### “工作打卡系统”详细设计说明书

### **引言**

#### **1.1编写目的**

本详细设计说明书旨在详细描述工作打卡系统的设计细节，为开发人员提供清晰的指导，确保系统的正确实现。

#### **1.2 范围**

本系统涵盖员工上班打卡、下班打卡、请假打卡、加班打卡等功能，以及相关的打卡数据记录、查询、统计和分析功能。

#### **1.3 定义、首字母缩写词和缩略语**

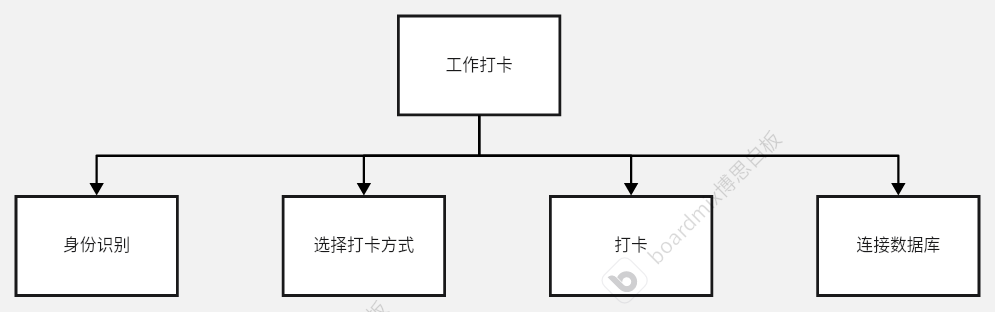
* ****打卡****：员工通过特定方式记录其工作出勤相关的时间点。
* ****NFC****：近场通信（Near Field Communication），一种可能用于打卡的技术。
* ****GPS****：全球定位系统（Global Positioning System），可用于定位打卡。

#### **1.4 参考资料**

* 《“工作打卡系统”软件需求说明书》 。
* 《“工作打卡系统”概要设计说明书》。

### **系统总体设计**

| **层次** | **模块名称** | **功能描述** |
| --- | --- | --- |
| 应用层 | 打卡模块 | 为员工提供上班、下班、请假、加班等打卡功能入口，收集打卡相关信息（如时间、类型）并传递给业务逻辑层 |
|  | 查询模块 | 接收员工或管理员的查询请求，包括打卡记录、请假和加班信息等，将请求传递给业务逻辑层，并展示查询结果 |
|  | 通知模块 | 接收来自服务器的消息（如打卡成功、请假审批结果）并向用户推送通知 |
|  | 设置模块 | 供员工设置个人信息（如密码修改）和打卡相关设置（如提醒时间） |
| 业务逻辑层 | 打卡业务逻辑 | 根据打卡类型（上班、下班、请假、加班）进行不同的逻辑处理，包括判断打卡时间是否符合规则（如迟到、早退、加班时长计算），检查请假加班与正常工作时间的冲突情况等 |
|  | 查询业务逻辑 | 根据查询条件（日期范围、打卡类型、员工部门等）生成查询指令，对从数据访问层获取的数据进行筛选、排序和统计处理 |
|  | 通知业务逻辑 | 处理消息的分类、优先级设定和推送规则，将消息内容与相应的用户进行匹配 |
|  | 设置业务逻辑 | 验证和处理员工在设置模块提交的信息，如修改密码时的密码强度验证，将更新的设置信息传递给数据访问层 |
| 数据访问层 | 员工信息管理 | 负责员工基本信息（员工编号、姓名、部门、职位等）的存储、查询、更新操作，与数据库中的员工信息表交互 |
|  | 打卡数据管理 | 将打卡记录（打卡编号、员工编号、打卡时间、打卡类型、打卡地点等）插入数据库的打卡记录表，以及查询、更新和删除相关记录 |
|  | 请假数据管理 | 对请假信息（请假编号、员工编号、请假开始时间、请假结束时间、请假类型、审批状态等）在数据库的请假信息表中进行增删改查操作 |
|  | 加班数据管理 | 处理加班信息（加班编号、员工编号、加班开始时间、加班结束时间、审批状态等）在数据库加班信息表中的存储和查询等操作 |
| 基础设施层 | 数据库连接 | 建立与数据库系统（如 MySQL、Oracle 等）的连接，管理连接的创建、关闭和复用，确保数据访问的稳定性 |
|  | 安全认证 | 实现员工登录认证机制（如用户名 / 密码验证、多因素认证），保障系统的安全性，防止非法访问 |
|  | 日志记录 | 记录系统中的关键操作（如打卡、查询、设置修改）和异常情况，用于系统监控、故障排查和审计 |
|  | 配置管理 | 存储和管理系统的配置参数（如数据库连接字符串、服务器地址、打卡规则参数等），支持参数的修改和加载 |



