

**黑马网众筹平台系统**

**详细设计书**

*创建时间: 22-Apr-2014*

*更新时间: 22-Apr -2014*

文档版本: 1.0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本/状态** | **作者** | **参与者** | **日期** | **备注** |
| V1.0 | 王飞 | 侯毓冉、林宇、樊执政、梁炜 | 2013.8.26 |  |
| V2.0 | 梁炜 | 侯毓冉、林宇、樊执政、王飞 | 2013.9.10 |  |
|  |  |  |  |  |

|  |
| --- |
|  |

**目 录**

[第一部分 总体设计 10](#_Toc396123782)

[一、技术结构设计 10](#_Toc396123783)

[二、核心控制流程 11](#_Toc396123784)

[三、开发环境集成配置 12](#_Toc396123785)

[第二部 分界面设计 13](#_Toc396123786)

[一、界面设计 14](#_Toc396123787)

[1、web端界面设计 14](#_Toc396123788)

[1.1、首页界面设计 14](#_Toc396123789)

[1.2、登录界面设计 14](#_Toc396123790)

[1.2、浏览项目界面设计 15](#_Toc396123791)

[1.3、项目详情界面设计 15](#_Toc396123792)

[1.4、查看已关注的项目界面设计 16](#_Toc396123793)

[2、Android界面设计 17](#_Toc396123794)

[2.1、登录界面设计 17](#_Toc396123795)

[2.2、浏览界面设计 18](#_Toc396123796)

[2.3、个人信息界面设计 19](#_Toc396123797)

[2.4、项目信息界面设计 20](#_Toc396123798)

[2.5、注册界面设计 20](#_Toc396123799)

[第三部分 单元模块设计 21](#_Toc396123800)

[一、数据访问层设计 21](#_Toc396123801)

[1、类图设计 21](#_Toc396123802)

[2、类的详细设计描述 22](#_Toc396123803)

[2.1、IUserDAO接口设计 22](#_Toc396123804)

[2.2、IProjectDAO接口设计 23](#_Toc396123805)

[2.3、IUserAttentionDAO接口设计 24](#_Toc396123806)

[2.4、IUserSupportDAO接口设计 25](#_Toc396123807)

[2.5、IProjectCategoryDAO接口设计 25](#_Toc396123808)

[二、用户管理业务功能模块 26](#_Toc396123809)

[1、用户登录功能 26](#_Toc396123810)

[1.1、功能汇总 26](#_Toc396123811)

[1.2、类图设计 26](#_Toc396123812)

[1.2.1、类关系结构图 26](#_Toc396123813)

[1.2.2、时序图 27](#_Toc396123814)

[1.3、类的详细设计描述 27](#_Toc396123815)

[1.3.1、Login.jsp描述 27](#_Toc396123816)

[1.3.2、UserAction描述 28](#_Toc396123817)

[1.3.3、UserLogin描述 29](#_Toc396123818)

[1.4、功能实现 30](#_Toc396123819)

[2、用户注册功能 30](#_Toc396123820)

[2.1、功能汇总 30](#_Toc396123821)

[2.2、类图设计 31](#_Toc396123822)

[2.2.1、类关系结构图 31](#_Toc396123823)

[2.2.2、时序图 31](#_Toc396123824)

[2.3、类的详细设计描述 31](#_Toc396123825)

[2.3.1、Land.jsp描述 31](#_Toc396123826)

[2.3.2、UserAction描述 32](#_Toc396123827)

[2.3.3、UserLand描述 32](#_Toc396123828)

[3、用户个人信息管理功能 33](#_Toc396123829)

[3.1、功能汇总 33](#_Toc396123830)

[3.2、类图设计 33](#_Toc396123831)

[3.2.1、类关系结构图 33](#_Toc396123832)

[3.2.2、时序图 34](#_Toc396123833)

[3.3、类的详细设计描述 34](#_Toc396123834)

[3.3.1、UserManage.jsp描述 34](#_Toc396123835)

[3.3.2、UserAction描述 35](#_Toc396123836)

[3.3.3、UserManage描述 35](#_Toc396123837)

[三、项目管理业务功能模块 36](#_Toc396123838)

[1、发起项目功能 36](#_Toc396123839)

[1.1、功能汇总 36](#_Toc396123840)

[1.2、类图设计 36](#_Toc396123841)

[1.2.1、类关系结构图 36](#_Toc396123842)

[1.2.2、时序图 37](#_Toc396123843)

[1.3、类的详细设计描述 37](#_Toc396123844)

[1.3.1、CreatePro.jsp描述 37](#_Toc396123845)

[1.3.2、ProjectMagAction描述 38](#_Toc396123846)

[1.3.3、CreateProject描述 39](#_Toc396123847)

[2、浏览项目功能 40](#_Toc396123848)

[2.1、功能汇总 40](#_Toc396123849)

[2.2、类图设计 40](#_Toc396123850)

[2.2.1、类关系结构图 40](#_Toc396123851)

[2.2.2、时序图 41](#_Toc396123852)

[2.3、类的详细设计描述 41](#_Toc396123853)

[2.3.1、ScanProject.jsp描述 41](#_Toc396123854)

[2.3.2、ProjectMagAction描述 42](#_Toc396123855)

[2.3.3、ScanProject描述 42](#_Toc396123856)

[3、关注项目功能 43](#_Toc396123857)

[3.1、功能汇总 43](#_Toc396123858)

[3.2、类图设计 43](#_Toc396123859)

[3.2.1、类关系结构图 43](#_Toc396123860)

[3.2.2、时序图 44](#_Toc396123861)

[3.3、类的详细设计描述 44](#_Toc396123862)

[3.3.1、ProjectInfo.jsp描述 44](#_Toc396123863)

[3.3.2、ProjectMagAction描述 45](#_Toc396123864)

[3.3.3、LikeProject描述 45](#_Toc396123865)

[4、编辑项目功能 46](#_Toc396123866)

[4.1、功能汇总 46](#_Toc396123867)

[4.2、类图设计 46](#_Toc396123868)

[4.2.1、类关系结构图 46](#_Toc396123869)

[4.2.2、时序图 47](#_Toc396123870)

[4.3、类的详细设计描述 47](#_Toc396123871)

[4.3.1、MyProject.jsp描述 47](#_Toc396123872)

[4.3.2、ProjectMagAction描述 47](#_Toc396123873)

[4.3.3、EditProject描述 47](#_Toc396123874)

[四、投资管理业务功能模块 48](#_Toc396123875)

[1、查看已关注、已投资项目功能 48](#_Toc396123876)

[1.1、功能汇总 48](#_Toc396123877)

[1.2、类图设计 49](#_Toc396123878)

[1.2.1、类关系结构图 49](#_Toc396123879)

[1.2.2、时序图 50](#_Toc396123880)

[1.3、类的详细设计描述 50](#_Toc396123881)

[1.3.1、AttentionedProjectList.jsp描述 50](#_Toc396123882)

[3.2、ProjectMagAction描述 51](#_Toc396123883)

[1.3.3、CancelAttentionedProject描述 52](#_Toc396123884)

[2、投资项目功能 53](#_Toc396123885)

[2.1、功能汇总 53](#_Toc396123886)

[2.2、类图设计 53](#_Toc396123887)

[2.2.1、类关系结构图 53](#_Toc396123888)

[2.2.2、时序图 54](#_Toc396123889)

[2.3、类的详细设计描述 54](#_Toc396123890)

[2.3.1、ProjectInfo.jsp描述 54](#_Toc396123891)

[2.3.2、ProjectMagAction描述 55](#_Toc396123892)

[2.3.3、InvestProject描述 56](#_Toc396123893)

[2.4、配置文件 57](#_Toc396123894)

[五、移动平台业务功能模块 62](#_Toc396123895)

[1.1、功能汇总 62](#_Toc396123896)

[1.2、类图设计 63](#_Toc396123897)

[1.2.1、类关系结构图 63](#_Toc396123898)

[1.2.2、时序图 63](#_Toc396123899)

[1.3、类的详细设计描述 64](#_Toc396123900)

[1.4、功能实现 66](#_Toc396123901)

[第六部分数据库设计 67](#_Toc396123902)

[一、数据库整体结构图 68](#_Toc396123903)

[二、业务功能功能模块 69](#_Toc396123904)

[1、用户信息表结构 69](#_Toc396123905)

[2、项目管理表结构 70](#_Toc396123906)

[3、关注项目信息表结构 71](#_Toc396123907)

[4、支持项目信息表结构 71](#_Toc396123908)

[5、目录信息表结构 71](#_Toc396123909)

[6、站内信息表结构 72](#_Toc396123910)

[第七部分 补充设计与说明 73](#_Toc396123911)

[一、编译运行环境设计 73](#_Toc396123912)

[二、包路径与Web目录结构设计 74](#_Toc396123913)

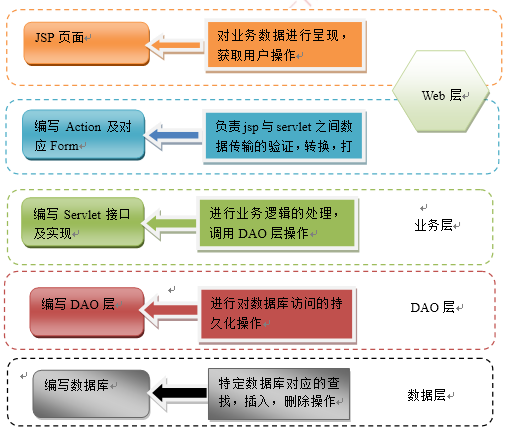
[1、包路径设计 74](#_Toc396123914)

