**基于Spring的社团管理系统的设计与实现**

**前言**

**社团管理系统，即社团办公自动化系统，简称OA，是将社团办公和互联网结合起来形成的一种办公管理系统，最早的OA系统是硬件配置阶段，主要以提高办公效率为目的。在最近的几年里，协同办公系统迅猛发展，以阿里的钉钉和移动的和飞信为首的OA系统已经霸占了国内乃至国际的市场。**

**我们的系统虽然基于协同办公系统，但是主要是通过结合社团的实际需求，开发的适用于本社团的系统。**

**1. 开发工具和相关技术介绍**

**1.1 IntelliJ IDEA 简介**

**IntelliJ IDEA简称IDEA， 是一个java开发集成环境，在业界被认为是Java最好的开发工具之一，支持自动代码补全、各种依赖管理工具，JUnit单元测试工具、各类版本工具(git、svn、github等)、diff、 等各种领先功能。目前为止IntelliJ IDEA稳定版已经更新到了2018.1.1，基本上每年都会更新3次，可见对IDEA的支持。IntelliJ IDEA问世之前开发Java应用大家普遍使用的Eclipse，接下来我们通过与Eclipse的对比来进行对IntelliJ IDEA的简要介绍：**

**1. 首先最大的优势在于IntelliJ IDEA的智能化，整个软件使用下来给人的感觉就是一个舒服，就像碰到了一个知心朋友；**

**2. 启动速度方面，IntelliJ IDEA完美的解决了Eclipse经常卡死的现象，而且代码自动保存不会丢失代码；**

**3. 完美整合了各种版本控制系统，如：Git和SVN，在本系统即使用Git，且将代码托管到GitHub；**

**4. 终端内置了，作为程序员，终端是常用的工具，这可以极大的提升开发人员的开发效率。**

**IntelliJ IDEA 的优点还有非常多，篇幅限制，在这里只介绍上述4条。**

**1.2 Java 技术介绍**

**java是一种开源的面向对象且具有良好可移植性和健壮性分布式动态解释型语言。**

**Java具备面向对象的一切优点，是使用人员最多的一种开发语言。**

**1.3 Spring介绍**

**Spring是一个开源的一站式java开发框架，用于降低业务逻辑层与其他各层之间的耦合度，整个框架均面向接口编程。Spring具有以下的优点：**

**1. Spring具有其他流行框架所不具有的定位领域，它提供了一种能管理业务对象的方法。**

**2. Spring是模块化的，也就是说你可以独立的使用它的任一部分并且仍然是稳定的，例如你可以使用它管理对象，或者简化数据库操作。**

**3. Spring拥有许多特性，如利用控制反转（IOC）技术降低耦合度。**

**4. Spring支持面向切面编程（AOP），可以对单个方法进行增强。**

**综合上述优点，本应用采用Spring框架进行开发，并会在系统实现模块详细介绍使用方式。**

**1.4 JBPM简介**

**JBPM，即业务流程管理，它是包含业务整个流程的一个开源灵活易扩展的可执行流程语言框架。下面对于用到的内容做一下简要的介绍：**

**1. 业务逻辑**

**它的业务逻辑采用了类似于uml状态图的流程逻辑，包括各种状态以及状态之间的转换，通过图的流程化来描绘业务流程。**

**2. 与Hibernate结合**

**可以与Hibernate一起使用。数据管理部分由Hibernate来管理，JBMP专注于业务流程处理。更好的进行职能分离。**

**3. 流程概念介绍**

**工作流概念抽象了流程定义，其并没有确切的执行者。而当其有执行者的时候，这个流程就具体化了，例如张三借款：张三为借款人，李四是张三的部门领导则为审批人，王五作为财务，这样一个流程就具象化了。在这样一个过程中，张三、李四、王五均为参与者，过程中需要介入的环节如李四审批则为一个任务，每个任务被称为任务实例，从一个任务到另外一个任务则称为流转，在流程中被预设的行为叫做活动。**

