

# 系统设计说明书

## 目录

系统设计说明书.....	1
1、 引言.....	2
1.1 编写目的.....	2
1.2 背景.....	3
1.3 定义.....	3
2、 程序系统的结构.....	4
2.1 类图.....	4
2.2 用例图.....	4
2.3 状态图.....	5
2.4 泳道图.....	5
2.5 顺序图.....	6
3、 程序设计说明.....	6
3.1 流程逻辑.....	6
3.2 程序描述.....	7
3.2.1Login()函数.....	7
3.2.2Clock()函数.....	7
3.2.3AddFriends()函数.....	7
3.2.4Charts()函数.....	7
3.2.5ShowTime()函数.....	8

3.2.6Weekends()函数.....	8
3.2.7DataMaintenance()函数.....	8
3.2.8DiaryMaintenance()()函数.....	8
4、 系统安全.....	8
4.1 系统安全设计原则.....	8
4.1.1 标识与确认.....	9
4.1.2 授权.....	9
4.1.3 日志.....	9
4.1.4 加密.....	9
4.2 系统级安全.....	9
4.2.1 操作系统平台的安全管理.....	10
4.2.2 数据库系统的安全管理.....	10
4.3 应用级安全.....	10
4.3.1 用户授权及安全访问控制.....	11
4.3.2 日志管理机制.....	11
4.3.3 数据加密及数据保护.....	11
5、 ER 图.....	11

## 1、引言

### 1.1 编写目的

在本阶段中，确定应该如何具体地实现所要求的系统，从而在编码阶段可以

把这个描述直接翻译成用具体的、程序语言书写的程序。主要工作有：根据在《需求分析说明书》中所描述的数据、功能、运行、性能需求，设计小程序的结构设计、逐个模块的程序描述（包括各模块的功能、性能、输入、输出等等），解决如何

- （1）接收：接收获取用户打卡的时间信息
- （2）用户打卡时长排行榜
- （3）用户周末统计报告
- （4）好友在线时长
- （5）系统管理员 日志维护
- （6）系统管理员 用户信息维护
- （7）系统管理员 后台维护
- （8）输出：相关提示信息