[2、各包下详细文件设计 74](#_Toc396123915)

第一部分 总体设计

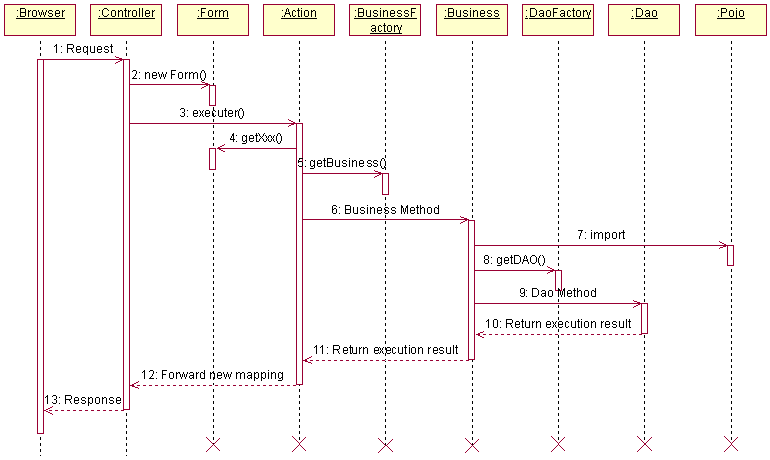
### 一、技术结构设计

我们根据小组成员的技术特点，选择了以Struts框架为主的架构设计，采用JDBC完成数据库及DAO层的设计.利用Action进行jsp页面不同请求的控制及跳转，体现了良好的维护性及可扩展性。



这样的MVC风格的架构基本可以满足平台的开发，在详细设计时需要着重考虑的是接口的设定，良好的接口将使servlet中的可能需要频繁更改的业务操作与其他DAO层，Web层相分离，提高可维护性，同样的通过Action与From也更好的封装了前端与后台间信息的传递，以及之后过滤器Filter的使用，都是力图形成接口与实现想分离，低耦合的系统架构。

### 二、核心控制流程



结合上图，针对每一个消息请求，按照消息请求/回复顺序，作如下说明。

|  |  |  |
| --- | --- | --- |
| **编号** | **消息名称** | **消息说明** |
| 1 | HttpServletRequest | 浏览器发来的HttpServletRequest 请求 |
| 2 | ActionForm | 通过ActionForm获取Struts加载的页面表单数据 |
| 3 | Execute | 根据struts-config.xml配置调用指定的Action指定的方法处理请求 |
| 4 | ActionForm.getXxx() | 通过ActionForm获取页面表单数据 |
| 5 | BusinessFactory .getBusiness | 获得Business对象 |
| 6 | Business Method | 调用业务逻辑 |
| 7 | Import | 使用Pojo进行数据操作 |
| 8 | DAOFactory.getDAO | 获得Dao对象 |
| 9 | Dao Method | 调用Dao方法，执行数据库操作 |
| 10 | Return execution result | 返回Dao执行结果，给Business |
| 11 | Return execution result | 返回Business执行结果，给Action |
| 12 | ActionForward.findForward | Action返回执行结果给新的画面（Struts-Config.xml配置） |
| 13 | HttpServletResponse | MappingDispatchAction返回处理结果给浏览器 |

### 三、开发环境集成配置

1、服务器环境

1）OS：Windows Server 2003

2）2G 以上内存，建议使用 4G 内存；

3）500GB 以上的硬盘空间，不包括数据存储空间

4）JDK：Java SE 6

5）DB：Database 10g

6）Application Server：WebLogic Server 10.3

2、WEB 客户机器环境

1）Pentium III 或以上微处理器（CPU）；

2）Microsoft Windows XP、 Windows Vista、Windows 7 操作系统；

3）512MB 以上内存，建议使用 1024MB 内存；

4）WEB：要求 IE7 以上版本，最好 IE8 以上版本。

3、Android移动平台环境

Android4.0及以上

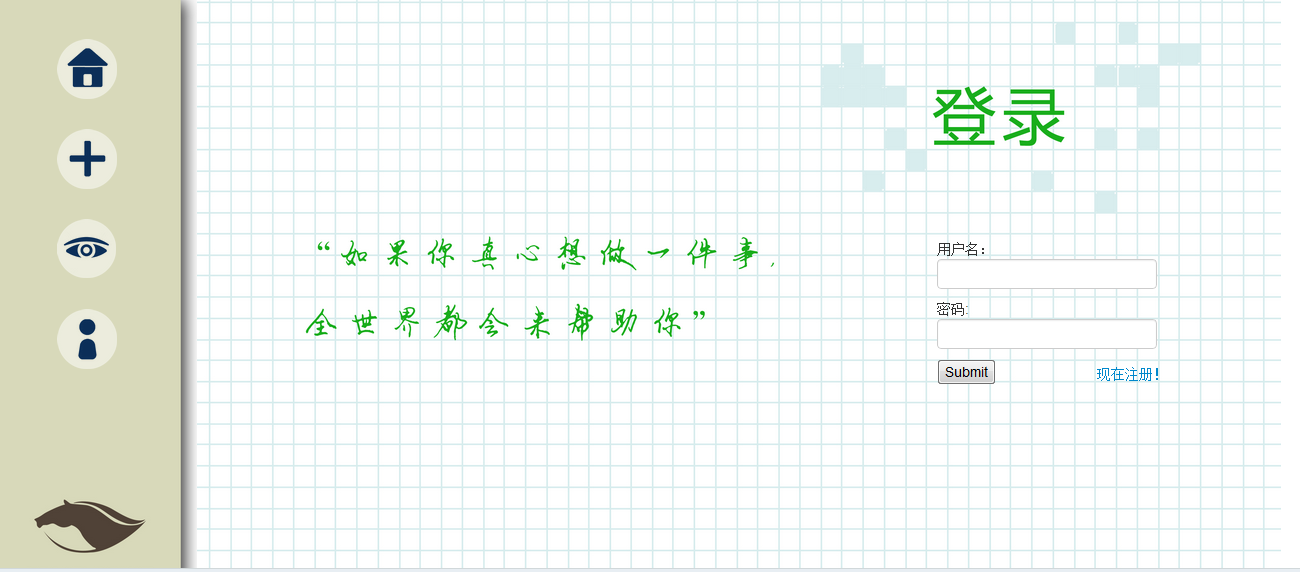
第二部 分界面设计

### 一、界面设计

#### 1、web端界面设计

##### 1.1、首页界面设计

##### 1.2、登录界面设计



##### 1.2、浏览项目界面设计



##### 1.3、项目详情界面设计





##### 1.4、查看已关注的项目界面设计



#### 2、Android界面设计

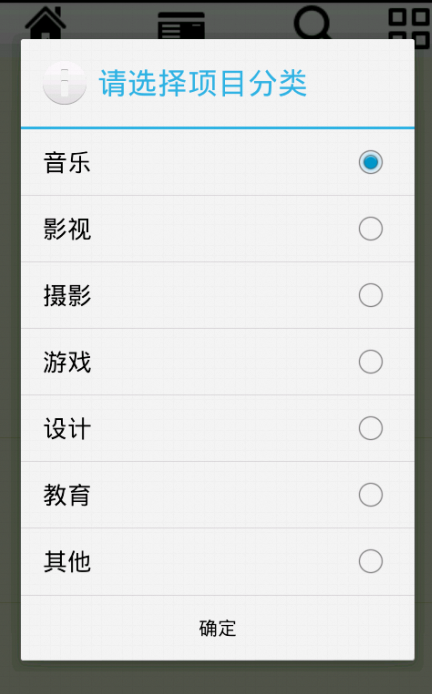
##### 2.1、登录界面设计



##### 2.2、浏览界面设计







##### 2.3、个人信息界面设计



##### 2.4、项目信息界面设计



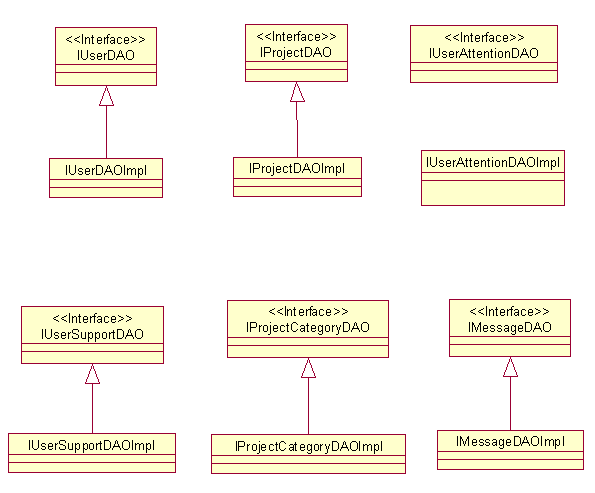
##### 2.5、注册界面设计



第三部分 单元模块设计

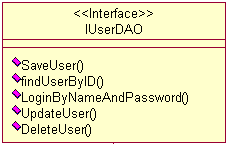
### 一、数据访问层设计

#### 1、类图设计



#### 2、类的详细设计描述

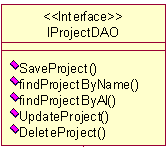
##### 2.1、IUserDAO接口设计



**详细描述：**

|  |
| --- |
| IUserDAO盘点操作接口  事务定义@Transactional(propagation=Propagation.REQUIRED)，声明在接口名上方，表示本接口内所有的方法如果没有定义事务类型，都以该事务类型执行，该事务为：如果当前没有事务，就新建一个事务，如果已经存在一个事务中，加入到这个事务中 |
| public void saveUser(User user)SQLException;  保存用户信息 |
| public void findUserByID(int id)SQLException;  通过用户编号查找用户 |
| public String LoginByNameAndPassword(String name , String password)SQLException;  通过用户名和密码登陆，并返回登录结果（成功或者失败） |
| public String UpdateUser(User user)SQLException;  更新个人信息 |

