要将Spring Boot Web应用程序快速转换为可在手机上使用的应用程序，您可以考虑以下几个步骤：

响应式设计：确保您的Web应用程序使用响应式设计，以便在不同尺寸和分辨率的移动设备上提供良好的用户体验。确保您的应用程序可以自动适应手机屏幕，并且界面元素在小屏幕上可读性良好。

移动优化界面：重新评估您的用户界面，使其在移动设备上更加友好和易于使用。考虑使用更大的按钮、简化的导航和更少的输入字段，以适应手机上的有限空间和触摸操作。

移动端框架：考虑使用专门为移动应用程序开发的框架，如React Native、Flutter或Ionic等。这些框架允许您使用Web技术（如HTML、CSS和JavaScript）构建跨平台的移动应用程序，从而节省开发时间和精力。

API适配器：如果您的Spring Boot应用程序提供API，可以创建一个适配器层，将现有的API端点适配为适合移动应用程序的格式。这样，移动应用程序可以通过调用适配器层来获取所需的数据。

推送通知：利用移动设备的推送通知功能，为移动应用程序添加即时通知功能，向用户发送重要信息或提醒。

测试和优化：确保在转换后的应用程序中进行充分的测试，以确保它在各种移动设备上的性能和功能良好。根据用户反馈和使用情况，进行必要的优化和改进。

请注意，这只是一个概述，并不详尽，具体的实施方法可能会因项目需求和技术选择而有所不同。您可能需要进一步研究和探索相关技术和工具，以确定最适合您的情况的方法

系统结构：描述系统的整体结构和组织，包括模块划分、层次结构和组件之间的关系。

架构设计：详细描述系统的架构设计，包括所采用的架构模式、设计原则和设计决策。

组件设计：对系统中的各个组件进行详细设计，包括组件的功能、接口、数据结构和算法等。

数据库设计：如果系统使用数据库，描述数据库的设计和结构，包括表结构、关系、索引等。

接口设计：描述系统与外部系统或组件的接口设计，包括API、通信协议、数据格式等。

用户界面设计：详细描述系统的用户界面设计，包括界面元素、布局、交互流程等。

算法设计：如果系统包含复杂的算法或数据处理过程，描述算法的设计和实现方法。

安全设计：考虑系统的安全性需求，描述安全设计措施和机制，如身份验证、访问控制、加密等。

性能设计：考虑系统的性能需求，描述性能设计方案，如缓存策略、并发控制、优化技术等。

错误处理和异常处理：描述系统对错误和异常情况的处理机制和策略，包括错误码、异常处理流程等。

日志和监控：描述系统的日志记录和监控机制，用于跟踪系统的运行状态和故障排查。

测试设计：详细描述系统的测试策略和测试计划，包括单元测试、集成测试、系统测试等。

部署和运维：描述系统的部署和运维策略，包括部署环境、配置管理、备份和恢复等。

设计文档应该清晰、详细地描述系统的设计，以便开发团队和利益相关者理解和实现系统。具体的设计文档内容和结构可能会因项目的规模、技术选型和特定需求而有所不同。因此，根据您的项目需求，可能需要对以上内容进行适当的调整和定制。