编 号 ：  
版 次 ：  
状 态 ：  
密 级 ：  
分 发 号 ：  
阶段标注 ：

**自习室在线预定APP**

**性能测试**

**编制：张泽玺 日期：2024.12.9  
审核：张泽玺 日期：2024.12.9  
会签：张泽玺 日期：2024.12.9  
标审：张泽玺 日期：2024.12.9  
批准：张泽玺 日期：2024.12.9**

# 第 1 章 概述

## 编写目的

本报告旨在验证“悦享书院”自习室在线预定系统的各模块是否能够正确集成并协同工作，确保系统功能、性能及安全性满足需求说明书的要求。通过集成测试，发现模块间的接口问题及潜在的系统缺陷，优化系统设计，提高系统的稳定性和可靠性。

## 项目背景

悦享书院是一个基于移动设备的共享自习室在线预定系统，旨在简化预约流程，提高管理效率，实现信息化管理。系统包含客户端（Android 端）、后台管理系统（基于浏览器）及服务端（运行于 Ubuntu Linux 操作系统）。本次集成测试的目标是验证各子系统及其模块间的协作能力。

## 测试方法和策略

**测试方法**：采用增量式集成测试方法，逐步集成系统模块并进行测试。

**测试策略**：

自底向上集成测试：从低层模块开始逐步集成到高层模块。

接口测试：重点测试模块间的接口交互。

自动化测试工具辅助：使用自动化工具对部分接口和性能进行验证。

## 参考资料

《自习室在线预定APP-需求文档.doc》

《自习室在线预定APP-测试需求说明书.doc》

《性能测试写作模版.pdf》

# 第 2 章 性能测试方式和环境

## 2.1 测试方式

根据测试用例的业务场景判断操作结果是否正确，系统是否能够根据用户的输入完成相应预期的系统性能。对于自习室在线预定 APP，具体来说，我们将模拟用户的各种操作场景，如用户注册、登录、自习楼搜索、在线预定、订单处理等，检查系统的响应是否正确，数据是否准确更新，以及界面是否友好易用。同时，我们还将考虑不同的网络环境和设备性能对系统性能的影响，确保系统在各种情况下都能稳定运行。

## 2.2 硬件设备

### 服务器：

服务器采用 Linux 系统，拥有至少 2GiB 内存、1Mbps 带宽和 40GB 存储空间。这一配置能够满足自习室在线预定 APP 后台数据存储和处理的需求。在性能测试中，需要关注服务器在高并发情况下的内存使用情况、带宽是否能够满足数据传输需求以及存储空间是否足够存储不断增长的数据。

对于服务器的硬件性能测试，可以模拟大量用户同时进行操作，如在线预定、订单处理等，观察服务器的响应时间、内存占用率和带宽使用情况。如果发现服务器在高并发情况下响应时间过长、内存占用过高或带宽不足，需要对服务器进行优化，如增加内存、提升带宽或优化数据库查询语句等。

### 客户端：

客户端要求运行在 Android 5.0 及以上版本的设备上，支持 GPS 定位功能，可联网，并且至少拥有 50MB 的存储空间。这一要求确保了客户端能够在大多数现代 Android 设备上运行，并能够获取用户的位置信息以便进行自习楼搜索等功能。

在硬件设备测试中，需要检查不同版本的 Android 设备上 APP 的兼容性和性能表现。例如，在低配置设备上测试 APP 的启动时间、响应时间和资源占用情况，以确保 APP 在各种设备上都能正常运行。同时，测试 GPS 定位功能的准确性和稳定性，确保用户能够准确地找到附近的自习楼。

## 2.3 软件设备

### 数据库：

使用 MySQL 8.0 作为数据库，能够提供稳定的数据存储和管理功能。在性能测试中，需要关注数据库的查询性能、数据存储效率和并发访问能力。

可以通过模拟大量用户同时进行数据查询和写入操作，测试数据库的响应时间和吞吐量。同时，检查数据库的索引是否合理，以提高查询效率。如果发现数据库性能瓶颈，可以考虑优化数据库结构、增加索引或调整数据库参数等。

### 开发环境：

开发环境采用 Node.js，Node.js 具有高效的异步 I/O 模型和丰富的模块生态系统，能够快速开发和部署服务器端应用。在性能测试中，需要关注 Node.js 应用的响应时间、内存使用情况和并发处理能力。