“上班打卡”程序层次结构

#### **2.1 系统架构**

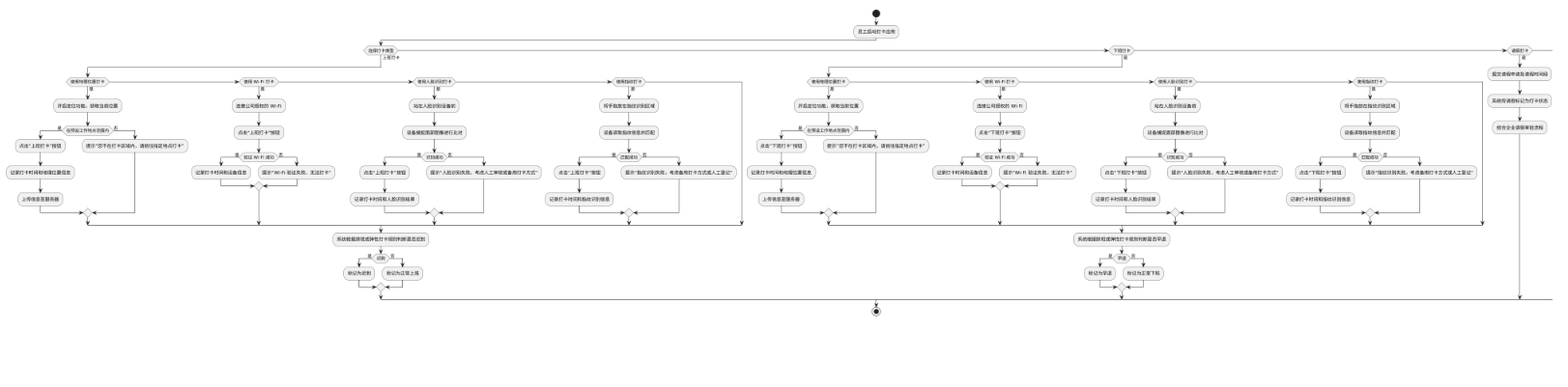
* ****客户端层****：员工使用的打卡终端，如手机 APP、打卡机等。
* ****网络层****：负责数据的传输，可采用 Wi - Fi、移动网络等。
* ****服务器层****：接收并处理打卡数据，存储数据，提供查询和统计服务。

#### **2.2 模块划分**

* ****打卡模块****：实现各种类型的打卡功能。
* ****数据存储模块****：负责将打卡数据存储到数据库。
* ****查询统计模块****：供管理员和员工查询打卡记录和统计信息。
* ****权限管理模块****：控制不同用户对系统功能的访问权限。

### **三、功能详细设计**

#### **3.1 打卡功能**



流程图

* ****上班打卡****
  + ****流程****：员工在规定上班时间附近启动打卡操作，系统获取当前时间和员工信息，判断打卡有效性（如是否在允许的迟到范围内）。
  + ****接口设计****：客户端与服务器之间的打卡请求和响应接口，包括传递员工 ID、打卡时间、打卡类型（上班）等参数。
* ****下班打卡****
  + ****类似上班打卡流程****：但需考虑早退判断逻辑。
* ****请假打卡****
  + ****流程****：员工提交请假申请，附带请假时间段，系统在该时间段内将请假标记为打卡状态，可与企业的请假审批流程相结合。
* ****加班打卡****
  + ****流程****：员工在正常下班时间后进行加班打卡，记录加班开始时间，加班结束时再次打卡，系统计算加班时长。