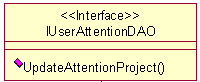
##### 2.2、IProjectDAO接口设计



**详细信息：**

|  |
| --- |
| IProjectDAO盘点操作接口  事务定义@Transactional(propagation=Propagation.REQUIRED)，声明在接口名上方，表示本接口内所有的方法如果没有定义事务类型，都以该事务类型执行，该事务为：如果当前没有事务，就新建一个事务，如果已经存在一个事务中，加入到这个事务中 |
| public void SaveProject(intuid, Project project)SQLException;  保存项目信息 |
| public List<ProjectSummary>findSaveProject()SQLException;  查看用户发起的项目 |
| public List<ProjectSummary>findProjectByNmae(String name)SQLException;  通过项目名称或者关键字查找项目，并获得的项目信息 |
| public List<ProjectSummary>findProjectByAI(String c\_name)SQLException;  通过关键字找到相关的项目，以数组形式列出 |
| public String UpdateProject(User user);  更新项目信息 |
| public List<ProjectSummary>getProjectList()SQLException;  获得所有项目概要信息的列表 |

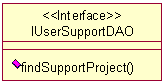
##### 2.3、IUserAttentionDAO接口设计



**详细信息：**

|  |
| --- |
| IUserDAO盘点操作接口  事务定义@Transactional(propagation=Propagation.REQUIRED)，声明在接口名上方，表示本接口内所有的方法如果没有定义事务类型，都以该事务类型执行，该事务为：如果当前没有事务，就新建一个事务，如果已经存在一个事务中，加入到这个事务中 |
| public void UpdateAttentionProject(intpid,,intuid, int flag);  更新关注列表里的项目信息（通过项目编号，用户编号将日志写进数据库） |
| Public List<ProjectSummary>ViewProject (intuid) throws SQLException  查看用户支持的项目列表 |

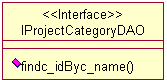
##### 2.4、IUserSupportDAO接口设计



**详细信息：**

|  |
| --- |
| IUserDAO盘点操作接口  事务定义@Transactional(propagation=Propagation.REQUIRED)，声明在接口名上方，表示本接口内所有的方法如果没有定义事务类型，都以该事务类型执行，该事务为：如果当前没有事务，就新建一个事务，如果已经存在一个事务中，加入到这个事务中 |
| public List<ProjectSummary>findSupportProject(intuid)throws SQLException;  找出已支持的项目信息 |
| Public booleanInvestion (intuid, intpid, double money)throws SQLException  对项目进行投资 |

##### 2.5、IProjectCategoryDAO接口设计



**详细信息：**

|  |
| --- |
| IUserDAO盘点操作接口  事务定义@Transactional(propagation=Propagation.REQUIRED)，声明在接口名上方，表示本接口内所有的方法如果没有定义事务类型，都以该事务类型执行，该事务为：如果当前没有事务，就新建一个事务，如果已经存在一个事务中，加入到这个事务中 |
| public List<ProjectSummary>findc\_idByc\_name(String category)throws SQLException;  通过关键字转换为目录表与项目的关联数字 |

### 二、用户管理业务功能模块

#### 1、用户登录功能

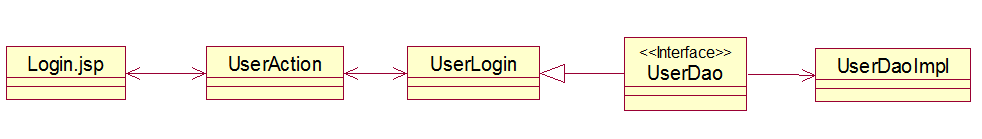
##### 1.1、功能汇总

提供用户的登录功能。

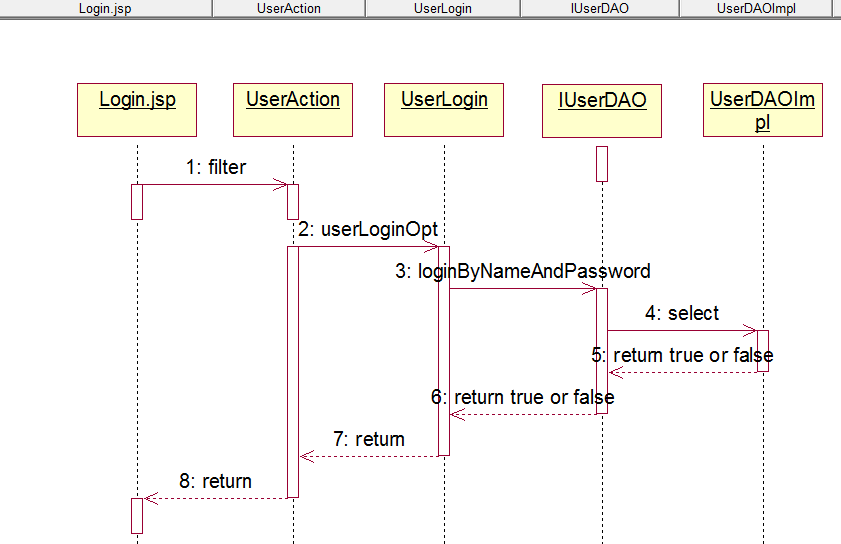
|  |  |
| --- | --- |
| 功能 | 描述 |
| 登录 | 用户输入正确的账号密码以登录进入系统 |

##### 1.2、类图设计

###### 1.2.1、类关系结构图



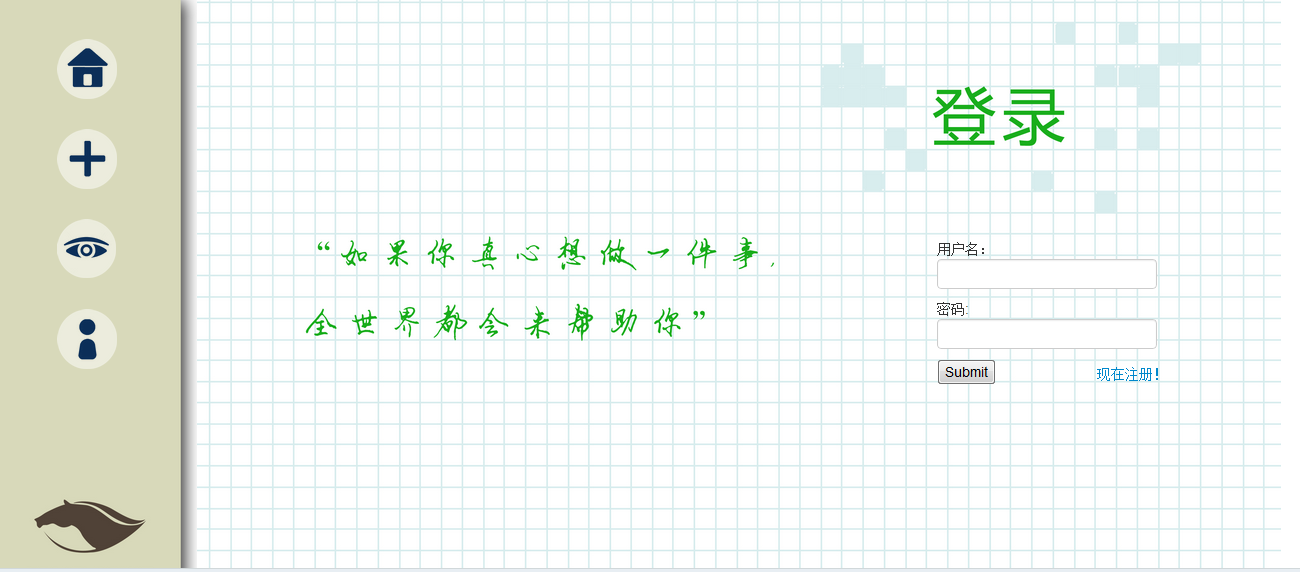
###### 1.2.2、时序图



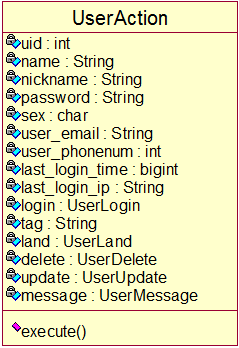
##### 1.3、类的详细设计描述

###### 1.3.1、Login.jsp描述

Login.jsp是用户登录页面，本页面包含一个表单内有username与userpass，在该页面用户触发的请求都会由服务器转发到UserAction该类处理。



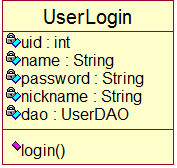
###### 1.3.2、UserAction描述



详细描述：

|  |
| --- |
| UserAction包含响应用户信息增加修改查询等请求。  实现【用户管理】中【用户登录】【用户注册】【个人信息修改】【站内信息管理】四项业务功能的转发及返回结果的响应。。 |
| 1. 成员变量   UserLoginlogin用户登录业务处理接口  UserLandland用户注册业务处理接口  UserDeletedelete用户注销业务处理接口  UserUpdateupdate 用户个人信息管理业务处理接口  UserMessagemessage 用户消息管理业务处理接口 |
| 1. public void execute()   Action类的主要执行方法，根据不同的提交表单的tag选择不同的处理接口 |