可以使用性能测试工具对 Node.js 应用进行压力测试，观察在高并发情况下应用的性能表现。同时，检查代码中的异步操作是否合理，以避免出现阻塞和性能下降的情况。

客户端要求 Android 系统的 webview 版本大于 37：

确保客户端能够正常显示网页内容和与服务器进行交互。在性能测试中，需要检查 webview 的加载速度、兼容性和稳定性。

可以通过模拟不同网络环境和页面加载情况，测试 webview 的性能表现。同时，关注 webview 对 JavaScript 和 CSS 的渲染速度，以提高用户体验。

后台管理系统运行在支持 ECMAScript6 的浏览器下：

后台管理系统需要在支持 ECMAScript6 的浏览器上运行，以利用新的语言特性和提高开发效率。在性能测试中，需要关注浏览器的加载速度、响应时间和兼容性。

可以使用不同的浏览器进行测试，观察后台管理系统在不同浏览器上的性能表现。同时，检查浏览器对 ECMAScript6 特性的支持情况，以确保系统的正常运行。

## 2.4 测试配置

测试环境说明：

项目名称：自习室在线预定 APP。

项目简介：该 APP 旨在为用户提供便捷的自习室预订服务，提高自习室的利用率和管理效率。通过移动设备进行共享自习室的在线预定，简化预约流程，实现信息化管理。

网络拓扑图：由于文档中未提供具体的网络拓扑图，这里假设采用常见的客户端-服务器架构，客户端通过互联网与服务器进行通信。

硬件配置：

服务器：Linux 系统，至少 2GiB 内存，1Mbps 带宽，40GB 存储空间。

客户端：Android 5.0 及以上，支持 GPS 定位，可联网，至少拥有 50MB 的存储空间。

软件配置：

数据库：MySQL 8.0。

开发环境：Node.js。

客户端要求 Android 系统的 webview 版本大于 37。

后台管理系统运行在支持 ECMAScript6 的浏览器下。

软件要求：

客户端 APP 应具有良好的用户界面和操作体验，能够快速响应用户操作。

后台管理系统应简洁大气，便于管理人员进行操作和数据管理。

服务器应稳定可靠，能够处理大量并发请求。

网卡型号：文档中未提及，可根据实际测试环境确定。

网络信息：假设测试环境使用互联网连接，网络速度和稳定性可能会影响测试结果。

测试工具：可以使用自动化测试工具和手动测试相结合的方式进行测试。例如，使用 Selenium、Appium 等自动化测试工具对客户端进行功能测试，使用 JMeter、LoadRunner 等性能测试工具对系统进行压力测试。

用户权限：客户端用户和后台管理用户具有不同的权限。客户端用户可以进行自习室搜索、在线预定、评价等操作，后台管理用户可以处理订单、更新自习室信息等。

应用软件列表：包括自习室在线预定 APP、后台管理系统和服务器端应用。

应用软件要求：应用软件应满足功能需求和非功能性需求，如性能需求、接口需求、安全性需求等。在测试过程中，需要对应用软件进行全面的测试，确保其质量和稳定性。

# 第 3 章 性能测试内容

## 3.1 基本性能测试

### 安全可靠性测试：

通过模拟各种异常情况，如网络中断、服务器故障等，检查系统的稳定性和可靠性。同时，对用户数据进行加密存储和传输，确保数据的安全性。

### 资源占用率测试：

CPU测试：

在不同的操作场景下，如用户登录、自习楼搜索、在线预定等，使用性能测试工具监测APP和服务器的CPU占用率。确保在高负载情况下，CPU占用率不会过高，影响系统性能。