**2. 系统需求分析**

**2.1 系统功能需求**

**用户打开系统后，通过登录/注册操作，进入主界面，根据不同用户的权限显示不同的操作项目。默认进入主界面，可点击展开某一需要操作的具体项，图示为相关的用例图：**

**通过系统管理员用户登录后，主页面会展示3个大项，分别是【系统管理】、【网上交流】和【审批流转】，整个界面分为五部分，上下左右中，每个部分展示不同的内容。**

**2.1.1 登录/注册页面**

**首次进入系统时进入登录页面，没有账号的用户应通过注册按钮进入注册页面来进行注册，注册页面需要重复输入密码来确保用户密码输入的准确性。登录或注册完成之后直接进入系统主界面。**

**2.1.2 主界面**

**（1）系统管理**

**【系统管理】模块，点击后会展开为角色管理、部门管理、用户管理三个小项。**

**（2）网上交流**

**【网上交流】模块，主要包括论坛和管理员管理论坛使用的论坛管理两个项目。**

**（3）审批流转**

**【审批流转】模块，包括个人使用的起草申请、我的申请查询和管理员使用的待我审批、审批流程管理，申请流程管理五个小项。**

**（4）个人设置**

**【个人设置】界面，包括个人资料修改及修改密码两个操作。**

**2.2 可行性分析**

**下面通过三个方面进行可行性分析。**

**2.2.1 技术可行性**

**Java语言可适用于各种计算机平台，可进行跨平台开发，广泛用于系统级开发。并且Java语言具有封装、继承和多态的特性。**

**（1） 面向对象**

**Java是一种面向对象的编程语言，与面向过程同样是程序设计方法的一种。面向对象类型的编程语言核心之一是开发人员可以在设计软件时使用自定义类型（对象）和相关操作（方法）。一个对象可以看作是由许多“行为（方法）”和“状态（属性）”组合而成的对象。对于数据结构的变化，我们需要通过方法来操作属性。面向对象的设计使得大型软件工程的规划和设计更易于管理，流程更加的规范，增强了工程的健壮性，减少了项目失败的概率。**

**（2） 封装**

**封装的定义就是把对象的属性和实现的细节隐藏起来，只对外暴露接口，控制各属性和方法的访问权限，然后通过接口操作对象。**

**（3） 继承**

**继承的主要目的就是实现代码的复用，即当两个类具有相同的属性或方法时，可以将相同的部分抽取出来放到一个共同的类中作为父类，这两个类继承这个父类。继承后子类自动拥有了父类的属性和方法，这样同样的代码就不必重复写多次，达到复用代码的目的。但特别注意的是，父类的私有属性和构造方法并不能被继承，所以需要继承的部分不能使用私有属性来声明。并且Java也有相应的权限控制子类和父类的访问机制。**

**（4） 多态**

**多态就是指程序中定义的变量所指向的具体类型或调用的方法在编程时并不明确，只有在程序运行时才确定，即一个变量到底会指向哪个类的实例对象，以及调用哪个对象的方法是不确定的，必须在由程序运行期间才能决定。在程序运行时才会确定具体的类，这样带来的优势就是：不用修改源程序代码，就可以让变量绑定到不同的实现类上，从而该引用调用的具体方法就会改变，即不修改程序代码就可以改变程序运行时所绑定的具体代码，让程序可以选择多个运行状态，这就是多态性。**

