

# Requirements Document

## 1. Introduction

### 1.1 Purpose of Writing

This project aims to develop a campus dormitory property management system to enhance the team's collaboration skills and software development capabilities, while also improving communication among members. The system will provide a web-based solution for potential users.

### 1.2 Project Background

Software Name: Electric Shock Boys Property Management System

Project Proposer: Lin Qifeng

Project Development Team: Electric Shock Boys Development Team

### 1.3 References

"Building Techniques"

## 2. Overall Description

## **2.1 Objectives**

### **2.1.1 Development Intent**

Using web development technology, create an intuitive and user-friendly dormitory property management system, allowing users to complete related functions through a simple operational interface.

### **2.1.2 Application Objectives and Scope**

The system serves on-campus students and property management staff, with main functions including dormitory information management, occupancy management, and repair requests.

### **2.1.3 Product Prospects**

Based on market research, there is a strong demand among students for dormitory management systems. The system will be developed according to these needs to enhance the user management experience.

## **3. Specific Requirements**

### **3.1 Attributes**

#### **3.1.1 Simplicity**

The user interface should be clear and concise, with simplified operational processes.

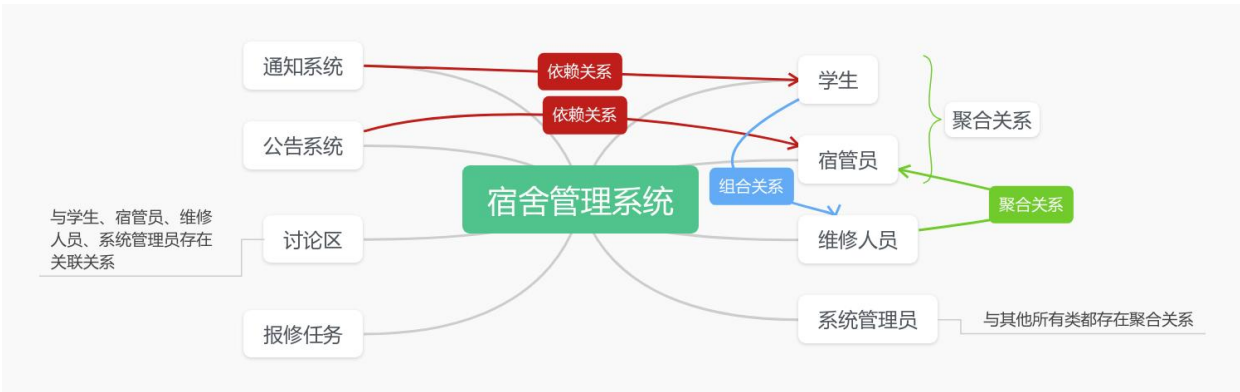
#### **3.1.2 Stability**

The system must maintain high stability to ensure that user operations are not interrupted by system failures.

#### **3.1.3 Data Authenticity**

Data will be jointly supervised by students, school leaders, dormitory management teachers, etc., to ensure reliable and trustworthy sources.

3.2 Class Diagram



4. Interface Prototype

In development, please stay tuned.

5. Function Description and Acceptance Criteria

## **5.1 Detailed Function Introduction**

### **5.1.1 Student Functions**

Dormitory application and cancellation

Repair registration and progress inquiry

Dormitory fee inquiry and payment

Accommodation information viewing (roommates, floor, etc.)

### **5.1.2 Dormitory Management/Maintenance Staff Functions**

Repair task management

Dormitory inspection and recording

Data statistics and report generation

### **5.1.3 Administrator Functions**

Dormitory allocation and adjustment

Repair order management

Dormitory occupancy information statistics

Fee management and payment reminders

### **5.1.4 Other Functions**

Message notification system (SMS/Email)

Real-time announcement publishing

System permissions management

Information sharing discussion area

## **5.2 Input and Output Formats**

### **5.2.1 Input Format**

Login to System:

Input Fields: Username (student/staff ID), Password

Validation: Input information must match the credentials stored in the system.

Dormitory Application:

Input Fields: Student ID, Personal Information (Name, Contact Information), Living Habits, Room Preferences

Validation: All required fields must be filled; fields cannot be left empty.

Repair Request:

Input Fields: Dormitory Number, Problem Description, Desired Repair Time

Validation: Problem description must be detailed; dormitory number must match the system.

Fee Payment:

Input Fields: Room Number, Payment Method, Payment Amount

Validation: Payment amount is auto-generated; the user must select a payment method.

### **5.2.2 Output Format**

Login to System:

Output: Displays a welcome message upon successful login; provides specific error messages upon failure.