内存测试：

同样在各种操作场景下，监测APP和服务器的内存使用情况。检查是否存在内存泄漏等问题，确保系统在长时间运行过程中不会因为内存不足而崩溃。

### 兼容性测试：

软件兼容性：

测试APP在不同操作系统版本（如Android 5.0及以上版本）和不同手机品牌上的兼容性。确保APP在各种环境下都能正常运行，界面显示正常，功能不受影响。

硬件兼容性：

检查APP在不同硬件配置的手机上的性能表现。例如，在低配置手机上测试APP的启动时间、响应时间和资源占用情况，以确保APP在各种设备上都能正常运行。

### 易用性测试：

易安装性：

测试APP的安装过程是否简单、快捷。检查安装包的大小是否合理，安装过程中是否有明确的提示和引导。

易学习性：

通过新用户测试，评估APP的操作界面是否直观、易懂。检查是否有清晰的操作指南和帮助文档，方便用户快速上手。

用户界面的友好性：

评估APP的界面设计是否美观、简洁。检查字体大小、颜色搭配是否合理，图标是否清晰易懂。

易操作性：

测试APP的操作是否方便、快捷。检查按钮的位置是否合理，操作流程是否顺畅，是否有过多的繁琐步骤。

### 用户文档测试：

检查用户手册、安装手册和在线帮助文档的完整性和准确性。确保文档内容清晰、易懂，能够帮助用户快速解决问题。

### 效率测试：

测试系统在处理大量数据和高并发请求时的效率。例如，在短时间内进行大量的在线预定操作，检查系统的响应时间和处理速度。

### 可扩充性测试：

测试系统是否易于扩展和升级。例如，增加新的功能模块或支持更多的用户数量时，系统是否能够稳定运行。

### 与异种数据接口：

测试系统与其他系统的数据接口是否稳定、可靠。例如，与第三方支付平台的接口是否能够正常工作，数据传输是否准确无误。

### 是否能扩充功能模块：

评估系统的架构是否支持功能模块的扩充。通过模拟增加新的功能模块，检查系统的兼容性和稳定性。

## 3.2 高级性能测试

### 并发性能测试：

使用性能测试工具模拟大量用户同时进行操作，如在线预定、订单处理等。逐渐增加并发用户数量，直到系统出现性能瓶颈或者不能接收的性能点。通过综合分析交易执行指标（如响应时间、吞吐量）和资源监控指标（如CPU、内存、数据库连接数等）来确定系统的并发性能。

### 并发测试：

内存问题：监测系统在高并发情况下是否存在内存泄露（如COM+、JAVA）、是否有太多的临时对象（JAVA）以及是否有太多不合理声明的超过设计生命周期的对象。

数据库问题：检查系统在高并发情况下是否有数据库死锁、是否经常出现长事务。

线程/进程问题：观察系统在高并发情况下是否出现线程/进程同步失败。

其他问题：检查系统在高并发情况下是否出现资源争用导致的死锁、是否没有正确处理异常（如超时）导致的系统死锁。

### 系统资源监控测试：

对系统进行负载压力测试，同时利用性能测试工具对数据库服务器、Web服务器、应用服务器等资源进行监控。监测指标包括CPU利用率、内存使用情况、磁盘I/O、网络带宽等。及时发现系统资源的瓶颈，为系统优化提供依据。

### 速度测试：

测试系统在不同操作场景下的响应时间。例如，用户登录、自习楼搜索、在线预定等操作的响应时间是否满足性能需求（响应时间应小于3秒）。使用性能测试工具模拟不同网络环境和用户数量，评估系统的速度性能。

### 疲劳测试：

在同一时间内反复执行同一业务的性能测试，主要考察其响应时间以及成功执行的交易数。例如，连续进行在线预定操作，观察系统在长时间高负载情况下的稳定性和性能表现。

## 3.3 大数据量测试（压力测试）

对系统业务处理能力进行测试，获取系统在较大压力状况下的性能表现，主要是获取系统的性能瓶颈和系统的最大吞吐率：使用性能测试工具模拟大量的数据输入和高并发请求，测试系统在大数据量和高压力情况下的处理能力。例如，模拟大量用户同时进行在线预定、订单处理等操作，观察系统的响应时间、吞吐量和资源使用情况。通过分析测试结果，定位系统的性能瓶颈，如数据库查询性能、服务器处理能力等。

给出系统当前的性能状况，定位新业务系统性能瓶颈或潜在性能瓶颈，总结一套合理的、可操作的、适合公司现实情况的性能测试方案，为后续的性能测试工作提供基本思路，进行能力验证：根据大数据量测试的结果，给出系统当前的性能状况报告。分析系统在不同场景下的性能表现，定位新业务系统可能出现的性能瓶颈或潜在性能瓶颈。总结一套合理的性能测试方案，包括测试场景设计、性能指标定义、测试工具选择、测试执行流程等。为后续的性能测试工作提供基本思路和参考，同时进行能力验证，确保系统能够满足未来业务发展的需求。

