****

技术方案说明书

**版本 1.1**

|  |  |
| --- | --- |
| **学院** | **软件学院** |
| **专业** | **软件工程** |
| **小组成员** | **欧若琪 宁炤宇 卓深伟朱怡超** |
| **提交日期** | **2025年 4月 21日** |

**目录**

[**1.简介** 1](#_Toc166742078)

[**1.1 内容** 1](#_Toc166742079)

[**1.1.1 项目名称** 1](#_Toc166742080)

[**1.1.2 项目文档简述** 1](#_Toc166742081)

[**1.2 适用对象** 1](#_Toc166742082)

[**1.3 参考文档** 1](#_Toc166742083)

[**2.系统整体架构** 2](#_Toc166742084)

[**2.1 系统技术方案及数据库设计** 2](#_Toc166742085)

[**2.1.1 系统技术方案** 2](#_Toc166742086)

[**2.1.2 数据库设计** 2](#_Toc166742087)

[**3.功能模块设计** 7](#_Toc166742089)

[**3.1 功能简介** 7](#_Toc166742090)

[**4.用例图** 8](#_Toc166742092)

[**4.1 总体用例图** 8](#_Toc166742093)

[**5.解决方案设计** 11](#_Toc166742096)

[**5.1 资源层解决方案** 11](#_Toc166742097)

[**5.1.1 资源层概述** 11](#_Toc166742098)

[**5.1.2 资源层详细解决方案** 11](#_Toc166742099)

[**5.2 业务层解决方案** 14](#_Toc166742100)

[**5.2.1 业务层概述** 14](#_Toc166742101)

[**5.2.2 业务规则** 14](#_Toc166742102)

[**5.2.3 业务层详细解决方案** 16](#_Toc166742104)

[**5.3 控制层解决方案** 18](#_Toc166742105)

[**5.3.1 控制层概述** 18](#_Toc166742106)

[**5.3.2 控制层详细解决方案** 21](#_Toc166742108)

[**5.4 表示层解决方案** 23](#_Toc166742109)

[**5.4.1 表示层概述** 23](#_Toc166742110)

[**5.4.2 表示层详细解决方案** 23](#_Toc166742111)

**1.简介**

**1.1 内容**

**1.1.1 项目名称**

项目名称：Sisyphus Tempus

**1.1.2 项目文档简述**

本项目主要基于打卡系统相关技术进行开发与设计，实现一款用于帮助用户实现在线打卡、管理打卡记录、查看打卡数据分析的系统——Sisyphus Tempus。该软件旨在为用户提供便捷、高效的打卡功能，同时支持人脸识别技术以提升打卡的准确性与安全性。

本文档主要对该系统的技术实现细节进行具体说明，从系统结构与功能模块设计、用例时序图设计、解决方案设计等方面进行详细的分析与设计。系统开发和测试人员可以通过本文档对系统有更深入的了解，从而为后续的相关工作（如二次开发、系统测试与维护等）提供便利和支持。

**1.2 适用对象**

本项目文档的适用对象主要有：

（1）开发人员：系统开发人员可以通过本文档提供的技术分析与说明，进行定制化开发与二次开发。

（2）测试与维护人员：系统测试与维护人员可以通过本文档了解到系统的具体架构与功能模块设计、技术栈等相关内容，从而便于他们进行系统的部署、测试与维护。

（3）普通用户：普通用户可以通过本文档了解到系统的基本功能与具体使用方法，以便其更好的使用Sisyphus Tempus

**1.3 参考文档**

本项目的技术开发主要参考了以下文档：

（1）jdk 1.8.0的安装与环境变量的配置：<https://blog.csdn.net/weixin_73320743/article/details/131166388>

