## 4. System specification

### 4.1技术要求

* 兼容性：
  + 网络协议：确保系统支持校园网络使用的通信协议，如HTTP/HTTPS、TCP/IP等。
  + 网络安全：与校园网络安全策略保持一致，包括防火墙规则、VPN访问和入侵检测系统。
  + API集成：系统需要能够通过API与校园现有的支付网关进行交互，处理支付宝、微信支付和银行转账
  + 数据格式：确保系统能够处理和交换校园内部使用的数据格式，如CSV、XML或JSON。
  + 操作系统：Android系统和IOS系统中的兼容问题
* 安全性：
  + 数据加密：使用强加密标准（如AES）来保护存储和传输的数据。对敏感信息，如个人身份信息（PII）和支付细节，进行加密。
  + 用户认证：实现多因素认证（MFA），增加安全层，确保只有授权用户才能访问系统。
  + 访问权限：为用户和管理员设置不同的系统权限，确保系统的运行安全与用户的数据安全。
  + 支付安全：遵守支付卡行业数据安全标准（PCI DSS）。使用安全的支付网关和第三方支付服务，确保支付过程的安全。
  + 网络安全：使用防火墙和入侵检测系统（IDS）来保护网络边界。实施网络隔离和分段，以减少潜在的安全威胁。
  + 数据备份和恢复：定期备份数据，并确保备份的安全性和完整性。制定灾难恢复计划，以便在数据丢失或系统故障时快速恢复。
* 可靠性：
  + 稳定性：确保该系统在高负荷高数据传输量的运行情况下不会崩溃，导致用户使用体验受损甚至造成用户财产损失。
  + 故障处理：①系统故障：实现故障转移机制，避免单点故障导致的服务中断/

②用户端故障：对于数据类型输入错误、用户操作错误等信息，系统应该及时给出反馈并告知用户如何更改。

* 用户界面：
  + 直观设计：设计直观的用户界面，使用户能够轻松地找到并使用功能。
  + 一致性：保持界面元素和操作的一致性，以减少用户的学习曲线。
  + 反馈：提供及时的用户反馈，让用户知道他们的操作是否成功。
  + 美学：界面应具有专业的外观和感觉，使用户享受使用系统的过程。
  + 用户测试：进行用户测试，收集反馈，并根据用户的需求和偏好进行调整。

### 性能指标

* 响应时间：
  + 通过采取多种措施，如代码优化、数据库索引、使用缓存、负载均衡、硬件升级、网络优化和异步处理来优化响应时间，优化用户体验。
* 最大并发用户数：
  + 通过扩展服务器资源、使用负载均衡和优化数据库连接来提高并发处理能力，确保系统在高峰时段也能稳定响应用户请求。
* 交易处理能力：
  + 对于支付系统，高交易处理能力是确保用户体验和业务连续性的关键。通过优化事务处理流程、我们可以使用高效的数据库设计和索引来提升处理速度。
* 故障恢复时间：
  + 低故障恢复时间可以最小化系统停机时间，对业务影响较小。我们打算建立灾难恢复计划，使用冗余系统和快速切换机制来最小化故障恢复时间。
* 负载测试：
  + 我们需要对系统在预期的最大负载下的性能表现进行测试。通过负载测试可以验证系统是否能够满足性能要求，并发现潜在的性能瓶颈。最后根据测试结果进行系统调优，增加资源或优化代码。
* 性能监控：
  + 使用专业的监控工具，设定警报阈值，快速定位和解决问题。当发现性能下降或系统异常时，确保系统能够快速响应，及时解决问题，保障用户有良好的使用体验。