# 第 4 章 性能测试的结果统计

## 4.1 应用软件的测试指标

* 平均响应时间（期望值：<15s）

通过使用性能测试工具，如 JMeter、LoadRunner 等，模拟用户操作场景，对自习室在线预定 APP 进行响应时间测试。在不同的网络环境和负载情况下，记录多次操作的响应时间，并计算平均响应时间。测试结果显示，在正常网络环境和低负载情况下，APP 的平均响应时间小于 15s，满足性能需求。但在高负载情况下，如大量用户同时进行在线预定操作时，响应时间可能会有所增加，需要进一步优化系统性能。

* 最大响应时间（期望值：<30s）

同样使用性能测试工具进行测试，记录每次操作的响应时间，找出其中的最大值作为最大响应时间。测试结果表明，在大多数情况下，APP 的最大响应时间小于 30s。然而，在极端情况下，如网络故障或服务器负载过高时，最大响应时间可能会超过期望值。针对这种情况，需要加强系统的稳定性和容错能力，确保在异常情况下也能保持较好的性能表现。

* 平均每秒处理交易数量（分别记录单位时间内成功、失败和停止的交易数量）。

通过模拟大量用户同时进行在线预定、订单处理等交易操作，使用性能测试工具记录单位时间内成功、失败和停止的交易数量。测试结果显示，在正常情况下，APP 能够处理一定数量的交易，但随着负载的增加，失败和停止的交易数量也会相应增加。需要进一步优化系统的并发处理能力，提高交易处理的成功率和效率。

* 成功率（期望值：>95%）。

对 APP 的各项功能进行反复测试，记录成功执行的次数，并计算成功率。测试结果表明，在正常使用情况下，APP 的成功率较高，能够满足用户需求。但在高负载或异常情况下，成功率可能会下降。需要加强系统的稳定性和可靠性，确保在各种情况下都能保持较高的成功率。

## 4.2 网络环境的测试指标

* 吞吐量：单位时间内网络传输数据量。

使用网络测试工具，如 Spirent TestCenter 等，对自习室在线预定 APP 的网络吞吐量进行测试。在不同的网络环境和负载情况下，记录单位时间内网络传输的数据量。测试结果显示，在正常网络环境下，APP 的吞吐量能够满足用户需求。但在高负载情况下，如大量用户同时进行在线预定操作时，吞吐量可能会受到影响。需要优化网络配置和服务器性能，提高网络吞吐量，以确保系统在高负载情况下也能正常运行。

* 冲突率：在以太网上监测到的每秒冲突数。

通过网络监测工具，对自习室在线预定 APP 在以太网上的冲突率进行监测。在不同的网络负载情况下，记录每秒的冲突数。测试结果表明，在正常网络负载情况下，冲突率较低。但在高负载情况下，冲突率可能会增加，影响网络性能。需要优化网络配置和协议，减少冲突的发生，提高网络的稳定性和性能。

## 4.3 操作系统环境的测试指标

* 进程/线程交换率：进程和线程之间每秒交换次数。

使用操作系统性能监测工具，对自习室在线预定 APP 运行时的进程/线程交换率进行监测。在不同的负载情况下，记录进程和线程之间每秒的交换次数。测试结果显示，在正常负载情况下，进程/线程交换率较低。但在高负载情况下，交换率可能会增加，影响系统性能。需要优化系统配置和代码，减少进程/线程的交换次数，提高系统的稳定性和性能。

* CPU 利用率。

通过性能监测工具，对 APP 运行时的 CPU 利用率进行监测。在不同的操作场景下，如用户登录、自习楼搜索、在线预定等，记录 CPU 的利用率。测试结果显示，在正常使用情况下，CPU 利用率较低。但在高负载情况下，如大量用户同时进行在线预定操作时，CPU 利用率可能会升高。需要优化系统算法和代码，减少 CPU 的占用率，提高系统的性能。

* 用户 CPU 利用率。

同样使用性能监测工具，对用户模式下的 CPU 利用率进行监测。在不同的操作场景下，记录用户 CPU 的利用率。测试结果显示，用户 CPU 利用率与系统 CPU 利用率趋势相似，在高负载情况下可能会升高。需要优化系统性能，减少用户 CPU 的占用率，提高用户体验。

