1. 登录模块

|  |  |  |
| --- | --- | --- |
| 类名称 | 方法名称 | 功能描述 |
| LoginFromUI | Show-Login | 显示用户登录的界面 |
| RetrievePasswordUI | Retrieve-psw | 显示用户找回密码的界面 |
| ResetPasswordUI | Reset-psw | 显示用户重置密码的界面 |
| ErrorPasswordUI | Show-error | 弹出用户密码错误 |
| ResultUI | Result-ui | 显示用户密码找回或重置结果最终界面 |
| LoginController | ToLogin | 跳转到用户登陆界面 |
| Login | 进行登录操作 |
| toerror | 跳转到登录失败界面 |
| RetrieveController | Toretieve | 跳转到用户密码找回界面 |
| Retrieve | 进行密码找回操作 |
| Toresult | 跳转到密码找回结果界面 |
| ResetController | Toreset | 跳转到用户密码重置界面 |
| reset | 进行密码重置操作 |
| Toresult | 跳转到密码重置结果界面 |
| LoginInformation | Login-Inform | 向数据库发送用户登录信息 |
| RetrieveIformation | Retieve-Inform | 向数据库发送密码找回信息 |
| ResetIformation | Reset-Inform | 向数据库发送密码重置信息 |
| LoginDAO | Login-data | 从数据库中获取用户信息的数据，返回用户登录结果 |
| RetrieveDAO | Retrieve-data | 从数据库中获取用户信息的数据，返回密码找回结果 |
| ResetDAO | Reset-data | 获取用户输入并写入数据库，返回密码重置结果 |



|  |  |  |  |
| --- | --- | --- | --- |
| 接口名称 | 关联的类 | 接口信息 | |
| LoginFramUI.  Show-Login | LoginFrameUI  LoginController | 语法 | Public void Show-Login(  Long user-id,  string password) |
| 前置条件 | 用户的输入不为空 |
| 后置条件 | 将用户的信息发送给控制器 |
| 不变量 | 用户的账号密码 |
| LoginController.  Login | LoginController  LloginInformation | 语法 | Public void Login(  Long user-id,  string password) |
| 前置条件 | 用户点击登录按钮 |
| 后置条件 | 将用户输入的账号、密码发送给LoginInformation |
| 不变量 | 用户输入的账号密码 |
| RetrievePasswordUI.  Retrieve-psw | RetrievePasswordUI  RetrieveController | 语法 | Public void Retrieve-psw(  Long user-id，  string mail-address，  string new-password) |
| 前置条件 | 用户的输入不为空 |
| 后置条件 | 将用户输入的信息发送给控制器 |
| 不变量 | 用户待找回账号、邮箱、新密码、验证码 |
| RetrieveControlller.  Retrieve | RetrieveController  RetrieveInformation | 语法 | Public void Show-Login(  Long user-id,  string mail,string new-password,string yanzhengma) |
| 前置条件 | 无 |
| 后置条件 | 将用户输入的账号邮箱地址、验证码、新密码发送给RetrieveInformation |
| 不变量 | 用户输入的账号、邮箱、密码 |
| ResetPasswordUI.  Reset-psw | ResetPasswordUI  ResetController | 语法 | Public void Reset-psw(  Long user-id，  string old-password，  string new-password) |
| 前置条件 | 用户的输入不为空 |
| 后置条件 | 将用户输入的信息发送给控制器 |
| 不变量 | 用户账号、旧密码、新密码 |
| ResetController.  Reset | ResetController  ResetInformation | 语法 | Public void Reset(  Long user-id,  string old-password  String new-password) |
| 前置条件 | 无 |
| 后置条件 | 将用户输入的账号旧密码、新密码发送给ResetInformation |
| 不变量 | 账号、旧密码、新密码 |
| ErrorPasswordUI.  Show-error | ErrorPasswordUI | 语法 | Public void Show-error(  ) |
| 前置条件 | 用户输入的密码错误 |
| 后置条件 | 无 |
| 不变量 | 错误信息 |
| ResultUI.  Result-ui | ResultUI  LoginFromUI | 语法 | Public void Result-ui(  ) |
| 前置条件 | 用户重置密码或找回密码成功 |
| 后置条件 | 无 |
| 不变量 | 无 |
| LoginInformation.  Login-Inform | LoginInformation  LoginController  LoginDAO | 语法 | Public void Login-Inform(  Long user-id,  String password) |
| 前置条件 | 数据库连接正常 |
| 后置条件 | 将用户登录的登录数据发送给数据库 |
| 不变量 | 用户的登录数据 |
| RetrieveIformation.  Retieve-Inform | RetrieveIformation  RetrieveController  RetrieveDAO | 语法 | Public void Retrieve-Inform(  Long user-id，  string mail-address，  string new-password) |
| 前置条件 | 数据库连接正常 |
| 后置条件 | 将用户的找回密码数据发送给数据库 |
| 不变量 | 用户的找回密码数据 |
| ResetIformation.  Reset-Inform | ResetIformation  ResetController  ResetDAO | 语法 | Public void Reset-Inform(  Long user-id，  string old-password，  string new-password) |
| 前置条件 | 数据库连接正常 |
| 后置条件 | 将用户的重置密码数据发送给数据库 |
| 不变量 | 用户的重置密码数据 |
| LoginDAO.  Login-date | LoginDAO  LoginInformation | 语法 | Public void Login-date(  Entity me) |
| 前置条件 | 数据库连接正常 |
| 后置条件 | 从数据库中获取信息数据并返回实体对象 |
| 不变量 | 登录id |
| RetrieveDAO.  Retrieve-date | RetrieveDAO  RetrieveInformation | 语法 | Public void Retrieve-date(  Entity me) |
| 前置条件 | 数据库连接正常 |
| 后置条件 | 从数据库中获取信息数据并返回实体对象 |
| 不变量 | 找回id |
| ResetDAO.  Reset-date | ResetDAO  ResetInformation | 语法 | Public void Reset-date(  Entity me) |
| 前置条件 | 数据库连接正常 |
| 后置条件 | 从数据库中获取信息数据并返回实体对象 |
| 不变量 | 重置id |