**上述分析后，在技术层次上可以可行的， Java语言作为系统级开发语言是可以的。**

**2.2.2 硬件条件的可行性**

**做Java开发，只需要普通家庭电脑就可一，由于要运行IDEA，内存占用较大，所以建议内存采用大于4GB，128GB的固态硬盘的解决方案。开发成本相对较小，所以在硬件方面是可行的。**

**2.2.3 用户使用可行性**

**本项目对于用户的要求，除了掌握使用电脑必备知识外，并不需要其他特别的知识，且使用对象主要为在校大学生，更不要对用户进行使用培训。**

**所以对于用户的使用是可行的。**

**3. 系统功能设计**

**3.1 系统总体功能设计**

**3.1.1 系统结构**

**根据以上对系统功能需求进行分析的基础上，基本上确定了本应用的各功能模块，如图所示：**

**3.1.2 系统总体设计**

**根据上述的功能需求，将功能分为5块，就是【系统管理】、【网上交流】、【审批流转】、【其他系统设置】和【注册、登录】，整体采用MVC的设计方案，包结构如下：**

**采用标准的分层方案，即：由jsp页面调用action， action调用service，service调用dao、dao进行CURD操作。**

**View，在本应用中是作为一个页面出现的，每个View都有一个相应的action负责控制，数据（Model）全部通过action显示到View上。**

**系统的初始化操作将在安装时完成，将完成初始化界面、初始化管理员用户和设置权限等的初始化操作。**

**3.1.3 登陆/注册部分**

**该部分在设计层次较为简单，重点需要做好的就是密码二次验证和用户重复性校验问题，实现中将详细说明。**

**3.1.4 系统管理部分**

**【系统管理】部分是对整个系统的一个管理界面，包括：**

**用户管理：可分配不同角色、部门及初始化密码操作。**

**部门管理：部门部分采用了树状结构显示，可针对不同层级的部门分别设置。**

**角色管理：可以通过对不同用户、不同角色分别设置某个父模块子模块权限。**

**3.1.5 网上交流部分**

**【网上交流】是用于社团内成员交流的一个模块，包括论坛及论坛管理。**

**具有管理权限的人员可以对论坛版块进行调整，通过调整不同的板块的顺序，来控制显示优先级。可以为专门的需求去创建专门的版块。**

**进入论坛界面时，会显示不同板块的最后发帖信息，以便于不错过任何新的消息。还会有每个板块的统计信息，包括文章数，贴数（包括回帖及发帖）。选择某个版块进入后，还能对不同的信息进行过滤，帖子可分为普通、精华、置顶等不同的状态。支持各种过滤操作。在发帖时，也采用了文本编辑器，采用了类似word的操作，更好的支持多种文本样式，而且对编辑器进行了防注入设计，弥补已知缺陷。**

**3.1.6 审批流转部分**

**【审批流转】部分是一个OA系统的核心，我们的系统也不例外。这个模块将流程定义，流程管理，发起，查看，审批都囊括在内，将日常的申请流程都在线化，可以满足社团的日常办公需要。**

**3.1.7 个人设置部分**

**个人设置部分，将一些个人信息和修改密码等常用功能放在这个位置。**

**3.2 数据库设计**

**数据库设计主要与实体类对应，本系统在设计的时候采用的是正向设计， 即先设计实体类、映射文件，然后建表。对于每个数据库表对应的都是本地的实体类，以Topic类为例，它的父类是Article，由于需要以对象的形式进行传输，所以实现了序列化的接口。**