（2）Vue.js 官方文档：<https://vuejs.org/v2/guide/>

（3）Maven使用教程：<https://blog.csdn.net/weixin_43888891/article/details/128622293>

（4）Mysql安装配置教程：<https://blog.csdn.net/Javachichi/article/details/132758551>

（5）Node.js安装配置教程：<https://blog.csdn.net/m0_54423289/article/details/133001799>

**2.系统整体架构**

**2.1 系统技术方案及数据库设计**

**2.1.1 系统技术方案**

（1）前端技术：

JavaScript：用于实现前端页面的交互逻辑与动态效果；

Vue.js：用于构建响应式和组件化的前端页面结构；

HTML/CSS：用于页面的基本结构搭建与样式美化。

（2）后端技术：

Java：作为主要的后端开发语言，处理业务逻辑与数据交互；

Spring Boot：用于构建高效、可扩展的后端服务框架；

Node.js：用于处理部分静态资源服务或中间层任务；

Nginx：作为反向代理服务器与前后端分离架构的协作组件。

（3）框架技术：

Vue.js：构建前端 UI 组件与页面逻辑；

Spring Boot：快速开发 RESTful 后端服务接口；

Element-UI：实现简洁高效的 UI 界面。

（4）数据库技术：

MySQL：用于存储用户信息、打卡记录、权限配置等结构化数据。

（5）人脸识别技术：

系统集成了基于 Python 和 OpenCV 的人脸识别模块，用于在打卡过程中进行用户身份验证，提升系统的安全性和便利性。通过摄像头采集人脸图像，与系统中已注册的人脸数据进行比对，实现无接触打卡和自动身份识别功能。

**2.1.2 数据库设计**

**（i）用户信息表（user\_info）**

| **字段名** | **字段类型** | **非空** | **默认值** | **注释** |
| --- | --- | --- | --- | --- |
| user\_id | INT(32) | √ | 自增 | 用户ID，主键 |
| user\_name | VARCHAR(64) | √ |  | 用户名 |
| password | VARCHAR(64) | √ |  | 用户密码 |
| email | VARCHAR(64) | √ |  | 用户邮箱 |
| phone\_number | VARCHAR(64) | √ |  | 用户手机号 |
| register\_at | DATETIME | √ | CURRENT\_TIMESTAMP | 用户注册时间 |

**（ii）打卡记录表（checkin\_record）**

| **字段名** | **字段类型** | **非空** | **默认值** | **注释** |
| --- | --- | --- | --- | --- |
| record\_id | INT(32) | √ | 自增 | 打卡记录ID，主键 |
| user\_id | INT(32) | √ |  | 用户ID，外键，指向用户表 |
| checkin\_time | DATETIME | √ | CURRENT\_TIMESTAMP | 打卡时间 |
| location | VARCHAR(128) |  |  | 打卡地点（如支持定位功能） |
| status | VARCHAR(32) | √ | 正常 | 打卡状态（如：正常、迟到） |
| face\_verified | BOOLEAN | √ | false | 是否通过人脸识别验证 |

**（iii）人脸数据表（face\_data）**

| **字段名** | **字段类型** | **非空** | **默认值** | **注释** |
| --- | --- | --- | --- | --- |
| face\_id | INT(32) | √ | 自增 | 人脸数据ID，主键 |
| user\_id | INT(32) | √ |  | 用户ID，外键，指向用户表 |
| image\_path | VARCHAR(256) | √ |  | 存储人脸图像或特征向量路径 |
| registered\_at | DATETIME | √ | CURRENT\_TIMESTAMP | 人脸信息录入时间 |

**（iv）打卡配置表（checkin\_config）**

| **字段名** | **字段类型** | **非空** | **默认值** | **注释** |
| --- | --- | --- | --- | --- |
| config\_id | INT(32) | √ | 自增 | 配置ID，主键 |
| start\_time | TIME | √ |  | 允许打卡开始时间 |
| end\_time | TIME | √ |  | 允许打卡结束时间 |
| allow\_late | BOOLEAN | √ | true | 是否允许迟到 |
| location\_range | VARCHAR(64) |  |  | 允许打卡的地点范围（可选） |

**3.功能模块设计**

**3.1 功能简介**

（1）**打卡设置与任务创建**  
Sisyphus Tempus 向管理员用户提供灵活的打卡任务创建功能。管理员可设置每日打卡的时间范围、地点要求（如有）、是否启用人脸识别验证等参数，从而满足不同组织或场景下的打卡管理需求。

（2）**打卡任务管理与修改**  
Sisyphus Tempus 支持对已创建但未生效的打卡任务进行修改，同时也允许管理员撤回正在进行中的打卡任务进行调整。管理员可以修改打卡时间、地点范围、人脸识别设置等内容，确保打卡策略的灵活性与准确性。