###### 1.3.3、UserLogin描述



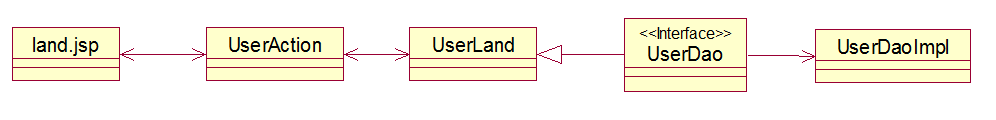
详细描述：

|  |
| --- |
| UserLogin包含响应用户信息增加修改查询等请求。  实现【用户管理】中【用户登录】【用户注册】【个人信息修改】【站内信息管理】四项业务功能的转发及返回结果的响应。。 |
| 1. 成员变量   UserDAOdao访问数据库进行处理接口 |
| 1. public String login(String，String)   Action类的主要执行方法，提交name与password给数据库进行数据是否存在的查询操作，并返回不同的值。 |

##### 1.4、功能实现

通过UserAction类调用UserLogin实现。

#### 2、用户注册功能



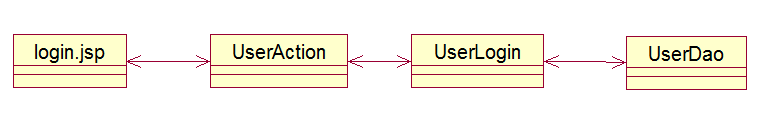
##### 2.1、功能汇总

提供用户的注册功能。

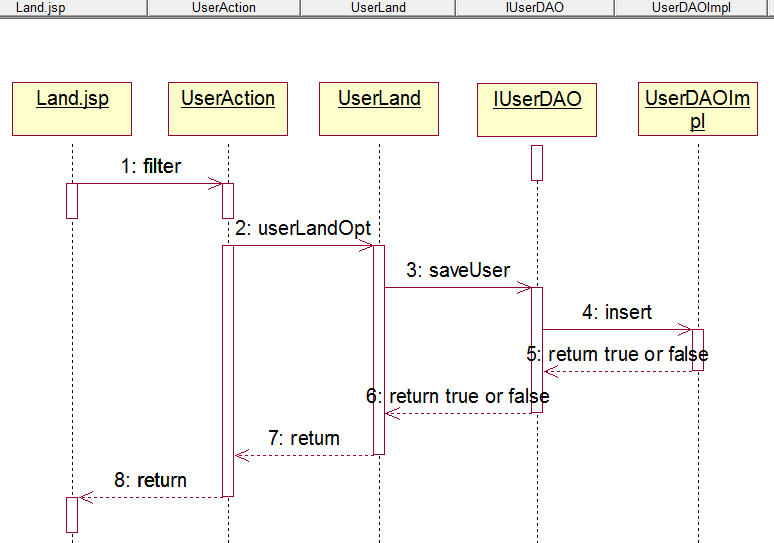
|  |  |
| --- | --- |
| 功能 | 描述 |
| 注册 | 用户输入必须的注册信息进行注册 |

##### 2.2、类图设计

###### 2.2.1、类关系结构图



###### 2.2.2、时序图



##### 2.3、类的详细设计描述

###### 2.3.1、Land.jsp描述

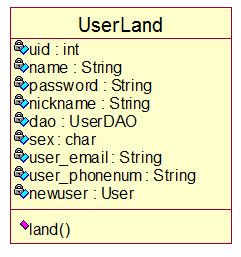
Land.jsp是用户注册页面，本页面包含用户需要的注册信息，在该页面用户触发的请求都会由服务器转发到UserAction该类处理。



###### 2.3.2、UserAction描述

见上

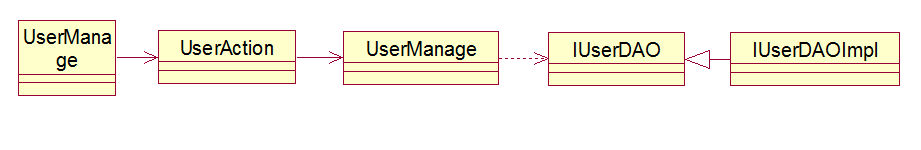
###### 2.3.3、UserLand描述



详细描述：

|  |
| --- |
| UserLand包含对用户注册请求的处理。 |
| 1. 成员变量  UserDAOdao 访问数据库进行处理接口  User newuser将用户注册信息打包 |
| 2.public String land(User)  该方法通过调用dao.saveuser(User)将User类型数据提交给数据库 |

#### 3、用户个人信息管理功能



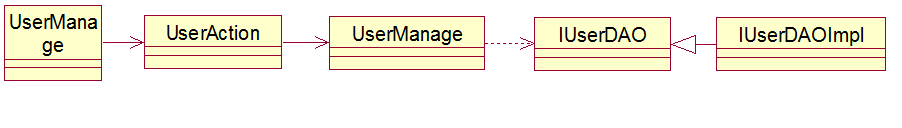
##### 3.1、功能汇总

提供用户的个人信息管理功能。

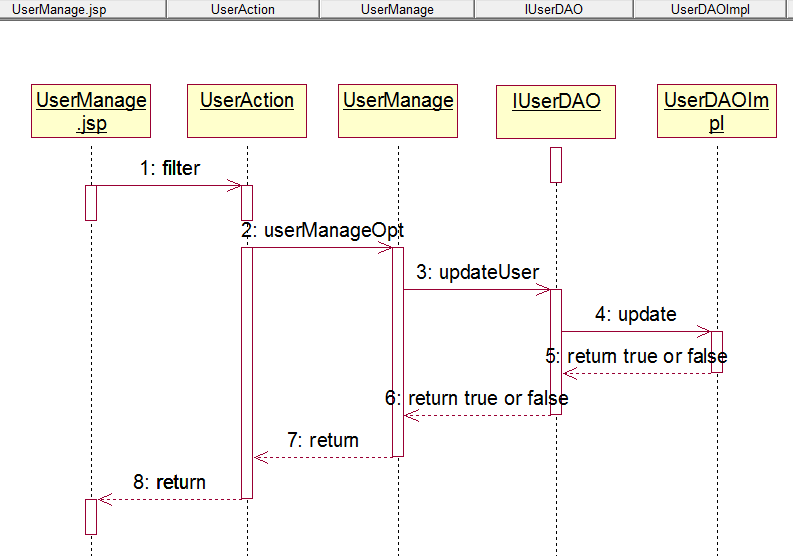
|  |  |
| --- | --- |
| 功能 | 描述 |
| 个人信息管理 | 进行用户个人信息的修改，更新。 |

##### 3.2、类图设计

###### 3.2.1、类关系结构图



###### 3.2.2、时序图



##### 3.3、类的详细设计描述

###### 3.3.1、UserManage.jsp描述

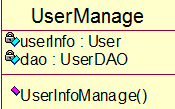
UserManage.jsp是用户管理页面，本页面包含用户个人信息修改，查看已发起的项目，已关注的项目，已投资的项目，在该页面用户触发的请求都会由服务器转发到UserAction该类处理。



###### 3.3.2、UserAction描述

见上

###### 3.3.3、UserManage描述



详细描述：

|  |
| --- |
| UserManage包含对用户更改个人信息请求的处理。 |
| 1. 成员变量  UserDAOdao 访问数据库进行处理接口  Useruserinfo将用户注册信息打包 |
| 2.public StringUserInfoManage(User)  该方法通过调用dao.updateUser(User)将User类型数据提交给数据库 |

### 三、项目管理业务功能模块

#### 1、发起项目功能

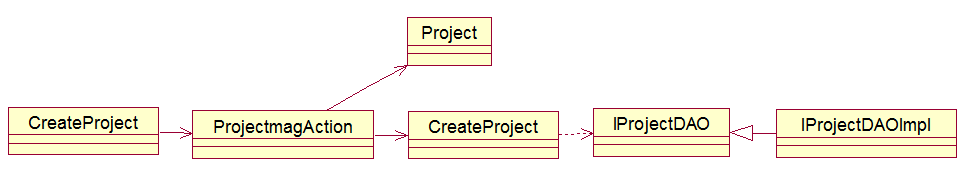
##### 1.1、功能汇总

提供用户发起项目的功能。

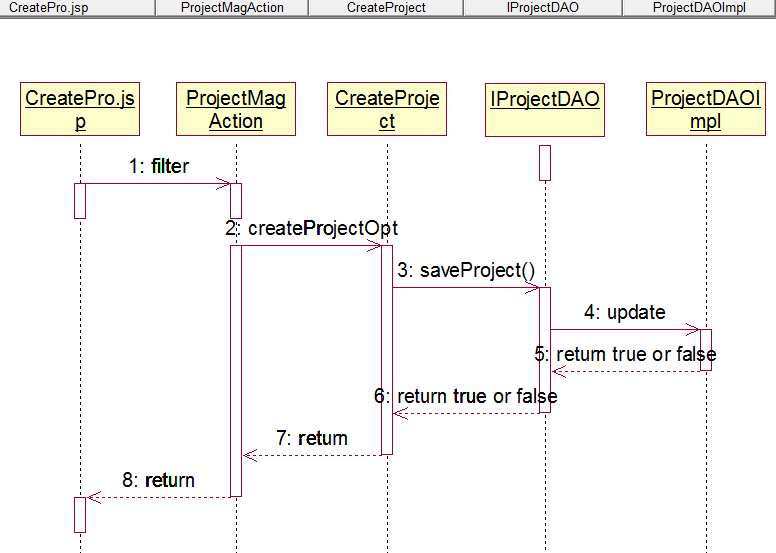
|  |  |
| --- | --- |
| 功能 | 描述 |
| 发起项目 | 对用户新建的项目进行处理。 |