接下来的各个阶段将与本阶段的工作密切结合，努力做到让用户易懂易学，同时将小程序开发的工作落到实处。

## 1.2 背景

项目名称：**wow** 打卡。

项目要求：帮助同学们更好的了解自己的学习情况。

## 1.3 定义

**SQL**：一种用于访问查询数据库的语言。

**事务流**：数据进入模块后可能有多种路径进行处理。

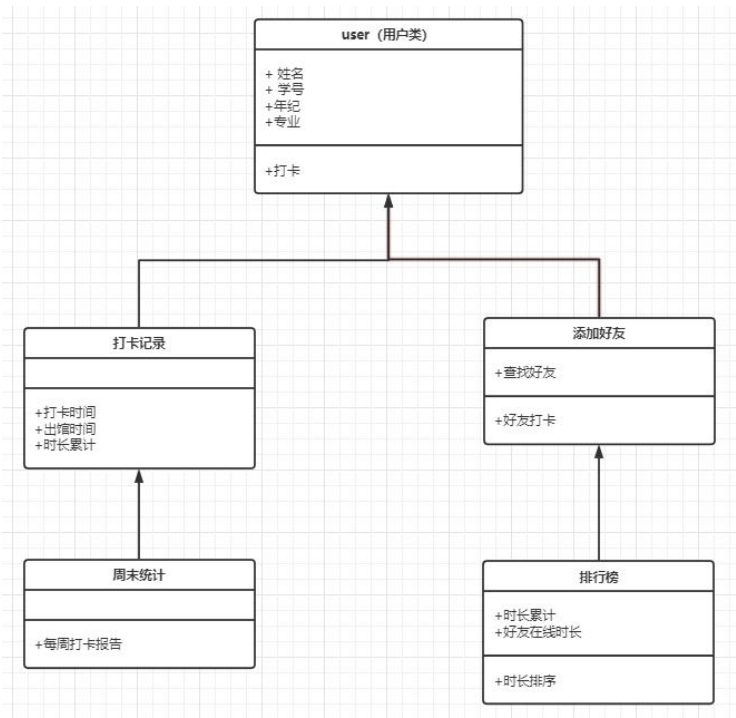
**主键**：数据库表中的关键域。值互不相同。

**外部主键**：数据库表中与其他表主键关联的域。

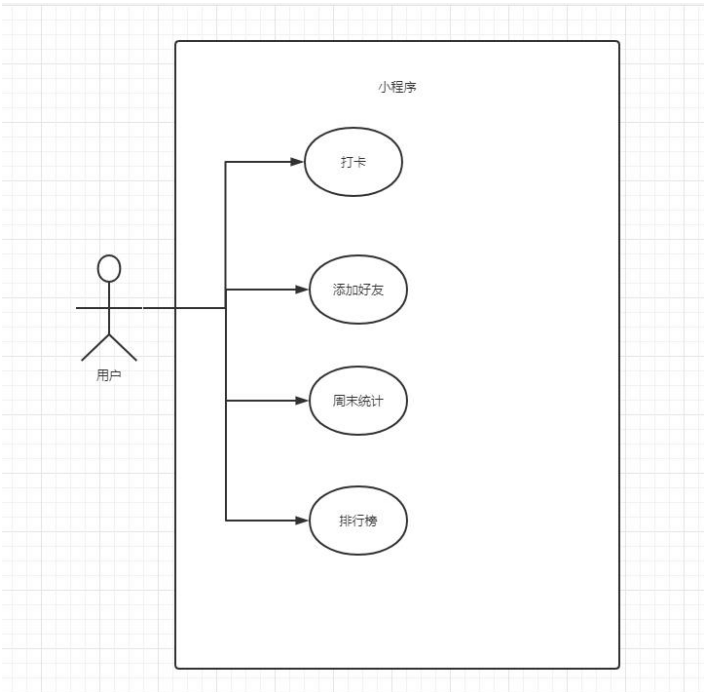
**ROLLBACK**：数据库的错误恢复机制。

# 1、程序系统的结构

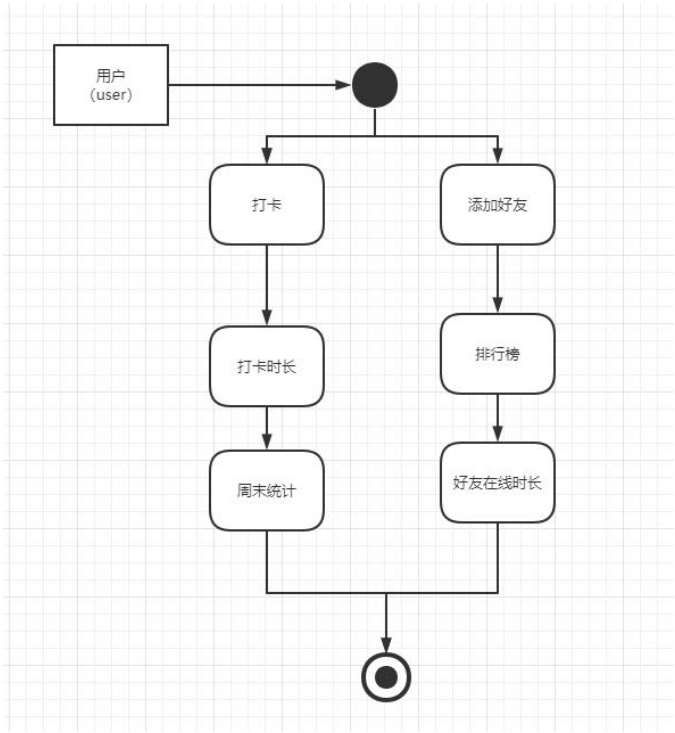