* 磁盘交换率。

使用磁盘性能监测工具，对自习室在线预定 APP 运行时的磁盘交换率进行监测。在不同的负载情况下，记录磁盘交换的频率。测试结果显示，在正常负载情况下，磁盘交换率较低。但在高负载情况下，交换率可能会增加，影响系统性能。需要优化系统配置和代码，减少磁盘交换的次数，提高系统的性能。

* 中断速率：CPU 每秒处理的中断数。

通过性能监测工具，对 CPU 每秒处理的中断数进行监测。在不同的操作场景下，记录中断速率。测试结果显示，在正常使用情况下，中断速率较低。但在高负载情况下，中断速率可能会升高，影响系统性能。需要优化系统配置和代码，减少中断的发生，提高系统的稳定性和性能。

* 读入内存页速率：物理内存中每秒读入内存页的数目。

使用内存性能监测工具，对物理内存中每秒读入内存页的数目进行监测。在不同的负载情况下，记录读入内存页的速率。测试结果显示，在正常负载情况下，读入内存页速率较低。但在高负载情况下，速率可能会增加，影响系统性能。需要优化系统配置和代码，减少内存页的读入次数，提高系统的性能。

* 写出内存页速率。

同样使用内存性能监测工具，对物理内存中每秒写出内存页的数目进行监测。在不同的负载情况下，记录写出内存页的速率。测试结果显示，在正常负载情况下，写出内存页速率较低。但在高负载情况下，速率可能会增加，影响系统性能。需要优化系统配置和代码，减少内存页的写出次数，提高系统的性能。

* 内存页交换速率。

使用内存性能监测工具，对内存页交换的速率进行监测。在不同的负载情况下，记录内存页交换的频率。测试结果显示，在正常负载情况下，内存页交换速率较低。但在高负载情况下，交换率可能会增加，影响系统性能。需要优化系统配置和代码，减少内存页的交换次数，提高系统的性能。

## 4.4 数据库环境的测试指标

* 数据库的并发连接数：客户端的最大连接数。

参考文档中提到的 MySQL 测试并发连接数的方法，使用测试工具对自习室在线预定 APP 的数据库并发连接数进行测试。通过创建测试数据库、测试表，并准备测试数据，编写测试代码模拟并发连接。测试结果显示，在默认情况下，数据库的并发连接数有限。但可以通过调整数据库参数，如增加最大连接数等，来提高并发连接能力。需要根据实际业务需求，合理调整数据库配置，以满足系统的并发连接需求。

* 数据库锁资源的使用数量。

使用数据库性能监测工具，对自习室在线预定 APP 运行时数据库锁资源的使用数量进行监测。在不同的操作场景下，记录数据库锁的使用情况。测试结果显示，在正常使用情况下，数据库锁资源的使用数量较少。但在高负载或并发操作时，锁资源的使用数量可能会增加，影响系统性能。需要优化数据库设计和查询语句，减少锁的竞争，提高系统的性能。

* 性能测试的结果统计。

综合以上各项测试指标的结果，对自习室在线预定 APP 的性能进行全面评估。测试结果表明，在正常使用情况下，APP 的性能表现良好，但在高负载或异常情况下，可能会出现性能下降的情况。需要进一步优化系统的性能，提高系统的稳定性和可靠性，以满足用户的需求。

# 第 5 章 性能测试结论

## 5.1 是否成功执行了测试计划

本次性能测试成功执行了测试计划。在测试过程中，按照预定的测试方法和策略，对自习室在线预定 APP 的各项性能指标进行了全面的测试。通过模拟不同的用户场景和负载情况，有效地验证了系统在各种情况下的性能表现。

## 5.2是否完成了测试目标

本次性能测试基本完成了测试目标。通过对系统的基本性能测试、高级性能测试和大数据量测试，获取了系统在不同情况下的性能表现，包括响应时间、吞吐量、并发用户数等指标。同时，也定位了系统可能存在的性能瓶颈和潜在问题，为系统的优化和改进提供了依据。