（3）**打卡功能**  
Sisyphus Tempus 提供简洁直观的打卡界面，用户可在指定时间范围内通过人脸识别完成打卡。系统自动记录打卡时间、位置及人脸验证结果，并支持多终端（网页、移动端）打卡操作，提升打卡便利性。

（4）**打卡记录与查询分享**  
系统为用户提供打卡记录的查询功能，用户可查看个人历史打卡数据，支持按日期筛选。管理员可对打卡记录进行统一查看与导出，便于数据整理与统计分析。同时，用户可将打卡结果截图或生成链接用于分享打卡状态（如：用于学习打卡群、晨跑记录等）。

（5）**数据统计与分析**  
Sisyphus Tempus 自动收集所有用户的打卡数据，并提供数据可视化分析功能。管理员可查看按天、按人统计的出勤率、迟到率、异常记录等分析图表，辅助进行团队出勤管理、异常行为识别和决策优化。

**4.用例图**

**4.1 总体用例图**

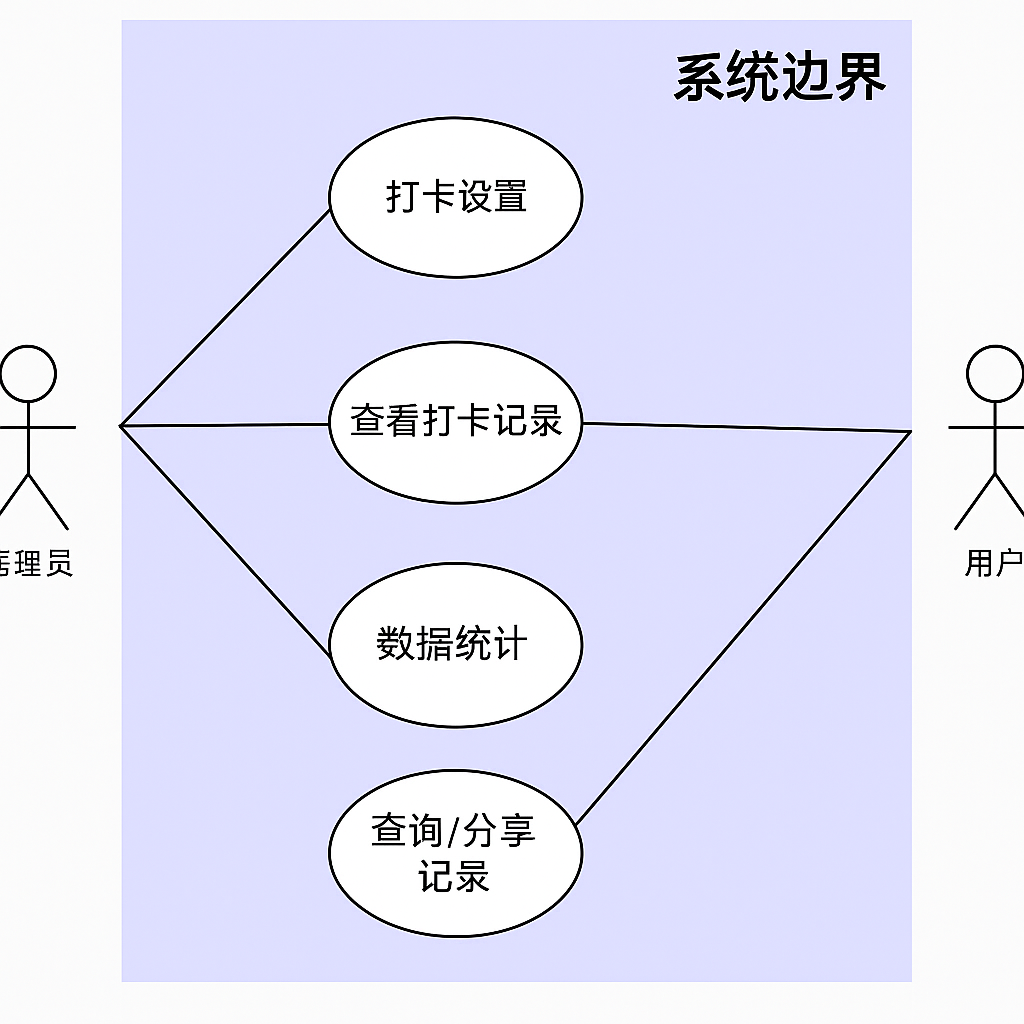


图 5 总体用例图

**5.解决方案设计**

**5.1 资源层解决方案**

**5.1.1 资源层概述**

资源层组件是指 Sisyphus Tempus 打卡系统中，专门负责与外部资源进行交互的一层，主要用于与数据库进行通信，实现系统的数据存储与访问功能。在系统的整体软件架构中，资源层扮演着关键角色，它向业务逻辑层提供统一的数据访问接口，使得业务逻辑层可以专注于处理业务规则，而不必关心数据是如何存储和获取的。

