

中山大学教学平台

软件设计文档

15332014 田桂雄 15331285 孙小旋
15331254 彭一峰 14331072 高炜东
15331268 邵柏卿 14331023 陈举平
15331279 苏菲



目录

1 架构设计	3
1.1 架构描述	3
1.2 架构图	3
1.3 关键抽象	3
2 用例分析	4
2.1 补充用例规约	4
2.2 用例中类的析取	5
2.2.1 用户注册用例析取	5
2.2.2 用户登陆用例析取	6
2.2.3 评论问题用例析取	7
2.2.4 发布问题用例析取	8
2.2.5 点赞问题用例析取	9
2.2.6 用户修改昵称用例析取	10
2.2.7 查看问题用例析取	11
2.3 分析机制	13
2.4 合并分析类	13
3 子系统及其接口设计	14
3.1 确定设计类	14
3.2 定义子系统	14
3.3 定义接口	15
3.3.1 评论模块子系统	15
3.3.2 点赞模块子系统	16
3.3.3 用户模块子系统	17
3.3.4 发布问题模块子系统	18
3.4 可重复使用子系统	19
4 描述系统运行时架构	19

1 架构设计

1.1 架构描述

本系统基于 Django 框架，采用 MTV (Model Template View) 三层架构。该架构本质上与 MVC (Model View Controller) 是一样的，可以保持各组件间的松耦合关系。该架构的使用实现了应用程序的分层管理，简化了后续对程序的修改和扩展，并且使程序某一部分的重复利用成为可能。

1) 数据存取层 (Model)

Model 层，即数据存取层，主要作用是处理与数据相关的所有事务：如何存取、如何验证有效性、包含哪些行为以及数据之间的关系等。

2) 业务逻辑层 (Template)

Template 层，即业务逻辑层，负责处理与表现相关的决定：如何在页面或其他类型文档中进行对用户的展示，在本架构中大多数使用 html 的形式。

3) 表示层 (View)

