | | | | |
| Overview | | | | | | |
| Test number | | | GN002 | | | |
| Functional description | | | User use reserve functions | | | |
| Module manager | | | Yikang Tao | | | |
| tester | | | Huiying Han | | | |
| Use purpose | | | Whether the user can use the reserve function normally | | | |
| Precondition | | | User login through the login interface | | | |
| Test operation | | | | | | |
| Number | Input/action | Expected output response | | The actual situation | Correct or not? | Error number |
| 1 | Click on the free seat to reserve a seat | The system prompts that the reservation is successful. | |  |  |  |
| 2 | Low credit users use the reserve function | The system prompts that the reservation is failed. | |  |  |  |
| 3 | Click on the damaged seat to reserve a seat | The system prompts that the reservation is failed. | |  |  |  |
| 4 | Click on the reserved seat to reserve a seat | The system prompts that the reservation is failed. | |  |  |  |
| 5 | Click on the using seat to reserve a seat | The system prompts that the reservation is failed. | |  |  |  |

## 6.4 Repair module test

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Function test | | | | | | |
| Overview | | | | | | |
| Test number | | | GN003 | | | |
| Functional description | | | User use repair functions | | | |
| Module manager | | | Yikang Tao | | | |
| tester | | | Hanyu Zhang | | | |
| Use purpose | | | Whether the user can use the repair function normally | | | |
| Precondition | | | User login through the login interface | | | |
| Test operation | | | | | | |
| Number | Input/action | Expected output response | | The actual situation | Correct or not? | Error number |
| 1 | The user fills in the repair information normally. | The system prompts that the report is successful. | |  |  |  |
| 2 | User does not fill in the repair information | The system prompts “Please fill in the report information.” | |  |  |  |

## 6.5 PIR sensor module test

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Function test | | | | | | |
| Overview | | | | | | |
| Test number | | | GN004 | | | |
| Functional description | | | The PIR sensor senses if there is someone in the seat and sends the data to the background. | | | |
| Module manager | | | TingHui Zhang | | | |
| tester | | | Huiying Han | | | |
| Use purpose | | | Whether the PIR sensor can be sensed normally and whether the data can be sent normally. | | | |
| Precondition | | |  | | | |
| Test operation | | | | | | |
| Number | Input/action | Expected output response | | The actual situation | Correct or not? | Error number |
| 1 | When someone is in the sensing range. | Prompt someone is here successful and the data is successfully sent to the background. | |  |  |  |
| 2 | No one appears in the sensing area for a long time | Always prompt nobody is here successful and the data is successfully sent to the background. | |  |  |  |

## 6.6 Indicate light module test

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Function test | | | | | | |
| Overview | | | | | | |
| Test number | | | GN005 | | | |
| Functional description | | | The indicator light will light according to the seat status | | | |
| Module manager | | | Huiying Han | | | |
| tester | | | Hanyu Zhang | | | |
| Use purpose | | | Whether the indicator light will light according to the seat status | | | |
| Precondition | | |  | | | |
| Test operation | | | | | | |
| Number | Input/action | Expected output response | | The actual situation | Correct or not? | Error number |
| 1 | Reserve a seat. | The indicator light is on normally. | |  |  |  |
| 2 | Confirm reserve succeed | The indicator light goes out | |  |  |  |
| 3 | Unconfirmed for a long time after reserve. | The indicator light goes out | |  |  |  |