## **打卡方式**

1. ****地理位置打卡****：
   1. 员工使用移动设备（如手机 APP），开启定位功能后，系统自动获取员工当前所在位置信息。当员工进入预设的工作地点范围内（例如公司园区、办公大楼周边一定区域），即可点击 “打卡上班” 或 “打卡下班” 按钮进行打卡操作。系统会记录打卡时的精确地理位置坐标、打卡时间，并将这些信息上传至服务器。
   2. 若员工位置不在允许打卡的范围内，系统将提示 “您不在打卡区域内，请前往指定地点打卡”，禁止打卡操作，以防止员工异地打卡作弊。
2. ****Wi-Fi 打卡****：
   1. 公司内部部署特定的 Wi-Fi 网络，员工的设备连接到公司 Wi-Fi 后，在打卡应用中点击打卡按钮，系统会验证设备所连接的 Wi-Fi 是否为公司授权的打卡 Wi-Fi。如果验证通过，则成功打卡，并记录打卡时间和设备信息。这种方式适用于在公司内部办公区域有稳定 Wi-Fi 覆盖的场景，相比地理位置打卡更加精准可靠，且能减少因网络信号不稳定或定位偏差导致的打卡异常情况。
3. ****人脸识别打卡****：
   1. 在公司办公区域的出入口或特定打卡点设置人脸识别设备（如智能考勤终端）。员工站在设备前，设备自动捕捉员工面部图像，通过与系统中预先存储的员工人脸信息进行比对识别。识别成功后，员工可选择 “上班打卡” 或 “下班打卡”，系统记录打卡时间和人脸识别结果。人脸识别技术具有较高的准确性和安全性，能够有效防止代打卡现象。同时，对于一些特殊情况，如员工面部受伤、化妆变化较大等可能导致识别失败的情况，系统应具备人工审核或备用打卡方式的切换机制，以确保员工正常打卡考勤。
4. ****指纹打卡****：
   1. 指纹打卡设备通常安装在公司的前台、考勤区域等位置。员工将手指放置在指纹识别区域，设备读取员工指纹信息并与系统中保存的指纹模板进行匹配。匹配成功后，员工可进行打卡操作，系统记录打卡时间和指纹识别信息。指纹打卡是一种传统且较为成熟的打卡方式，具有较高的准确性，但在一些特殊情况下，如员工手指受伤、蜕皮等可能影响指纹识别效果，此时也需要有相应的解决方案，如临时使用其他打卡方式或人工登记处理。

## **打卡时间设置**

1. ****排班制打卡****：
   1. 适用于采用排班工作制的企业或部门。管理员可以在系统中为每个员工或员工组设置不同的排班计划，包括工作日、班次、上班时间和下班时间等信息。例如，设置早班为 8:00 - 16:00，中班为 12:00 - 20:00，晚班为 16:00 - 0:00 等。员工按照自己所属的排班计划进行打卡，系统会根据排班设置判断打卡时间是否合规。如果员工在排班规定的上班时间之前或下班时间之后打卡，系统将标记为迟到、早退或加班等异常情况，并在考勤报表中体现。
2. ****弹性打卡****：
   1. 企业为了给予员工一定的工作时间灵活性，可设置弹性打卡规则。管理员设定一个核心工作时间段（如 10:00 - 16:00），员工在该时间段内必须出勤达到一定时长（如 4 小时），而上班和下班的具体时间则可以在一定范围内自由选择。例如，员工可以选择在 8:00 - 9:00 之间打卡上班，在 17:00 - 18:00 之间打卡下班。系统会根据员工的实际打卡时间计算工作时长，并判断是否满足弹性打卡要求。这种打卡方式有助于提高员工的工作满意度和工作效率，同时也能保证企业的正常运营秩序。
3. ****加班打卡****：
   1. 当员工需要加班时，在正常下班打卡之后，再次进行加班打卡操作。加班打卡可以选择加班类型（如工作日加班、周末加班、法定节假日加班），系统会根据加班打卡时间和加班类型计算加班时长，并按照企业的加班工资计算规则进行相应的薪酬核算。加班打卡时间应与正常下班打卡时间有明确的区分和记录，以便在考勤统计和工资结算时能够准确处理加班数据。

## **打卡提醒功能**

1. ****上班打卡提醒****：
   1. 员工可以在打卡应用中设置上班打卡提醒功能，系统会在预设的提醒时间（如上班前 15 分钟）向员工的移动设备发送推送通知、短信或语音提醒，提醒员工及时打卡上班。提醒方式可由员工根据个人喜好和设备使用习惯进行选择和配置。这样可以有效避免员工因疏忽而忘记打卡，减少考勤异常情况的发生。
2. ****下班打卡提醒****：
   1. 同样，员工也可以设置下班打卡提醒。系统在下班时间到达时或接近下班时间（如下班前 5 分钟）向员工发送提醒，确保员工不会因工作繁忙而遗漏下班打卡。下班打卡提醒的设置有助于员工养成良好的打卡习惯，同时也能提高考勤数据的准确性和完整性。

