校园共享单车管理系统

设计文档

张鑫成 20271055

张辰昕 20271053

王继航 20301019

王 炯 20231110

文档控制

变更记录

| 版本号 | 日期 | 作者 | 段落、图或表 | 增加/修改/删除 | 简单描述 | 更改申请单号 |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
| 1.0 | 20230520 | 张鑫成、张辰昕、王继航、王炯 |  | 创建 | 新建文字、图、表 |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

目录

[校园共享单车管理系统 1](#_Toc7945)

[文档控制 2](#_Toc15964)

[1. 系统结构 4](#_Toc30946)

[1.1系统功能结构图 4](#_Toc3267)

[2. 架构设计 7](#_Toc8177)

[2.1系统整体框架设计 7](#_Toc28256)

[3. 数据库设计 8](#_Toc19928)

[3.1数据库ER图 8](#_Toc26385)

[3.2表结构 9](#_Toc27831)

[3.3数据库关系 9](#_Toc16200)

[3.4索引 9](#_Toc22204)

[4. 接口设计： 10](#_Toc30168)

[4.1内部接口清单 10](#_Toc28492)

[4.2用户信息管理模块 10](#_Toc10504)

[4.2.1用户登录 10](#_Toc19001)

[4.2.2用户注册 11](#_Toc31074)

[4.2.3用户信息查询 12](#_Toc31205)

[4.3单车信息管理模块 13](#_Toc15320)

[4.3.1新增单车 13](#_Toc9024)

[4.3.2删除单车 14](#_Toc15922)

[4.3.3单车信息查询 14](#_Toc12040)

[5. 用户界面设计：详细描述系统的用户界面设计，包括界面元素、布局、交互流程等。 15](#_Toc14318)

[6. 安全设计 15](#_Toc27061)

[6.1身份验证： 15](#_Toc4841)

[6.2访问控制： 15](#_Toc12263)

[6.3数据加密： 16](#_Toc21133)

[6.4日志和监控： 16](#_Toc29544)

[7. 错误处理和异常处理 16](#_Toc21862)

[7.1错误码： 16](#_Toc1262)

[7.2异常处理： 16](#_Toc19518)

[7.3异常日志记录： 16](#_Toc12614)

[8. 测试设计 17](#_Toc23345)

[测试计划： 17](#_Toc29848)

# 系统结构

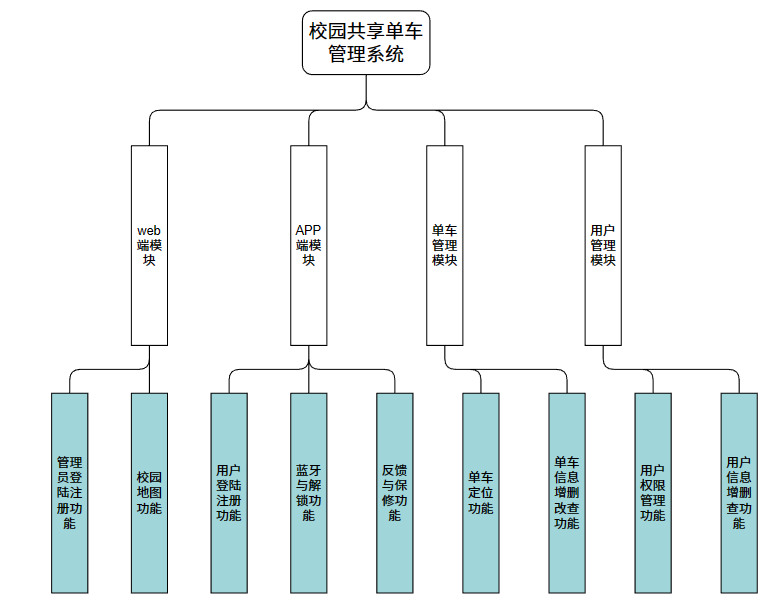
## 1.1系统功能结构图

该景区推荐系统主要分为四大模块共九个细分功能。

信息管理模块由用户注册登陆、用户信息修改两个功能。用户注册登陆主要负责用户和管理员账号的登陆注册；用户信息修改让是用户可在个人信息管理界面对邮箱、账号密码等信息进行修改。

