概要设计说明书

1. 引言

1.1 编写目的

本概要设计说明书旨在提供访客闭环管理系统的高层次设计细节，为开发和测试团队提供清晰的指导，确保系统的高效开发和可靠运行。

1.2 项目背景

访客闭环管理系统是一个面向企业访客管理解决方案。其主要目标是提高访客管理的效率，增加对访客活动的可追溯性，强化对访客的安全管理。

1.3 定义

访客闭环管理系统（VAMS）: VAMS 是一个综合管理系统，涵盖了普通访客、VIP 访客、物流司机（长期/临时）、施工人员等多种访客类型的全生命周期管理。
KAMS（车辆识别系统）: KAMS 用于识别车辆信息，与 VAMS 实现无缝集成，确保访客与车辆信息同步。

1.4 参考资料

墨刀相关文章：https://modao.cc/proto/design/pb2ln7j0e6yh72rn2
https://modao.cc/community/mtlcr0etybzu9bv7
https://blog.csdn.net/Ateasedodo/article/details/119852488

2. 任务概述

2.1 目标

访客闭环管理系统的总体目标是提供一个集成、高效、安全的访客管理平台，实现以下具体目标：

实现访客的预约、到访、离开等全生命周期管理。
区分普通访客、VIP 访客、物流司机和施工人员，提供相应的功能。
与 KAMS 实现数据交互，确保车辆信息与访客信息的同步。

2.2 运行环境

访客闭环管理系统将在以下环境中运行：

操作系统：Microsoft Windows10
数据库：Navicat for MySQL2023

2.3 需求概述

普通访客
预约：提供在线预约功能，填写姓名、公司、手机号、访问部门、车牌号等信息。
审批：被访人通过 PC 端或手机小程序审批预约，生成二维码。
到访：访客通过扫描二维码进入。
访问结束：被访人通过 PC 端或小程序点击“访问结束”。
离开：访客通过扫描二维码确认离开。
VIP 访客
预约：被访人预约，仅填写车牌号和来访单位。
审批：审批流程包括申请部门、人事总务部、总经理。
离开：被访人在 PC 端或小程序点击“访问结束”。
物流司机（长期）
登记：司机登记车牌号、姓名、手机号，成功后自动导入到 KAMS 系统。
装卸货：KAMS 人员登录系统，登记货物信息并拍照，部门班长审核后放行。
离厂：保安核对信息后点击放行，KAMS 抬杆放行。
物流司机（临时）
预约：提前预约，填写公司、姓名、手机号、车牌号、当天计划进出次数。
预约验证：扫描预约二维码，系统核验后生成访客单，成功后可正常进出一次。
装卸货：同物流司机（长期）。
离厂：保安收取访客单，核对信息后点击放行，KAMS 抬杆放行。
施工人员
预约：单人填写姓名、公司、手机号、身份证号、车牌号、拜访部门、被访人。
审批：部门审批至课长或相应部门领导，成功后生成二维码。
入厂：扫描二维码进入。
离厂：刷二维码确认离厂。

2.4 条件与限制

项目团队需具备相关技术和经验。
用户需在规定时间前提供系统需求和访客信息。
KAMS 系统的接口必须保持稳定。

3. 总体设计

3.1 处理流程

系统的处理流程如下：

访客发起预约请求。
被访人审批预约请求，生成二维码。
访客到访，扫描二维码进入系统。
访问结束，被访人确认。
访客离开，扫描二维码确认。

3.2 总体结构和模块外部设计

前端用户界面：提供各种访客类型的预约、审批、记录查看等功能。
后端业务逻辑：处理预约审批、访客验证、审批流程等业务逻辑。
数据库：存储访客信息、审批记录、访问日志等。

3.3 功能分配

预约模块：处理访客的预约请求。
审批模块：处理审批流程，生成审批结果。
验证模块：负责验证访客身份，生成访问凭证。
记录模块：记录访客的访问历史，包括到访、访问结束和离开的时间点。

