# 论软件的可靠性设计

摘要：

2023年4月，我所在的单位启动同城一码项目开发，该项目集合公交、地铁、景区等过闸机的二维码于一体，做到在一个城市凭一个二维码即可通过大部公共设施，无需注册多余的账号。本人在项目中担任系统架构设计师，负责本项目的架构设计工作。在本系统的开发设计过程中，我们严格遵循架构规则与过程，得到一个适合本系统的高可靠性架构，确保项目的准时上线稳定运行，提升用户满意度，从而获得用户的一致好评。

本文以同城一码项目为例，介绍软件的可靠性设计在本项目中的应用。在同城一码软件系统开发项目中，提高软件的的可靠性设计方法有避错设计、检错设计、容错设计。我们也为开发团队从结构上解决了业务难题，保障项目的顺利进行，为以来类似的系统开发积累了宝贵的经验。

正文：

我就职于某市的公交系统中，长期以来，我市的公交、地铁、各大景区有自己独立的二维码，各单位自行管理，但这对广大市民用户带来了多应用多入口的问题，市民通勤需要使用公交软件，地铁软件，游玩景区需要关注各大景区公众号预约。2023年4月，为积极满足市民用户需求，我所在单位启动同城一码项目的开发，项目工期为9个月。本项目意在将各大公共设施过闸机二维码集中于一体，做到一码可全市同行。用户只需下载一个应用即可凭此应用全市通行。公司领导层经过开会决议，由我部门牵头设计实施该项目。我作为系统架构设计师主要负责项目架构设计和管理工作。

基于对可靠性的设计，同时考虑到未来的功能扩展。同城一码项目基于JavaEE开发，前后端分离，前端开设微信小程序、支付宝小程序、安卓应用、IOS应用，同时基于React.js搭建网页端，网页端使用Ant Design提供组件库支持；后端统一开发平台使用IDEA集成开发平台；开发环境使用JDK1.8+Java8构建，同时使用SpringBoot搭建后端框架；数据库采用MySql，管理数据和持久化使用MyBatisPlus工具；部署环境使用CentOS；使用Nginx实现负载均衡。并且使用Redis作为分布式缓存，用户数据，开放公示等数据存放到Redis中，减少对数据库的请求，提高系统响应速度，前后端通信使用HTTPS进行通信，通信内容基于JSON的序列化和反序列化。

项目开始之初，我们通过对市民用户的需求调研，需求分析，收集资料以及公司领导开会谈论项目需求，最后明确系统需要具备一系列质量属性，如负载正常的情况下，系统需在0.5s内响应用户请求，系统宕机需要在5s内进行过备用站点切换，项目必须满足365天正常运行（高可靠性），系统故障必须于2小时内修复。其中高可靠性尤为重要，任何一个时间点系统无法正常运行导致的二维码无法生成进而导致市民无法出行，对公司的业务影响是极大的。对此我们对项目的可靠性进行了非常严格的设计。

可靠性是指软件系统在意外或者错误使用的情况下，维持软件系统的功能特性的基本能力，可靠性包含三个层面的含义，即成熟性、容错性、易恢复性。成熟性是指软件最好不要出现故障。容错性是指软件出现故障依然不影响主要的功能及业务。即使软件对故障的容忍度。易恢复性是指软件出现故障影响主营业务后，软件系统具有尽快定位故障点并恢复的能力。如在同城一码项目中，主要的功能性业务生成过机二维码必须保障不受影响。对此我们分析多种提高可靠性的技术。常见的由避错设计、检错设计以及容错设计。

避错设计：预防是提高可靠性的首要方法，我们都希望系统能稳定正常的运行，因此，提前发现问题，解决问题显得尤为重要。在开发阶段我们严格控制代码规范，采用代码走查、代码审查等方法，提前规避问题。同时我们坚持开发测试并重，子模块开发完成进行单元测试，系统开发完成进行系统测试、确认测试、回归测试等。在系统交付前，我们规避了绝大部分错误。

检错设计：由于不能完全避免错误的发生，我们需要系统能拥有自我检测错误并定位错误的能力。检错主要分为主动检测与被动检测。主动检测是主动检查程序状态，如CPU使用情况等。被动检错只指程序具有错误的监测能力，当系统疑似受到攻击时，发出预警，如常见的XSS攻击、DoS攻击等，当用户频繁访问该系统，系统可以及时发现并且提示用户操作过于频繁，同时短时间内拒绝该用户的请求。在同城一码项目中，我们做对XSS攻击做出了处理，在用户提交自定义信息时，对用户信息进行格式检验。同时对DoS攻击作于故障报警功能，若系统出现频繁长时间的报警，将禁止本用户的所用服务，待审核无问题后恢复该用户使用。

容错设计：容错设计系统最后的保障，当个系统出现故障时候，系统具备一定的故障容忍度。常见的容错设计有N版本程序设计、恢复快设计、冗余设计。

1. N版本程序设计：设计多个模块或不同版本，在同样的初始条件下运行。得到结果，通过多数表决判定结果的正确性，进行前向恢复。
2. 恢复快设计：恢复块设计是每次摸出处理结束时，都要检验结果的正确性，一旦发现问题则启用备用模块替换运行，进行后向恢复。
3. 冗余设计：在一套完成的系统之外，用不同的方法设计一台备用系统，当软件系统发生故障时替换运行。但此法成本太高。

在同城一码系统项目中，我们采用了恢复快设计和冗余设计。在主要生成二维码步骤，我们设计了两个生码模块，当其中主生码模块功能异常则启用备用生码模块代替使用。由于项目的重要性，关乎本市市民出行，项目同时也采用了冗余设计，我们部署了两台本地服务器为用户提供服务，机房配备供电系统，在停电时，发电机继续为服务器供电。备用服务器每隔5s向主服务器发出心跳信号，主服务器作出响应则表示其正常运行。如果主服务器连续3次不响应备用服务器的心跳信号，备用服务器将会代替接管主服务器运行。同时对系统运维人员发出警报，尽快抢修服务器。

通过全面细致的设计，整个开发项目耗时9月最终按时上线并且稳定运行至今。系统上线后得到市民用户和领导的一直好评。项目的成功与我始终严格遵循系统架构规则和过程进行开发设计是没密不可分的。但是我们并不满足于现状，发现了很多不足，如在B/S设计的基于React.js的前端网页中，出现首页查询性能不佳的问题，我们最后对前端项目进行代码分割，并进行按需加载。同时利用CDN提高用户访问网站的响应速度，经过多种优化，系统的响应速度大幅提升。

在该项目中，我认识到，软件架构不只是技术，更重要的是解决用户的核心关注点，遵循一定的系统架构过程和原则才能设计一个好的系统框架，让我们的系统具有更高的可用性、可扩展性。本人更进一步的了解到系统架构设计的重要性。也为今后类似的系统设计积累了宝贵的经验。在以后的系统架构工作中，我将继续努力，为祖国的信息化建设贡献绵薄之力。