##### 1.2、类图设计

###### 1.2.1、类关系结构图



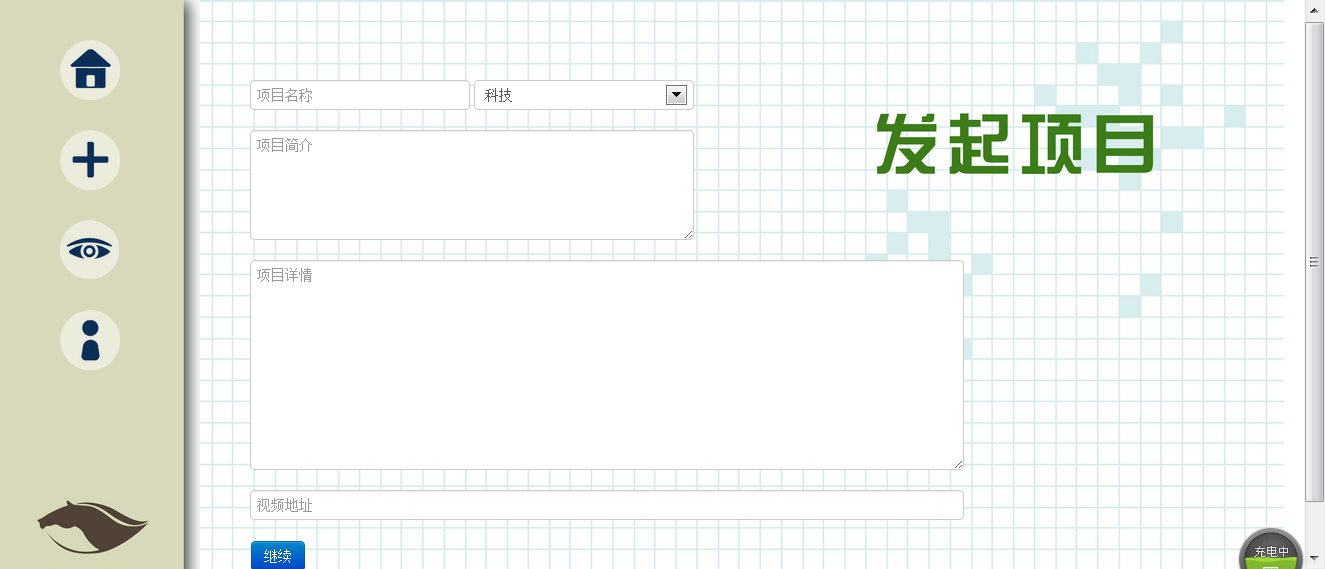
###### 1.2.2、时序图



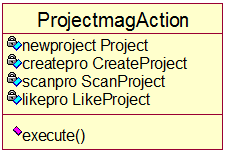
##### 1.3、类的详细设计描述

###### 1.3.1、CreatePro.jsp描述

CreateProject.jsp,CreateProject1.jsp,CreateProject2.jsp,CreateProject3.jsp分四个页面供项目发起人填写新建一个项目所需要的各项信息，点击提交按钮，待系统审核员通过可完成发起项目功能。



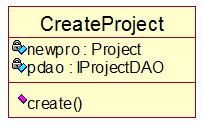
###### 1.3.2、ProjectMagAction描述



详细描述：

|  |
| --- |
| ProjectMagAction针对不同页面关于项目的请求进行转发，给不同的业务处理类去处理，完成项目管理中的【发起项目】，【浏览项目】，【关注项目】功能。 |
| 1. 成员变量  Projectnewproject用于对项目信息进行打包  CreateProjectcreatepro发起项目处理的接口  ScanProjectscanpro浏览项目处理的接口  LikeProjectlikepro关注项目处理的接口 |
| 2.public Stringexecute()  Action的执行函数，对不同tag的信息进行分类处理，并根据结果返回页面。 |

###### 1.3.3、CreateProject描述



详细描述：

|  |
| --- |
| CreateProject包含对用户新建一个项目操作的处理。 |
| 1. 成员变量  ProjectDAOpdao 访问数据库进行处理接口  Projectnewpro传递项目信息 |
| 2.public Stringcreate(Project)  该方法通过调用DAO层的pdao.saveProject(Project)将Project类型数据存储到数据库 |

#### 2、浏览项目功能

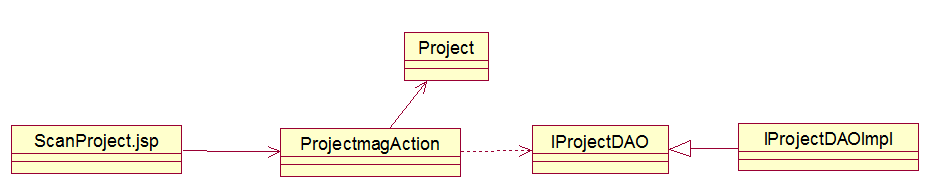
##### 2.1、功能汇总

提供用户通过关键字和标签过滤，浏览项目的功能。

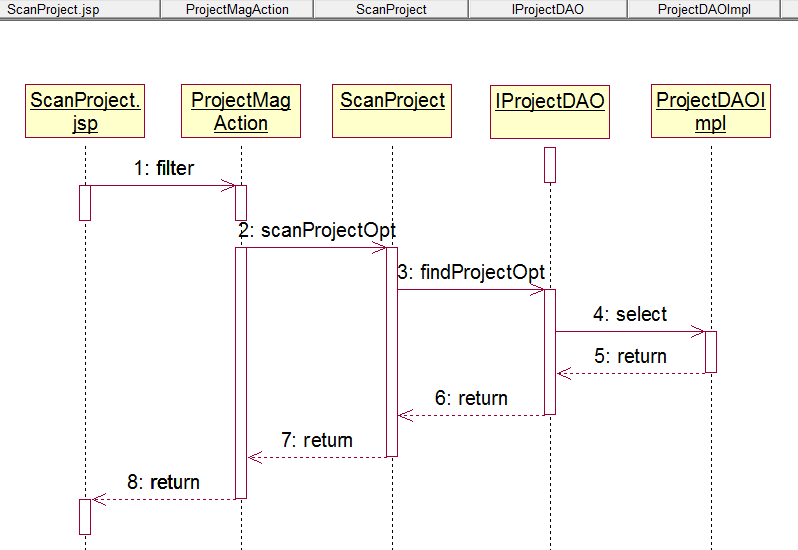
|  |  |
| --- | --- |
| 功能 | 描述 |
| 通过标签过滤项目 | 用户点击不同的标签将向用户呈现该标签分类下的项目。 |
| 通过搜索关键字过滤项目 | 用户在搜索栏输入关键字，提交后得到符合条件的项目 |

##### 2.2、类图设计

###### 2.2.1、类关系结构图



###### 2.2.2、时序图



##### 2.3、类的详细设计描述

###### 2.3.1、ScanProject.jsp描述

项目以图片列表的形式呈现，用户可以对浏览的项目进行筛选，查看的操作。顶部的导航条具有滚动监听功能，可自动变换当前项目的标签分类。

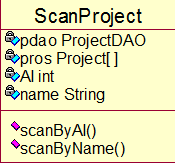


###### 2.3.2、ProjectMagAction描述

见上1.3.2

###### 2.3.3、ScanProject描述

详细描述：



|  |
| --- |
| ScanProject包含对用户筛选，浏览项目操作的处理。 |
| 1. 成员变量  ProjectDAOpdao 访问数据库进行处理接口  Project[]pros传递项目信息  IntAI 标签对应的标签序号  String name 要搜索的关键字 |
| 2.public Project[]scanByAI(int)  该方法通过调用DAO层的pdao.findProjectByAI(int)从数据库中寻找到合适的项目数据，以数组的形式返回。 |
| 2.public Project[]scanByName(String)  该方法通过调用DAO层的pdao.findProjectByname(String)从数据库中寻找到合适的项目数据，以数组的形式返回。 |

#### 3、关注项目功能

##### 3.1、功能汇总

提供用户通过点击关注按钮，对感兴趣的项目进行关注的功能。。

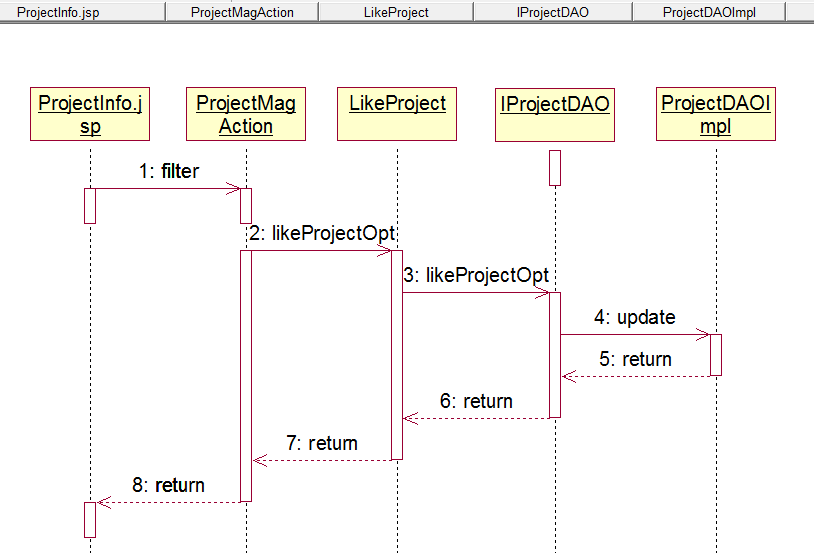
|  |  |
| --- | --- |
| 功能 | 描述 |
| 关注项目 | 用户点击关注按钮，将关注该项目。 |