在 Sisyphus Tempus 中，数据存储是打卡功能的核心之一。我们使用关系型数据库（如 MySQL）来持久化存储用户信息、打卡记录、人脸识别结果、任务配置等数据。

**5.1.2 资源层详细解决方案**

**1. PunchRecordMapper 类**

| **方法名称** | **参数** | **返回值** | **简单描述** |
| --- | --- | --- | --- |
| addPunchRecord | @Param("uid") int uid, @Param("tid") int tid | int | 添加一条用户打卡记录 |
| updatePunchStatus | @Param("recordId") int recordId | int | 更新打卡状态（如迟到、正常等） |
| getDailyPunchStatus | @Param("uid") int uid, @Param("date") String date | List<PunchRecord> | 获取某用户某日的打卡记录 |
| deleteRecordByTid | @Param("tid") int tid | int | 删除某打卡任务的所有打卡记录 |

**2. TaskMapper 类**

| **方法名称** | **参数** | **返回值** | **简单描述** |
| --- | --- | --- | --- |
| getAllTasks | 无 | List<Task> | 获取所有打卡任务 |
| getTasksByUid | @Param("uid") int uid | List<Task> | 获取指定用户发布的打卡任务 |
| addTask | Task task | int | 新增打卡任务 |
| updateTaskInfo | @Param("tid") int tid, @Param("title") String title | int | 修改任务标题 |
| deleteTask | @Param("tid") int tid | int | 删除指定打卡任务 |
| searchTasks | @Param("keyword") String keyword | List<JSONObject> | 通过关键字搜索打卡任务 |

**3. UserMapper 类**

| **方法名称** | **参数** | **返回值** | **简单描述** |
| --- | --- | --- | --- |
| getAllUsers | @Param("username") String username | List<User> | 获取所有用户（支持模糊查询） |
| findUser | @Param("username") String username, @Param("password") String password | User | 用户登录验证 |
| findUserById | @Param("id") int id | User | 通过用户 ID 获取用户信息 |
| addUser | User newUser | int | 注册新用户 |
| getUserByPhone | @Param("phone") String phone | User | 通过手机号获取用户信息 |
| getBeforePhone | @Param("id") int id | String | 获取指定用户的原手机号 |

**4. FaceRecognitionMapper 类**

| **方法名称** | **参数** | **返回值** | **简单描述** |
| --- | --- | --- | --- |
| addFaceModel | @Param("uid") int uid, @Param("model") BLOB model | int | 添加用户人脸模型数据 |
| getFaceModel | @Param("uid") int uid | BLOB | 获取用户人脸模型数据 |
| updateFaceModel | @Param("uid") int uid, @Param("model") BLOB model | int | 更新用户人脸模型数据 |
| deleteFaceModel | @Param("uid") int uid | int | 删除用户人脸模型 |

**5. StatisticsMapper 类**

| **方法名称** | **参数** | **返回值** | **简单描述** |
| --- | --- | --- | --- |
| getDailyStats | @Param("tid") int tid | JSONObject | 获取某打卡任务的日打卡统计数据 |
| getUserStats | @Param("uid") int uid | JSONObject | 获取某用户所有打卡任务的统计数据 |
| getOverallStats | 无 | List<JSONObject> | 获取系统整体打卡统计（如出勤率） |

**5.2 业务层解决方案**

**5.2.1 业务层概述**

业务层是系统的核心，负责实现系统的业务逻辑和功能。业务层包含了系统的业务规则、算法等核心功能。在业务层中可定义业务对象、服务和操作，以实现系统的业务目标。

**5.2.2 业务规则**

（1）**打卡任务创建与管理**：

* 管理员权限用户可创建、编辑和删除打卡任务；
* 管理员可设置任务类型（如日常打卡、签到、活动打卡等）、打卡时间范围、打卡地点或人脸识别要求等；
* 管理员可管理打卡任务的状态，如“草稿”、“已发布”、“已结束”等，灵活控制任务生命周期。