## 2.1 类图



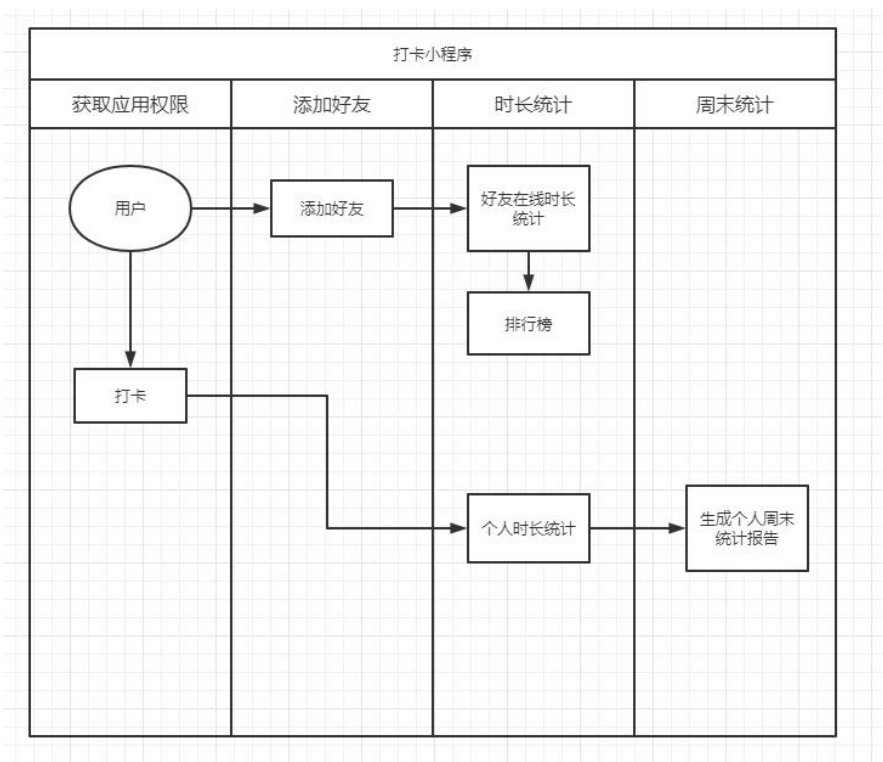
## 2.2 用例图



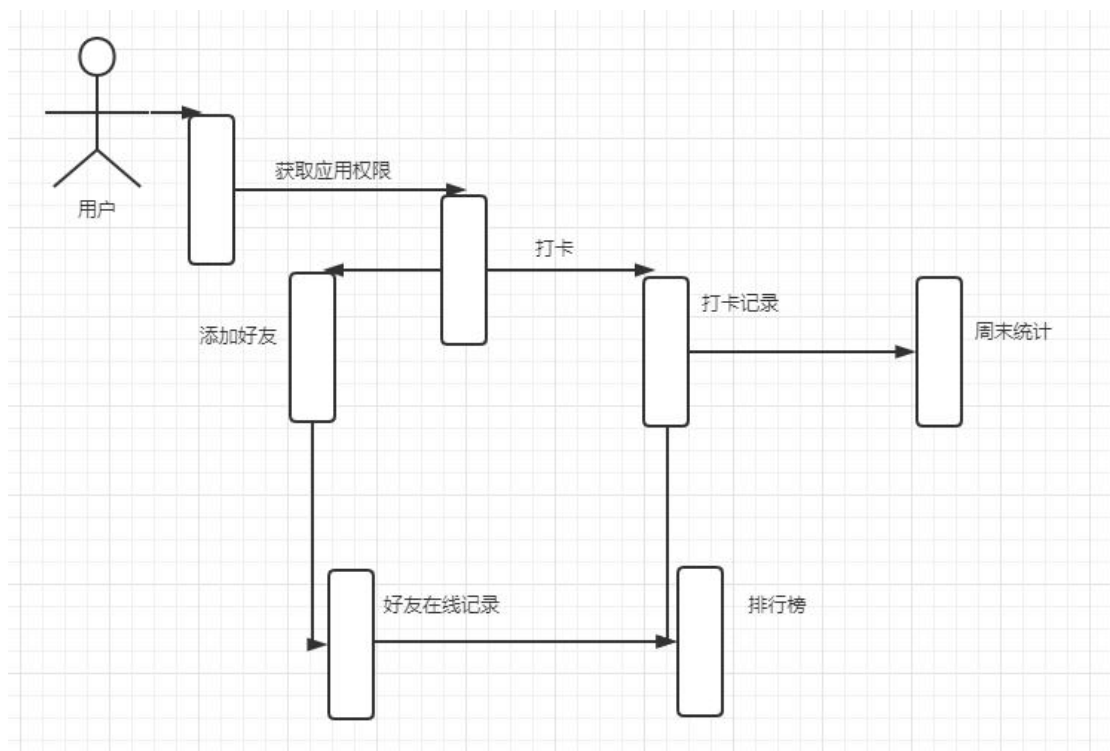
2.3 状态图



2.4 泳道图



## 2.5 顺序图

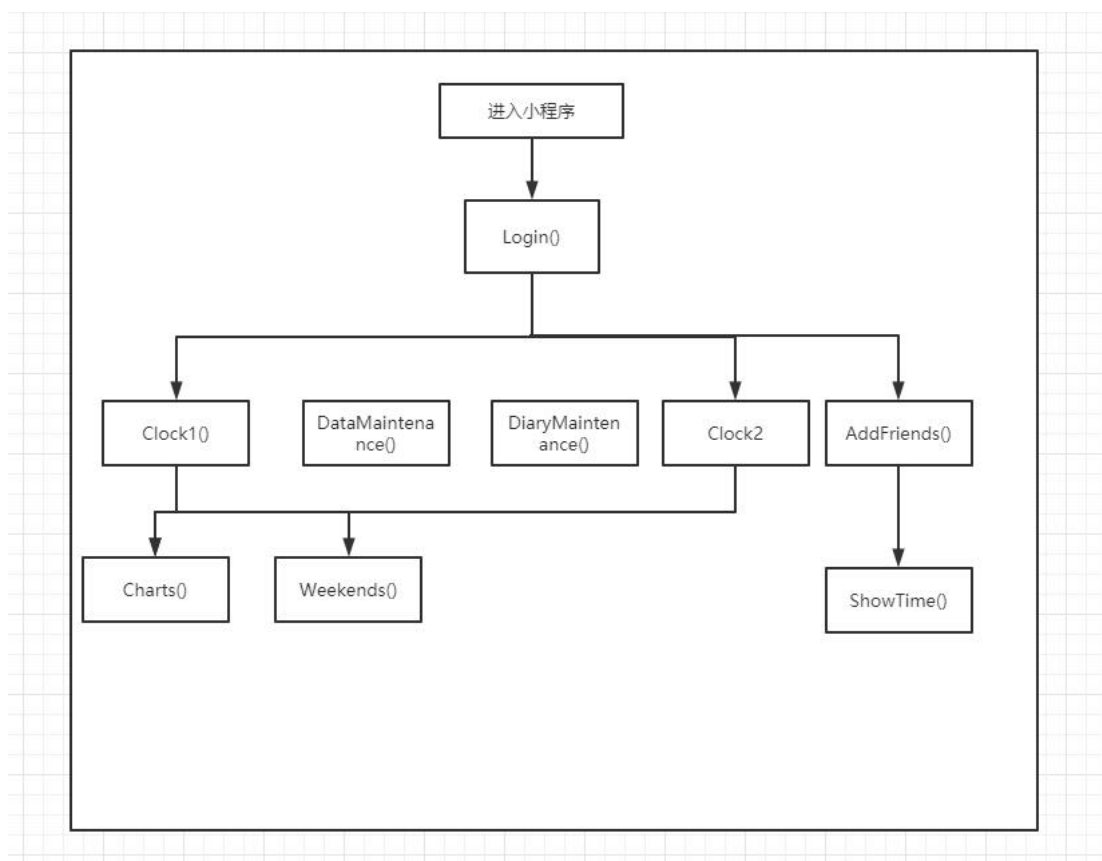


## 2、程序设计说明

### 3.1 流程逻辑

#### 3.1.1 用户模块

首先，调用 `Login()` 函数实现对用户权限的获取，对要实现的每一项功能，都有相对应的函数：在打卡时，调用 `Clock1()` 函数；`Clock2()` 实现签退功能；`AddFriends()` 函数实现添加好友的功能；`Charts()` 实现排行榜的功能；`ShowTime()` 实现显示好友在线时长的功能；`Weekends()` 实现对用户一周打卡时长的统计功能；`DataMaintenance()` 实现对数据的维护；`DiaryMaintenance()` 实现对日志的维护等。



## 3.2 程序描述

### 3.2.1 Login()函数

用户通过在微信端允许小程序访问权限，使得小程序获取到所需要的信息，例如地点、时间、用户名等，从而使小程序的功能能够正常运行。

### 3.2.2 Clock()函数

用户通过点击打卡按钮，调度 Clock1()函数，从而使小程序能够获取到当前的时间，点击签退按钮，调动 Clock2()函数，获取签退时的时间，然后通过对签退时间和签到时间的时间差计算得到打卡时长，保存数据，以便其他函数获取信息，当用户多次进行打卡则进行累计。