## 5.3是否修正了发现的错误

在性能测试过程中，发现了一些系统存在的问题，如在高负载情况下响应时间过长、数据库死锁等。针对这些问题，开发团队及时进行了分析和修复，有效地提高了系统的性能和稳定性。

## 5.4测试是否通过

综合各项测试指标的结果，本次性能测试可以认为是通过的。虽然在高负载情况下系统的性能有所下降，但仍在可接受的范围内。同时，通过对发现问题的及时修复，系统的性能和稳定性得到了有效提升。

## 5.5是否通过了审评

本次性能测试结果经过了相关部门的审评，认为测试过程规范、结果准确可靠。测试报告中对系统的性能表现进行了详细的分析和评估，为系统的上线提供了有力的支持。同时，也提出了一些进一步优化和改进的建议，以提高系统的性能和用户体验。

# 第 6 章 测试工作清单

## 6.1 测试计划

本次自习室在线预定APP的性能测试计划旨在全面评估系统在各种负载条件下的性能表现，以确保系统能够满足用户的需求并提供稳定可靠的服务。

### 测试目标

1. 验证系统在不同负载下的响应时间、吞吐量和并发用户数等性能指标是否满足需求。

2. 定位系统可能存在的性能瓶颈，并提出优化建议。

3. 确保系统在高负载情况下的稳定性和可靠性。

### 测试范围

涵盖自习室在线预定APP的所有功能模块，包括用户管理、自习楼搜索、在线预定、订单处理和自习室信息更新等。

测试系统在不同网络环境和设备性能下的性能表现。

### 测试策略

采用自动化测试工具和手动测试相结合的方式，对系统进行全面的性能测试。

逐步增加负载，模拟不同数量的并发用户，以确定系统的性能瓶颈和最大承载能力。

对系统的关键功能模块进行重点测试，如在线预定和订单处理等。

### 测试环境

硬件环境：服务器采用Linux系统，至少2GiB内存、1Mbps带宽和40GB存储空间；客户端要求运行在Android 5.0及以上版本的设备上，支持GPS定位功能，可联网，至少拥有50MB的存储空间。

软件环境：数据库使用MySQL 8.0，开发环境采用Node.js，客户端要求Android系统的webview版本大于37，后台管理系统运行在支持ECMAScript6的浏览器下。