1. 首页模块

|  |  |  |
| --- | --- | --- |
| 类名称 | 方法 | 功能描述 |
| MainFromUI | Show-mainfm | 显示软件的首页 |
| SendUI | Show-send | 显示用户发布动态的界面 |
| SendResultUI | Show-sendresult | 显示显示用户动态发送结果信息 |
| MainController | Tosend | 跳转到用户发布动态的界面 |
| Tomain | 跳转到软件的首页 |
| Send | 将界面接受的信息数据保存 |
| Tosendresult | 跳转到发送结果界面 |
| MessageIformation | Message-inform | 向数据库发送信息操作请求 |
| SendDAO | Send-datasave | 保存操作信息到数据库，返回动态发布结果 |



|  |  |  |  |
| --- | --- | --- | --- |
| MainFromUI.  Show-mainfm | MainFromUI | 语法 | Public void Show-mainfm(  Long uid) |
| 前置条件 | 无 |
| 后置条件 | 无 |
| 不变量 | 用户id |
| SendUI.  Show-sendui | SendUI  SendController | 语法 | Public void Show-sendui(  Long uid ,string context) |
| 前置条件 | 用户的输入不为空 |
| 后置条件 | 将用户待发布的信息发送给控制器 |
| 不变量 | 用户待发布的信息，用户id |
| SendResultUI.  Show-sendresult | SendResultUI  SendController | 语法 | Public void Show-sendresult  (long uid,long message-id  ) |
| 前置条件 | 用户发布的动态发布成功 |
| 后置条件 | 将用户信息发送给控制器 |
| 不变量 | 用户id，用户发布的动态的id |
| SendController.  Savesendmsg | SendController  MessageInformation | 语法 | Public void Savesendmsg(  Long uid,string context) |
| 前置条件 | 无 |
| 后置条件 | 将用户填写的动态内容发送给MessageInformation |
| 不变量 | 用户填写的动态内容 |
| MessageInformation.  Message-inform | MessageInformation  Send DAO | 语法 | Public void Message-inform  (long uid ,string context  ) |
| 前置条件 | 数据库连接正常 |
| 后置条件 | 将用户填写的动态内容和用户id发送给数据库 |
| 不变量 | 用户的id和动态内容 |
| SendDAO.  Send-datesave | SendDAO  MessageInformation | 语法 | Public void Send-datesave(  Entity me) |
| 前置条件 | 数据库连接正常 |
| 后置条件 | 从数据库获取信息数据并返回实体对象 |
| 不变量 | 用户id |

