<景区预约管理模拟软件>

补充规约

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 2024/9/7 | <1.0> | <景区预约管理模拟软件补充规约> | <王笑然> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[1.       简介](#jj)

[1.1     目的](#md)

[1.2     范围](#fw)

[1.3     定义、首字母缩写词和缩略语](#dy)

[1.4     参考资料](#参考资料)

[1.5     概述](#gs)

[2.       功能性](#功能性)

[2.1     <功能性需求一>](#gnxxq)

[3.       可用性](#kyx)

[3.1     <可用性需求一>](#可用性需求)

[4.       可靠性](#可靠性)

[4.1     <可靠性需求一>](#可靠性需求)

[5.       性能](#性能)

[5.1     <性能需求一>](#性能需求)

[6.       可支持性](#kzcx)

[6.1     <可支持性需求一>](#kzcxxq)

[7.       设计约束](#sjys)

[7.1     <设计约束一>](#sjysxq)

[8.       联机用户文档和帮助系统需求](#ljyhwd)

[9.       购买的构件](#gmddj)

[10.            接口/界面](#jkjm)

[10.1     用户界面](#用户界面)

[10.2     硬件接口](#yjjk)

[10.3     软件接口](#rjjk)

[10.4     通信接口](#txjk)

补充规约

**1. 简****介**

**1.1 目****的**

旨在详细阐述“景区预约管理模拟软件”的补充需求，包括功能性需求、可用性、可靠性、性能、可支持性、设计约束等，为开发团队提供明确的方向和标准。

**1.2范****围**

适用于景区预约管理模拟软件，包括游客预约、景区管理等模块。

**1.3定义****、首字母缩写词和缩略语**

**SRS**: 软件需求规约

**FR**: 功能性需求

**PR**：功能性-性能需求

**SR**：功能性-安全性需求

**A、B**：可用性需求

**R**：可靠性需求

**P**：性能需求

**S**：可支持性

**C**：设计约束

**MTBF**：平均故障间隔时间

**MTTR**：平均修复时间

**bugs/KLOC**：每千行代码的错误数目

**1.4 参考资****料**

项目组选题说明

软件需求规约

用例规约

**1.5 概****述**

本规约包含以下内容：

功能性需求

可用性需求

可靠性需求

性能需求

可支持性需求

设计约束

**2.  功能****性**

**2.1  <功能****性需求一>**

**2.1.1 用户注册与认证**

FR1.1 系统应提供一个用户友好的注册页面，供用户填写姓名、邮箱、密码等信息。

FR1.2 系统必须验证用户提交的注册信息，包括邮箱地址的有效性和密码的复杂度。

FR1.3 密码复杂度要求：密码必须包含至少一个字母、一个数字和一个特殊字符，且长度至少为8个字符。

FR1.4 系统应在验证信息无误后创建新账号，并引导用户至登录页面。

FR1.5 系统应提供一个登录页面，供用户输入账号和密码进行登录。

FR1.7 系统应为管理员提供特殊的注册流程，包括通过内部代码获得特殊身份的步骤。

FR1.8 管理员登录时，系统应验证其特殊身份，并授予相应的权限。

**2.1.2 密码管理**

FR2.1 系统应提供一个密码找回功能，允许用户通过邮箱接收重置链接。

FR2.2 用户通过点击重置链接后，系统应引导用户设置新密码。

**2.1.3 景区地图浏览**

FR3.1 系统应提供一个景区地图页面，供用户查看景区的卫星地图。

FR3.2 系统必须能够在用户访问地图页面时，快速加载卫星地图。

FR3.3 用户应能通过地图上的按钮或控件实现即时浏览，包括平移和缩放操作。

FR3.4 地图浏览页面应包含搜索框，允许用户输入查询条件以检索特定景点或设施。

FR3.5 用户应能够通过缩放和旋转功能，更清晰地查看景区的各个部分。

**2.1.4 地图搜索与信息检索**

FR4.1 系统应在用户输入搜索关键词后，提供相关的搜索结果。

