内大信件管理系统

体系结构

指导教师：孟和吉雅

成员：王林 齐琦 张淑云 史泽弘

2018 年 3月 27日

在软件分析与设计中，我们通常会使用“4+1”模式，也就是：逻辑视图、开发视图、物理视图、处理视图再加上场景视图。以下是各视图的简单介绍：

**逻辑试图：** 用来描述系统的功能需求，即在为用户提供服务方面系统所应该提供的功能。在逻辑视图中，系统分解成一系列的功能抽象、功能分解与功能分析，这些主要来自问题领域（ProblemDefinition)。在面向对象技术中，表现为对象或对象类的形式，采用抽象、封装和继承的原理。用对象模型来代表逻辑视图，可以用类图（Class Diagram）来描述逻辑视图。借助于类图和类模板的手段 ，类图用来显示一个类的集合和它们的逻辑关系：关联、使用、组合、继承等。相似的类可以划分成类集合。类模板关注于单个类，它们强调主要的类操作，并且识别关键的对象特征。

**处理视图：** 过程架构考虑一些非功能性的需求，如性能和可用性。它解决并发性、分布性、系统完整性、容错性的问题，以及逻辑视图的主要抽象如何与进程结构相配合在一起，即定义逻辑视图中的各个类的具体操作是在哪一个线程（Thread）中被执行。过程视图侧重系统的运行特性。服务于系统集成人员，方便后续性能测试。

**物理视图：** 主要描述硬件配置。服务于系统工程人员，解决系统的拓扑结构、系统安装、通信等问题。主要考虑如何把软件映射到硬件上，也要考虑系统性能、规模、可靠性等。可以与进程视图一起映射。物理架构主要关注系统非功能性的需求，如可用性、可靠性（容错性），性能（吞吐量）和可伸缩性。

**开发视图：** 描述了在开发环境中软件的静态组织结构，即关注软件开发环境下实际模块的组织，服务于软件编程人员。将软件打包成小的程序块（程序库或子系统），它们可以由一位或几位开发人员来开发。子系统可以组织成分层结构，每个层为上一层提供良好定义的接口。

**场景视图：** 它综合所有的视图。用于刻画构件之间的相互关系，将四个视图有机地联系起来。可以描述一个特定的视图内的构件关系，也可以描述不同视图间的构件关系。四种视图的元素通过一组重要场景（更常见的是用例）进行无缝协同工作，我们为场景描述相应的脚本（对象之间和过程之间的交互序列）。在某种意义上场景是最重要的需求抽象，它们的设计使用对象场景图和对象交互图来表示。