Dormitory Application Status:

Output: Displays application confirmation information, allocates room or places on waiting list.

Repair Request:

Output: Generates a repair request number, displays task status updates (e.g., “In Progress” or “Completed”).

Fee Payment:

Output: Generates an electronic receipt upon successful payment and sends confirmation notification via email or SMS.

### **5.3 Interface Acceptance Criteria**

User Interface:

The design should be clear and concise, with buttons and field labels being distinct. Navigation should be convenient, with no more than two clicks needed to access the main functional modules from the homepage.

Responsiveness:

The interface should have a responsive design that adapts to various device screen sizes (desktop, tablet, mobile).

Accessibility:

The system should provide alternative text for images, use high-contrast color schemes, and support screen readers for easy operation by visually impaired users.

Consistency:

The interface style should be uniform, with consistent headers, footers, fonts, and colors.

Error Handling:

The system should provide clear prompts when errors occur (e.g., “Invalid dormitory number” or “Password too short”), and error messages should be displayed below input fields without affecting user experience.

### **5.4 Functional Acceptance Criteria**

Login/Authentication Function:

Users should be able to complete login within 5 seconds. If login fails 5 times, the system should lock the account and provide unlocking instructions via email.

#### Dormitory Application:

Users can submit, view, and edit dormitory applications, and the system should notify users when the application status changes.

#### Repair Request:

Users can submit repair requests and track progress; maintenance staff can update task statuses, and the system should send status update notifications to users.

#### Fee Payment:

Students should be able to view fee details, choose payment methods, and complete payments. After successful payment, the system should generate an electronic receipt and automatically send confirmation via email.

#### Administrator Functions:

Administrators should be able to quickly complete dormitory allocation, manage repair tasks, and generate various statistical reports (e.g., occupancy rates, repair statuses, and payment statuses).

## **6 Interface Prototype**

### **6.1 User Function Module**

Users can click to enter the system operation interface, where they can access various functional modules, including the homepage, personal center, owner information management, fee information management, building information management, repair information management, parking space information management, parking information management, complaint number management, and announcement information management. In the personal information section, users can view and modify their account details, including account number, name, gender, age, ID number, phone number, license plate number, and photo.

工号: 22

性别: 男

身份证号: 22366555412512630

员工姓名: 李明

年龄: 30

手机号码: 12544780590

照片: [Upload Button]

部门: 部门名称2

## 6.2 Building Information Management

Users can retrieve information such as information number, building name, building location, number of floors, layout, number of units, and publication date from the list. They can also perform detail views and delete operations.

信息编号	楼栋名称	楼栋位置	楼层数	户型	户型数量	发布日期	操作
信息编号1	楼栋名称1	楼栋位置1	1	户型1	1	2021-03-22	修改 删除
信息编号2	楼栋名称2	楼栋位置2	2	户型2	2	2021-03-22	修改 删除
信息编号3	楼栋名称3	楼栋位置3	3	户型3	3	2021-03-22	修改 删除
信息编号4	楼栋名称4	楼栋位置4	4	户型4	4	2021-03-22	修改 删除
信息编号5	楼栋名称5	楼栋位置5	5	户型5	5	2021-03-22	修改 删除
信息编号6	楼栋名称6	楼栋位置6	6	户型6	6	2021-03-22	修改 删除

## 6.3 Administrator Function Module

Administrators can log in by filling in their username, password, and role, as shown in Figure 5-5. After a successful login, administrators enter the system operation interface, where they can perform corresponding operations on various functional modules, including the homepage, personal center, user management, employee management, owner information management, fee information management, building information management, repair information management, parking space information management, parking information management, complaint number management, announcement information management, and department information.



management.

首页 / 员工

工号

工号





员工姓名

员工姓名

搜索

+新增

删除

<input type="checkbox"/>	工号	员工姓名	性别	年龄	身份证号码	手机号码	照片	部门	操作
<input type="checkbox"/>	员工1	员工姓名1	男	1	440300199101010001	13823888881		部门1	<div><div>详情</div><div>修改</div><div>删除</div></div>
<input type="checkbox"/>	员工2	员工姓名2	男	20	440300199202020002	13823888882		部门2	<div><div>详情</div><div>修改</div><div>删除</div></div>
<input type="checkbox"/>	员工3	员工姓名3	男	3	440300199303030003	13823888883		部门3	<div><div>详情</div><div>修改</div><div>删除</div></div>
<input type="checkbox"/>	员工4	员工姓名4	男	4	440300199404040004	13823888884		部门4	<div><div>详情</div><div>修改</div><div>删除</div></div>