1. 信息模块

|  |  |  |
| --- | --- | --- |
| 类名称 | 方法 | 功能描述 |
| MessageUI | Showmessage | 显示该条动态信息 |
| MessageController | ToMessage | 显示可显示动态信息 |
| Transmit | 转发该条动态 |
| Comment | 评论该条动态 |
| View | 点击该条动态表示已浏览 |
| Praise | 点赞该条动态 |
| Feedback | 向管理员反馈该条动态 |
| MessageIformation | Msg-get | 从数据层获取信息 |
| Msg-save | 向数据层发送信息 |
| MessageDAO | Message-datasave | 将用户的动态信息保存到数据库，返回信息操作结果 |
| Message-dataget | 从数据库获取用户的动态信息，传入界面显示 |



|  |  |  |  |
| --- | --- | --- | --- |
| MessageUI.  Show-message | MessageUI  MessageController | 语法 | Public void Show-message(  Long uid) |
| 前置条件 | 用户选择查看动态信息 |
| 后置条件 | 将用户请求发送给控制器 |
| 不变量 | 用户id |
| Messagecontroller.  Transmit | Messagecontroller  MessageInformation | 语法 | Public void Transmit(  Long message-id,  string context,long circle-id) |
| 前置条件 | 无 |
| 后置条件 | 将待转发的动态的id，动态内容，待转发的目的圈子信息发送给MessageInformation |
| 不变量 | 动态id，圈子id，动态内容 |
| Messagecontroller.  Comment | Messagecontroller  MessageInformation | 语法 | Public void Comment(  Long message-id,  string comment-context  ，long uid) |
| 前置条件 | 无 |
| 后置条件 | 将评论用户的id、动态id、评论内容发送给MessageInformation |
| 不变量 | 用户id，动态id，评论内容 |
| Messagecontroller.  View | Messagecontroller  MessageInformation | 语法 | Public void View(  Long user-id,  Long message-id) |
| 前置条件 | 无 |
| 后置条件 | 将用户的id和动态id发送给MessageInformation |
| 不变量 | 用户id，动态id |
| Messagecontroller.  Praise | Messagecontroller  MessageInformation | 语法 | Public void Praise(  Long user-id,  Long message-id) |
| 前置条件 | 无 |
| 后置条件 | 将用户点赞的动态id和点赞用户的id发送给MessageInformation |
| 不变量 | 动态id，用户id |
| Messagecontroller.  Feedback | Messagecontroller  MessageInformation | 语法 | Public void Feedback(  Long user-id,  Long message-id  string context) |
| 前置条件 | 无 |
| 后置条件 | 将用户选择要反馈的动态id，反馈内容，反馈人的id发送给MessageInformation |
| 不变量 | 动态id，反馈内容，反馈人的id |
| MessageInformation.  Msg-save | MessageInformation  MessageController  MessageDAO | 语法 | Public void Msg-save(  Long uid,string context) |
| 前置条件 | 数据库连接正常 |
| 后置条件 | 将用户id和用户请求发送给数据库 |
| 不变量 | 用户id |
| MessageDAO.  Dateget | MessageInformation  MessageDAO | 语法 | Public void Dateget(  Entity me) |
| 前置条件 | 数据库连接正常 |
| 后置条件 | 从数据库获取信息数据并返回实体对象 |
| 不变量 | 用户id |

1. 个人主页模块

|  |  |  |
| --- | --- | --- |
| 类名称 | 方法 | 功能描述 |
| OwnFromUI | Show-ownfm | 显示个人主页界面 |
| OwnSetUI | Show-ownset | 显示个人设置界面 |
| FriendFromUI | Show-friendfm | 显示好友个人主页界面 |
| OwnSetController | To-ownfm | 跳转到个人主页界面 |
| To-ownset | 跳转到个人设置主页界面 |
| To-friendfm | 跳转到好友个人主页界面 |
| Save-ownset | 保存个人设置信息 |
| OwnSetInformation | Own-Informget | 从数据层获取个人设置信息 |
| Own-Informsave | 保存需要更改的个人设置信息 |
| OwnSetDAO | Ownset-save | 将需要保存的数据写入数据库，返回更改结果 |
| Ownset-get | 从数据库中读取将需要展示的数据 |