景区推荐模块由景区信息展示、人流量预测、景区推荐功能三个功能组成。景区信息展示是将每个景区的评论评分等信息展示给用户，供用户参考；人流量预测功能是通过模型对每个景区后一年的人流量进行预测，然后存储进数据库供后端读取；景区推荐功能是读取用户选定日期的每个景区的人流量，之后用折线图、柱状图和地图散点热力图的方式可视化地将各个景区当天人流量和近期人流量变化呈现给用户，之后根据一定原则推荐给用户五个景区。

交互展示则分为景点定位、景点收藏、系统反馈、用户评价四个功能组成。景点定位功能是用户可以点击某一个推荐结果中的景区，地图就会跳转到该景区所在的地理位置，方便用户更直观便捷地了解景区信息；景区收藏功能就是用户可以收藏喜欢的景区，并且持久化存储；景区反馈功能是用户可以对景区人流量预测系统做出评级，该评级会反馈给管理员用于调整系统；用户评价功能是用户可以对系统、景区等做出评价，评价会呈现给之后的用户，提供一定的参考。



**图1.2 系统功能结构图**

如图所示，本系统有web展示功能，APP端展示模块，单车模块功能以及用户管理模块。其主要功能为校园地图展示功能、单车定位功能、蓝牙与解锁功能功能。

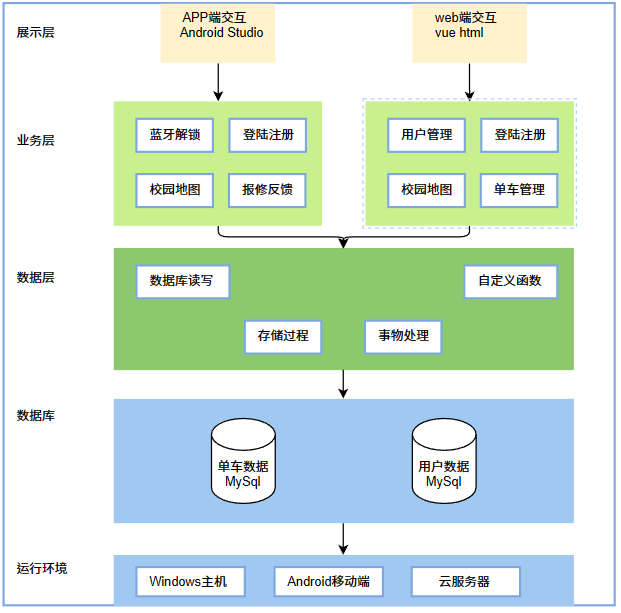
系统功能描述表如下。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | **序号** | **功能名称** | **标识符** | **用途与说明** | | --- | --- | --- | --- | |  | 管理员登录功能 | ZCYDL | 用户可以通过注册获得新账号，根据正确的用户名密码登录系统 | |  | 校园地图功能 | YHXXXG | 用户可以在个人中心处修改账号密码 | |  | 用户登录注册功能 | JQXXGL | 用户可以在个人中心处查看收藏的景点，可以在百度地图上查看景区信息，在主页查看景点综合评分 | |  | 蓝牙与解锁功能 | JQXXZS | 主页使用百度地图，点击景点即可获取景区信息 | |  | 人流量预测功能 | RLLYC | 通过训练完成的模型对选择日期当天的人流量进行预测 | |  | 景区推荐功能：折线图 | JQTJA | 通过选择日期获取推荐前三的景点往后五天的流量绘制折线图，让用户可以看到景区流量后续发展趋势，更好地做旅游规划 | |  | 景区推荐功能：柱状图 | JQTJB | 通过选择日期获取当天推荐前五的景点，绘制柱状图，让用户可以直观看到第一推荐梯队的景点流量信息与对比 | |  | 景区推荐功能：百度地图+散点图 | JQTJC | 通过选择日期获取当天所有景点的流量与经纬度信息，将景点位置与流量用散点的方式表现，使用户可以全局查看城市景区流量与位置，同时可以在这上面查看景区信息 | |  | 景区推荐功能：文字展示 | JQTJD | 通过选择日期获取当天推荐前十的景区名以及景区评分，在这里可以使用跳转按钮将百度地图的中心定位到相应的景区位置，也可以使用收藏按钮对景点进行收藏 | |  | 信息管理单元 | XXGL | 存储景区的流量、经纬度、景区名等信息，按流量升序存储 | |  | 景区信息单元 | JQXX | 存储景区的综合评分与景区名，由于加入百度地图api不需要存储太多景区信息。 | |  | 预测推荐单元 | YCTJ | 通过获取的数据将预测推荐结果以图文的形式展现 | |  | 社区评论单元 | SQPL | 用户可以对选择的景点进行评分，系统将统计综合评分并在预测处展现 | |  | 用户评价功能 | YHPJ | 用户可以根据使用感受对系统进行反馈，管理员根据反馈升级系统 | |