**Topic类的映射文件：**

**^hibernate-mapping^**

**^!--配置类和表对应 name属性：实体类全路径 table属性：数据库名称--^**

**^class name="com.ezra.gechuang.domain.Topic" table="t\_topic"^**

**^!--配置实体类id和表id对应，hibernate要求一个属性具有唯一值--^**

**^id name="id"^**

**^!--设置数据库表id增长策略**

**navtive:设置id就是主键自增长--^**

**^generator class="native"^^/generator^**

**^/id^**

**^property name="title"^^/property^**

**^property name="content" type="text" length="65535"^^/property^**

**^property name="faceIcon"^^/property^**

**^property name="postTime" type="timestamp"^^/property^**

**^property name="ipAdd"^^/property^**

**^property name="type"^^/property^**

**^property name="lastUpdateTime" type="timestamp"^^/property^**

**^property name="replyCount"^^/property^**

**^!--author,本类与User的多对一关系--^**

**^many-to-one name="author" column="authorId" class="com.ezra.gechuang.domain.User"/^**

**^!--forum,本类与Forum的多对一关系--^**

**^many-to-one name="forum" column="forumId" class="com.ezra.gechuang.domain.Forum"^^/many-to-one^**

**^!--replys,本类与Reply的一对多关系--^**

**^set name="replys"^**

**^key column="topicId"^^/key^**

**^one-to-many class="com.ezra.gechuang.domain.Reply"/^**

**^/set^**

**^!--lastReply,本类与Reply的一对一关系--^**

**^many-to-one name="lastReply" column="lastReplyId" class="com.ezra.gechuang.domain.Reply" unique="true"/^**

**^/class^**

**^/hibernate-mapping^**

**Topic类对应的数据库表如图所示：**

**其他实体类的实现与本类类似，所以本文不再赘述。**

**4. 系统的实现**

**根据上一章节中的系统功能设计，在本章将主要介绍各系统功能的实现。**

**4.1 系统初始化操作**

**由于我们的系统使用了SSH框架，easyUI，fckeditor，JBPM等框架，分别针对不同的流程。他们的初始化尤为重要，更是实现本系统的大前提，接下来我们将介绍这些框架的初始化操作。**

**4.1.1 SSH框架整合**

**SSH框架主要包含Spring、Struts2、Hibernate，其中Struts2主要用在web层，用的最多的是action；Spring主要用在service层，用的最多的是控制反转和面向切面编程，以及对象的管理操作；Hibernate主要用在dao层，用于对数据库进行CRUD操作。**

**主要整合思想为两两整合，通过Spring作为承接点，将Struts2中的action等创建操作交给Spring进行管理，需注意多实例问题；将Hibernate中的Session工厂交给Spring管理，将数据库配置也交给Spring管理，并且管理dao对象。这里的整合都是通过jar包进行整合的。**