|  |  |  |  |
| --- | --- | --- | --- |
| OwnFromUI.  Show-ownfm | OwnFromUI  OwnSetcontroller | 语法 | Public void Show-ownfm(  Long uid) |
| 前置条件 | 用户选择查看个人主页界面 |
| 后置条件 | 无 |
| 不变量 | 用户id |
| OwnSetUI.  Show-ownset | OwnSetUI  OwnSetcontroller | 语法 | Public void Show-ownset(  Long uid ,string settings) |
| 前置条件 | 用户选择的设置信息不为空 |
| 后置条件 | 将设置信息发送给控制器 |
| 不变量 | 用户的设置信息 |
| FriendFromUI.  Show-friendfm | FriendFromUI  OwnSetcontroller | 语法 | Public void Show-friendfm(  Long uid) |
| 前置条件 | 用户选择查看好友的个人空间 |
| 后置条件 | 用户具有查看的权限 |
| 不变量 | 用户查看的好友id |
| OwnSetController.  Save-own-msg | OwnSetController  OwnInformation | 语法 | Public void Save-own-msg(  Long uid ,string settings) |
| 前置条件 | 无 |
| 后置条件 | 将设置信息发送给OwnSetInformation |
| 不变量 | 用户id，设置信息 |
| OwnSetController.  Save-ownset | OwnSetController  OwnSetInformation | 语法 | Public void Save-ownset(  Long user-id,  string settings) |
| 前置条件 | 无 |
| 后置条件 | 将用户id和用户的设置信息发送给OwnSetInformation |
| 不变量 | 用户id，设置信息 |
| OwnSetInformation.  Own-Informsave | OwnSetInformation  OwnSetDAO | 语法 | Public void Own-Informsave  (Long uid,string settings) |
| 前置条件 | 数据库连接正常 |
| 后置条件 | 将设置信息或用户id发送给数据库 |
| 不变量 | 设置信息、用户id |
| OwnSetDAO.  Ownset-get | OwnSetInformation  OwnSetDAO | 语法 | Public void Ownset-get(  Entity me) |
| 前置条件 | 数据库连接正常 |
| 后置条件 | 从数据库中获取信息数据并返回实体对象） |
| 不变量 | 用户id |

1. 搜索模块

|  |  |  |
| --- | --- | --- |
| 类名称 | 方法 | 功能描述 |
| SearchFromUI | Show-search | 显示初始搜索界面 |
| SearchResultUI | Show-search\_result | 显示搜索结果界面 |
| SearchController | Toshow-search | 跳转到初始搜索界面 |
| toshow-search\_result | 跳转到搜索结果界面 |
| SearchAccount | 对用户进行搜索 |
| SearchCircle | 对圈子进行搜索 |
| SearchInformation | Search-informget | 从数据层获取搜索信息信息 |
| SearchDAO | Search-get | 通过用户输入从数据库中获取搜索的信息，返回搜索结果 |



|  |  |  |  |
| --- | --- | --- | --- |
| SearchFromUI.  Show-search | SearchFromUI  SearchController | 语法 | Public void Show-search(  String search-key) |
| 前置条件 | 用户的输入不为空 |
| 后置条件 | 将用户输入的关键字发送给控制器 |
| 不变量 | 用户搜索的关键字 |
| SearchResultUI.  Show-search\_result | SearchResultUI  SearchController | 语法 | Public void Show-search-result  (string result) |
| 前置条件 | 用户的搜索有结果 |
| 后置条件 | SearchController将数据库的返回内容送给了SearchresultUI |
| 不变量 | 数据库的返回内容 |
| SearchController.  SearchAccount | SearchController  SearchInformation | 语法 | Public void SearchAccount(  string search-keys，int flag) |
| 前置条件 | 无 |
| 后置条件 | 将用户搜索的关键字和搜索对象类型（如人、圈子、动态）发送给SearchInformation |
| 不变量 | 搜索的关键字、搜索对象类型 |
| SearchController.  SearchCircle | SearchController  SearchInformation | 语法 | Public void SearchCircle(  string search-keys，int flag) |
| 前置条件 | 无 |
| 后置条件 | 将用户搜索的关键字和搜索对象类型（如人、圈子、动态）发送给SearchInformation |
| 不变量 | 搜索的关键字、搜索对象类型 |
| Search Information.  Search-informget | Search Information  SearchController  **Search DAO** | 语法 | Public void Search-information(  String search-key) |
| 前置条件 | 数据库连接正常 |
| 后置条件 | 将用户搜索的关键字发送给数据库 |
| 不变量 | 用户搜索的关键字 |
| Search DAO.  Search-get | SearchDAO  SearchController | 语法 | Public void Search-get(  Entity me) |
| 前置条件 | 数据库连接正常 |
| 后置条件 | 从数据库获取信息数据并返回实体对象 |
| 不变量 | 用户id |