##### 3.2、类图设计

###### 3.2.1、类关系结构图



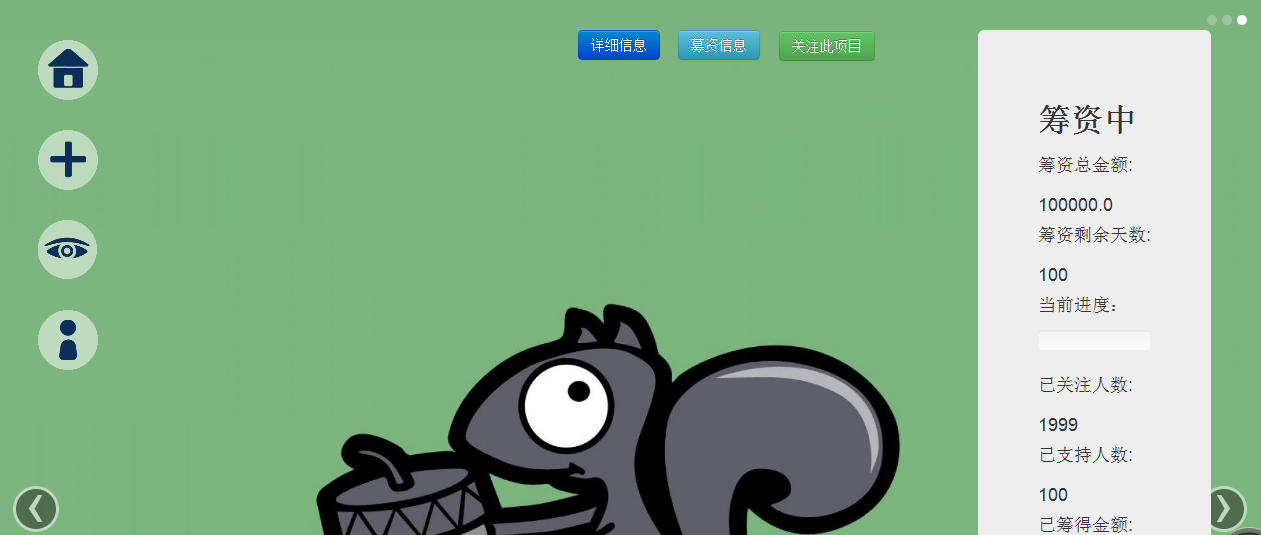
###### 3.2.2、时序图



##### 3.3、类的详细设计描述

###### 3.3.1、ProjectInfo.jsp描述

显示项目的详细信息，包括视频，图片，文字，以及已获投资情况。并且提供关注按钮，和支持按钮。

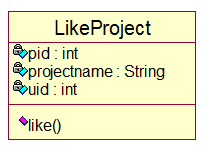


###### 3.3.2、ProjectMagAction描述

见上1.3.2

###### 3.3.3、LikeProject描述

详细描述：



|  |
| --- |
| LikeProject完成用户关注操作的处理。 |
| 1. 成员变量  ProjectDAOpdao 访问数据库进行处理接口  Intpid项目ID  String projectname项目名称  Intuid用户ID |
| 2.public Boolean like(int,int)  该方法通过调用DAO层的pdao.updateAttentionProject(int,int)完成用户关注项目的更新操作。 |

#### 4、编辑项目功能

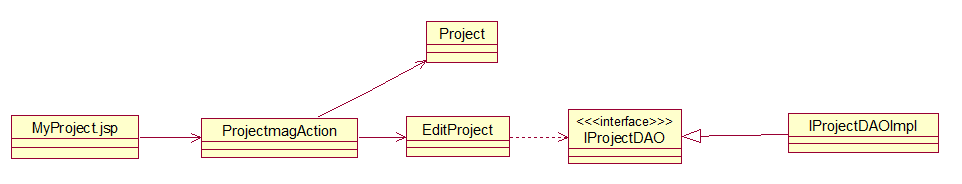
##### 4.1、功能汇总

用户进入我的项目界面，点击发起的项目选项，可以进行发起的项目的查看和编辑。

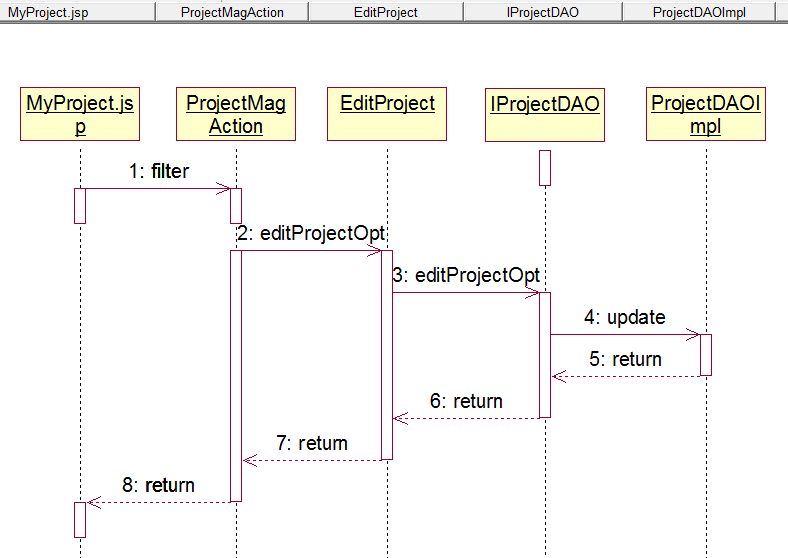
|  |  |
| --- | --- |
| 功能 | 描述 |
| 编辑项目 | 用户进入我的项目界面，点击发起的项目选项，可以进行发起的项目的查看和编辑。 |

##### 4.2、类图设计

###### 4.2.1、类关系结构图



###### 4.2.2、时序图



##### 4.3、类的详细设计描述

###### 4.3.1、MyProject.jsp描述

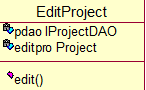
显示用户发起项目的信息，用户可以对已发起的项目进行修改编辑。

###### 4.3.2、ProjectMagAction描述

见上1.3.2

###### 4.3.3、EditProject描述

详细描述：



|  |
| --- |
| EditProject完成用户编辑项目操作的处理。 |
| 1. 成员变量  ProjectDAOpdao 访问数据库进行处理接口  Projecteditpro项目信息 |
| 2.public Boolean edit(Project)  该方法通过调用DAO层的pdao.updateProject(Project)完成用户编辑项目的更新操作。 |

### 四、投资管理业务功能模块

#### 1、查看已关注、已投资项目功能

##### 1.1、功能汇总

提供用户查看关注、投资项目的功能。

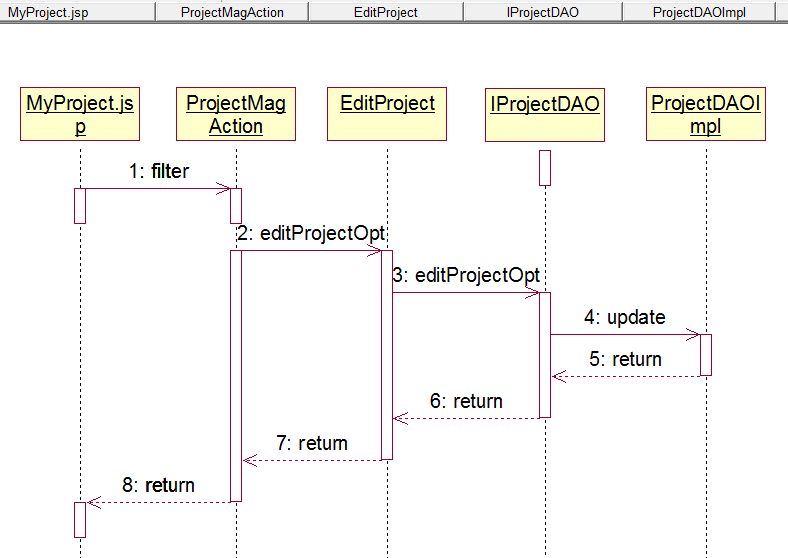
|  |  |
| --- | --- |
| 功能 | 描述 |
| 查看已投资的项目 | 用户点击已投资项目选项，可以查看当前已投资的项目。 |
| 查看已关注的项目 | 用户点击已关注项目选项，可以查看当前已关注的项目。 |
| 取消关注 | 用户可以点击取消按钮，取消对该项目的关注 |
| 取消投资 | 用户可以点击取消按钮，取消对该项目的投资 |

