User（用户）：

User user = new User();

User user = new User(String name , String password, String gender, String tel , String email, String address , String header);

|  |  |
| --- | --- |
| 通过用户名查询用户 | UserDAO.selectByName(String user\_name)  :User |
| 通过用户ID查询用户 | UserDAO.selectByID(String user\_ID)  :User |
| 注册用户 | UserDAO.insert(User user)  :void |
| 注销用户 | UserDAO.delete(String user\_name  :void |
| 查看用户名是否已存在 | UserDAO.nameIsExist(String user\_Name)  :boolean  存在则返回true |
| 更新用户信息 | UserDAO.update(User user)  :void  参数为新的User对象 |
| 获取当前用户的头像 | user.getheader()  :String  返回头像的存储地址 |
| 查看当前用户的所有评论(目前不能查看评论条数) | user.getCommentList()  :ArrayList<Comment> |
| 查看当前用户在论坛中发布的所有帖子（目前不能查看帖子个数） | user.getForumList()  :ArrayList<Forum> |
| 查询当前用户管理的所有社团 | user.getSocietyList()  ArrayList<Society> |
| 查询当前用户参加的所有旅游小组 | user.getTravelGroupList()  :ArrayList<TravelGroup> |

Attraction（景区）:

Attraction attraction = new Attraction();

Attraction attraction = new Attraction(String attract\_name, String address, String tel, String email, int level, float price, int play\_time, Time start, Time end, String image\_ID)

|  |  |
| --- | --- |
| 通过景区ID查询景区 | AttractionDAO.selectByID(String attract\_ID)  :Attraction |
| 通过景区名模糊查询景区 | AttractionDAO.selectByName(String attract\_name)  :ArrayList<Attraction> |
| 通过地理位置模糊查询景区 | AttractionDAO.selectByAddress(String address)  :ArrayList<Attraction> |
| 通过景区等级查询景区 | AttractionDAO.selectByLevel(int level)  :ArrayList<Attraction> |
| 管理员删除一个景区 | AttractionDAO.delete(String attract\_ID)  :void |
| 管理员新增一个景区 | AttractionDAO.insert(Attraction attraction)  :void |
| 管理员修改景区名称 | AttractionDAO.updateName  (String attract\_ID , String attract\_name)  :void |
| 管理员修改景区地理位置 | AttractionDAO.updateAddress  (String attract\_ID , String address)  :void |
| 管理员修改景区联系电话 | AttractionDAO.updateTel(String attract\_ID , String tel)  :void |
| 管理员修改景区邮箱 | AttractionDAO.updateEmail(String attract\_ID , String email)  :void |
| 管理员修改景区等级（0~5） | AttractionDAO.updateLevel(String attract\_ID , int level)  :void |
| 管理员修改景区的门票价格 | AttractionDAO,updatePrice(String attract\_ID , float price)  :void |
| 更新景区的点赞人数（用户给景区点赞或撤销点赞时用）（目前不能知道哪个用户对景区点赞了） | AttractionDAO.updateLikeNum(String attract\_ID , int like\_num)  :void |
| 更新景区页面的浏览量（每次打开景区页面时调用） | AttractionDAO.updateViews(String attract\_ID , int views)  :void |
| 管理员修改景区的建议游玩时间（单位：小时） | AttractionDAO.updatePlayTime(String attract\_ID , int play\_time)  :void |
| 管理员修改景区的开放时间 | AttractionDAO.updateStartTime(String attract\_ID , Time start)  :void |
| 管理员修改景区的关闭时间（缺少关闭时间在开放时间后的判定） | AttractionDAO.updateEndTime(String attract\_ID , Time end)  :void |
| 管理员修改景区的封面图片存储地址 | AttractionDAO.updateImageID  (String attract\_ID , String image\_ID)  :void |
| 查看当前景区名称是否已经存在 | AttractionDAO.nameIsExist(String attract\_name)  :Boolean（存在则返回true） |
| 获取当前景区的所有标签 | attraction.getLabelList()  :ArrayList<Label> |
| 获取景区的封面图片 | attraction.getCover()  :Image(com.bean) |
| 获取与当前景区相关的所有图片 | attraction.getImageList()  :ArrayList<Image>(com.bean) |
| 获取当前景区的所有评论 | attraction.getCommentList()  :ArrayList<Comment> |
| 获取当前景区聊天群中的最后num个聊天记录 | attraction.getChatList(int num)  :ArrayList<Chat> |
| 获取当前景区相关的所有论坛帖子 | attraction.getForumList()  :ArrayList<Forum> |
| 获取与当前景区相关的所有攻略 | attraction.getStrategyList()  :ArrayList<Strategy> |