View 层，即表示层，负责存取模型及调取恰当模板的相关逻辑，并在恰当时候调用 Model 和 Template，是模型与模板之间的桥梁。

1.2 架构图

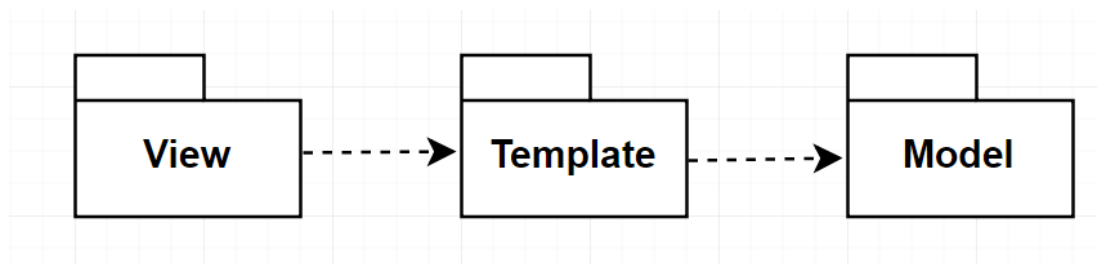


图 1-1 中山大学师生教学平台架构图

1.3 关键抽象

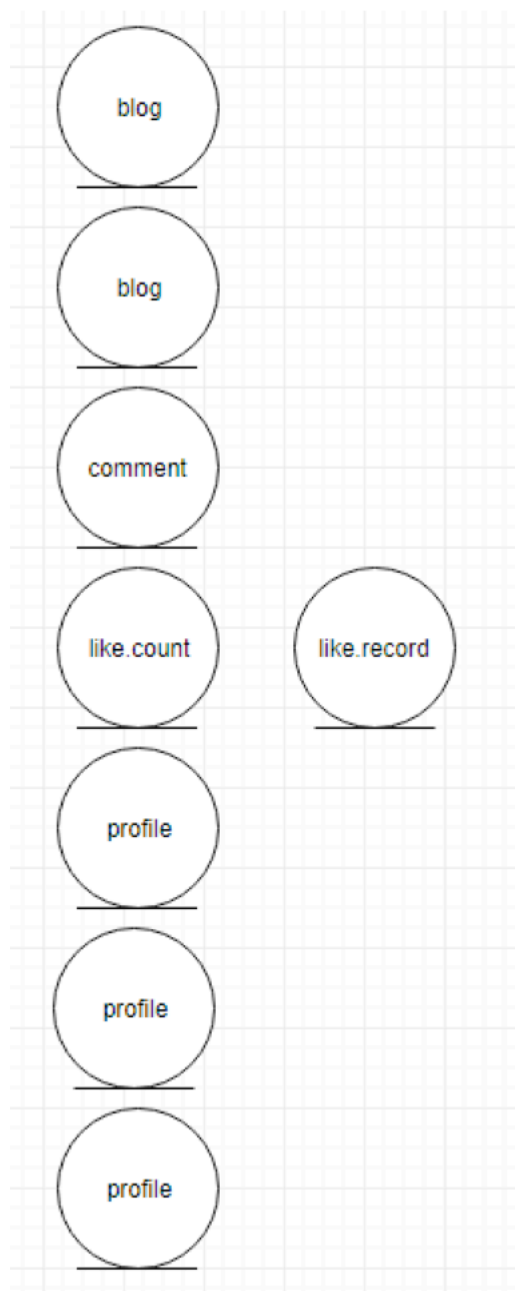


图 1-1 关键抽象（实体类）

2 用例分析

2.1 补充用例规约

经检查，本项目还应补充“查看问题”这个用例。

2.2 用例中类的析取

2.2.1 用户注册用例析取

画出用例析取图：

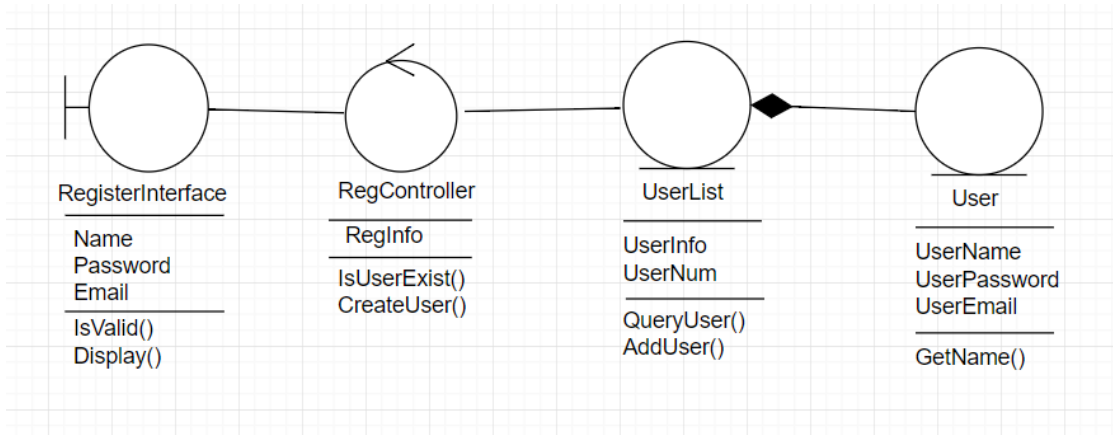


图 2-1 用户注册用例析取图

画出时序图：

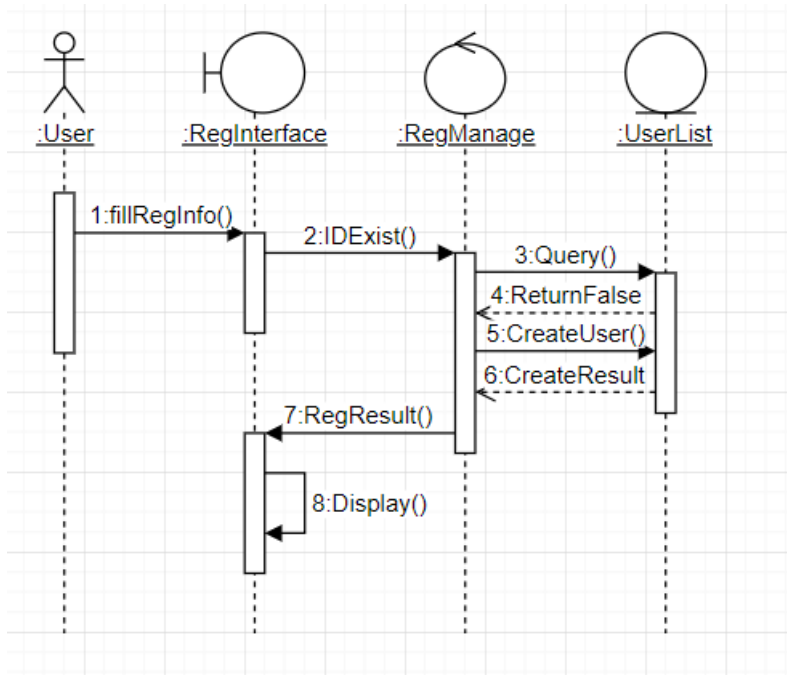


图 2-2 用户注册时序图

用户填写注册页面所需要的信息，之后系统会进入用户列表查看用户名是否已经存在，返回不存在的结果之后，在用户列表中创建该用户，创建成功后将注册成功的结果返回到注册页面上。

