# 软件系统设计

1. **系统概述**

知否-知识问答社区系统要求的主要功能分别是登录、注册、提问、回答、评论，要求用数据库保存用户的数据，数据主要内容为用户的ID、帐户密码、浏览记录等以及用户使用系统功能时储存所用的数据，并跟据内容为其推荐相应的信息，系统性能方面，要求有合理的流畅度让户体有舒适的体验。

1. **运行环境**

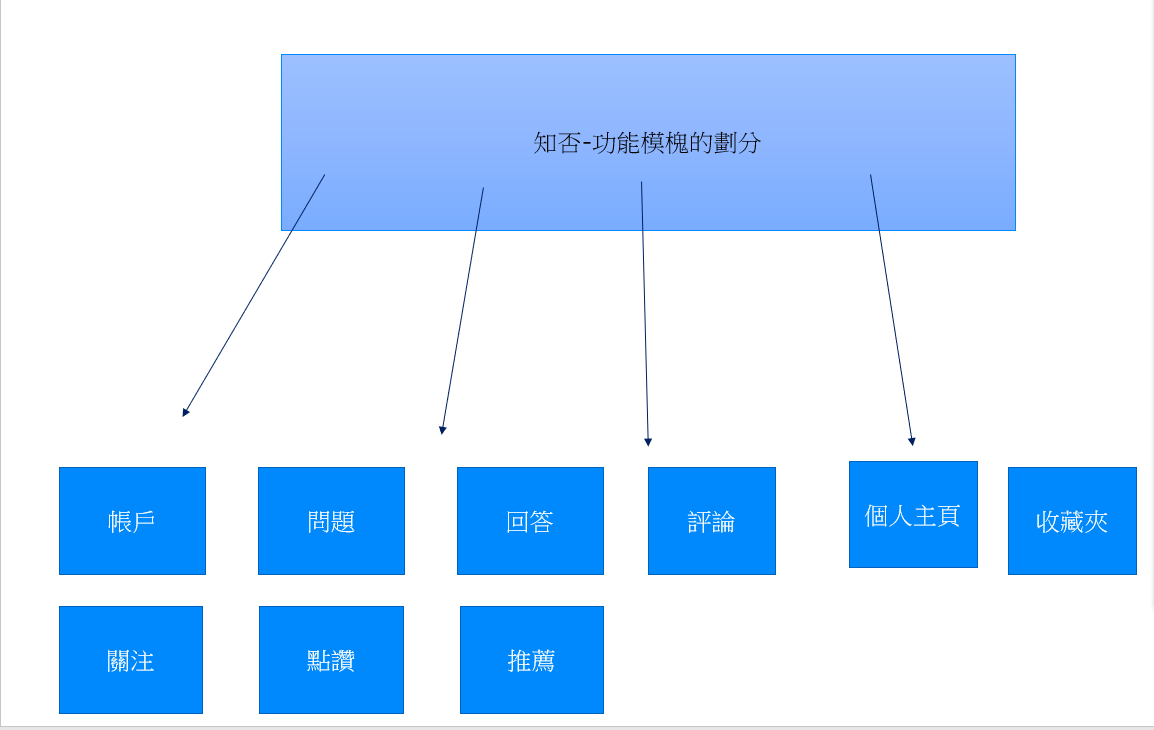
系统运行的操作系统：windows

数据库：MySQL数据库，性能卓越，服务稳定。前端技术：HTML+CSS+JSP后前端编程语言:HTML、CSS、JSP

后端编程语言：JAVA

java框架：SSM(Spring+SpringMVC+MyBatis)

1. **系统模块**



账户:该功能模槐负责用户向平台登录、注删等的功能。

问题:该功能模块负责用户向平台上提问、和修改提问的功能。

回答: 该功能模块负责该用户向平台发布提问后，别的用在该提问下回答问题，支持修该回答和删除回答等功能。

评论: 该功能模槐负责该用户回答其他用户问题，并在该问题下容许用户进行回答

话题: 该功能模槐负责将用户提出的回答进行分类，让用户可以在该分类下对有关的提问进行浏览、回答。

收藏: 该功能模槐负责将用户的回答行收藏或取消收藏，并可以根据随时查看收藏夹来浏览该回答。

点赞: 该功能模槐负责将用户的回答进行点赞，将其回答下的点赞数加一，每人只可点赞一次。

关注: 该功能模槐负责用户关注其他用户，点击关注后，用户能查看该用户的回答或点赞记录。

推荐: 该功能模槐负责向用户推荐内容，根据用户的浏览、点赞、关注的话题等，用户推荐其感兴趣的内容。

个人主页: 该功能模槐负责用户展示自己的个人资料，如出生日期、工作、城等等。

1. **模块功能**

1.账户：一个账户包含用户的用户名，密码，手机号等。每名用户拥有一个账户，每位用户可以注册一个账户，设置自身的用户名跟密码，同时绑定自己的手机号。每位注册了账户的用户可以通过自己在注册时所用的账户名与密码进行登陆操作，也可通过自己在注册时绑定的手机号申请验证码进行短信登陆验证从而实现登陆。每个账户只有一个用户id，是用户在使用该网站时的身份凭证。

