# **网吧管理系统软件工程文档**

## **一、引言**

### **1.1 目的**

本软件工程文档旨在详细描述网吧管理系统的设计、开发、测试和维护过程，为开发团队和相关利益者提供清晰的指导，确保系统满足网吧日常管理的需求。

### **1.2 项目背景**

随着网吧行业的发展，传统的手工管理方式难以满足高效运营的要求。本网吧管理系统旨在实现网吧的自动化管理，包括顾客信息管理、计算机资源管理、计费管理等功能，提高网吧的管理效率和服务质量。

## **二、需求分析**

### **2.1 功能需求**

1. ****顾客管理****
   1. 顾客注册：记录顾客基本信息，如姓名、身份证号、联系方式等。
   2. 顾客登录：顾客使用注册信息登录系统。
   3. 顾客信息查询与修改：管理员可查询和修改顾客信息。
2. ****计算机资源管理****
   1. 计算机状态监控：实时显示计算机的使用状态（空闲、使用中、故障）。
   2. 计算机分配与释放：为顾客分配空闲计算机，并在顾客下机时释放。
3. ****计费管理****
   1. 计时计费：根据顾客上机时长计算费用。
   2. 收费记录：记录每次收费的详细信息，包括顾客信息、上机时间、下机时间、费用等。
   3. 价格设置：管理员可设置不同时段的收费标准。
4. ****商品销售管理（可选）****
   1. 商品库存管理：记录网吧内商品的库存信息。
   2. 商品销售记录：记录顾客购买商品的信息，包括商品名称、价格、数量等。

### **2.2 性能需求**

1. ****响应时间****：系统应在短时间内响应顾客的操作请求和管理员的管理请求，一般操作响应时间不超过 3 秒。
2. ****并发处理能力****：能够支持多个顾客同时登录和使用系统，至少支持 50 个并发操作。

### **2.3 安全性需求**

1. ****数据保护****：对顾客的个人信息和网吧的经营数据进行加密存储，防止数据泄露。
2. ****权限管理****：区分管理员和普通顾客的权限，管理员具有系统的全部管理权限，顾客只能进行与自身相关的操作。

## **三、系统设计**

### **3.1 总体架构设计**

网吧管理系统采用三层架构，包括表示层、业务逻辑层和数据访问层。

1. ****表示层****：负责与用户进行交互，提供用户界面，接收用户输入并显示系统输出。可以采用桌面应用程序（如使用 JavaFX、WinForms 等）或 Web 应用程序（如使用 HTML、CSS、JavaScript 等）的形式。
2. ****业务逻辑层****：处理系统的核心业务逻辑，包括顾客管理、计算机资源管理、计费管理等功能。它接收表示层的请求，调用数据访问层的接口进行数据操作，并将处理结果返回给表示层。
3. ****数据访问层****：负责与数据库进行交互，实现数据的存储、查询、更新和删除操作。可以使用关系型数据库（如 MySQL、SQL Server 等）或非关系型数据库（如 Redis、MongoDB 等）。

### **3.2 模块设计**

1. ****顾客管理模块****
   1. 顾客注册子模块：设计用户界面，获取顾客输入的信息，进行合法性验证后，将信息传递给业务逻辑层。业务逻辑层调用数据访问层将顾客信息存储到数据库中。
   2. 顾客登录子模块：提供登录界面，接收顾客输入的用户名和密码，与数据库中的数据进行比对，验证通过后登录系统。
   3. 顾客信息查询与修改子模块：管理员通过特定界面查询顾客信息，可对顾客信息进行修改操作，修改操作通过业务逻辑层和数据访问层更新数据库中的数据。
2. ****计算机资源管理模块****
   1. 计算机状态监控子模块：通过定时查询计算机的连接状态或使用专门的网吧管理软件接口获取计算机的使用状态，将状态信息存储在数据库中，并在表示层实时显示。
   2. 计算机分配与释放子模块：根据计算机的空闲状态，为顾客分配计算机，并在顾客下机时释放计算机资源。分配和释放操作通过更新数据库中的计算机状态信息实现。
3. ****计费管理模块****
   1. 计时计费子模块：在顾客登录计算机时开始计时，下机时结束计时，根据设定的价格标准计算费用。
   2. 收费记录子模块：将每次顾客的上机、下机时间和费用等信息记录到数据库中，形成收费记录。
   3. 价格设置子模块：提供管理员界面，用于设置不同时段（如高峰时段、低谷时段）的收费标准，设置信息存储在数据库中。
4. ****商品销售管理模块（可选）****
   1. 商品库存管理子模块：记录商品的进货、销售和库存数量，通过数据库操作实现库存信息的更新。
   2. 商品销售记录子模块：在顾客购买商品时，记录商品销售信息，包括商品名称、价格、数量等，存储到数据库中。