1. 圈子模块

|  |  |  |
| --- | --- | --- |
| 类名称 | 方法 | 功能描述 |
| CircleListUI | Show-circlelist | 显示圈子列表 |
| CircleMainUI | Show-circle | 显示圈子主界面 |
| CircleControlUI | Show-circlecontrol | 显示圈子管理界面 |
| CircleJoinUI | Show-circlejoin | 显示圈子加入申请界面 |
| CircleController | Toshow-circle | 跳转圈子主界面 |
| Toshow-circle\_cotrol | 跳转圈子管理界面 |
| Createcircle | 创建圈子 |
| DissolveCircle() | 解散圈子 |
| Removemember | 移除圈子成员 |
| Joinmember | 加入新成员 |
| Transfer | 给与成员权限 |
| JoinController | toShow-circle\_join | 跳转圈子加入申请界面 |
| Joincircle | 提出加入圈子申请 |
| CircleInformation | Controlmsg-save | 向数据层发送圈子管理信息 |
| JoinInformation | Joinmsg-save | 向数据层发送圈子加入信息 |
| CircleDAO | Circlecontrol-data | 与数据库进行交互，获取圈子管理员有关圈子管理的信息，写入数据库，返回圈子管理结果 |
| CircleJoinDAO | Circlejoin-data | 与数据库进行交互，获取用户圈子的申请信息，写入数据库，返回圈子加入结果 |



|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CircleListUI.  Show-circle | CircleListUI  CircleController | 语法 | | | | | Public void Show-circle(  Long uid) | |
| 前置条件 | | | | | 用户选择查看个人的圈子信息 | |
| 后置条件 | | | | | 将用户的id发送给控制器 | |
| 不变量 | | | | | 用户id | |
| CircleMainUI.  Show-circle | CircleMainUI  CircleController | 语法 | | | | | Public void show-circle(  Long uid,long circle-id) | |
| 前置条件 | | | | | 用户选择查看某个圈子 | |
| 后置条件 | | | | | 将用户id和待查看圈子id发送给控制器 | |
| 不变量 | | | | | 用户id，待查看的圈子id | |
| CircleControllerUI.  Show-circle-controll | CircleControllerUI  CircleController | 语法 | | | | | Public void Show-circle-controll(  Long uid,string context) | |
| 前置条件 | | | | | 用户对于设置信息的选择不为空 | |
| 后置条件 | | | | | 将用户id和用户的设置信息发送给控制器 | |
| 不变量 | | | | | 用户id以及用户的设置信息 | |
| CircleController.createcircle | CircleController  CircleControllerui | 语法 | | | | | Public void createcircle(Long uid,string context) | |
| 前置条件 | | | | | 用户选择创建圈子按钮 | |
| 后置条件 | | | | | 将用户id和圈子创建请求信息发送给控制器 | |
| 不变量 | | | | | 用户id | |
| CircleController.DissolveCircle | CircleController  CircleControllerui | | | 语法 | | Public void DissolveCircle(Long uid,string context) | | |
| 前置条件 | | 用户选择解散圈子按钮 | | |
| 后置条件 | | | 将用户id和圈子解散请求信息发送给控制器 | |
| 不变量 | | | 用户id | |
| CircleController.Removemember | CircleController  CircleControllerui | | | | 语法 | | Public void Removemember(Long uid,string context) | |
| 前置条件 | | 用户选择移除成员按钮 | |
| 后置条件 | | | 将用户id和移除出圈子的成员信息发送给控制器 |
| 不变量 | | | 用户id |
| CircleController.Joinmember | CircleController  CircleControllerui | | 语法 | | | | Public void Joinmember (Long uid,string context) | |
| 前置条件 | | | | 用户选择加入新成员按钮 | |
| 后置条件 | | | | 将用户id和加入圈子的新成员的信息（id）发送给控制器 | |
| 不变量 | | | | 用户id | |
| CircleJoinUI.  Show-circle-join | CircleJoinUI  JoinController | 语法 | | | | | Public void Show-circle-join(  Long uid,string context) | |
| 前置条件 | | | | | 用户申请加入圈子的信息不为空 | |
| 后置条件 | | | | | | 将用户的id和申请信息发送给控制器 |
| 不变量 | | | | | | 用户id和申请信息 |
| JoinController. Joincircle | JoinController.  CircleJoinUI | 语法 | | | | | | Public void Joincircle (Long uid,string context) |
| 前置条件 | | | | | | 用户选择提出加入圈子申请按钮 |
| 后置条件 | | | | | | 将用户id和申请信息发送给控制器 |
| 不变量 | | | | | | 无 |
| ControlInformation.  Controlmsg-save | ControlInformation  CircleController  CircleDAO | 语法 | | | | | | Public void Controlmsg-save  (long uid ,string context) |
| 前置条件 | | | | | | 数据库连接正常 |
| 后置条件 | | | | | 将用户id和设置信息发送给数据库 | |
| 不变量 | | | | | 用户id和设置信息 | |
| JoinInformation.  Joinmsg-save | JoinInformation  JoinController  CircleDAO | 语法 | | | | | Public void Joinmsg-save(  Long uid,string context) | |
| 前置条件 | | | | | 数据库连接正常 | |
| 后置条件 | | | | | 将用户id和申请信息发送给数据库 | |
| 不变量 | | | | | 用户id和申请信息 | |
| CircleDAO.  Circlecontrol-data | CircleInformation  CircleDAO | 语法 | | | | | Public void Circlecontrol-date(  Entity me) | |
| 前置条件 | | | | | 数据库连接正常 | |
| 后置条件 | | | | | 从数据库获取信息数据并返回实体对象 | |
| 不变量 | | | | | 用户id | |
| CircleJoinDAO.  Circlejoin-data | Circle JoinDAO  JoinInformation | 语法 | | | | | Public void Circlejoin-data(  Entity me) | |
| 前置条件 | | | | | 数据库连接正常 | |
| 后置条件 | | | | | 从数据库获取信息数据并返回实体对象 | |
| 不变量 | | | | | 用户id | |