2.2.2 用户登陆用例析取

画出用例析取图：

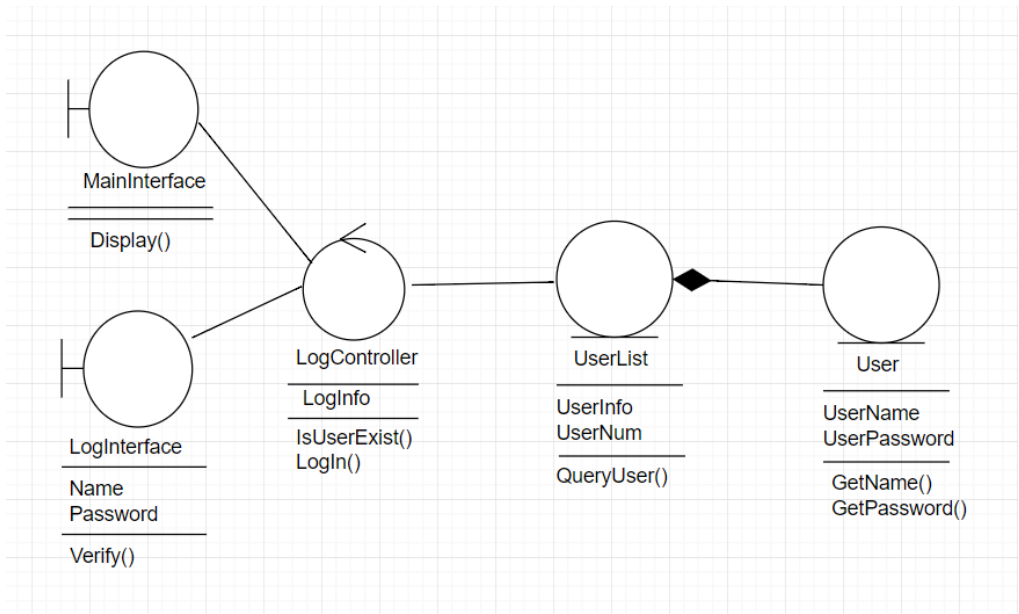


图 2-3 用户登录用例析取图

画出时序图：

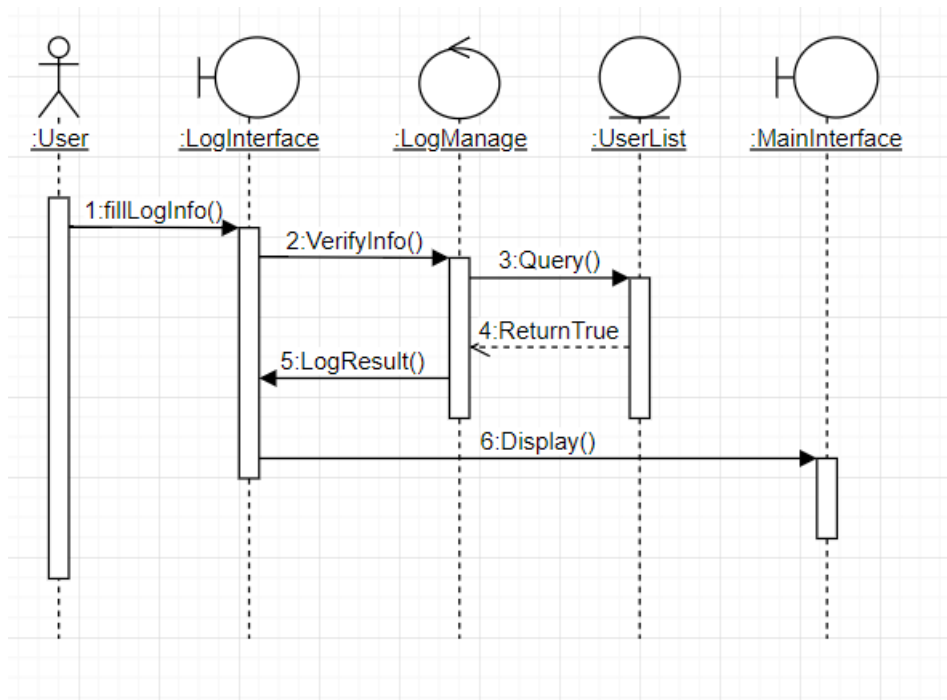


图 2-4 用户登录时序图

用户输入登录所需信息，系统会在用户列表中进行数据比对，将信息匹