### **3.3 数据库设计**

1. ****顾客表（Customer）****
   1. 顾客 ID（CustomerID，主键）
   2. 姓名（Name）
   3. 身份证号（IDCardNumber）
   4. 联系方式（ContactInfo）
   5. 用户名（Username）
   6. 密码（Password）
2. ****计算机表（Computer）****
   1. 计算机编号（ComputerNumber，主键）
   2. 状态（Status，如空闲、使用中、故障）
3. ****计费表（Billing）****
   1. 计费 ID（BillingID，主键）
   2. 顾客 ID（CustomerID，外键，关联 Customer 表）
   3. 上机时间（StartTime）
   4. 下机时间（EndTime）
   5. 费用（Cost）
4. ****价格表（Price）****
   1. 价格 ID（PriceID，主键）
   2. 时段（TimePeriod，如高峰时段、低谷时段）
   3. 单价（UnitPrice）
5. ****商品表（Goods）****（可选）
   1. 商品 ID（GoodsID，主键）
   2. 商品名称（GoodsName）
   3. 库存数量（InventoryQuantity）
   4. 进货价格（PurchasePrice）
   5. 销售价格（SalesPrice）
6. ****商品销售记录表（GoodsSales）****（可选）
   1. 销售记录 ID（SalesRecordID，主键）
   2. 商品 ID（GoodsID，外键，关联 Goods 表）
   3. 销售数量（SalesQuantity）
   4. 销售时间（SalesTime）

## **四、系统实现**

### **4.1 开发环境与技术选型**

1. ****开发环境****：根据系统设计选择合适的开发环境，如 Visual Studio（用于.NET 开发）、Eclipse 或 IntelliJ IDEA（用于 Java 开发）等。
2. ****技术选型****
   1. ****编程语言****：可以选择 Java、C++、Python 等。
   2. ****数据库****：根据需求选择 MySQL、SQL Server、Oracle 等关系型数据库或 Redis、MongoDB 等非关系型数据库。
   3. ****用户界面技术****：如果是桌面应用程序，可以选择 JavaFX、WinForms、WPF 等；如果是 Web 应用程序，可以选择 HTML、CSS、JavaScript 等前端技术和 Spring Boot、Django 等后端框架。

### **4.2 代码实现**

1. ****按照模块设计分别实现各个功能模块的代码。****
   1. 在顾客管理模块中，实现顾客注册、登录、信息查询与修改的功能代码，确保输入验证、数据库操作等功能的正确性。
   2. 在计算机资源管理模块中，编写计算机状态监控和分配与释放的代码，与底层的计算机管理接口或网络通信机制相结合。
   3. 在计费管理模块中，实现计时计费、收费记录和价格设置的代码，处理时间计算和数据库记录的相关逻辑。
   4. 在商品销售管理模块（如果有）中，编写商品库存管理和销售记录的代码，保证库存更新和销售数据记录的准确性。
2. ****实现各个模块之间的接口调用，确保系统的整体性和连贯性。****

## **五、系统测试**

### **5.1 测试计划**

1. ****制定详细的测试计划，包括测试目标、测试范围、测试方法、测试资源和测试进度安排。****
2. ****确定测试类型，如功能测试、性能测试、安全性测试等。****

### **5.2 功能测试**

1. ****对系统的各个功能模块进行详细的功能测试，确保每个功能都能正常运行。****
   1. 测试顾客注册、登录、信息查询与修改功能，检查输入各种合法和非法数据时系统的反应。
   2. 测试计算机资源管理功能，包括计算机状态监控的准确性、计算机分配与释放的正确性。
   3. 测试计费管理功能，检查计时计费的准确性、收费记录的完整性和价格设置的有效性。
   4. 测试商品销售管理功能（如果有），验证商品库存管理和销售记录的正确性。
2. ****使用测试用例来覆盖各种功能场景，记录测试结果并及时修复发现的问题。****

### **5.3 性能测试**

1. ****使用性能测试工具（如 JMeter、LoadRunner 等）对系统进行性能测试。****
   1. 模拟多个用户同时登录、操作计算机和进行计费等操作，测试系统的响应时间和并发处理能力。
   2. 根据性能需求，调整系统参数和优化代码，确保系统满足性能要求。

### **5.4 安全性测试**

1. ****对系统进行安全性测试，检查数据加密和权限管理的有效性。****
   1. 尝试对数据库中的数据进行非法访问，检查加密机制是否能有效保护数据。
   2. 测试不同权限用户的操作限制，确保权限管理的正确性。

## **六、系统部署与维护**

### **6.1 系统部署**

1. ****根据网吧的实际网络环境和硬件设施，选择合适的部署方式。****
   1. 如果是桌面应用程序，在网吧的每台管理终端和客户机上安装相应的程序，并配置好数据库连接等参数。
   2. 如果是 Web 应用程序，将服务器端程序部署在网吧的服务器上，配置好 Web 服务器（如 Apache、Nginx 等）和数据库服务器，确保网络畅通，客户机可以通过浏览器访问系统。
2. ****对部署后的系统进行全面测试，确保系统在实际环境中正常运行。****

### **6.2 系统维护**

1. ****建立系统维护计划，包括定期备份数据、更新系统软件、修复漏洞等。****
2. ****设立技术支持渠道，及时处理用户在使用过程中遇到的问题。****
3. 最后根据网吧的业务发展和用户反馈，对系统进行功能升级和优化。