### 3.2.3 AddFriends()函数

用户通过对微信通讯录现存好友的记录，添加自己想要添加的好友。

### 3.2.4 Charts()函数

通过对打卡信息的保存记录以及好友，将累计打卡时长按从高到低的顺序排列出来，方便用户了解今天一天自己的学习情况。

### **3.2.5ShowTime()函数**

通过添加好友，以及好友打卡的记录，可以查阅你想查阅的好友的打卡记录，从而形成对良好的学习氛围。

### **3.2.6Weekends()函数**

将用户 7 天的打卡时长进行总的统计，形成一份周报告，用户只需要每周进行查看就可以。

### **3.2.7DataMaintenance()函数**

系统管理员可以选择要维护的数据信息时间范围，确认后将显示该时间范围内的相关数据信息，管理员可以对其进行删除或备份，同时更新日志信息。

### **3.2.8DiaryMaintenance()()函数**

系统管理员可以选择要维护的日志信息时间范围，确认后将显示该时间范围内的相关信息，管理员可以对其进行删除或备份，同时更新日志信息。

## **3、系统安全**

### **4.1 系统安全设计原则**

在网络环境下，由于任何用户对任何资源包括硬件和软件资源的共享，所以必须通过制定相关的安全策略来防止非法访问者访问数据资源，对数据资源的存储以及传输进行安全性保护。在小程序打卡系统中，参考 OSI 的七层协议，从网络安全、传输级协议、系统级安全和应用级安全等几方面进行考虑，主要遵循以下设计原则：



#### **4.1.1 标识与确认**

任何用户访问系统资源，必须得到系统的身份认证以及身份标识，如用户的数据证书、用户号码、密码。当用户信息与确认信息一致时，才能获准访问系统。在本系统中，对操作系统，数据库系统和应用系统都有相应的用户和权限的设置。

#### **4.1.2 授权**

对系统资源，包括程序、数据文件、数据库等，根据其特性定义其保护等级；对不同的用户，规定不同的访问资源权限，系统将根据用户权限，授予其不同等级的系统资源的权限。

#### **4.1.3 日志**

为了保护数据资源的安全，在系统中所保护的资源进行任何存取操作，都做相应的记录，形成日志存档，完成基本的审计功能。

#### **4.1.4 加密**

为了保护数据资源的安全，在系统中所在网络中传输的信息必须经过高强度的加密处理来保证数据的安全性。通过整体考虑来保证网络服务的可用性、网络信息的保密性和网络信息的完整性。

### **4.2 系统级安全**

系统级安全主要体现在系统软件平台的安全设置上。

#### 4.2.1 操作系统平台的安全管理

在操作系统平台上，应进行如下设置：系统的超级用户口令应由专人负责。建立数据库的专用用户，系统在与数据库打交道时，应使用专用用户的身份，避免使用超级用户身份。在系统的其他用户的权限设置中，应保证对数据库的数据文件不能有可写、可删除的权限。选用较高安全级别的操作系统，时刻了解操作系统以及其他系统软件的动态，对有安全漏洞的，及时安装补丁程序。

#### 4.2.2 数据库系统的安全管理

数据库系统是整个系统的核心，是所有业务管理数据以及清算数据等数据存放的中心。数据库的安全直接关系到整个系统的安全。在本系统中对此考虑如下：数据库管理员（SA）的密码应由专人负责。客户端程序连接数据库的用户绝对不能使用数据库管理员的超级用户身份。客户端程序连接数据库的用户在数据库中必须对其进行严格的权限管理，控制对数据库中每个对象的读写权限。利用数据库的审计功能，以对用户的某些操作进行记录。充分使用视图以及存储过程，保护基础数据表。对于不同的应用系统应建立不同的数据库用户，分配不同的权限。

### 4.3 应用级安全

针对本系统，我们在考虑其应用级安全时，主要针对以下几个方面：

- 系统用户授权及安全访问控制

- 全面的日志管理机制

- 各种类型数据的加密

### 4.3.1 用户授权及安全访问控制

对于用户授权和安全访问控制的有关需求，我们在实现本系统时，利用系统的基本定制功能实现对用户属性的定制，可新建用户及用户组，新建角色，用户组可为多层嵌套结构，可按不同用户级别和组级别进行权限分配；角色可以按不同用户级别和组级别进行权限分配，支持 PKI 技术的 CA 安全认证功能及基于 LDAP 的目录服务等功能，支持 HTTPS 安全链路加密访问，支持 SSL 证书安全认证功能。系统设有 XSS 过滤器防止 XSS 漏洞攻击，同时设有 SQL 过滤器，防止 SQL 注入攻击。

### 4.3.2 日志管理机制

实现系统使用情况的日志记录，系统对重要的操作都自动进行日志记录，管理员对日志进行查询、管理。

### 4.3.3 数据加密及数据保护

系统将对传输过程中的信息进行加密处理，对信息进行保护，以防止信息泄露。

## 4、ER 图