（2）**打卡任务发布与执行**：

* 管理员用户可一键发布打卡任务，并生成任务链接或二维码，便于用户访问；
* 普通用户可通过链接或系统入口，进入打卡页面，进行打卡操作（如定位打卡、人脸识别打卡、拍照上传等）；
* 系统支持设置打卡频次与提醒机制，如每日固定时间提醒打卡。

（3）**打卡数据收集与存储**：

* 系统自动收集并存储用户提交的打卡数据，包括时间、打卡方式、设备信息、定位信息等；
* 打卡数据结构化存储于数据库中，确保数据的完整性和可追溯性，为后续分析提供基础。

（4）**打卡数据统计与分析**：

* 系统提供丰富的数据统计与分析功能，管理员可查看某任务的打卡率、异常打卡情况、迟到/早退等指标；
* 支持生成数据统计报表与可视化图表，如出勤曲线、每日打卡人数分布图等；
* 分析结果支持导出为 Excel、PDF 等格式，便于汇报和归档。

（5）**用户管理与权限控制**：

* 系统支持多级用户权限机制，包括管理员、普通用户、访客等角色；
* 不同角色用户拥有不同操作权限，如任务发布、数据查看、打卡等；
* 系统实现用户认证与授权机制，确保用户只能访问其权限范围内的打卡任务与数据。

**5.2.3 业务层详细解决方案**

**1. ClockInTaskService**

| **方法名称** | **参数** | **返回值** | **简单描述** |
| --- | --- | --- | --- |
| getAllTasks | void | List<ClockInTask> | 获取所有打卡任务信息 |
| getTasksByAdminId | adminId: int | List<ClockInTask> | 获取某位管理员创建的所有打卡任务 |
| getTasksByKeyword | keyword: String | List<ClockInTask> | 通过关键字模糊搜索打卡任务 |
| addTask | task: ClockInTask | int | 添加一个新的打卡任务 |
| deleteTaskById | tid: int | int | 删除指定 ID 的打卡任务 |
| updateTaskTitle | tid: int, title: String | int | 更新打卡任务标题 |
| setTaskStatus | tid: int, status: int | int | 设置打卡任务状态（如草稿、已发布、已结束） |
| getTaskInfo | void | List<JSONObject> | 获取打卡任务信息（json 格式） |

**2. UserService**

| **方法名称** | **参数** | **返回值** | **简单描述** |
| --- | --- | --- | --- |
| findUserByUsername | username: String | User | 根据用户名查找用户 |
| findUserById | id: int | User | 根据用户 ID 查找用户 |
| addUser | user: User | int | 添加一个新用户 |
| deleteUser | username: String | void | 删除指定用户名的用户 |
| updateUsername | id: int, username: String | int | 更新指定用户的用户名 |
| updatePassword | username: String, pwd: String | void | 设置/修改用户密码 |

**3. CheckInOptionService**

| **方法名称** | **参数** | **返回值** | **简单描述** |
| --- | --- | --- | --- |
| getOptionsByTid | tid: int | List<Option> | 获取指定打卡任务的打卡选项 |
| deleteOptionsByTid | tid: int | int | 删除指定打卡任务的所有打卡选项 |
| addOption | value: String, tid: int | int | 添加打卡选项 |
| getOptionById | oid: int | String | 获取指定选项内容 |
| updateOption | option: String, oid: int | int | 更新指定打卡选项 |
| getOptionsJson | tid: int | List<JSONObject> | 获取指定打卡任务的选项信息（json 格式） |

**4. TaskCheckInService**

| **方法名称** | **参数** | **返回值** | **简单描述** |
| --- | --- | --- | --- |
| submitCheckIn | data: CheckInRecord | int | 提交打卡记录 |
| deleteCheckInByTid | tid: int | void | 删除指定任务下的所有打卡记录 |
| deleteCheckInByOid | oid: int | void | 删除与某一选项相关的打卡记录 |
| getCheckInByTid | tid: int | List<JSONObject> | 获取某任务的所有打卡记录（json 格式） |