##### 1.2、类图设计

###### 1.2.1、类关系结构图



###### 1.2.2、时序图

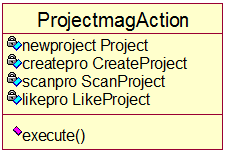


##### 1.3、类的详细设计描述

###### 1.3.1、AttentionedProjectList.jsp描述

显示当前登录用户的关注的项目列表，用户可以对其进行查看，以及取消关注。

###### 3.2、ProjectMagAction描述

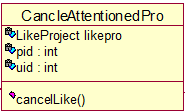


以及取消关注，取消投资对应的接口函数

详细描述：

|  |
| --- |
| ProjectMagAction针对不同页面关于项目的请求进行转发，给不同的业务处理类去处理，完成项目管理中的【发起项目】，【浏览项目】，【关注项目】功能。 |
| 1. 成员变量  Projectnewproject 用于对项目信息进行打包  CreateProjectcreatepro发起项目处理的接口  ScanProjectscanpro浏览项目处理的接口  LikeProjectlikepro关注项目处理的接口  CancelAttentionedProjectcalattpro取消关注的项目业务处理的接口  CancelInvestedProjectcalainvpro取消投资的项目业务处理的接口 |
| 2.public String execute()  Action的执行函数，对不同tag的信息进行分类处理，并根据结果返回页面。 |

###### 1.3.3、CancelAttentionedProject描述



详细描述：

|  |
| --- |
| CreateProject包含对用户新建一个项目操作的处理。 |
| 1. 成员变量  LikeProjectclike关注项目业务处理接口 |
| 2.public StringcancelLike(int,int)  该方法通过调用关注项目业务处理likepro.like(int,int)的函数，通过参数值的判断进行取消关注操作 |

#### 2、投资项目功能

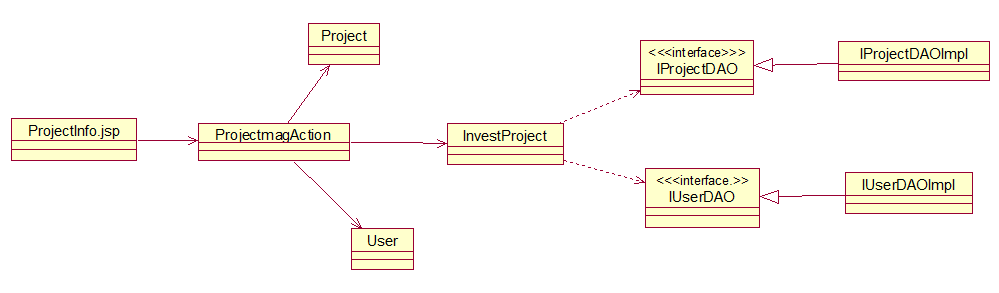
##### 2.1、功能汇总

提供用户投资项目的功能。

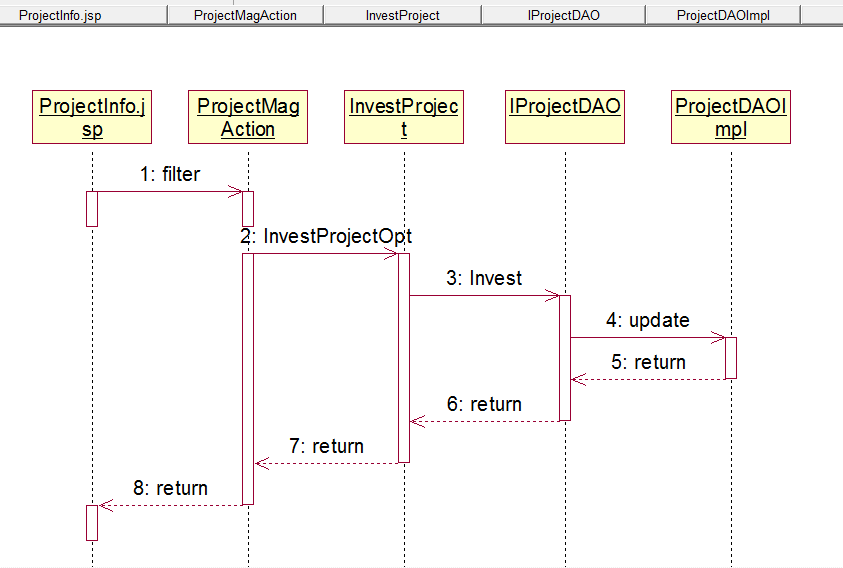
|  |  |
| --- | --- |
| 功能 | 描述 |
| 投资项目 | 用户点击项目介绍中的支持按钮，可以选择不同的金额进行支持，付款后完成投资项目业务功能。 |

##### 2.2、类图设计

###### 2.2.1、类关系结构图



###### 2.2.2、时序图



##### 2.3、类的详细设计描述

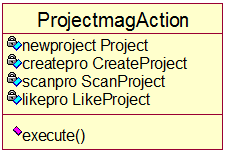
###### 2.3.1、ProjectInfo.jsp描述

显示项目的详细信息，包括视频，图片，文字，以及已获投资情况。并且提供关注按钮，和支持按钮。其中用户可根据项目发起人的设置选择不同金额的支持按钮进行支持操作。





###### 2.3.2、ProjectMagAction描述

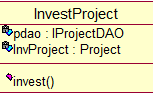


新增了投资的接口函数InvestProjectinvpro

详细描述：

|  |
| --- |
| ProjectMagAction针对不同页面关于项目的请求进行转发，给不同的业务处理类去处理，完成项目管理中的【发起项目】，【浏览项目】，【关注项目】【投资项目】功能。 |
| 1. 成员变量  Projectnewproject 用于对项目信息进行打包  CreateProjectcreatepro发起项目处理的接口  ScanProjectscanpro浏览项目处理的接口  LikeProjectlikepro关注项目处理的接口  CancelAttentionedProjectcalattpro取消关注的项目业务处理的接口  CancelInvestedProjectcalainvpro取消投资的项目业务处理的接口  InvestProjectinvpro 投资业务处理的接口函数 |
| 2.public String execute()  Action的执行函数，对不同tag的信息进行分类处理，并根据结果返回页面。 |

###### 2.3.3、InvestProject描述

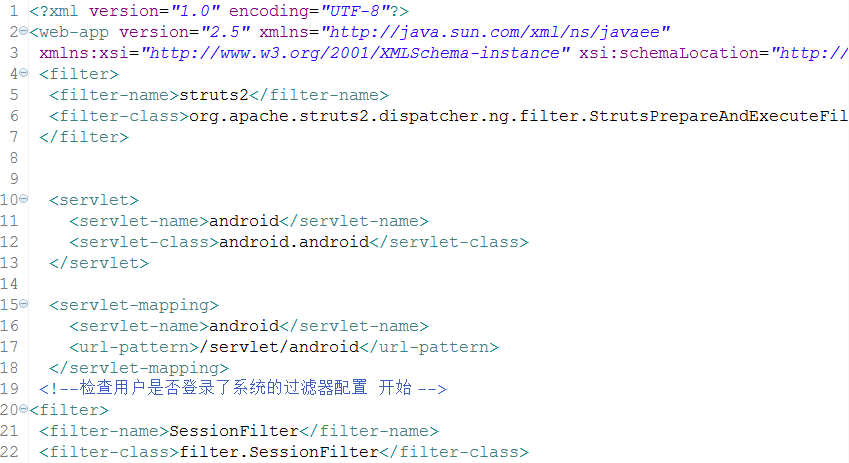


详细描述：

|  |
| --- |
| InvestProject包含对用户投资项目操作的处理。 |
| 1. 成员变量  Project InvProject待投资的项目信息  IProjectDAOpdao数据库接口函数 |
| 2.public Stringinvest(Project)  该方法通过调用pdao.investProject(Project)，通过参数值的判断进行投资操作 |

##### 2.4、配置文件

Web.xml





Struts.xml

<?xmlversion=*"1.0"*encoding=*"UTF-8"*?>

<!DOCTYPEstrutsPUBLIC

"-//Apache Software Foundation//DTD Struts Configuration 2.3//EN"