**4.1.2 JBPM配置**

**（1）准备工作所需环境**

**需要环境如下：jdk1.5+；ant1.7.0+；tomcat；mysql，由于本系统开发所使用即为jdk1.8，tomcat7.0.56，mariaDB10.1.13,所以只需再下载ant即可。**

**（2）准备开发环境**

**在maven中添加所需jar包；**

**添加并制定配置文件；**

**初始化数据库；**

**4.2 系统各界面的实现**

**4.2.1 登录/注册页面**

**jsp页面本身没有什么可说，所以，只是简单演示一下登陆注册逻辑，下面的代码是省略后的代码，只有大体的框架：**

**4.2.2系统管理部分**

**本部分的角色管理中，采用了jqery中的treeview的方式，来对权限进行展示，效果如图：**

**下面是点击选中时的处理代码逻辑：**

**在treeview中，每个节点都有其父节点，子节点，兄弟节点的概念，对每个节点的修改等操作，都要考虑对其关联节点的影响。关于这点实现起来还是需要比较细心地，不然实现的就会比较混乱。**

**4.2.3 论坛部分**

**根据论坛模块的设计，它的实现比较复杂，该部分将分步骤详细说明：**

**（一） 界面实现**

**首先界面如图所示：**

**如图所示，界面分为展示区，筛选过滤区，分页等基本操作部分。由于在本系统中多个地方都用到了分页操作，所以也将此代码提取成了一个公用的代码。**

**在这个模块中，为了优化数据库操作，增加了一些特殊字段，如：**

**4.2.4 审批流转部分**

**审批流转部分是整个系统的重中之重，因此，对整个审批流程进行了详细的调查研究，下面是研究结果：**

**1。审批开始：开始应该是提交申请，然后是审批环节。审批者的数量和审批者是由不同的过程决定的。这是造成了审批流程的不同。**

**2。当填写完申请后应该是“提交申请”，申请人默认使用当前用户。**

**三。在我们的社团管理系统中，每个用户都是唯一的一个对象，在JBPM中，每个人也是采用一个字符串的形式来表示。所以在本系统中统一使用对象的用户名作为标识符。**

**4。在“提交申请”时需要保存本次申请的信息，而且在后面的所有环节中都应该能方便的获取到本次申请的全部信息，以便于查看与审核使用。JBPM中的流程变量即可以满足这个功能：在“提交申请环节”，会生成一个form对象，然后把form对象设置为流程变量，以后的所有环节只需要取出这个变量，就能得到了申请信息。于是“提交申请”的环节中的申请人就可以写成：{form.user.uName}**

**4.2.5 我的界面**

**根据设计，主要是个人信息和修改密码部分，实现逻辑比较简单，因此不再赘述。**

**4.3 其他注意事项**

**4.3.1 Android 7.0 FileProvider适配**

**参考文献**

**[1] Bruce Eckel.Java编程思想 [M]. 陈昊鹏 译. 4版 北京：机械工业出版社，2007.**

**[2]James Gosling，等.Java编程语言[M]. 3版. 北京：中国电力出版社，2004.**

**[3]Y Daniel Liang. Java语言程序设计 进阶篇 [M]. 万波，郑海红，潘蓉，李娜，等译. 6版. 北京：机械工业出版社，2008**

**[4]Y Daniel Liang. Java语言程序设计 基础篇 [M]. 万波，郑海红，潘蓉，李娜，等译. 6版. 北京：机械工业出版社，2008**

**[5]Gay S Horstmann，Gary Cornell. Java核心技术卷Ⅰ 基础知识 [M]. 叶乃文，等译. 8版 北京：机械工业出版社**

**[6]Gay S Horstmann，Gary Cornell. Java核心技术卷Ⅱ 高级特性 [M]. 陈昊鹏，等译. 8版 北京：机械工业出版社**

**[7]弗里曼.HeadFirst设计模式(中文版) [M].中国电力出版社, 20074**

**[8]福勒(MartinFowler),潘加字. UML精粹(第三版)[M].第一版.电子工业出版社, 2012.**

**[9]拉曼. UML和模式应用(原书第3版)[M].第一版.机械工业出版社, 2006.**

**[10]张跃平,耿祥义. JsP程序设计[M].清华大学出版社, 2009**

**[11]鲍格斯坦,林琪译. JsP设计[M].中国电力出版社, 2009**

**[12]刘京华等. JavaWeb整合开发王者归来[M].清华大学出版社, 2010-01**

**[13]Budi Kumiawan,崔毅,俞哲皆,俞黎敏译. Servlet和 JSP学习指南[M].机械工业出版社, 2013.**

**[14]郭美l1州,陈军红. SQL Server2008完全学习手册[M].清华大学出版社,2011.**

**[15]卫琳. SQL Server2008数据库应用与开发教程(第2版) [M].清华大学出版社, 2011.**

**[16]Mike Hotek,潘玉琪译. SQL Server2008从入门到精通[M].清华大学出版,,2011.**

**[17]ComelDouglas Intemetworking withTCP/「P:Vo1. I11, Client-selver programming and applications Qinghua University Press, 2006.**

**[18]Petzold.Charles Programming Windows[M].人民邮电出版社, 2009.**

**[19](美) RonPatton.软件测试[M].机械工业出版社, 2010**

**致谢**

**经过几个月的努力，终于顺利的完成了基于Spring的社团管理系统的设计与实现。在完成本项目开发设计的整个过程中，发现自己js界面写的只能算是好多年前的水平，对前端技术并不擅长，但得到了学院多位老师的帮助，特别是李寰老师更是提出了许多宝贵的意见和建议，并指出了许多不足之处，使系统功能更加完善合理，在此表示特别感谢。**