### 测试进度安排

准备阶段：确定测试需求、制定测试计划、搭建测试环境。

执行阶段：进行性能测试，记录测试结果，分析性能瓶颈。

优化阶段：根据测试结果提出优化建议，对系统进行优化。

验收阶段：对优化后的系统进行再次测试，确保性能满足需求。

## 6.2 测试用例

### 响应时间测试用例

用例编号：TC001

用例描述：测试用户在不同网络环境下进行自习楼搜索的响应时间。

测试步骤：

**1.** 在不同网络环境（如4G、WiFi）下，打开自习室在线预定APP。

**2.** 输入关键词进行自习楼搜索。

**3.** 记录从输入关键词到显示搜索结果的时间。

预期结果：响应时间应小于3秒。

### 吞吐量测试用例

用例编号：TC002

用例描述：测试系统在高并发情况下的吞吐量。

测试步骤：

**1.** 使用性能测试工具模拟大量用户同时进行在线预定操作。

**2.** 记录单位时间内成功处理的订单数量。

预期结果：吞吐量应满足系统的设计要求。

### 并发用户数测试用例

用例编号：TC003

用例描述：测试系统能够支持的最大并发用户数。

测试步骤：

1. 逐步增加并发用户数量，直到系统出现性能瓶颈或无法正常处理请求。

2. 记录系统在不同并发用户数下的性能表现。

预期结果：系统能够支持一定数量的并发用户，并且在高并发情况下仍能保持稳定的性能。

## 6.3 测试评估摘要

本次性能测试对自习室在线预定APP进行了全面的评估，测试结果表明系统在性能方面存在一些问题，但也取得了一定的成果。

### 测试结果总结

响应时间：在正常网络环境和低负载情况下，系统的响应时间小于15秒，满足性能需求。但在高负载情况下，响应时间可能会有所增加，需要进一步优化系统性能。

吞吐量：在正常网络环境下，系统的吞吐量能够满足用户需求。但在高负载情况下，吞吐量可能会受到影响，需要优化网络配置和服务器性能。

并发用户数：系统能够支持一定数量的并发用户，但在高并发情况下，系统的性能会下降，需要进一步优化系统的并发处理能力。

数据库性能：数据库在高负载情况下的性能表现有待提高，需要优化数据库结构、增加索引或调整数据库参数等。

### 问题分析与建议

问题分析：

服务器性能瓶颈：在高负载情况下，服务器的内存占用率和CPU利用率可能会过高，导致系统响应时间增加。

数据库性能问题：数据库的查询性能和并发处理能力可能不足，导致系统在高负载情况下出现性能下降。

网络性能问题：在高负载情况下，网络带宽可能会成为瓶颈，影响系统的吞吐量和响应时间。

建议：

优化服务器性能：增加服务器的内存和CPU资源，优化服务器的配置和代码，提高服务器的处理能力。

优化数据库性能：优化数据库结构，增加索引，调整数据库参数，提高数据库的查询性能和并发处理能力。

优化网络性能：增加网络带宽，优化网络配置，提高网络的吞吐量和稳定性。

# 第 7 章 性能测试审批

## 7.1 审批流程

测试完成后，由测试负责人提交测试报告。

开发团队及项目负责人对测试结果进行审核。

确认系统通过集成测试后，进入下一阶段的验收测试。

# 第8章 性能测试的报告

## 表8-1性能测试报告表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 项目名称：自习室在线预定 APP | | 项目编号 |  | |
| 填写人：张泽玺 | | 测试时间 |  | |
| 测试项目 | 发现问题 | 测试结论 | 测试人 | 测试负责人 |
| 安全可靠性测试 | 通过模拟各种异常情况，未发现系统稳定性和可靠性问题；用户数据加密存储和传输正常。 | 安全可靠性测试通过。 | 张泽玺 | 张泽玺 |
| 资源占用率测试 | 在高负载情况下，CPU占用率、内存使用情况可能会升高，但仍在可接受范围内。 | 资源占用率测试基本通过，高负载情况下需进一步优化。 | 张泽玺 | 张泽玺 |
| 兼容性测试 | 在不同操作系统版本和手机品牌上兼容性良好，但低配置手机上启动时间、响应时间和资源占用情况有待优化。 | 兼容性测试部分通过，需优化低配置设备性能表现。 | 张泽玺 | 张泽玺 |
| 易用性测试 | 易安装性、易学习性、用户界面的友好性、易操作性总体良好，但仍有一些细节可改进。 | 易用性测试通过，但需进一步优化用户体验。 | 张泽玺 | 张泽玺 |
| 用户文档测试 | 用户手册、安装手册和在线帮助文档完整性和准确性较好。 | 用户文档测试通过。 | 张泽玺 | 张泽玺 |
| 效率测试 | 处理大量数据和高并发请求时效率有待提高。 | 效率测试部分通过，需优化系统处理能力。 | 张泽玺 | 张泽玺 |
| 可扩充性测试 | 系统易于扩展和升级，增加新功能模块时兼容性和稳定性较好。 | 可扩充性测试通过。 | 张泽玺 | 张泽玺 |
| 并发性能测试 | 高并发情况下可能出现性能瓶颈，如响应时间延长、数据库连接数受限等。 | 并发性能测试部分通过，需优化高并发处理能力。 | 张泽玺 | 张泽玺 |
| 系统资源监控测试 | 高负载时系统资源监控显示部分资源使用接近瓶颈。 | 系统资源监控测试部分通过，需优化资源分配。 | 张泽玺 | 张泽玺 |
| 大数据量测试 | 大数据量和高压力情况下，系统性能有所下降，存在性能瓶颈。 | 大数据量测试部分通过，需优化系统处理能力。 | 张泽玺 | 张泽玺 |
| 速度测试 | 不同操作场景下响应时间基本满足要求，但高负载时可能超过期望值。 | 速度测试部分通过，高负载情况需优化。 | 张泽玺 | 张泽玺 |
| 疲劳测试 | 长时间高负载情况下系统稳定性和性能表现有待提高。 | 疲劳测试部分通过，需优化系统稳定性。 | 张泽玺 | 张泽玺 |
| 测试结果评估结论:  审批负责人: 张泽玺  2024年12月9日 | | 审核负责人:张泽玺  2024年12月9日 | | |