2019年7月，我参与了某医疗信息系统有限公司移动护理项目的开发，担任项目技术负责人，负责系统整体架构设计及开发工作。该系统主要包含患者信息查对，医嘱执行，体征数据采集，护理文书管理等功能。本文以移动护理项目为例，介绍了软件系统架构风格在项目中的具体应用。在系统层次设计时，采用了层次结构风格，将系统分成持久化层、服务层、控制层以及表示层。在系统通知提醒功能设计时，采用隐式调用风格，降低了系统之间的耦合度。在医嘱扫码执行功能设计时，采用解释器风格，实现了医嘱条码规则的自定义。通过以上架构风格的使用，我们完成了软件项目开发工作，保证了项目质量。最终项目在某省三甲医院成功落地，而后也陆续收到其它医院的上线申请。

移动医疗时代的到来，为实现医疗无纸化，实现医嘱的全生命周期的跟踪，有效解决护士站到病房的＂最后１０米＂的信息延伸问题提供了更好的解决方案．最终达到改变医疗的存在方式，改善医疗业务流程，提升医疗服务质量，提高医疗安全性的目的．

2019年７月，某医疗信息系统有限公司启动了移动护理系统项目。此系统以电子病历系统ＥＭＲ和医嘱录入系统ＣＰＯＥ为平台，借助无线手持终端设备PDA进行信息的实时化传输和移动采集，实现患者信息查对，医嘱执行，体征数据采集，护理文书管理等功能．患者信息查对主要实现PDA扫描患者床头卡二维码或患者腕带条形码查询患者基本信息，核对患者身份；医嘱执行主要实现PDA扫描药品信息二维码，进行药品领取核对、记录护士医嘱执行过程，完成皮试药品双人核对，进行检验样本管扫码采集等功能。体征数据采集主要实现体征任务提醒，患者体征数据录入存库。护理文书管理主要实现患者量表，特护单，通用单，评估单等文档的编辑。我在项目中担任技术负责人，负责系统整体架构设计及开发工作，最终项目在某省著名三甲医院成功落地。

架构风格用于描述系统的组织方式和惯用模式，反映了多系统共有的结构和语义特征，强调了架构设计的复用。对于系统的分析和实现过程起着至关重要的作用。常用的软件架构风格有数据流风格、调用/返回风格、独立构件风格、虚拟机风格、数据仓库风格。数据流风格包括批处理序列及管道-过滤器，交互性差，数据分阶段处理，上阶段的结果是下一个阶段的输入。调用/返回风格包括主程序/子程序风格、面向对象风格及层次结构风格，系统层次明确，能够降低系统耦合度。独立构件风格包括进程通信及事件驱动，构件风格特性为重用提供基础服务。虚拟机风格包括解释器及基于规则的系统，构建自定义语言的执行引擎，具有良好的扩展性及灵活性。数据仓库风格包括数据库系统、黑板系统及超文本系统，以数据为中心，其它职能围绕数据展开。

我们综合考虑以上架构风格的优势特点，并认真分析了移动护理项目的实际业务场景，决定应用多种架构风格进行移动护理项目的开发。下面分别从层次结构、隐式调用及解释器三种架构风格在项目中的应用展开具体说明。  
1.层次结构风格

为了使系统整体各项职能清晰合理、便于扩展维护，我们整体采用层次结构风格。整体系统被分为数据存储层、持久化层、服务层、控制层以及表示层。数据存储层采用了Oracle数据库用于存储系统功能模块的基础数据和业务数据。例如患者基本信息、医嘱药品信息以及医嘱执行过程信息等。持久化层采用了MyBatis开源框架，能够灵活的使用SQL语句，对数据存储层的信息进行增删改查的操作。服务层主要进行各种业务逻辑的处理，比如患者医嘱信息获取，默认情况下从CPOE系统获取，也可能从医院的其它系统中获取。因此，我们遵循依赖倒置原则，面向接口编程，提供医嘱信息获取接口，进行默认实现，医院根据情况进行本地化接口实现。控制层是表现和服务层的接口，用于将表现层的请求转发给服务层进行业务逻辑处理，再将请求结果返回给表现层。表现层是用户交互的界面，向用户显示相关数据，并接收用户的输入数据。我们主要实现了移动端表现层、Web端表现层、H5端表现层。移动端用于护士医嘱扫码执行相关交互；Web端用于管理员进行系统配置交互；H5端用于护理文书功能的相关交互。通过层次结构风格的使用，系统各层各司其职。  
2.隐式调用风格

在医院的实际场景中，对于新患者入院的事件需要及时关注，因此，系统需要提供新患者入院通知提醒功能。患者入院的相关的业务流程由HIS系统实现，通过远程过程调用实现通知，系统之间的耦合度过高，容易相互影响。针对这一场景，我们决定采用隐式调用风格，来降低服务之间的耦合度。综合考虑各种常用的消息中间件，我们决定采用ActiveMQ的发布订阅功能实现事件订阅和触发。HIS系统完成患者入院相关操作后，将新入院患者事件发布到ActiveMQ中。移动护理系统对新患者入院的事件进行订阅，当有新患者入院时，就能够接收到ActiveMQ的通知。护士通过移动护理系统接收到新患者入院的提醒，可以立即采取响应的医疗措施。通过隐式调用风格的使用，降低了系统之间的耦合度，具备更好的维护性和扩展性。  
3.解释器风格

移动护理使用PDA扫描二维码识别患者信息和医嘱信息，并且针对不同类型的医嘱有不同的执行流程。比如口服类型医嘱，直接扫码完成医嘱执行，而输液类型医嘱，第一次扫码表示输液穿刺动作，再次扫码表示输液拔针动作并且完成医嘱执行。因此，二维码中的数据表示不同的信息，对应不同的业务执行流程，需要对二维码信息进行解析。针对这一场景，我们决定采用解释器架构风格。我们定义了一系列二维码信息的语法规则，使用PDA扫码二维码获取信息后，根据规则进行解析。如果解析为患者信息码则查询患者基本信息，进行身份核对，如果解析为医嘱信息码则调用相关医嘱执行流程，完成医嘱执行操作。通过解释器架构风格的使用，大大增强了医嘱条码的灵活性。

基于软件架构风格的运用，项目顺利开发完成，于2021年1月在某省三甲医院的成功上线。项目运行期间，各项性能指标均达标，系统保持95%的使用率，获得患者，护士及医院领导的一致好评。可以说明系统软件架构风格在此项目中的运用是一次成功的实践．为我们提供了一套理论结合实践的成功经验。

尽管如此，我们为了减轻数据库服务器压力而采用了Redis缓存数据库，但在系统运维中发现，部分针对缓存的查询，未在缓存系统中找到对应的key,从而引发了大量对数据库服务器的查询请求，最严重时甚至导致了数据库服务宕机。一开始解决方案是在数据库中未查走到该key时，在缓存系统中key设置空值，防止对数据库服务器发起重复查询。但后面发现，缓存中有大量的key值都为空，消耗大量内存资源，导致系统的性能急剧下降。最终我们采用布隆过滤器，将所有可能非空数据的key值哈希到bitmap中，查询缓存之前，过滤掉不在bitmap中的key值，从而避免对缓存和数据库的查询。通过这次软件开发工作，让我认识到优秀的理论知识和实践相结合才能成就好的软件项目，这敦促我在工作中也不忘提升理论知识，最终成为一名优秀的软件架构师。