1. 好友模块

|  |  |  |
| --- | --- | --- |
| 类名称 | 方法 | 功能描述 |
| FriendListUI | Show-friend | 显示好友列表和对好友操作按钮 |
| FriendMainUI | Show-friendmain | 显示好友主界面  （不是个人空间，是聊天界面） |
| FriendAddUI | Show-friendadd | 显示好友申请界面 |
| AddDealUI | Show-Adddeal | 显示处理好友申请界面 |
| FriendController | Tofriendmain | 跳转好友主界面 |
| Toshow-friendlist | 跳转好友列表和好友操作界面 |
| DeleteFriend | 删除好友 |
| FriendAddController | ToFriendadd | 跳转好友申请界面 |
|  | ToAdddeal | 跳转处理好友申请界面 |
|  | Adddeal | 处理好友申请 |
|  | addfriend | 发送好友请求 |
| FriendInformation | Friendmsg-save | 向数据层发送申请对好友操作信息 |
| AddInformation | Friendmsgadd-save | 向数据层发送好友申请信息 |
| FriendDAO | Friend-data | 与数据库进行交互，返回对好友操作结果 |
| FriendAddDAO | Friendadd-data | 与数据库进行交互，返回好友申请结果 |



|  |  |  |  |
| --- | --- | --- | --- |
| FriendListUI.  Show-friend | FriendListUI  FriendController | 语法 | Public void Show-friend(  Long uid) |
| 前置条件 | 用户选择查看好友列表 |
| 后置条件 | 将用户id发送给控制器 |
| 不变量 | 用户id |
| FriendMainUI.  Show-friendmain | FriendMainUI  FriendController | 语法 | Public void Show-friend(  Long send-uid，  long aceept-uid，  string context) |
| 前置条件 | 用户选择好友聊天界面 |
| 后置条件 | 将自己和好友id、聊天信息发送给控制器 |
| 不变量 | 用户id、好友id、聊天信息 |
| FriendController.DeleteFriend | FriendController.  FriendControllerui | 语法 | Public void DeleteFriend (Long uid,string context) |
| 前置条件 | 用户选择好友删除按钮 |
| 后置条件 | 将用户id和删除好友信息发送给控制器 |
| 不变量 | 用户id |
| FriendAddUI.  Show-friendadd | FriendAddUI  FriendAddController | 语法 | Public void Show-friendadd(  Long uid,context) |
| 前置条件 | 用户的输入不为空 |
| 后置条件 | 将用户的好友申请信息发送给控制器 |
| 不变量 | 用户id和好友申请信息 |
| FriendAddController.Adddeal | FriendAddController.  FriendAddControllerui | 语法 | Public void Adddeal (Long uid,string context) |
| 前置条件 | 用户选择处理好友申请按钮 |
| 后置条件 | 将用户id和用户操作信息发送给控制器 |
| 不变量 | 用户id |
| FriendAddController.Adddfriend | FriendAddController.  FriendAddControllerui | 语法 | Public void Adddfriend (Long uid,string context) |
| 前置条件 | 用户选择好友申请按钮 |
| 后置条件 | 将用户id和好友申请请求信息发送给控制器 |
| 不变量 | 用户id |
| FriendInformation.  Friendmsg-save | FrendInformation  FriendController | 语法 | Public void Show-friend(  Long send-uid，  long aceept-uid，  string context) |
| 前置条件 | 数据库连接正常 |
| 后置条件 | 将好友和自己id以及好友操作信息发送给数据库 |
| 不变量 | 好友、自己的id以及好友操作信息 |
| AddInformation.  Friendmsgadd-save | FrendAddInformation  FriendAddController | 语法 | Public void Show-friend(  long uid，  string context) |
| 前置条件 | 数据库连接正常 |
| 后置条件 | 将待添加好友的id和申请信息发送给数据库 |
| 不变量 | 好友id，申请信息 |
| FriendDAO.  Friend-data | FriendDAO  FriendInformation | 语法 | Public void Friend-data(  Entity me) |
| 前置条件 | 数据库连接正常 |
| 后置条件 | 从数据库获取信息数据并返回实体对象 |
| 不变量 | 用户id |
| FriendAddDAO.  Friendadd-data | FriendAddDAO  FriendAddInformation | 语法 | Public void Friendadd-data(  Entity me) |
| 前置条件 | 数据库连接正常 |
| 后置条件 | 从数据库获取信息数据并返回实体对象 |
| 不变量 | 用户id |