## **打卡记录查询与统计**

1. ****员工个人打卡记录查询****：
   1. 员工可以在打卡应用或企业内部管理系统中查看自己的打卡历史记录，包括打卡日期、打卡时间、打卡地点（地理位置打卡时）、打卡方式（如地理位置、Wi-Fi、人脸识别、指纹等）以及打卡状态（正常、迟到、早退、加班等）。员工能够随时了解自己的考勤情况，以便核对工资计算的准确性或对异常打卡情况进行申诉。打卡记录查询界面应简洁明了，支持按日期范围、打卡类型等条件进行筛选和搜索，方便员工快速定位所需的打卡信息。
2. ****管理员打卡记录查询与统计****：
   1. 管理员拥有更高权限，可以查询和统计所有员工的打卡记录。系统提供丰富的查询和统计功能，管理员可以按照部门、员工组、时间段等多种维度对打卡记录进行查询和分析。例如，查询某个部门在某一周内的考勤情况，统计全体员工在某个月内的迟到次数、早退人数等。同时，系统能够生成详细的考勤报表，报表内容包括员工姓名、工号、部门、打卡日期、打卡时间、考勤状态等信息，并支持以 Excel、PDF 等格式导出报表，以便管理员进行存档、上报或进一步的数据处理。通过对打卡记录的深入分析，管理员可以及时发现考勤管理中存在的问题，如个别员工频繁迟到早退、某个部门整体考勤异常等，并采取相应的措施加以改进和优化。

## **异常打卡处理**

1. ****异常打卡提示****：
   1. 当员工打卡出现异常情况（如迟到、早退、不在打卡区域内打卡、人脸识别或指纹识别失败等）时，系统会立即向员工的移动设备发送异常打卡提示消息，告知员工打卡异常的原因和类型。同时，在员工个人打卡记录界面和管理员考勤管理界面中，异常打卡记录会以特殊颜色或标识进行突出显示，以便引起注意。
2. ****申诉与审批流程****：
   1. 员工如果认为自己的异常打卡是由于特殊原因导致的（如设备故障、临时外出办公等），可以在系统中提交打卡异常申诉。申诉时需要填写申诉原因、上传相关证明材料（如外出申请单、故障设备照片等）。申诉提交后，系统会将申诉信息推送给相应的审批人（如部门主管、人力资源经理等）。审批人收到申诉通知后，可以在系统中查看申诉详情和相关证明材料，并根据实际情况进行审批处理，审批结果包括通过、驳回或要求补充材料等。审批过程和结果会在系统中进行记录，员工和管理员均可随时查看申诉的处理进度和结果。通过建立完善的申诉与审批流程，可以保证考勤管理的公正性和人性化，避免因误判导致员工权益受损。
3. ****自动补卡与人工补卡****：
   1. 对于一些符合特定条件的轻微异常打卡情况（如因网络波动导致的打卡延迟几分钟），系统可以设置自动补卡功能。管理员可以预先定义自动补卡的规则和时间范围，当员工的打卡异常符合自动补卡条件时，系统会自动将异常打卡记录修正为正常打卡状态，无需员工手动申请。而对于一些较为复杂或不符合自动补卡条件的异常打卡情况，则需要员工通过人工补卡流程进行处理。员工在提交打卡异常申诉并获得审批通过后，系统会根据审批结果为员工进行人工补卡操作，将异常打卡记录更新为正常打卡记录，确保考勤数据的准确性和完整性。

## **打卡数据同步与备份**

1. ****实时数据同步****：
   1. 无论是员工通过移动设备打卡还是在公司内部设备上打卡，打卡数据都会实时上传至服务器，并在各个相关系统和终端之间进行同步。例如，员工在手机 APP 上打卡后，打卡数据会立即同步到企业内部的人力资源管理系统、考勤管理系统以及管理员的监控终端等，确保所有相关人员能够及时获取最新的考勤信息，实现数据的一致性和及时性。
2. ****数据备份与恢复****：
   1. 为了防止打卡数据因系统故障、硬件损坏、人为误操作或其他意外情况而丢失，系统应具备定期数据备份功能。数据备份可以采用全量备份和增量备份相结合的方式，将打卡数据备份到本地存储设备（如服务器硬盘）和异地存储介质（如云存储）中，以确保数据的安全性和可靠性。在发生数据丢失或损坏的情况下，系统管理员可以通过数据恢复机制从备份中还原打卡数据，最大限度地减少数据损失对考勤管理工作的影响。同时，系统应记录数据备份和恢复的操作日志，以便对数据管理过程进行追溯和审计。

#### **3.2 数据存储功能**

* ****数据库设计****
  + ****员工信息表****：存储员工 ID、姓名、部门等基本信息。
  + ****打卡记录表****：包含打卡 ID、员工 ID、打卡时间、打卡类型、打卡地点（如果有）等字段。
  + ****请假信息表****：记录请假申请 ID、员工 ID、请假开始时间、请假结束时间、审批状态等。
  + ****加班信息表****：存储加班申请 ID、员工 ID、加班开始时间、加班结束时间等。
* ****数据存储逻辑****：每次打卡操作成功后，将相关数据插入到打卡记录表；请假和加班申请审批通过后，更新相应表的数据。