配成功的结果返回后，在登录页面显示登录成功，之后用户自动进入主界面。

2.2.3 评论问题用例析取

画出用例析取图：

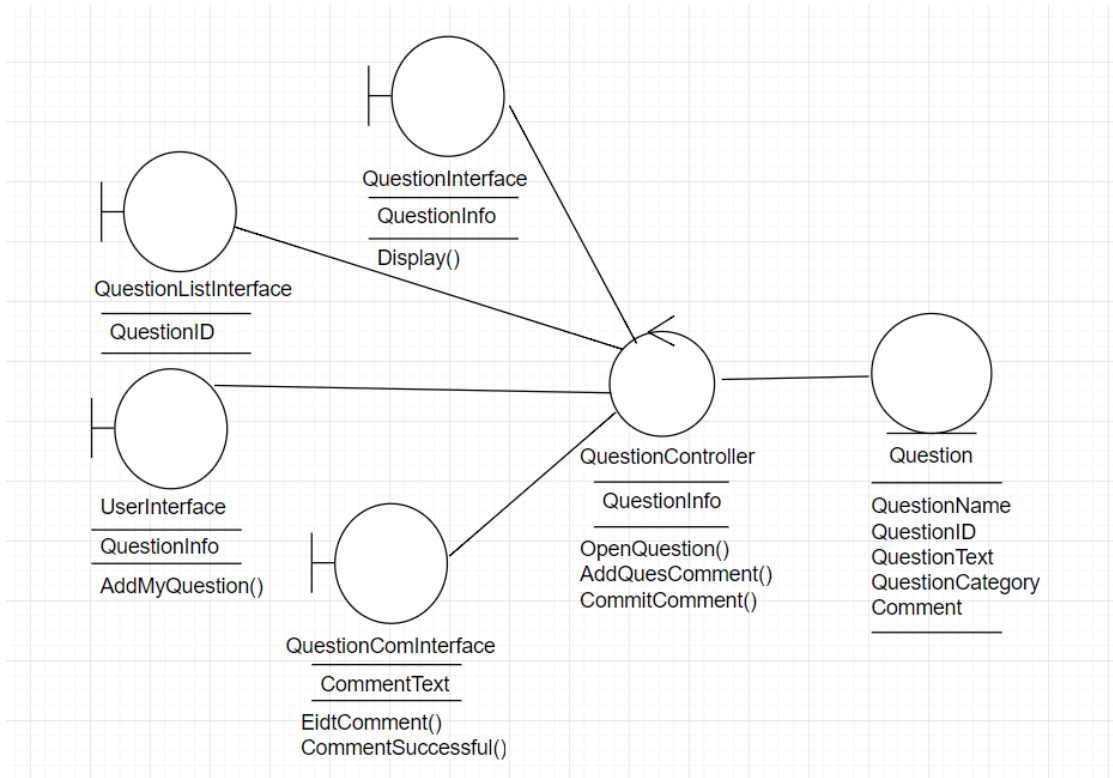


图 2-5 评论问题用例析取图

画出时序图：

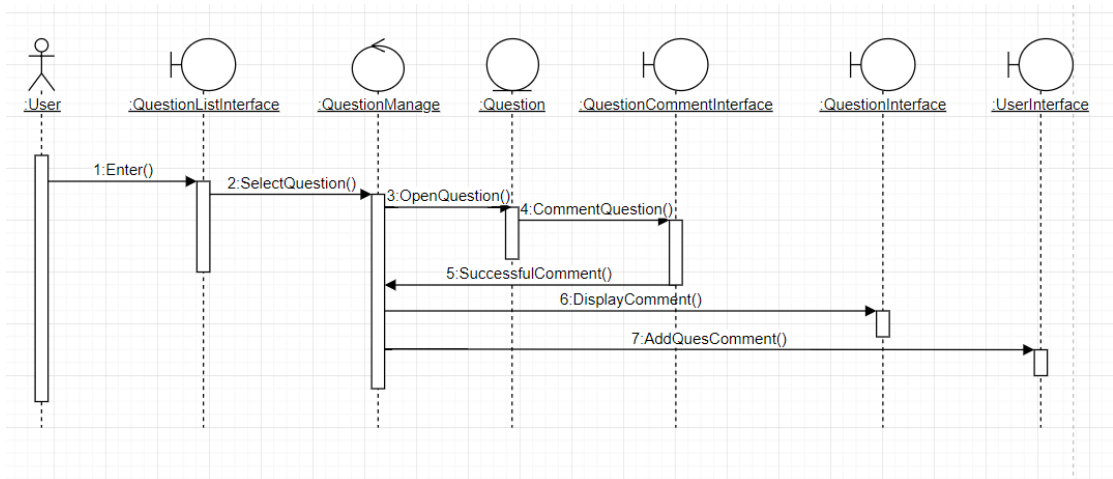


图 2-6 评论问题时序图