Label（标签：美食、摄影、历史、古建筑……）

Label label = new Label(String attract\_ID , String label)

|  |  |
| --- | --- |
| 管理员给景区添加一条标签 | LabelDAO.insert(Label label)  :void |
| 管理员删除一条标签 | LabelDAO.delete（Label label）  :void |
| 查看一个景区的所有标签列表 | LabelDAO.selectLabelList(String attract)  :ArrayList<Label> |
| 查看一个标签的所有景区 | LabelDAO.selectAttractionListBylabel(String label)  :ArrayList<Attraction> |
| 判断一个标签是否已经存在 | LabelDAO.labelIsExist()  :boolean（若已经存在则返回true） |

Image（图片）：

Image image = new Image();

Image image = new Image(String attract\_ID, String image\_adr);

|  |  |
| --- | --- |
| 添加一张与景区有关的图片 | ImageDAO.insert(Image image)  :void |
| 删除一张图片 | ImageDAO.delete(String image\_ID)  :void |
| 通过图片ID查询图片 | ImageDAO.selectByID(String image\_ID)  :Image(com.bean) |
| 查询一个景区对应的所有图片 | ImageDAO.selectByAttractID(String attract\_ID)  :ArrayList<Image> |
| 查看当前存储地址是否被占用(缺少一个自动生成图片 地址+名字 的函数) | ImageDAO.addressOccupied(String adr)  :boolean（被占用则返回true） |
| 读取当前bean对象对应的图片 | image.getImage()  :java.awt.Image |

Comment（评论）：

Comment comment = new Comment();

Comment comment = new Comment(String user\_ID , String attract\_ID , int score , String content);

|  |  |
| --- | --- |
| 通过评论ID查询评论 | CommentDAO.selectByID(String comment\_ID)  :Comment |
| 查询一个用户的所有评论 | CommentDAO.selectByUserID(String user\_ID)  :ArrayList<Comment> |
| 查询一个景区的所有评论 | CommentDAO.selectByAttractID(String attract\_ID)  :ArrayList<Comment> |
| 删除一条评论 | CommentDAO.delete(comment\_ID)  :void |
| 添加一条评论 | CommentDAO.insert(Comment comment) |

Chat（聊天群里的一条消息）

Chat chat = new Chat();

Chat chat = new Chat(String attract\_ID , String user\_ID , String message);

|  |  |
| --- | --- |
| 通过消息ID查询消息 | ChatDAO.selectByID(String chat\_ID)  :Chat |
| 查询一个景区的聊天群中前num条聊天记录 | ChatDAO.selectByAttractID(String attract\_ID , int num)  :ArrayList<Chat> |
| 撤回一条消息（发送超过2分钟则不能撤回） | ChatDAO.delete(String chat\_ID)  :void |
| 添加一条消息 | ChatDAO.insert(Chat chat)  :void |

Strategy（攻略）：

Strategy strategy = new Strategy();

Strategy strategy = new Strategy(String attract\_ID, String address, String topic, Blob content, String image);

|  |  |
| --- | --- |
| 通过攻略的ID查询攻略 | StrategyDAO.selectByID(String strategy\_ID)  :Strategy |
| 查询一个景区对应的所有攻略 | StrategyDAO.selectByAttractID(String attract\_ID)  :ArrayList<Strategy> |
| 查询对应一个地区的所有攻略 | StrategyDAO.selectByAddress(String address)  :ArrayList<Strategy> |
| 删除一条攻略 | StrategyDAO.delete(String strategy\_ID)  :void |
| 添加一条攻略 | StrategyDAO.insert(Strategy strategy)  :void |
| 修改攻略对应的景区编码 | StrategyDAO.updateAttractID  (String strategy\_ID , String attract\_ID)  :void |
| 修改攻略对应的地域 | StrategyDAO.updateAddress  (String strategy\_ID , String address)  :void |
| 修改攻略的内容 | StrategyDAO.updateContent  (String strategy\_ID , Blob content)  :void |
| 修改攻略封面的图片地址 | StrategyDAO.updateImage  (String strategy\_ID , String image)  :void |
| 修改攻略的点赞数（目前不能查看点赞用户） | StrategyDAO.updateLikeNum  (String strategy\_ID , int like\_num)  :void |

Forum（论坛帖子）：

Forum forum = new Forum();