#### **3.3 查询统计功能**

* ****员工查询****
  + ****个人打卡记录查询****：员工可查询自己的打卡历史，按日期范围、打卡类型等条件筛选。
  + ****请假和加班查询****：查看自己的请假和加班申请状态和历史记录。
* ****管理员查询****
  + ****部门打卡情况查询****：可查看某个部门或所有部门在特定时间段内的打卡情况，包括出勤人数、迟到早退人数等统计信息。
  + ****请假和加班统计****：统计各部门或员工的请假天数、加班时长等，可生成报表。

#### **3.4 权限管理功能**

* ****员工权限****：允许打卡、查询个人打卡记录、提交请假和加班申请。
* ****管理员权限****：具有所有员工的打卡记录查询权、统计权，以及审批请假和加班申请的权力。

### **四、用户界面设计**

#### **4.1 打卡界面**

* 设计简洁，有明显的打卡按钮，显示当前时间和员工姓名（如果已登录）。
* 对于不同类型的打卡（如请假、加班）有相应的入口或提示。

#### **4.2 查询界面**

* 提供搜索框和筛选条件选项，方便用户查询打卡记录。
* 以列表形式展示查询结果，包括关键信息如打卡时间、类型等。

#### **4.3 管理员界面**

* 有部门选择、时间范围选择等功能用于统计查询。
* 显示审批请假和加班申请的操作界面，可查看申请详情。

### **五、数据结构设计**

#### **5.1 主要数据结构**

* ****员工结构体****：{员工 ID，姓名，部门，职位，密码（用于登录）}。
* ****打卡记录结构体****：{打卡 ID，员工 ID，打卡时间，打卡类型，打卡地点，是否有效}。
* ****请假结构体****：{请假申请 ID，员工 ID，请假开始时间，请假结束时间，请假事由，审批状态}。
* ****加班结构体****：{加班申请 ID，员工 ID，加班开始时间，加班结束时间，加班事由，审批状态}。

### **六、算法设计**

#### **6.1 打卡有效性判断算法**

* ****迟到早退判断****：根据企业规定的上班时间和下班时间，以及允许的迟到早退时间范围，判断打卡是否有效。例如，上班打卡时间晚于上班时间但在迟到允许范围内，则标记为迟到但有效打卡。
* ****请假加班时间冲突判断****：当员工进行打卡操作时，检查是否与已批准的请假或加班时间冲突。

#### **6.2 统计算法**

* ****出勤人数统计****：遍历打卡记录表，统计在指定日期内有上班打卡记录的员工人数。
* ****迟到早退人数统计****：根据打卡有效性判断结果，统计迟到和早退人数。
* ****请假加班时长计算****：对于请假，计算请假开始时间和结束时间的差值；对于加班，计算加班结束时间和加班开始时间的差值。

### **七、性能设计**

#### **7.1 响应时间要求**

* 打卡操作在正常网络情况下应在 [X] 秒内完成响应，确保员工打卡过程的流畅性。
* 查询操作在大规模数据下（如查询一个月内所有员工的打卡记录）应在 [Y] 秒内返回结果。

#### **7.2 可扩展性**

* 系统应能方便地扩展以适应企业规模的扩大，如增加新的部门和员工。
* 能够支持新的打卡方式（如未来可能出现的新技术）的接入。

### **八、安全设计**

#### **8.1 数据安全**

* 对员工的打卡数据、个人信息等进行加密存储，防止数据泄露。
* 定期备份数据库，防止数据丢失。

#### **8.2 访问安全**

* 员工登录需要进行身份验证，采用安全的登录机制（如密码加盐等）。
* 限制不同权限用户的访问范围，防止非法访问和操作。

### **九、测试计划**

#### **9.1 单元测试**

* 对各个功能模块编写单元测试用例，如打卡模块的打卡有效性测试、数据存储模块的插入和查询功能测试等。
* 使用测试框架进行自动化单元测试。

#### **9.2 集成测试**

* 将各个模块集成后，测试整个系统的功能流程是否正常，如打卡 - 存储 - 查询的流程测试。
* 检查模块之间的接口是否正确交互。

#### **9.3 用户验收测试**

* 邀请企业员工和管理员对系统进行试用，收集反馈意见，对系统进行优化和改进，确保系统满足用户的实际需求。