2.问题：拥有账户的用户可以在平台上发表自己的问题，供他人浏览或者回答，每个问题都有其问题内容，发布时间，提问者的id，若问题被回答过，下方会显示回答者的回答内容。问题可以被提问者自行修改

3.回答：拥有账户的用户在平台浏览问题时可对其感兴趣的话题进行回答，回答的内容会显示在问题的下方同时也包括回答用户的id以及回答时间。回答的问题可以被其他用户进行评论。回答内容可以被回答者自行修改。

4.评论：拥有账户的用户在平台浏览问题时可对其感兴趣的问题的答案进行评论，评论的内容会显示在回答的答案的下方同时也包括评论用户的id以及评论时间。评论的问题可以被多个其他用户进行评论。评论内容可以被评论者自行修改。

5.话题：系统会根据用户提出的若干问题进行分类，问题的内容或者性质相关性很强的问题形成一类归为一个话题。当用户对某一特点感兴趣的时候可以选择对应特点的话题进行浏览，提问，回答，评论。

6.收藏：拥有账户的用户可以在浏览，回答，评论问题时对该问题经行收藏，收藏之后的问题会添加到用户自身的收藏夹中，用户可通过个人主页轻松找到收藏夹并点击问题以回顾该问题。

7.关注：拥有账户的用户可以在浏览，回答，评论问题时对问题的提出者或者回答的回答者或评论的评论者经行关注，关注之后的问用户会添加到用户自身的关注列表中，用户可通过个人主页轻松找到关注列表并点击该用户以浏览该用户所提出的问题以及其对其他问题的评论回答。

8.点赞：拥有账户的用户在浏览问题时，如果遇到自己感兴趣或者自己赞同的回答或者评论可以对该回答或评论进行点赞操作，每个人对一个回答或者评论只能点赞一次。

9.推广：系统会根据用户自身点赞评论收藏浏览的话题次数自动推选出可能符合用户自身喜好的话题进行推广，方便用户浏览查看。推广的话题或者问题内容一般会在主页显示

10.个人主页：个人主页包括了用户详细信息，用户的收藏夹，用户的关注列表，用户自身所提出的问题，对问题的回答以及对问题回答的评论。用户详细信息显示了用户的基本信息，如昵称，出生日期等等

1. **模块的属性和方法**
2. **账户：**

|  |  |  |  |
| --- | --- | --- | --- |
| 方法名 | 参数 | 返回值 | 方法说明 |
| login | telephone : String  password : String | message : String  userId : int | 登录，根据电话号与密码进行信息验证，返回用户ID |
| signup | telephone : String  password : String  name : String | message : String  userId : int | 注册，将用户电话、密码、用户名注入数据库，返回用户ID |
| checkTelephone | telephone : String | message : String | 验证电话号是否已注册 |
| getVerification | null | message : String | 申请获取验证码 |
| checkVerification | verification : String | message : String | 检测用户输入的验证码 |
| modifyPassword | userId : int  password : String | Message : String | 修改密码，根据用户ID来修改密码 |

1. **用户：**

|  |  |  |  |
| --- | --- | --- | --- |
| 方法名 | 参数 | 返回值 | 方法说明 |
| viewHomepage | userId : int | message : String  targetUser : User | 访问主页，以目标用户ID为参数，返回该用户的信息 |
| modifyPersonalInfo | userId : int  name : String  sex : String  birthday : Date | message : String | 修改个人信息 |
| queryUser | name : String | message : String  userList : List<User> | 根据用户名来搜寻用户 |

1. **关注：**

|  |  |  |  |
| --- | --- | --- | --- |
| 方法名 | 参数 | 返回值 | 方法说明 |
| followUser | fanId : int  userId : int | message : String | 关注用户 |
| unfollowUser | fanId : int  userId : int | message : String | 取消关注用户 |
| followerList | userId : int  start : int  count : int | message : String  userList : List<User> | 获取粉丝列表，根据用户ID查询，获取从start开始，count位粉丝信息 |
| followingList | userId : int  start : int  count : int | message : String  userList : List<User> | 获取关注用户列表，根据用户ID查询，获取从start开始，count位关注用户信息 |

1. **话题：**

|  |  |  |  |
| --- | --- | --- | --- |
| 方法名 | 参数 | 返回值 | 方法说明 |
| bringUpTopic | userId : int  topicName : String  topicContent : Object | message : String  topicId : int | 创建话题 |
| checkTopicName | topicName : String | message : String  topic : Topic | 检测话题是否已存在，若存在则进行导航 |
| queryTopic | topicName : String | message : String | 根据关键字搜索话题 |