**表4-1 景区推荐模块功能描述**

# 架构设计

## 2.1系统整体框架设计



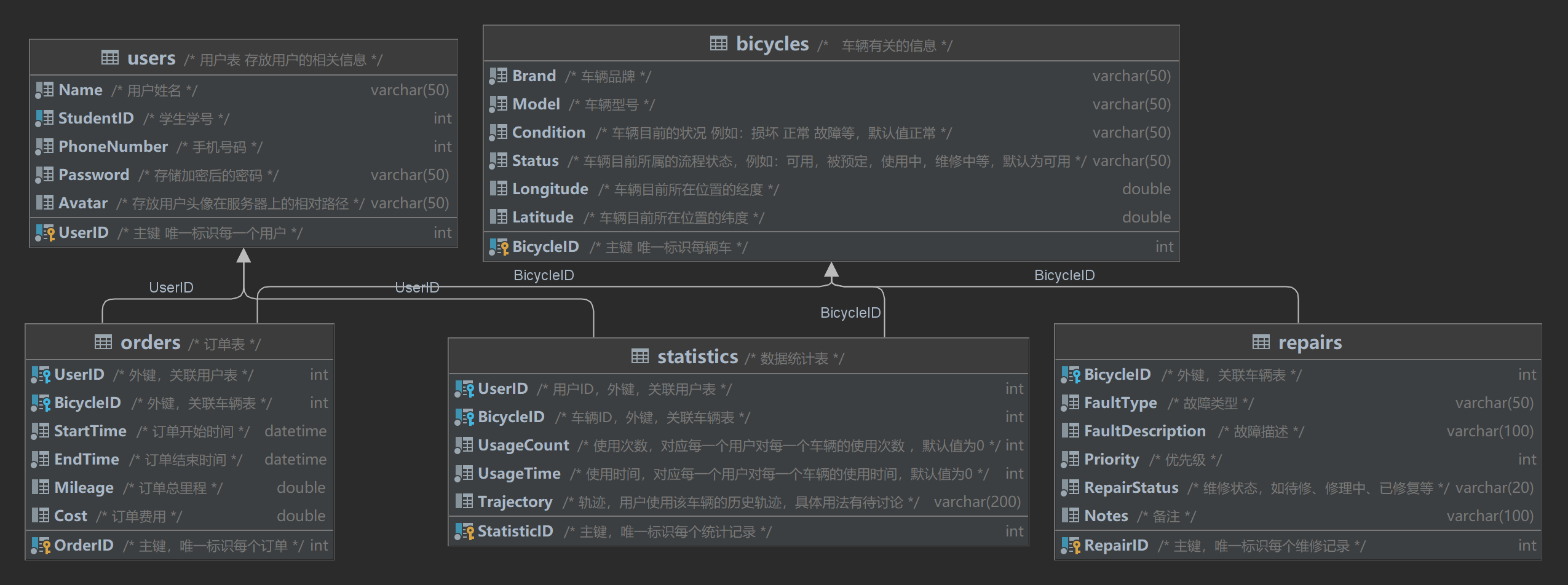
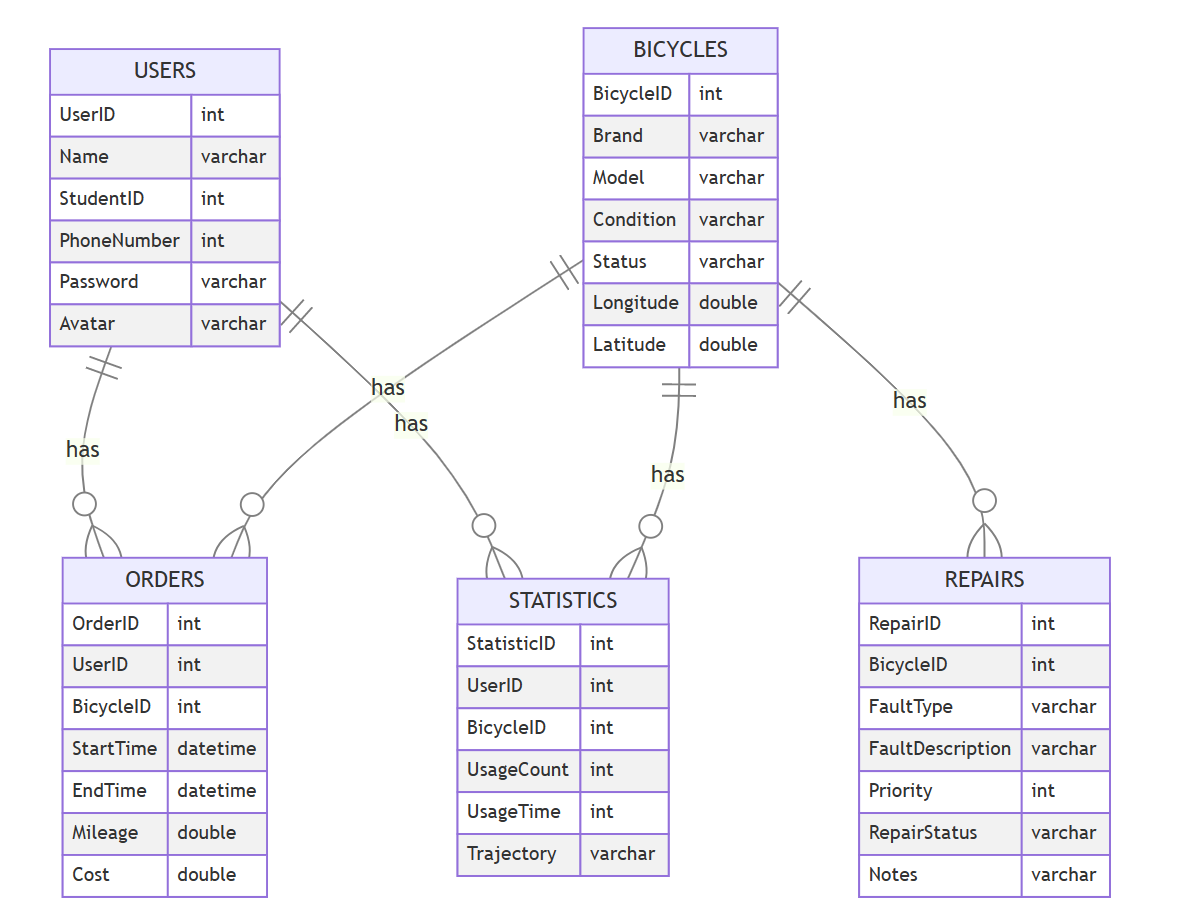
**图1.1 系统整体框架图**