4. 接口设计

4.1 外部接口

与 KAMS 的数据交互接口：确保车辆信息与访客信息同步。

4.2 内部接口

系统内部各模块之间通过定义的接口进行数据交互，确保数据一致性。

5. 数据结构设计

5.1 逻辑结构设计

访客信息表：存储访客的基本信息。
审批记录表：记录审批流程的详细信息。
访问日志表：记录访客的访问历史。

5.2 物理结构设计

逻辑结构映射到数据库中，确保数据的持久性和安全性。
访客信息表（Visitors）
字段：
VisitorID (int, 主键)
Name (nvarchar(50))
Company (nvarchar(100))
PhoneNumber (nvarchar(20))
PlateNumber (nvarchar(15))
Department (nvarchar(100))
VisitedPerson (nvarchar(50))
AppointmentTime (datetime)
ApprovalStatus (nvarchar(20))
VisitStatus (nvarchar(20))
审批记录表（ApprovalRecords）
字段：
RecordID (int, 主键)
VisitorID (int, 外键关联 Visitors 表)
Approver (nvarchar(50))
ApprovalTime (datetime)
ApprovalResult (nvarchar(20))
ApprovalComments (nvarchar(max))
访问日志表（VisitLogs）
字段：
LogID (int, 主键)
VisitorID (int, 外键关联 Visitors 表)
EntryTime (datetime)
EndTime (datetime)
LeaveTime (datetime)
VisitStatus (nvarchar(20))

5.3 数据结构与程序的关系

程序模块：前端用户界面、后端业务逻辑。
关系：通过 ORM（对象关系映射）工具，如 Entity Framework，将程序中的对象与数据库中的表进行映射，确保数据的正确存储和访问。

6. 运行设计

6.1 运行模块的组合

前端用户界面
模块：
预约模块
审批模块
访问验证模块
记录查看模块
后端业务逻辑
模块：
预约处理模块
审批处理模块
验证处理模块
记录处理模块
数据库模块
模块：
数据存储模块
数据查询模块
数据更新模块

6.2 运行控制

运行时参数配置
参数：
数据库连接字符串
访客预约时间限制
审批流程配置
访客验证规则配置
错误处理
策略：
记录错误日志
提供用户友好的错误提示

自动恢复机制，尽可能保障系统的稳定性

6.3 运行时间

高峰期处理
策略：
提前预约系统资源
数据库优化以提高查询速度
负载均衡策略，确保系统稳定运行
低谷期处理
策略：
执行数据库备份和维护任务
系统更新和补丁安装

7. 出错处理设计

7.1 出错输出信息

系统在出错时向用户和管理员输出详细信息，以便及时发现和解决问题。

用户输出：错误提示信息显示在用户界面，引导用户操作或提供联系方式。
管理员输出：错误日志记录在系统后台，包括错误信息、发生时间、用户信息等。

7.2 出错处理对策

自动恢复机制
策略：监测系统状态，自动尝试恢复发生错误的模块或功能，减少人工干预。
报警通知
策略：通过邮件、短信等方式及时通知管理员，使其能够迅速响应并处理问题。

8. 安全保密设计

8.1 身份验证

访客身份验证
方法：使用用户名密码、短信验证码等方式确保访客身份的真实性。
系统管理员身份验证
方法：使用双因素认证等高级方式，确保系统管理员的身份安全。

8.2 数据加密

数据传输加密
方法：使用 SSL/TLS 等协议对数据进行加密，防止数据在传输过程中被窃取。
数据存储加密
方法：对敏感数据采用加密算法存储在数据库中，增加数据安全性。

9.维护设计

9.1 系统更新

更新策略
策略：定期发布系统更新，包括功能增强、性能优化、Bug 修复等。
更新流程
流程：系统管理员通过后台界面或自动更新工具执行更新，确保更新流程简单可控。

9.2 Bug 修复

Bug 追踪
工具：使用 Bug 追踪系统，记录和跟踪 Bug 的处理过程。
定期维护
计划：定期进行系统维护，包括清理无用数据、优化数据库等，确保系统稳定运行。

9.3 数据库维护

数据库备份
策略：定期进行数据库备份，确保系统数据的安全性。
数据库优化
策略：根据数据库查询性能，进行索引优化等操作，提高数据库查询速度。