本系统为《内大信件管理系统》，主要用于内大日常收取信件所使用，需求分析已经在之前的需求分析文档中介绍过。下面来谈谈该系统的“4+1”模式设计。

**逻辑视图：**本系统主要使用人员为传达室工作人员。主要包括用户登录，信件登记，收信人登记及核对功能。

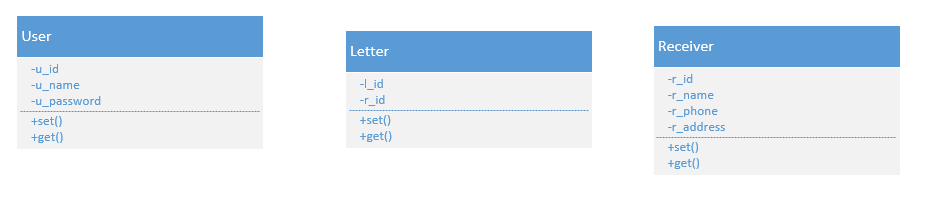


图1.1 实体类图

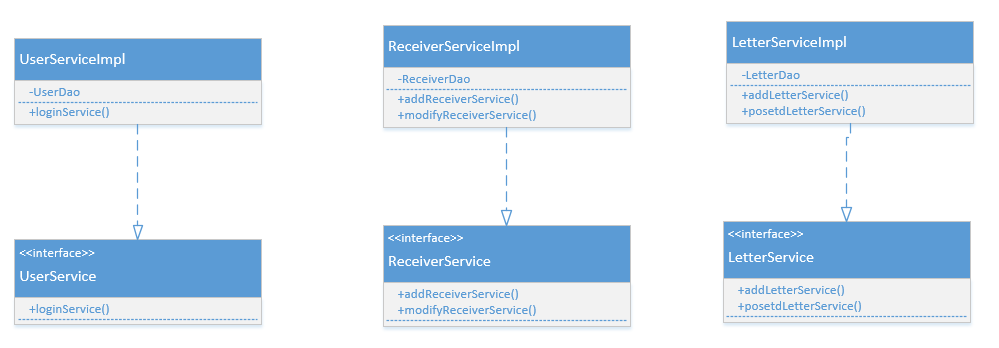


图1.2 实现类图

**处理视图：**本系统主要面向传达室工作人员，由于岗位的特殊性，不存在大并发的可能性。

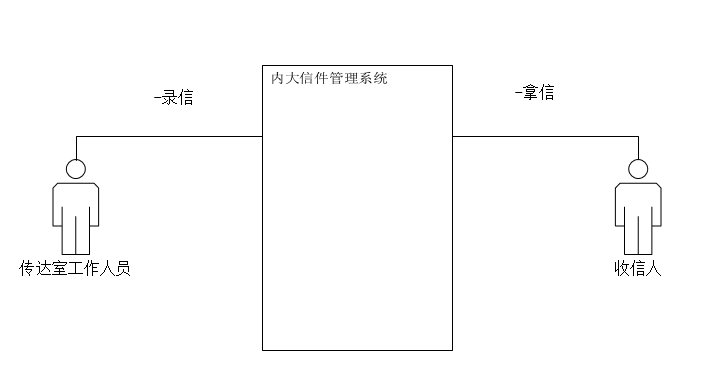


图 1.3 系统体系处理视图

**物理视图：**将做好的系统部署到一台或多台PC机上， 传达室工作人员通过Web服务来连接应用程序服务器以便使用信件管理系统。

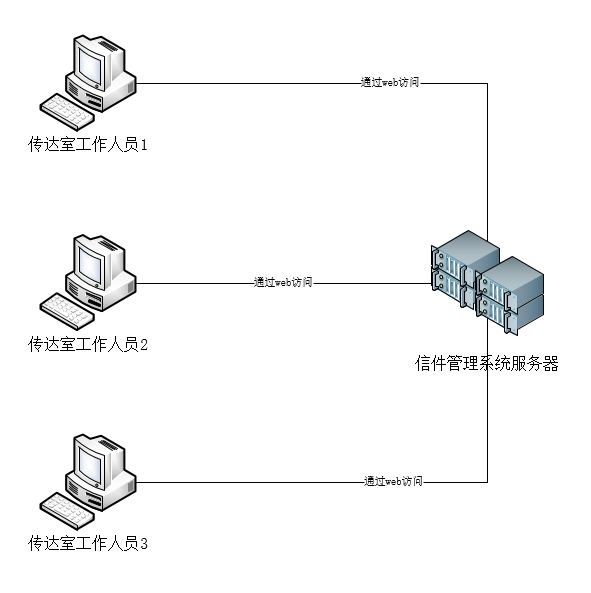


图1.4 物理视图

**开发视图：**本系统采用基于MVC的SSH架构来实现。主要分为三层，Aation(Struts2)主要负责处理前端请求以及将结果返回前端。Service层主要负责业务逻辑处理，比如登录存在用户名不存在，或者密码错误等结果，其处理就在Service层。Dao(Hibernate)层负责程序与数据库的交互，及增删改查操作。而Spring是一个无侵入式框架，主要负责各层间的通信。还有一个VO类，即将数据库每张表映射为一个实体类，以供系统使用。

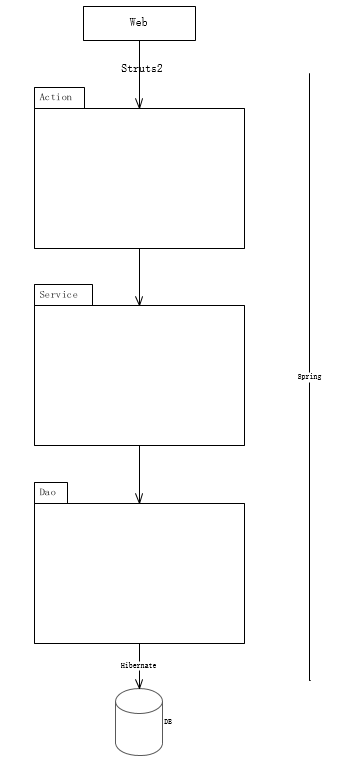


图1.5 系统体系开发视图

**场景视图：**本系统使用场景主要为传达室人员的信件收取。

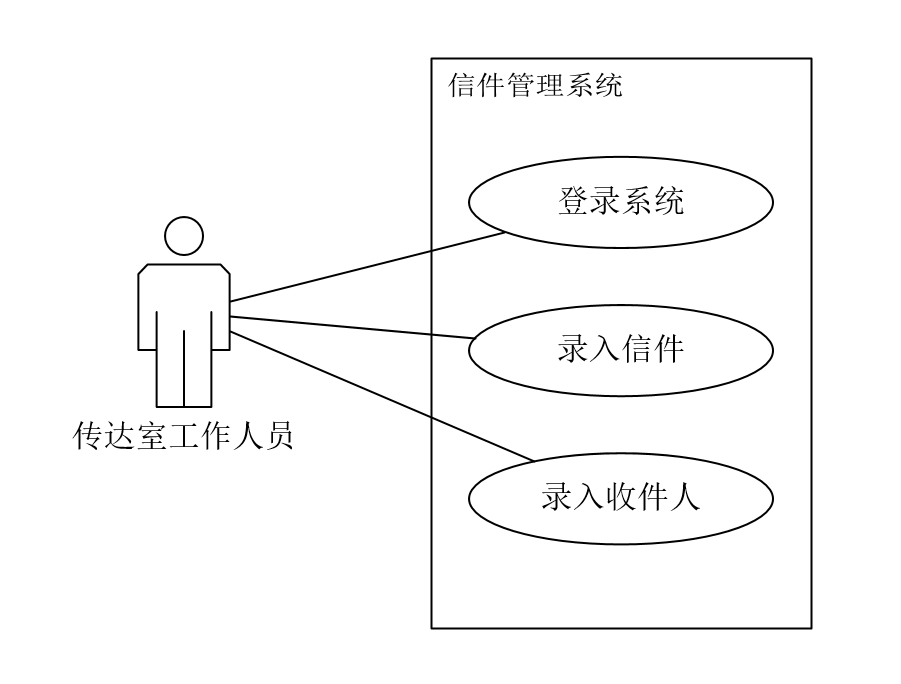


图1.6系统用例图

**总结**

以上就是《内大信件管理系统》“4+1”模式视图的分析。通过此次分析，使系统的逻辑及开发进度具有的很好的透明性和计划性。提高的软件开发的稳定性。