如图所示，本系统分为展示层、业务层、数据层、数据库层以及运行环境层。

1. 展示层主要面向用户，向用户展示最终信息并接收用户的输入。
2. 业务层主要包括了用户以及管理员需要使用的功能。
3. 数据层负责对数据库进行读写以及处理、传输。
4. 数据库层负责存储单车数据、用户数据。
5. 运行环境层有Windows主机、Android移动端、云服务器负责为用户、管理员提供服务。

# 数据库设计

## 3.1数据库ER图



**图3.1数据库ER图**

## 3.2表结构

1. 用户表（Users）：UserID (主键)、Name、StudentID、PhoneNumber、Password、Avatar
2. 车辆表（Bicycles）：、BicycleID (主键)、Brand、Model、Condition、Status、Longitude、Latitude
3. 订单表（Orders）：、OrderID (主键)、UserID (外键，关联用户表的UserID)、BicycleID (外键，关联车辆表的BicycleID)、StartTime、EndTime、Mileage、Cost
4. 维修表（Repairs）：、RepairID (主键)、BicycleID (外键，关联车辆表的BicycleID)、FaultType、FaultDescription、RepairType、Priority、RepairStatus、Notes
5. 数据统计表（Statistics）：、StatisticID (主键)、UserID (外键，关联用户表的UserID)、BicycleID (外键，关联车辆表的BicycleID)、UsageCount、UsageTime、Trajectory