FR4.2 如果用户搜索无结果，系统应显示消息告知用户，并允许用户修改搜索关键词或清除搜索框重新尝试。

**2.1.5 景点详情浏览**

FR5.1 系统应提供一个景点详情页面，展示各个景点的图片、视频和周边设施信息。

FR5.2 系统应确保景点详情页面的信息能够实时更新，以保持信息的准确性和时效性。

FR5.3 系统应支持用户通过搜索框输入景点名称或关键词进行搜索，并显示相关景点的详细信息。

FR5.4 系统应提供切换不同景点链接的功能，允许用户查看不同景点的详细信息。

FR5.5 系统应允许用户对景点发表评论，并展示其他用户的评论和回复。

**2.1.6 预约景点**

FR6.1 系统应提供一个预约按钮，位于景点详情页面，供用户选择预约特定景点。

FR6.2 系统应显示预约表单，允许用户填写预约信息，包括选择日期和时间。

FR6.3 用户提交预约表单后，系统应验证预约信息的有效性，包括日期、时间和数量限制。

FR6.4 如果预约信息有效，系统应确认预约并显示预约成功消息。

FR6.5 系统应更新用户的预约记录和景点的预约状态。

FR6.6 系统应提供错误消息，告知用户预约信息无效，并允许用户修正信息并重新提交。

FR6.7 用户在预约确认前可以选择取消预约，系统应取消预约并返回景点详情页。

**2.1.7 预约限制和提醒**

FR7.1 系统应实施预约数量限制，确保每个账户每天只能预约一定数量的景点。

FR7.2 系统可以提供预约提醒功能，通过电子邮件或短信通知用户即将到来的预约。

**2.1.8 预约时间冲突和变更**

FR8.1 用户不能预约同一时间段内的多个景点，系统应自动检测并阻止此类预约。

FR8.2 用户在预约生效前一定时间内可以变更预约信息，系统应允许用户进行修改。

**2.1.9 个人中心管理**

FR9.1 系统应提供一个个人中心入口，供用户访问个人中心页面。

FR9.2 个人中心页面应包括用户信息、预约记录、系统通知等模块。

FR9.3 用户可以查看和编辑个人信息，包括姓名、联系方式、密码等。

FR9.4 用户可以查看和管理预约记录，包括预约的景点、时间、状态等。

FR9.5 用户可以查看系统发送的通知或消息。

FR9.6 用户可以修改个人信息，系统应验证信息格式是否正确。

FR9.7 用户可以取消预约，系统应提示用户确认取消并更新预约状态。

**2.1.10 管理员后台管理**

FR10.1 管理员登录后可以通过内部代码获取特殊身份

FR10.1 管理员可以访问后台管理页面。

FR10.2 管理员可以查看后台管理页面，包括游客信息、景点信息等。

FR10.3 管理员可以查看游客信息，包括用户基本信息、预约记录等。

FR10.4 管理员可以管理游客信息，如添加、删除、修改游客信息。

FR10.5 管理员可以查看景点信息，包括景点基本信息、预约情况等。

FR10.6 管理员可以修改景点信息，如景点描述、开放时间等。

**2.2 性能**

PR2.1 系统应在2秒内响应用户的操作请求。

PR2.2 系统应能够处理高峰时段的高并发访问。

PR2.3 系统应在5秒内完成卫星地图的加载，以提供流畅的用户体验

PR2.4 系统应在用户提交预约表单后，即时处理预约请求，并在2秒内给出响应。

PR2.5 系统应能够处理高峰时段的高并发预约请求，确保系统稳定性。

PR2..6 个人中心页面应能够快速加载，保证用户在5秒内能够访问和操作。

PR2.7 系统应在管理员登录后台管理页面后，迅速加载游客和景点信息，响应时间不超过3秒。

PR2.8 管理员执行添加、删除或修改游客和景点信息的操作时，系统应在1秒内完成处理并给出反馈。

PR2.9 系统应支持高效的数据备份和恢复操作，确保在管理员执行这些操作时，系统性能不受显著影响。

**2.3 安全性**

SR3.1 系统必须确保注册和登录过程中的数据传输安全，采用HTTPS等加密协议。

SR3.2 系统应实施账户锁定机制，防止连续失败的登录尝试。

SR3.3 系统应确保地图数据的传输安全，防止数据泄露。

SR3.4 系统应确保用户在发表评论时，评论内容不包含任何恶意代码或不当内容

SR3.5 系统应确保预约信息在传输过程中的安全性，采用加密协议进行数据传输。

SR3.6 系统应实施登录尝试限制，防止暴力破解攻击。

SR3.7 系统应记录所有预约相关的操作日志，以便审计和监控。