1. **问题：**

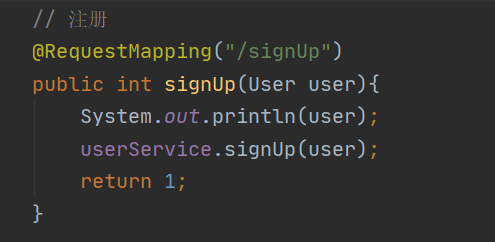
|  |  |  |  |
| --- | --- | --- | --- |
| 方法名 | 参数 | 返回值 | 方法说明 |
| askQuestion | fanId : int  userId : int | message : String | 关注用户 |
| modifyQuestion | fanId : int  userId : int | message : String | 取消关注用户 |
| queryQuestionById | userId : int  start : int  count : int | message : String  userList : List<User> | 获取粉丝列表，根据用户ID查询，获取从start开始，count位粉丝信息 |
| queryQuestionByTitle | userId : int  start : int  count : int | message : String  userList : List<Question> | 获取关注用户列表，根据用户ID查询，获取从start开始，count位关注用户信息 |
| queryQuestionListByTopicId | topicId : int  start : int  count : int | message : String  userList : List<Question> | 根据话题ID获取问题列表，从start起，count个 |
| queryQuestionListByUserId | userId : int  start : int  count : int | message : String  userList : List<Question> | 根据用户ID获取问题列表，从start起，count个 |
| queryQuestionNumberByTopicId | topicId : int | message : String  count : int | 根据话题ID获取问题数量 |
| queryQuestionNumberByUserId | userId : int | message : String  count : int | 根据用户ID获取问题数量 |

1. **模块的编码实现和单元测试**

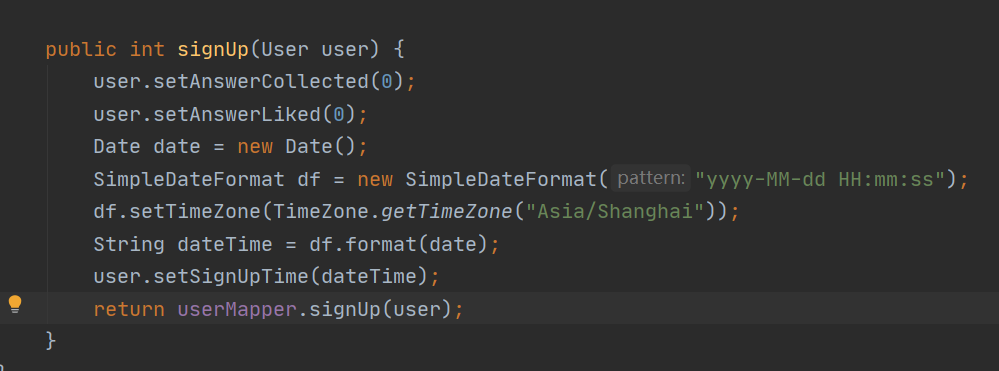
对于每个模块，要实现模块下的方法就是为前端编写调用的controller和接口，从而实现对数据库的增删改查。

项目的调用层级主要为controller->service->dao，service层和dao层为controller编写操作数据库的接口，而controller函数中通过接口实现类的对象进行数据交互。

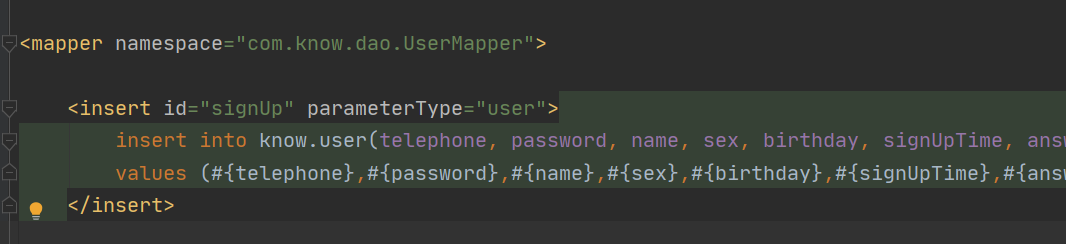
比如账户模块下的注册接口：



前端传回的是一个User类的对象，接着我们调用userService的实现类来进行插表操作



Service层调用了dao层UserMapper的对象并将处理过的User对象传给它进行注册操作，dao层对数据库的操作如下：



可以看到进行了插表的操作，对应数据库中user表的插入，将注册的信息比如电话号码，密码等信息插入数据库。

以上就是我们项目模块下方法实现的实例，其他的诸如登录等方法或者其他模块，回答问题等方法的代码实现同上所示。

单元测试方面，对于controller和service层我们可以使用Junit进行测试，方便有效，对于整个模块的功能，我们编写jsp文件进行单元测试，比如上述注册方法的jsp文件



编写好url方便调用接口，并且传输user对象下的属性，telephone，password等，然后运行Tomcat，并在网页中执行注册操作：



查看数据库中是否成功插入数据：



测试成功。