Forum forum = new Forum(String user\_ID, String attract\_ID, String topic, String forum\_content);

|  |  |
| --- | --- |
| 通过ID查找帖子 | ForumDAO.selectByID(String forum\_ID)  :Forum |
| 查询一个用户在论坛上发布的所有帖子 | ForumDAO.selectByUserID(String user\_ID)  :ArrayList<Forum> |
| 查询与一个景区相关的所有论坛帖子 | ForumDAO.selectByAttractID(String attract\_ID)  :ArrayList<Forum> |
| 删除一个帖子 | ForumDAO.delete(String forum\_ID)  :void |
| 添加一条帖子 | ForumDAO.insert(Forum forum)  :void |
| 查询当前帖子的所有评论（目前不能查看评论个数） | forum.getForumAnswerList()  :ArrayList<ForumAnswer> |

ForumAnswer（帖子的一条评论）

ForumAnswer forumAnswer = new ForumAnswer();

ForumAnswer forumAnswer = new ForumAnswer(String forum\_ID, String user\_ID, String answer);

|  |  |
| --- | --- |
| 通过ID查询一条评论 | ForumAnswerDAO.selectByID(String answer\_ID)  :Forumanswer |
| 查询一条帖子的所有评论 | ForumAnswerDAO.selectByForumID(String forum\_ID)  :ArrayList<ForumAnswer> |
| 删除一条对帖子的评论 | ForumAnswerDAO.delete(String answer\_ID)  :void |
| 向数据库中插入一条对帖子的评论 | ForumAnswerDAO.insert(ForumAnswer forumanswer)  :void |

Society（旅游社团）：

Society society = new Society();

Society society = new Society(String society\_name, String user\_ID, String introduce, String tel, String email, String address);

|  |  |
| --- | --- |
| 通过社团ID查询社团 | SocietyDAO.selectByID(String society\_ID)  :Society |
| 通过社团名称查询旅游社团(模糊查询) | SocietyDAO.selectBySocietyName(String society\_name)  :ArrayList<Society> |
| 查询一个用户管理的所有社团 | SocietyDAO.selectBuUserID(String user\_ID)  ArrayList<Society> |
| 查询一个区域内的所有旅游社团 | SocietyDAO.selectByAddress(String addredd>  ArrayList<Society> |
| 注销一个社团 | SocietyDAO.delete(String society\_ID)  :void |
| 注册一个社团 | SocietyDAO.insert(Society society) |
| 修改社团的名称 | SocietyDAO.updateSocietyName  (String society\_ID , String society\_name)  :void |
| 修改社团的管理员 | SocietyDAO.updateUserID(String society , String user\_ID)  :void |
| 修改社团的介绍 | SocietyDAO.updateIntroduce  (String society , String introduce)  :void |
| 修改社团的联系电话 | SocietyDAO.updateTel(String society , String tel)  :void |
| 修改社团的email | SocietyDAO.updateEmail(String society , String email)  :void |
| 修改社团的地址 | SocietyDAO.updateAddress(String society , String address)  :void |
| 判断当前社团名称在数据库中是否存在 | SocietyDAO.nameIsExist(String society\_name)  :boolean（若存在则返回true） |
| 获取当前社团发布的所有旅游团信息 | society.getTravelGroupList()  :ArrayList<TravelList> |

TravelGroup（旅游社团发布的一个团体游小组）

TravelGroup travelGroup = new TravelGroup();

TravelGroup travelGroup = new TravelGroup(String society\_ID, String topic, String description, Timestamp start, Timestamp end);

|  |  |
| --- | --- |
| 通过旅游小组ID查询旅游社团 | TravelGroupDAO.selectByID(String group\_ID)  :TravelGroup |
| 查询一个旅游社团发布的所有组团游列表 | TravelGroupDAO.selectBySocietyID(String society\_ID)  :ArrayList<TravelGroup> |
| 删除一个Travel Group | TravelGroupDAO.delete(String group\_ID)  :void |
| 向数据库中插入一个旅游小组 | TravelGroupDAO.insert(TravelGroup travelGroup)  :void |
| 目前不能通过旅游时间查询旅游团 |  |
| 查询当前小组的所有成员 | travelGroup.getGroupMemberList()  :ArrayList<User> |

GroupMenber（旅游团小组成员）

GroupMenber groupMenber = new GroupMenber();

GroupMenber groupMenber = new GroupMenber(String group\_ID, String user\_ID)

|  |  |
| --- | --- |
| 查询旅游小组内的所有小组成员 | GroupMenberDAO.selectByGroupID(String group\_ID)  ArrayList<User> |
| 查询一个用户加入的所有旅游小组 | GroupMenberDAO.selectByUserID(String user\_ID)  ArrayList<TravelGroup> |
| 删除一个小组成员 | GroupMenberDAO.delete(String group\_ID , String user\_ID)  :void |
| 添加一个小组成员 | GroupMenberDAO.insert(GroupMenber groupMenber)  :void |