1. 聊天模块

|  |  |  |
| --- | --- | --- |
| 类名称 | 方法 | 功能描述 |
| ChatUI | Show-chat | 显示聊天界面 |
| ChatController | Toshow-chat | 跳转聊天界面 |
| Sendchat | 发送聊天信息 |
| ChatInformation | Chatget-infor | 从数据库获取聊天相关信息 |
| Chatsave-infor | 将聊天相关信息保存到数据库 |
| ChatDAO | Chat-history | 获取历史聊天信息，返回聊天历史查询结果 |



|  |  |  |  |
| --- | --- | --- | --- |
| ChatUI.  Show-chat | ChatUI  ChatController | 语法 | Public void Show-chat(  Long uid ,long chat-uid) |
| 前置条件 | 用户选择了某个用户进行聊天 |
| 后置条件 | 将待聊天对象的id发送给控制器 |
| 不变量 | 用户id，待聊天用户的id |
| ChatController. Sendchat | ChatController.  Chatui | 语法 | Public void Sendchat (Long uid,string context) |
| 前置条件 | 用户发送聊天信息按钮 |
| 后置条件 | 将用户id和发送的聊天信息发送给控制器 |
| 不变量 | 用户id |
| ChatInformation.  Chatget-infor | ChatController  ChatInformation  ChatDAO | 语法 | Public void Chatget-infor(  String context,  long chat-uid) |
| 前置条件 | 数据库连接正常 |
| 后置条件 | Chat DAO将聊天相关信息发送给ChatInformation |
| 不变量 | 聊天信息 |
| ChatInformation.  Chatsave-infor | ChatController  ChatInformation  ChatDAO | 语法 | Public void Chatsave-infor(  String context,  long chat-uid) |
| 前置条件 | 数据库连接正常 |
| 后置条件 | ChatInformation将聊天相关信息发送给数据库 |
| 不变量 | 聊天信息 |
| ChatDAO.  Chat-history | ChatInformation  ChatDAO | 语法 | Public void Chat-history(  Entity me) |
| 前置条件 | 数据库连接正常 |
| 后置条件 | 从数据库获取信息数据并返回实体对象 |
| 不变量 | 用户id |

1. 反馈模块

|  |  |  |
| --- | --- | --- |
| 类名称 | 方法 | 功能描述 |
| FeedbackUI | Show-feedback | 显示用户反馈界面 |
| FeedbackController | toShow-feedback | 跳转用户反馈界面 |
| SendFeedback | 发送反馈信息 |
| FeedbackInformation | Feedbackmsg-get | 从数据层获得数据库反馈信息 |
| Feedbackmsg-save | 向数据层发送用户反馈相关信息 |
| FeedbackDAO | Feedback-write | 获取用户输入并写入数据库，存储用户反馈操作信息 |
| Feedback-read | 从数据库读取信息 |