**5. DataAnalysisService**

| **方法名称** | **参数** | **返回值** | **简单描述** |
| --- | --- | --- | --- |
| getCheckInStatsByTid | tid: int | JSONObject | 获取某任务的打卡数据统计结果 |
| getDailyCheckInRate | tid: int | List<JSONObject> | 获取某任务每日打卡率信息 |
| exportTaskReport | tid: int, format: String | File | 导出打卡数据报表（Excel、PDF 等） |

**5.3 控制层解决方案**

**5.3.1 控制层概述**

控制层负责处理外部请求，并将其转发给业务层进行处理。控制层在问卷星项目中起到承前启后的作用，负责接收和处理客户端请求，调用相应的业务逻辑，并返回处理结果给业务层。

**5.3.2 控制层详细解决方案**

**1. TaskCtrl（打卡任务控制器）**

**类变量定义（Public → protected → private）：**

| **名称** | **类型** | **描述** |
| --- | --- | --- |
| taskService | ClockInTaskService | 打卡任务服务接口 |
| optionService | CheckInOptionService | 打卡选项服务接口 |
| checkInService | TaskCheckInService | 打卡记录服务接口 |
| userService | UserService | 用户服务接口 |

**方法：**

| **名称** | **参数** | **返回值** | **简单描述** |
| --- | --- | --- | --- |
| addTaskAndOptions | data: JSONObject | task: JSONObject | 添加新的打卡任务及选项信息 |
| editTask | data: JSONObject | task: JSONObject | 编辑打卡任务内容 |
| deleteTask | data: JSONObject | taskList: JSONObject | 删除打卡任务，返回剩余任务列表 |

**2. CheckInCtrl（打卡记录控制器）**

**类变量定义：**

| **名称** | **类型** | **描述** |
| --- | --- | --- |
| taskService | ClockInTaskService | 打卡任务服务接口 |
| optionService | CheckInOptionService | 打卡选项服务接口 |
| checkInService | TaskCheckInService | 打卡记录服务接口 |
| userService | UserService | 用户服务接口 |
| analysisService | DataAnalysisService | 数据统计分析服务接口 |

**方法：**

| **名称** | **参数** | **返回值** | **简单描述** |
| --- | --- | --- | --- |
| submitCheckIn | data: JSONObject | record: JSONObject | 用户提交打卡记录 |
| getCheckInByTask | taskId: JSONObject | recordList: JSONObject | 获取某个任务下的打卡记录列表 |

**3. TaskManageCtrl（打卡任务管理控制器）**

| **名称** | **类型** | **描述** |
| --- | --- | --- |
| taskService | ClockInTaskService | 打卡任务服务接口 |
| optionService | CheckInOptionService | 打卡选项服务接口 |
| checkInService | TaskCheckInService | 打卡记录服务接口 |

**方法：**

| **名称** | **参数** | **返回值** | **简单描述** |
| --- | --- | --- | --- |
| getAllTasks | adminId: JSONObject | taskList: JSONObject | 加载所有打卡任务（管理员视图） |
| searchTasks | keyword: String | taskList: JSONObject | 根据关键词搜索任务 |
| getTasksByUserId | userId: JSONObject | taskList: JSONObject | 加载当前用户的所有任务 |
| getTaskPage | task: JSONObject | task: JSONObject | 获取任务详细内容 |
| addTaskInfo | task: JSONObject | task: JSONObject | 添加打卡任务信息 |
| getTaskTitle | task: JSONObject | title: JSONObject | 获取任务标题 |
| editTaskTitle | data: JSONObject | task: JSONObject | 修改任务标题 |
| editPublishStatus | task: JSONObject | task: JSONObject | 修改任务发布/未发布状态 |
| deleteTaskInfo | data: JSONObject | taskList: JSONObject | 删除打卡任务并返回剩余任务列表 |

**4. UserCtrl（用户控制器）**

| **名称** | **参数** | **返回值** | **简单描述** |
| --- | --- | --- | --- |
| getUserList | 无 | userList: JSONObject | 获取系统所有用户信息 |
| login | loginUser: User | res: JSONObject | 用户登录验证 |
| register | registerUser: User | res: JSONObject | 用户注册 |
| getUsername | user: User | name: JSONObject | 获取当前用户的用户名 |
| getUserInfo | user: User | user: JSONObject | 获取当前用户的完整信息 |
| resetPassword | user: User | user: JSONObject | 重置密码 |
| setPassword | user: User | user: JSONObject | 修改密码 |
| setEmail | user: User | user: JSONObject | 设置绑定邮箱 |
| deleteUser | user: User | userList: JSONObject | 删除指定用户 |
| saveUser | data: JSONObject | user: User | 创建/修改用户信息 |