SR3.8 系统应确保用户个人信息在存储和传输过程中的安全性，采用加密技术进行保护。

SR3.9 系统应实施访问控制，限制非授权用户访问个人中心。

SR3.10 系统应记录所有个人中心相关的操作日志，以便审计和监控。

SR3.11 系统应实现强密码策略，确保管理员账户的安全性。

SR3.12 管理员登录后台管理页面时，系统应采用多因素认证机制，以增强身份验证的安全性。

SR3.13 系统应对管理员的所有操作进行权限验证，确保只有获得授权的管理员才能访问和操作后台管理功能。

SR3.14 系统应加密存储和传输管理员和游客的敏感信息，如个人身份信息、预约记录等。

SR3.15 系统应记录管理员的所有操作日志，包括登录、数据修改、备份和恢复操作，以便进行安全审计和监控。

SR3.16 系统应实施访问控制，防止未授权访问后台管理页面和数据。

SR3.17 系统应提供数据备份和恢复的加密功能，确保备份数据的安全性。

**3.    可****用性**

**3.1   <可****用性需求一>**

**3.1. 培训需求**

A1：普通用户完成基本预约操作所需的培训时间不应超过30分钟。

A2：高级用户（如管理员）进行复杂操作（如批量管理预约或数据报告生成）所需的培训时间不应超过1小时。

**3.1.2 可评测任务次数**

B1：普通用户完成一次景点预约的任务次数不应超过5步。

B2：管理员审核和确认预约的任务次数不应超过3步。

B3: 管理员完成景点或用户信息修改的任务次数不应超过5步

**4.     可****靠性**

**4.1   <可靠****性需求一>**

**4.1.1 可用性**

R1：系统的可用性应达到80%以上（即80%的时间系统都是可用的）。

R2：系统应保证每天24小时不间断运行，且每周维护时间不超过1小时。

R3：在维护访问期间，系统应提供降级模式操作，确保核心功能仍然可用。

**4.1.2 平均故障间隔时间 (MTBF)**

R4：系统的MTBF应不少于100小时。

**4.1.3 平均修复时间 (MTTR)**

R5：系统发生故障后，MTTR不应超过60分钟。

**4.1.4 精确度**

R6：系统输出数据的精确度应达到90%以上，分辨率至少为5%。

R7：系统应按照已知标准提供精确的数据处理和报告。

**4.1.5 错误或缺陷率**

R8：系统的最高错误率不应超过10 bugs/KLOC（每千行代码的错误数目）。

R9：系统的错误率应按照小错误、大错误和严重错误进行分类。

**4.1.6 错误分类**

R10：小错误指不影响系统正常运行，但可能影响用户体验的问题。

R11：大错误指影响系统部分功能，需要立即修复的问题。

R12：严重错误定义为导致数据完全丢失或系统核心功能完全不可用的错误。

严重错误的界定如下：

数据完全丢失，包括预约记录、用户信息等关键数据。

系统核心功能完全不可用，如无法进行预约、无法查看景点信息等。

**5.****性能**

**5.1  <性能****需求一>**

**5.1.1 响应时间**

P1：系统对用户事务的响应时间平均不应超过10秒。

P2：系统对用户事务的最长响应时间不应超过10秒。

P3：针对“注册与登录”用例，系统应在10秒内完成用户认证过程。

P4：针对“管理个人中心”用例，系统应在10秒内加载用户的个人中心和预约记录。

**5.1.2 吞吐量**

P5：系统应能够每秒处理至少10个并发事务。

P6：在高峰时段，系统吞吐量应能够支持至少20个并发用户进行预约操作。

**5.1.3 容量**

P7：系统应能够支持至少20名活跃用户同时使用。

P8：系统数据库应能够存储至少1000条预约记录。

**5.1.4 降级模式**

P9：在系统资源受限时，系统应进入降级模式，保证核心功能（如预约和登录）的可用性。

P10：在降级模式下，非核心功能（如系统通知推送）可能会被暂时禁用。

**5.1.5 资源利用情况**

P11：系统运行时应确保内存利用率不超过80%。

P12：磁盘空间利用率在任何时候都不应超过70%，以留有足够空间进行数据备份和恢复。

P13：系统应优化网络通信，确保数据传输效率，减少延迟。

P14：系统应监控资源利用情况，并在资源接近极限时发出警告，以便及时进行优化或扩展。

**6.     可支持****性**

**6.1   <可****支持性需求一>**

**6.1.1 编码标准**

S1：所有源代码应遵循统一的编码标准，以增强代码的可读性和可维护性。