|  |  |  |  |
| --- | --- | --- | --- |
| FeedbackUI.  Show-feedback | Feedback  FeedbackController | 语法 | Public void Show-feedback(  Long uid ) |
| 前置条件 | 用户选择查看反馈界面 |
| 后置条件 | 将用户id发送给控制器 |
| 不变量 | 用户id |
| FeedbackController. SendFeedback | FeedbackController. Feedbackui | 语法 | Public void SendFeedback (Long uid,string context) |
| 前置条件 | 用户选择发送反馈信息按钮 |
| 后置条件 | 将用户id和发送的反馈信息发送给控制器 |
| 不变量 | 用户id |
| FeedbackInformation.  Feedback-get | FeedbackInformation  FeedbackDAO  FeedbackController | 语法 | Public void Feedback-get(  Entity me) |
| 前置条件 | 数据库连接正常 |
| 后置条件 | 将获取的实体对象转换成数据库可以存储的对象 |
| 不变量 | 用户id |
| FeedbackInformation.  Feedback-save | FeedbackInformation  FeedbackDAO  FeedbackController | 语法 | Public void Feedback-save(  Long uid,string context) |
| 前置条件 | 数据库连接正常 |
| 后置条件 | 将用户的反馈信息发送给数据库 |
| 不变量 | 反馈信息和用户id |
| FeedbackDAO.  Feedback-wirte | FeedbackInformation  FeedbackDAO | 语法 | Public void Feedback-write(  Long uid,string context) |
| 前置条件 | 数据库连接正常 |
| 后置条件 | 将用户的反馈信息封装成数据库可以存储的数据对象 |
| 不变量 | 无 |
| FeedbackDAO.  Feedback-read | FeedbackInformation  FeedbackDAO | 语法 | Public void Feedback-read(  Entity me) |
| 前置条件 | 数据库连接正常 |
| 后置条件 | 从数据库获取信息数据并返回实体对象 |
| 不变量 | 用户id |

1. 管理员模块

|  |  |  |
| --- | --- | --- |
| 类名称 | 方法 | 功能描述 |
| AdminUI | Show-admin | 显示管理员专属界面 |
| AdminController | Toshowadmin | 跳转管理员的专属界面 |
|  | DealFeedback | 处理反馈信息 |
|  | deletemsg | 删除动态 |
|  | removemsg | 恢复动态 |
| AdminInformation | Adminmsgset | 从数据层获取管理员界面显示的信息 |
| Adminmsgget | 向数据层发送管理员的处理信息 |
| AdminDAO | admin-read | 从数据库读取管理员的信息 |
| admin-write | 获取管理员的操作信息，根据管理员操作修改存储数据库中信息，保存管理员对反馈信息的处理信息 |



|  |  |  |  |
| --- | --- | --- | --- |
| AdminUI.  Show-admin | AdminUI  AdminController | 语法 | Public void Shwo-admin(  Long uid) |
| 前置条件 | 用户登录的账号身份属于管理员 |
| 后置条件 | 将用户id发送给控制器 |
| 不变量 | 用户id |
| AdminController. DealFeedback | AdminController.  Adminui | 语法 | Public void DealFeedback (Long uid,string context) |
| 前置条件 | 用户选择处理反馈信息按钮 |
| 后置条件 | 将用户id和处理反馈操作发送给控制器 |
| 不变量 | 用户id |
| AdminController. deletemsg | AdminController.  Adminui | 语法 | Public void deletemsg (Long uid,string context) |
| 前置条件 | 用户选择删除动态按钮 |
| 后置条件 | 将用户id和删除的动态信息发送给控制器 |
| 不变量 | 用户id |
| AdminController. removemsg | AdminController Adminui. | 语法 | Public void removemsg (Long uid,string context) |
| 前置条件 | 用户选择恢复动态按钮 |
| 后置条件 | 将用户id和需要恢复的动态信息发送给控制器 |
| 不变量 | 用户id |
| AdminInformation.  Adminmsgsave | AdminController  AdminInformation  AdminDAO | 语法 | Public void Adminmsgsave(  Long uid,string context) |
| 前置条件 | 数据库连接正常 |
| 后置条件 | 将用户id和管理员的处理消息发送给数据库 |
| 不变量 | 用户id和处理消息 |
| AdminInformation.  Adminmsgget | AdminController  AdminInformation  AdminDAO | 语法 | Public void Adminmsgget(  Entirty me) |
| 前置条件 | 数据库连接正常 |
| 后置条件 | 将数据的实体类型转换成数据库可以存储的对象 |
| 不变量 | 无 |
| AdminDAO.  Admin-read | AdminInformation  AdminDAO | 语法 | Public void Admin-read(  Entity me) |
| 前置条件 | 数据库连接正常 |
| 后置条件 | 从数据库获取信息数据并返回实体对象 |
| 不变量 | 用户id |
| AdminDAO.  Admin-write | AdminInformation  AdminDAO | 语法 | Public void Admin-wirte(  Long uid , string context) |
| 前置条件 | 数据库连接正常 |
| 后置条件 | 将AdminInformation获取的处理消息和用户id封装成数据库可以存储的数据类型 |
| 不变量 | 无 |