## 3.3数据库关系

1. 用户表（Users）与订单表（Orders）之间是一对多的关系，一个用户可以有多个订单，但每个订单只能对应一个用户。
2. 用户表（Users）与数据统计表（Statistics）之间也是一对多的关系，一个用户可以有多个统计记录，但每个统计记录只能对应一个用户。
3. 车辆表（Bicycles）与订单表（Orders）之间是一对多的关系，一辆车可以有多个订单，但每个订单只能对应一辆车。
4. 车辆表（Bicycles）与维修表（Repairs）之间是一对多的关系，一辆车可以有多个维修记录，但每个维修记录只能对应一辆车。
5. 车辆表（Bicycles）与数据统计表（Statistics）之间也是一对多的关系，一辆车可以有多个统计记录，但每个统计记录只能对应一辆车。

## 3.4索引

在用户表（Users）中，可以为UserID列添加主键索引，为PhoneNumber列添加唯一索引。

在车辆表（Bicycles）中，可以为BicycleID列添加主键索引。

在订单表（Orders）中，可以为OrderID、UserID和BicycleID列添加主键索引，根据查询需求可能还需要根据StartTime和EndTime添加索引。

在维修表（Repairs）中，可以为RepairID和BicycleID列添加主键索引，根据查询需求可能还需要根据RepairStatus添加索引。

在数据统计表（Statistics）中，可以在StatisticID、UserID和BicycleID字段上创建索引，以便根据统计ID、用户ID和车辆ID进行快速查询。

# 接口设计：

## 4.1内部接口清单

| **序号** | **模块** | **说明** |
| --- | --- | --- |
| 1 | 用户信息管理模块 | 主要包含用户的信息查询修改等操作 |
| 2 | 单车信息管理模块 | 主要包含单车轨迹、单车维修信息等查询修改 |

## 4.2用户信息管理模块

| **序号** | **接口** | **说明** |
| --- | --- | --- |
| 1 | login | 用户登录 |
| 2 | register | 用户注册 |
| 3 | userInfo | 用户信息查询 |

### **4.2.1用户登录**

1）功能说明

用户登录进入系统

1. 接口说明

请求方式：POST

请求地址：http://ip:port/user/login

请求参数： username , password

3）请求参数

示例:

http://127.0.0.1:8080/user/login

{ "username": "zzz" , "password": "123" }

4）响应参数

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| data | User | 用户对象 |
| status | String | 状态：1-成功，0-失败 |
| httpCode | String | Http返回码 |
| errCode | String | 错误码 |
| errMessage | String | 错误信息 |

示例：

{ "status": 1, "data": { "username": "zzz", "password": "123" }, "httpCode": 200, "errCode": 0, "errMessage": "" }

### **4.2.2用户注册**

1）功能说明

新用户注册

2）接口说明

请求方式：POST

请求地址：http://ip:port/user/register

请求参数： name , password

3）请求参数

示例:

http://127.0.0.1:8080/user/register

{ "username": "zzz" , "password": "123"}

4）响应参数

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| data | User | 用户对象 |
| status | String | 状态：1-成功，0-失败 |
| httpCode | String | Http返回码 |
| errCode | String | 错误码 |
| errMessage | String | 错误信息 |

示例：

{ "status": 1, "data": { "username": "zzz", "password": "123" }, "httpCode": 200, "errCode": 0, "errMessage": "" }

### **4.2.3用户信息查询**

1）功能说明

用户信息查询

1. 接口说明

请求方式：GET

请求地址：http://ip:port/user/userInfo

请求参数： username

3）请求参数

示例:

http://127.0.0.1:8080/user/userInfo

{ "username": "zzz" }

4）响应参数

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| data | User | 用户对象 |
| status | String | 状态：1-成功，0-失败 |
| httpCode | String | Http返回码 |
| errCode | String | 错误码 |
| errMessage | String | 错误信息 |

示例：

{ "status": 1, "data": { "name": "wjh", "password": "123" }, "httpCode": 200, "errCode": 0, "errMessage": "" }

## 4.3单车信息管理模块

| **序号** | **接口** | **说明** |
| --- | --- | --- |
| 1 | add | 新增单车 |
| 2 | delete | 删除单车 |
| 3 | bicycleInfo | 单车信息查询 |

