# **《上班打卡系统软件概要设计说明书》**

## **一、引言**

### **（一）编写目的**

本概要设计说明书旨在描述上班打卡系统软件的总体设计架构、模块划分以及各模块之间的接口关系等，为后续的详细设计、编码实现以及系统测试等阶段提供基础框架和指导依据。

### **（二）项目背景**

随着企业管理的规范化和信息化需求的不断增加，传统的人工签到方式已难以满足高效、准确记录员工出勤情况的要求。因此，开发一套上班打卡系统成为提高企业管理效率、降低人力成本的必要举措。

### **（三）定义、首字母缩写词和缩略语**

列出本说明书中用到的特定术语、定义、首字母缩写词和缩略语及其解释，例如：

* API：应用程序编程接口（Application Programming Interface），用于不同软件组件之间的通信和交互。

### **（四）参考资料**

列举编写本概要设计说明书所参考的相关文档、标准、规范等资料，如需求规格说明书、相关行业标准等。

## **二、总体设计**

### **（一）系统概述**

简要描述上班打卡系统的功能和目标，例如：该系统主要用于实现员工上下班的打卡操作，准确记录打卡时间、地点等信息，并能生成相应的考勤报表供管理人员查看和分析，以实现对员工出勤情况的有效监控和管理。

### **（二）系统架构**

采用分层架构（如三层架构：表现层、业务逻辑层、数据访问层）或其他合适的架构模式对系统进行设计，并描述各层的主要职责和功能：

* ****表现层****：负责与用户进行交互，提供友好的打卡界面（如移动端 APP 界面、网页端打卡页面等），接收用户的打卡操作请求，并将处理结果反馈给用户。
* ****业务逻辑层****：包含各种业务处理逻辑，如打卡规则的验证（是否在规定时间内打卡、是否在指定地点打卡等）、考勤数据的计算（迟到、早退、旷工等情况的判定）、员工信息的管理等。
* ****数据访问层****：负责与数据库进行交互，实现对考勤数据、员工信息等数据的存储、查询、更新和删除操作。

同时，可绘制系统架构图来直观展示各层之间的关系和交互流程。

### **（三）运行环境**

说明系统运行所需的硬件环境（如服务器配置、终端设备要求等）和软件环境（如操作系统、数据库管理系统、应用服务器等），例如：

* ****硬件环境****：服务器端要求具备一定的处理器性能、内存容量和存储空间，以满足多用户并发访问的需求；终端设备可以是智能手机、平板电脑或具备网络访问功能的 PC 机。
* ****软件环境****：服务器端操作系统可选用 Windows Server 或 Linux，数据库管理系统可采用 MySQL 或 Oracle，应用服务器可选用 Tomcat 或 Apache 等；移动端 APP 需支持主流的安卓和 iOS 操作系统。

## **三、模块设计**

### **（一）模块划分**

根据系统功能需求，将上班打卡系统划分为若干个相对独立的模块，如打卡模块、员工信息管理模块、考勤报表生成模块、系统设置模块等，并对每个模块的主要功能进行简要描述：

* ****打卡模块****：实现员工上下班的打卡操作，包括获取当前时间和地点信息，按照打卡规则进行打卡验证，并将打卡数据上传至服务器。
* ****员工信息管理模块****：负责对员工的基本信息（如姓名、工号、部门、岗位等）进行录入、查询、修改和删除操作，同时为其他模块提供员工信息的查询服务。
* ****考勤报表生成模块****：根据考勤数据和打卡规则，定期（如每日、每周、每月）生成各类考勤报表（如出勤报表、迟到早退报表、旷工报表等），供管理人员查看和分析。
* ****系统设置模块****：用于设置打卡规则（如上班时间、下班时间、允许迟到时间、打卡地点范围等）、系统参数（如数据备份周期、日志记录级别等）以及对系统用户（管理员、普通员工等）的权限进行管理。

### **（二）模块接口**

描述各模块之间的接口关系，包括接口的输入输出参数、调用方式等。例如：

* 打卡模块与员工信息管理模块之间通过员工工号进行关联，打卡模块在进行打卡操作时需要调用员工信息管理模块的查询接口获取员工的基本信息，以便进行打卡验证；员工信息管理模块的修改和删除接口可由系统管理员在需要时调用。
* 考勤报表生成模块需要调用打卡模块的考勤数据接口获取原始打卡数据，并结合员工信息管理模块提供的员工信息，按照设定的报表格式和计算规则生成各类考勤报表。

## **四、数据库设计**

### **（一）数据库概念模型**

采用实体 - 关系模型（E-R 模型）等方法绘制数据库的概念模型图，展示系统中主要实体（如员工、部门、打卡记录、考勤报表等）之间的关系，例如：员工属于某个部门，员工有多次打卡记录，考勤报表是根据打卡记录和员工信息生成的等。