**5.4 表示层解决方案**

**5.4.1 表示层概述**

Sisyphus Tempus 打卡系统的表示层（前端）旨在为用户提供**直观、高效、易用且美观的打卡界面**，使用户能够便捷地进行日常打卡、查看任务状态和历史记录。我们采用现代前端技术（如 **HTML5、CSS3、JavaScript、Vue.js**）与后端技术（如 **Java、Node.js、Nginx**）协同构建系统，确保表示层与后端各层（控制层、服务层、数据层等）顺畅交互。

**表示层设计目标与核心需求：**

1. **响应性（Responsiveness）**
   * 系统界面需快速响应用户操作，支持异步加载与数据缓存机制，避免因请求延迟导致的等待时间过长，提升用户打卡流畅度。
   * 支持移动端和PC端的自适应布局，确保在多种设备上都有良好的交互体验。
2. **安全性（Security）**
   * 所有用户输入内容（如打卡备注、补卡申请）需进行前端校验，防止非法输入。
   * 实现基础的数据加密与登录令牌机制，保护用户隐私与账号安全。
3. **易用性（Usability）**
   * 用户界面设计注重简洁与直观，用户可快速定位“打卡”、“查看历史”、“个人信息”等功能入口。
   * 对管理员用户提供图形化操作界面，用于管理任务、查看数据报表等，无需繁琐命令。
4. **一致性（Consistency）**
   * 全系统界面风格统一，按钮、表单、提示框等组件样式保持一致，减少用户认知负担。
   * 所有交互反馈（如打卡成功/失败、提交中、加载中等）均使用一致的提示方式和颜色标识。
5. **灵活性（Flexibility）**
   * 系统支持响应式设计，兼容不同分辨率与平台（PC端、平板、手机等）。
   * 用户可自定义打卡提醒方式（如每日弹窗、消息提醒、邮件通知等）。

**表示层与系统其他层的交互：**

表示层不仅负责与用户交互，还通过 **HTTP 请求或前后端接口规范** 与系统的 **控制层、服务层、资源层** 等后端组件紧密协作，完成如下关键任务：

* **任务展示与打卡操作**：用户可查看已分配任务，并提交打卡记录至后端数据库；
* **数据统计可视化**：通过与统计模块交互，用户可查看打卡趋势图、个人/团队出勤图表；
* **权限管理反馈**：根据用户登录身份，展示对应功能按钮（如管理员可见“任务发布”功能）；
* **历史记录展示与搜索**：用户可快速筛选历史打卡记录，支持关键词、日期、状态等过滤。

**5.4.2 表示层详细解决方案**

| **候选的关键抽取** | **选定的名字** | **功能描述** |
| --- | --- | --- |
| 用户 | User | 存储用户的个人信息，支持用户注册、登录、权限管理与账号注销等功能。 |
| 打卡任务 | CheckTask | 打卡活动的载体，定义打卡时间、打卡要求、可参与用户、打卡状态（进行中、已完成）等。 |
| 打卡记录 | CheckRecord | 用户提交的每次打卡信息记录，包括打卡时间、任务ID、用户ID、打卡内容（如照片、备注）等。 |
| 日常打卡 | DailyCheck | 固定周期（如每天）进行的常规打卡任务，适用于日常出勤、签到等场景。 |
| 自定义打卡 | CustomCheck | 用户或管理员发起的个性化打卡任务，可自由配置时间段、打卡形式（如打卡照片、签到码等）。 |
| 补卡申请 | RecheckRequest | 用户在错过打卡时提交的补卡请求，需由管理员进行审批。 |
| 打卡统计 | CheckStatistic | 对用户或任务的打卡数据进行统计，如打卡率、补卡次数、连续打卡天数等，并生成图表报表。 |
| 系统管理后台 | ManageSystem | 管理整个打卡系统的界面，提供任务发布、用户管理、数据查看与权限配置等功能。 |
| 打卡内容 | CheckContent | 打卡记录中的实际内容，如打卡位置、打卡备注、上传图片、评分等，支持多种内容格式组合。 |