### **4.3.1新增单车**

1）功能说明

系统存储新加入单车信息

1. 接口说明

请求方式：POST

请求地址：http://ip:port/bicycle/add

请求参数： BluetoothId

3）请求参数

示例:

http://127.0.0.1:8080/bicycle/add

{ "BluetoothId": "03" }

4）响应参数

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| data | Bicycle | 单车对象 |
| status | String | 状态：1-成功，0-失败 |
| httpCode | String | Http返回码 |
| errCode | String | 错误码 |
| errMessage | String | 错误信息 |

示例：

{ "status": 1, "data": { "BluetoothId": "03", "track": "", "maintenance": "" }, "httpCode": 200, "errCode": 0, "errMessage": "" }

### **4.3.2删除单车**

1）功能说明

系统删除对应存储单车信息

1. 接口说明

请求方式：DELETE

请求地址：http://ip:port/bicycle/delete

请求参数： BluetoothId

3）请求参数

示例:

http://127.0.0.1:8080/bicycle/delete

{ "BluetoothId": "03" }

4）响应参数

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| status | String | 状态：1-成功，0-失败 |
| httpCode | String | Http返回码 |
| errCode | String | 错误码 |
| errMessage | String | 错误信息 |

示例：

{ "status": 1, "httpCode": 200, "errCode": 0, "errMessage": "" }

### **4.3.3单车信息查询**

1）功能说明

查询单车轨迹、维修等信息

1. 接口说明

请求方式：GET

请求地址：http://ip:port/bicycle/bicycleInfo

请求参数： BluetoothId

{ "BluetoothId": "03" }

3）请求参数

示例:

http://127.0.0.1:8080/bicycle/bicycleInfo

4）响应参数

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| data | Bicycle | 单车对象 |
| status | String | 状态：1-成功，0-失败 |
| httpCode | String | Http返回码 |
| errCode | String | 错误码 |
| errMessage | String | 错误信息 |

示例：

{ "status": 1, "data": { "BluetoothId": "03", "track": "逸夫楼-思源楼-18号楼", "maintenance": "待维修" }, "httpCode": 200, "errCode": 0, "errMessage": "" }

# 用户界面设计：详细描述系统的用户界面设计，包括界面元素、布局、交互流程等。

# 安全设计

# 6.1身份验证：

1. 使用安全的身份验证机制，如用户名和密码、令牌（JWT）等，来进行用户的身份验证。
2. 在数据库使用加密算法对密码进行存储和传输保护。

# 6.2访问控制：

1. 实施适当的访问控制机制，根据用户的角色和权限限制对系统资源的访问。
2. 使用基于角色的访问控制模型来管理用户对不同功能和数据的访问权限。
3. 对敏感操作和敏感数据实施额外的访问控制，例如使用双重授权或审批流程来限制高风险操作的权限。

# 6.3数据加密：

1. 对敏感数据进行加密，包括用户密码、个人身份信息和敏感业务数据等。
2. 确保在数据库、文件系统、通信协议等多个层面上实施数据加密措施。

# 6.4日志和监控：

1. 记录系统操作日志，包括用户登录、权限变更、敏感操作等，以便追踪和审计系统的安全事件。
2. 实施实时监控和告警机制，检测异常行为和潜在的安全威胁，并及时采取措施进行响应和阻止。

# 错误处理和异常处理

# 7.1错误码：

1. 定义一套错误码来标识不同类型的错误和异常情况。
2. 每个错误码具有唯一性，以便在出现错误时能够准确地识别和处理。
3. 错误码可以按照一定的规范进行命名和分类，以便开发人员和用户理解和处理错误。

# 7.2异常处理：

1. 在代码中捕获和处理异常，以避免系统崩溃或无法正常运行。
2. 使用适当的异常类来表示不同类型的异常情况，以便能够对其进行特定的处理。
3. 实施异常处理机制，包括try-catch语句块、异常捕获和异常处理程序，以便在出现异常时能够进行相应的处理逻辑。

# 7.3异常日志记录：

1. 在出现异常情况时，记录相关的异常信息，包括异常类型、时间戳、堆栈跟踪等。
2. 用于故障排除、问题分析和系统维护，有助于定位和解决异常情况。

# 测试设计

# 测试计划：

1. 根据测试策略制定详细的测试计划，包括测试任务、时间表和里程碑。
2. 确定测试的输入和预期输出，制定测试用例和测试数据。
3. 设计和实施测试环境，包括搭建开发环境和测试环境。
4. 分配测试资源，确定测试团队的组成和责任分工。