### **（二）数据库表结构**

根据数据库概念模型，设计具体的数据库表结构，列出各表的名称、字段名称、字段类型、是否为主键、是否可为空等信息，例如：

* ****员工表****：
  + 工号：字符型，主键，不为空
  + 姓名：字符型，不为空
  + 部门：字符型，可为空
  + 岗位：字符型，可为空
* ****打卡记录表****：
  + 打卡记录 ID：整型，主键，不为空
  + 工号：字符型，外键，指向员工表的工号字段，不为空
  + 打卡时间：日期时间型，不为空
  + 打卡地点：字符型，可为空
* ****考勤报表表****：
  + 考勤报表 ID：整型，主键，不为空
  + 报表类型：字符型，不为空
  + 生成时间：日期时间型，不为空
  + 工号：字符型，外键，指向员工表的工号字段，可为空

### **（三）数据库操作**

说明对数据库进行的主要操作，如数据的插入、查询、更新和删除操作，以及相应的 SQL 语句示例或使用的数据库操作框架（如 MyBatis、Hibernate 等），例如：

* 插入员工信息：

收起

sql

复制

INSERT INTO employees (employee\_id, name, department, position) VALUES ('001', '张三', '市场部', '销售代表');

* 查询某员工的打卡记录：

收起

sql

复制

SELECT \* FROM punch\_records WHERE employee\_id = '001';

## **五、性能设计**

### **（一）性能指标**

确定系统的主要性能指标，如响应时间（用户发起打卡操作到系统反馈结果的时间）、吞吐量（单位时间内系统能够处理的打卡请求数量）、并发处理能力（系统能够同时处理的并发打卡请求数量）等，并明确各项性能指标的具体要求，例如：响应时间应控制在 3 秒以内，吞吐量应满足每小时处理 1000 次打卡请求，并发处理能力应达到 100 个并发打卡请求。

### **（二）性能优化措施**

针对上述性能指标，提出相应的性能优化措施，如采用缓存技术（如在服务器端缓存员工信息、打卡规则等常用数据，以减少数据库查询次数）、数据库优化（如索引优化、查询语句优化等）、分布式系统设计（如采用分布式数据库、分布式应用服务器等，以提高系统的并发处理能力）等。

## **六、安全设计**

### **（一）安全威胁分析**

分析系统可能面临的安全威胁，如数据泄露（员工考勤数据、个人信息等被非法获取）、非法访问（未经授权的用户试图访问系统资源）、数据篡改（打卡记录、员工信息等被恶意修改）等。

### **（二）安全措施**

针对上述安全威胁，采取相应的安全措施，如用户认证和授权（采用用户名和密码、指纹识别、面部识别等方式对用户进行身份认证，并根据用户角色分配相应的权限）、数据加密（对员工考勤数据、个人信息等进行加密处理，防止数据泄露）、访问控制（设置系统资源的访问权限，限制未经授权的用户进入系统或访问特定区域）、日志记录（对系统的重要操作（如打卡操作、系统设置修改等）进行详细的日志记录，以便在发生安全问题时进行追溯和调查）等。

## **七、系统部署**

### **（一）部署方案**

描述系统的部署方案，包括服务器的部署位置（本地机房、云计算平台等）、服务器的数量及配置、终端设备的安装和配置要求等，例如：

* 服务器可部署在企业内部的本地机房，选用两台高性能服务器，一台作为应用服务器，负责处理用户的打卡请求和业务逻辑运算；另一台作为数据库服务器，负责存储考勤数据和员工信息。
* 终端设备如智能手机，需从官方应用商店下载安装上班打卡系统的 APP，并按照提示进行注册和登录操作。

### **（二）部署流程**

详细说明系统的部署流程，包括服务器的安装和调试步骤、终端设备的安装和调试步骤、系统的初始化步骤（如导入初始员工信息、设置打卡规则等）等，以便于系统运维人员进行系统部署操作。

## **八、系统维护**

### **（一）维护内容**

列举系统维护的主要内容，如软件更新（修复软件漏洞、添加新功能等）、硬件维护（服务器硬件升级、终端设备维修等）、数据维护（数据备份、数据恢复等）、用户培训（对新员工进行系统使用培训、对管理人员进行系统管理培训等）等。

### **（一）维护流程**

描述系统维护的流程，包括问题发现（用户反馈问题、系统监控发现问题等）、问题诊断（分析问题原因、确定问题所在模块等）、问题解决（采取相应措施解决问题、如更新软件、更换硬件等）、效果验证（验证问题是否解决、系统是否恢复正常等）等，以便于系统运维人员进行系统维护操作