"http://struts.apache.org/dtds/struts-2.3.dtd">

<struts>

<!--上传的全部图片的最大限制-->

<constantname=*"struts.multipart.maxSize"*value=*"1024102400"*/>

<!--临时存放文件的路径 -->

<constantname=*"struts.multipart.saveDir"*value=*"d:/MyEclipse-9.0M1"*/>

<packagename=*"useraction"*extends=*"struts-default"*>

<actionname=*"login"*class=*"action.LoginAction"*>

<resultname=*"logsucc"*>/homePage.jsp</result>

<resultname=*"erro"*>/error.jsp</result>

</action>

<actionname=*"land"*class=*"action.LandAction"*>

<resultname=*"landsucc"*>/homePage.jsp</result>

<resultname=*"nameexisted"*>/NameExisted.jsp</result>

</action>

</package>

<packagename=*"projectaction"*extends=*"struts-default"*namespace=*"/"*>

<actionname=*"createPro"*class=*"action.CreateProjectAction"*>

<resultname=*"createsucc"*>/CreateSucc.jsp</result>

<resultname=*"erro"*>/error.jsp</result>

<resultname=*"next1"*>/CreateProject1.jsp</result>

<resultname=*"next3"*>/CreateProject3.jsp</result>

<resultname=*"getsucc"*>/CreatedProjectList.jsp</result>

</action>

<actionname=*"fileUpload"*class=*"action.FileUploadAction"*>

<!--限制图片的格式和图片的大小 -->

<interceptor-refname=*"fileUpload"*>

<paramname=*"allowedTypes"*>

image/bmp,image/png,image/gif,image/jpeg,image/pjpeg

</param>

</interceptor-ref>

<!--默认的拦截器，必须要写 -->

<interceptor-refname=*"defaultStack"*/>

<resultname=*"uploadsucc"*>/CreateProject2.jsp</result>

</action>

<actionname=*"browsePro"*class=*"action.BrowseProjectAction"*>

<resultname=*"prolist"*>/BrowseProject.jsp</result>

</action>

<actionname=*"getpro"*class=*"action.GetProjectAction"*>

<resultname=*"showpro"*>/ShowProject.jsp</result>

</action>

</package>

<packagename=*"attentionaction"*extends=*"struts-default"*>

<actionname=*"attentionpro"*class=*"action.AttentionProjectAction"*>

<resultname=*"addsucc"*type=*"chain"*>

<paramname=*"actionName"*>getpro</param>

<paramname=*"namespace"*>/</param>

</result>

<resultname=*"getsucc"*>/AttentionedProjectList.jsp</result>

<resultname=*"cancelsucc"*type=*"chain"*>

<paramname=*"actionName"*>getpro</param>

<paramname=*"namespace"*>/</param>

</result>

</action>

</package>

<packagename=*"investaction"*extends=*"struts-default"*>

<actionname=*"investpro"*class=*"action.InvestProjectAction"*>

<resultname=*"confirm"*>/InvestConfirm.jsp</result>

<resultname=*"investsucc"*>/InvestSucc.jsp</result>

<resultname=*"getsucc"*>/InvestedProjectList.jsp</result>

<resultname=*"deletesucc"*>/ManageUser.jsp</result>

</action>

</package>

<packagename=*"android"*extends=*"struts-default"*>

</package>

</struts>



### 五、移动平台业务功能模块

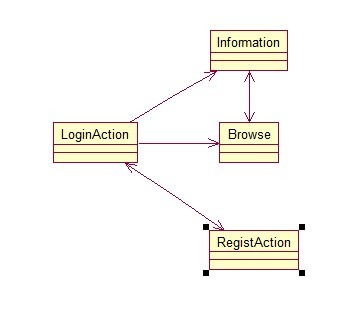
##### 1.1、功能汇总

管理系统中的所有用户的信息。

|  |  |
| --- | --- |
| 功能 | 描述 |
| 登陆 | 用户打开android端app，输入用户名和密码，从而进行登陆 |
| 注册 | 用户如果还没账号，选择登陆页面的注册按钮，填写信息注册为用户 |
| 添加关注 | 用户在浏览项目的时候，如果对项目感兴趣，可以将项目添加到自己的关注菜单中 |
| 取消关注 | 用户对自己关注菜单中的项目进行取消操作 |
| 查看个人信息 | 用户通过android客户端查看自己用户信息，包括关注菜单 |

##### 1.2、类图设计

###### 1.2.1、类关系结构图

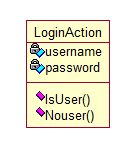


###### 1.2.2、时序图

图见1.4

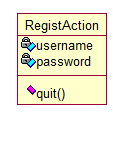
##### 1.3、类的详细设计描述

1.LoginAction.class



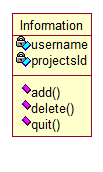
|  |
| --- |
| LoginAction  客户登录功能 |
| 1. 成员变量   username 用户姓名  password 用户密码 |
| 1. public boolenisUser() throws Exception   检查是否登陆成功 |
| 1. public void NoUser() throws Exception   调用注册界面进行注册 |

2.RegistAction.class



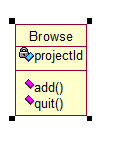
|  |
| --- |
| 1. 成员变量  username 用户姓名  password 用户密码 |
| 2.public void quit() throws Exception  退出当前界面 |

3.informatin.class



|  |
| --- |
| 1. 成员变量  username 用户姓名  projectId投资项目的ID号 |
| 2.public voidadd() throws Exception  将项目ID号加入到用户的关注菜单 |
| 3. public voidadd() throws Exception  将用户名下的关注项目ID号删除 |
| 4. public void quit() throws Exception  退出当前界面 |

4.Browse.class

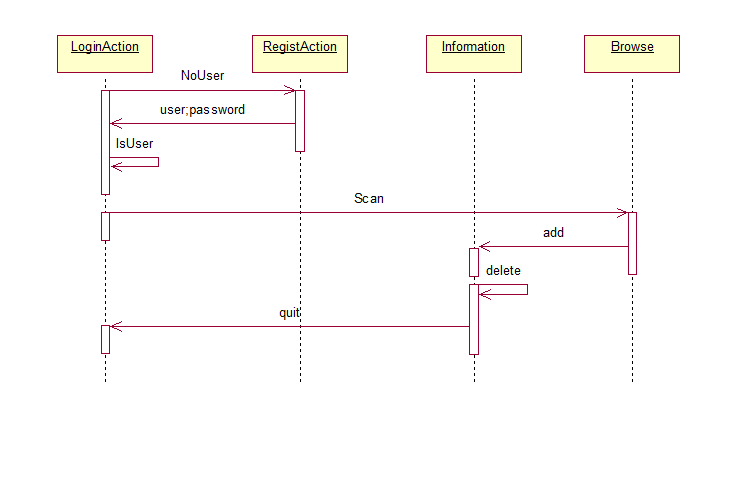


|  |
| --- |
| 1. 成员变量  projectId投资项目的ID号 |
| 2.public voidadd() throws Exception  将项目ID号加入到用户的关注菜单 |
| 3. public void quit() throws Exception  退出当前界面 |

##### 1.4、功能实现

LoginAction.java文件对用户登陆进行了判断，RegistAction.java文件将用户注册的信息发送给服务器，调用数据库进行保存操作,BrowseAction.java将从服务器端接收到的项目信息显示在android平台上，Information.Java将接收的个人信息进行显示

时序图：



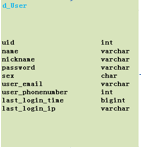
第六部分数据库设计

### 一、数据库整体结构图



### 二、业务功能功能模块

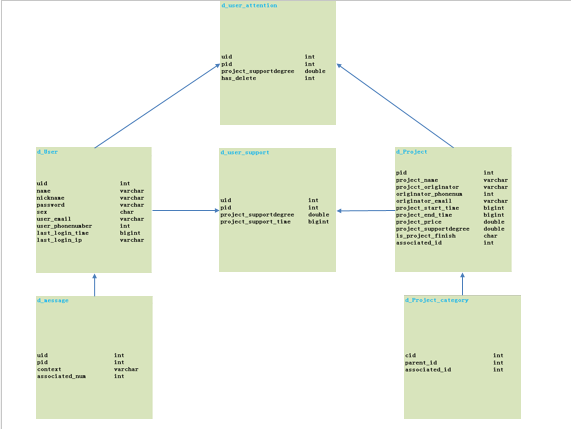
#### 1、用户信息表结构



**d\_user表（用户信息表）**



#### 2、项目管理表结构

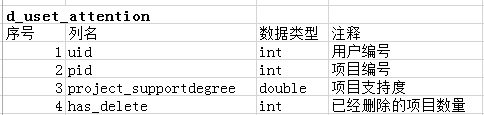


**d\_project表（项目信息表）**



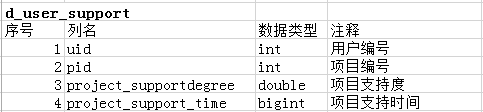
#### 3、关注项目信息表结构

**d\_user\_attention（关注项目信息）**



#### 4、支持项目信息表结构

**d\_user\_support（支持项目信息）**

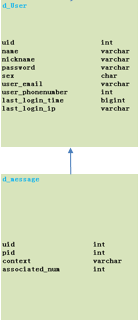


#### 5、目录信息表结构

**d\_project\_category（目录信息表）**



#### 6、站内信息表结构



**d\_message(消息管理)**



第七部分 补充设计与说明

### 一、编译运行环境设计

1、服务器环境

1）OS：Windows Server 2003

2）2G 以上内存，建议使用 4G 内存；

3）500GB 以上的硬盘空间，不包括数据存储空间

4）JDK：Java SE 6

5）DB：Database 10g

6）Application Server：WebLogic Server 10.3

2、WEB 客户机器环境

1）Pentium III 或以上微处理器（CPU）；

2）Microsoft Windows XP、 Windows Vista、Windows 7 操作系统；

3）512MB 以上内存，建议使用 1024MB 内存；

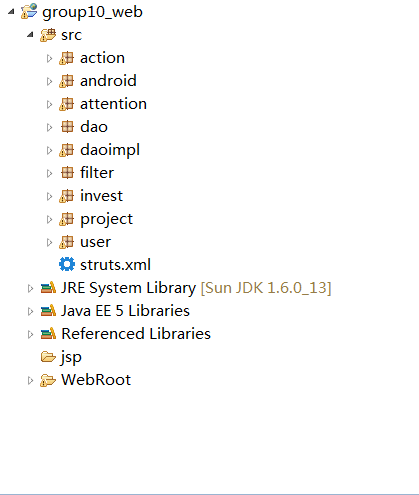
4）WEB：要求 IE7 以上版本，最好 IE8 以上版本。

3、Android移动平台环境

Android2.3及以上

### 二、包路径与Web目录结构设计

#### 1、包路径设计



#### 2、各包下详细文件设计