S2：编码标准应包括但不限于代码缩进、注释规范、语句长度限制等。

**6.1.2 命名约定**

S3：系统应采用一致的命名约定，包括变量、函数、类和文件的命名。

S4：命名约定应简洁明了，能够反映其用途和功能，避免使用缩写或含糊不清的名称。

**6.1.3 类库和框架**

S5：系统开发应使用标准化的类库和框架，以减少自定义代码的数量和提高代码的重用性。

S6：所选用的类库和框架应具有良好的社区支持和持续更新。

**6.1.4 维护访问权**

S7：系统应提供维护访问权，允许授权维护人员访问系统进行必要的维护和升级操作。

S8：维护访问权应通过角色权限管理进行控制，确保只有授权人员能够执行维护任务。

**6.1.5 维护实用程序**

S9：系统应包含一系列维护实用程序，用于日常的系统监控、日志分析、性能调优等。

S10：维护实用程序应提供友好的用户界面，便于维护人员操作和使用。

S11：系统应支持自动化部署和版本控制，简化维护和升级过程。

**7.     设计约****束**

**7.1   <设计约****束一>**

**7.1.1 软件语言**

C1：系统的前端开发应使用HTML、CSS 和 JavaScript，后端开发应使用 Java 或 Python。

C2：数据库开发应使用 SQL 语言，具体数据库管理系统（如 MySQL、PostgreSQL）需根据项目需求和技术栈确定。

**7.1.2 软件流程需求**

C3：系统开发必须遵循敏捷开发流程，包括迭代开发、持续集成和持续部署。

C4：所有代码提交前必须通过代码审查，确保代码质量和安全。

**7.1.3 开发工具的指定用途**

C5：前端开发工具必须支持最新的前端框架，如 React 或 Vue.js。

C6：后端开发工具应支持所选编程语言和相关的开发框架，如 Spring Boot（Java）或 Django（Python）。

**7.1.4 架构及设计约束**

C7：系统应采用微服务架构，以提高可扩展性和可维护性。

C8：所有服务必须是无状态的，以便于水平扩展和故障恢复。

**7.1.5 购买的构件和类库**

C9：系统使用的第三方构件和类库必须是商业许可或开源许可下的合法产品。

C10：购买的构件和类库必须经过严格的安全审查和性能测试。

**8. 联机用户文档****和帮助系统需求**

**无**

1. **购买****的构件**

**无**

**10. 接****口/界面**

**10.1 用户****界面**

**Web界面：**

协议：HTTP/HTTPS

端口：标准端口80（HTTP）和443（HTTPS）

逻辑地址：

功能：提供用户注册、登录、预约景区、查看预约记录、管理个人资料等操作界面。

设计要求：响应式设计，支持PC、平板和手机访问。

**移动应用界面：**

平台：iOS和Android

功能：提供与Web界面相似的功能，同时包括推送通知和离线访问能力。

设计要求：符合平台的设计指南，提供流畅的用户体验。

**10.2 硬件****接口**

**条码扫描器：**

逻辑结构：USB或无线连接

物理地址：根据具体设备而定

预期行为：用于快速扫描门票上的条码，实现快速入园。

**POS终端：**

逻辑结构：网络连接

物理地址：根据网络配置而定

预期行为：处理现场购票和退款操作。

**10.3 软****件接口**

**Google Maps API：**

协议：HTTPS

端口：443

逻辑地址：

接口规范：遵循Google Maps API文档。

**Stripe Payment Gateway：**

协议：HTTPS

端口：443

逻辑地址：

接口规范：遵循Stripe API文档。

**SendGrid Email Service：**

协议：HTTPS

端口：443

逻辑地址：

接口规范：遵循SendGrid API文档。

**10.4 通****信接口**

**内部服务通信：**

协议：HTTP/HTTPS

端口：内部服务端口（例如，8000/8443）

逻辑地址：

预期行为：支持微服务之间的内部通信。

**外部系统集成：**

协议：HTTPS

端口：443

逻辑地址：预期行为：与外部系统集成，如CRM系统、财务系统等，进行数据交换。

**局域网通信：**

协议：TCP/IP

端口：根据具体设备和服务配置

预期行为：支持局域网内设备和服务之间的通信。