用户先进入问题列表界面，选择一个问题并进入，之后在问题下方进行评论，成功提交评论后，评论内容会在该问题界面中显示出来，同时会在用户个人信息界面中添加此被评论过的问题。

2.2.4 发布问题用例析取

画出用例析取图：

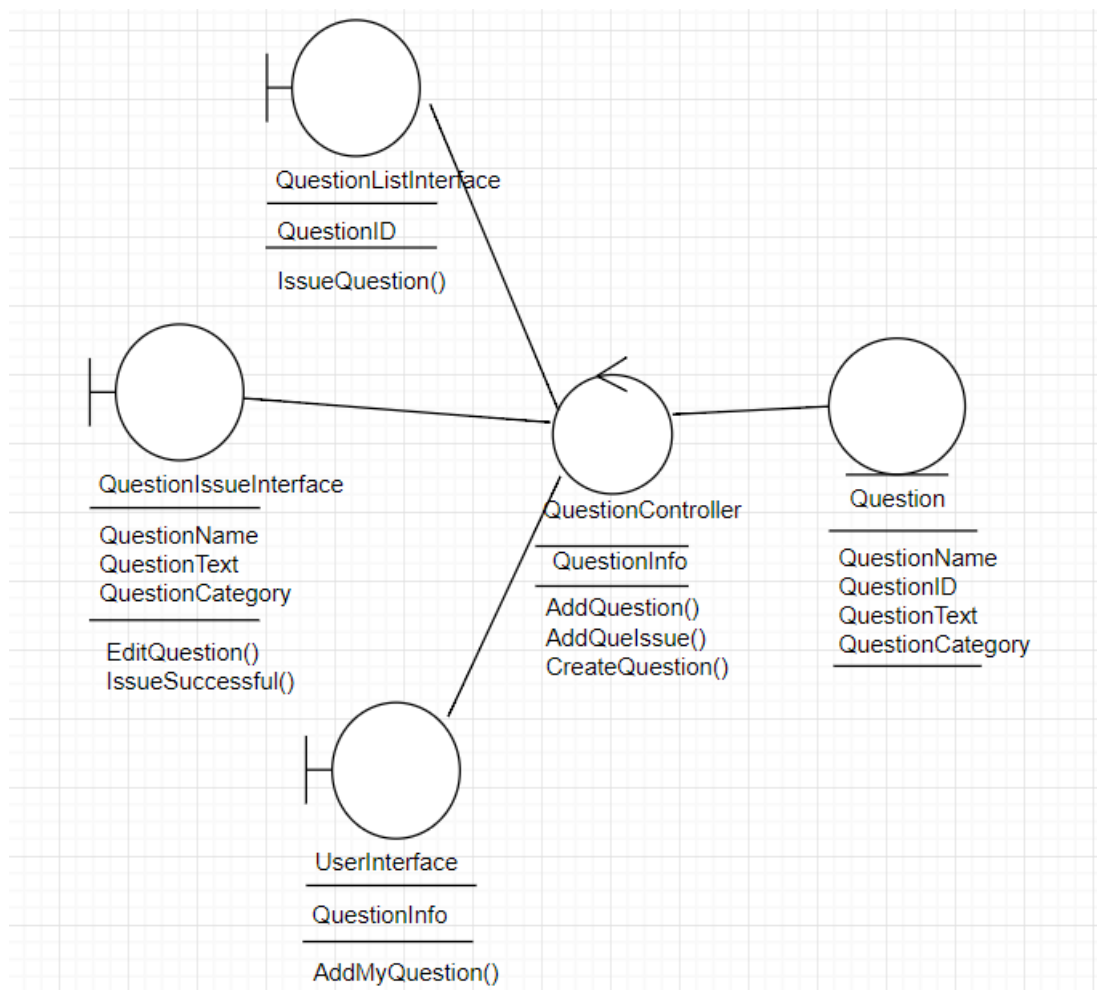


图 2-7 发布问题用例析取图

画出时序图：

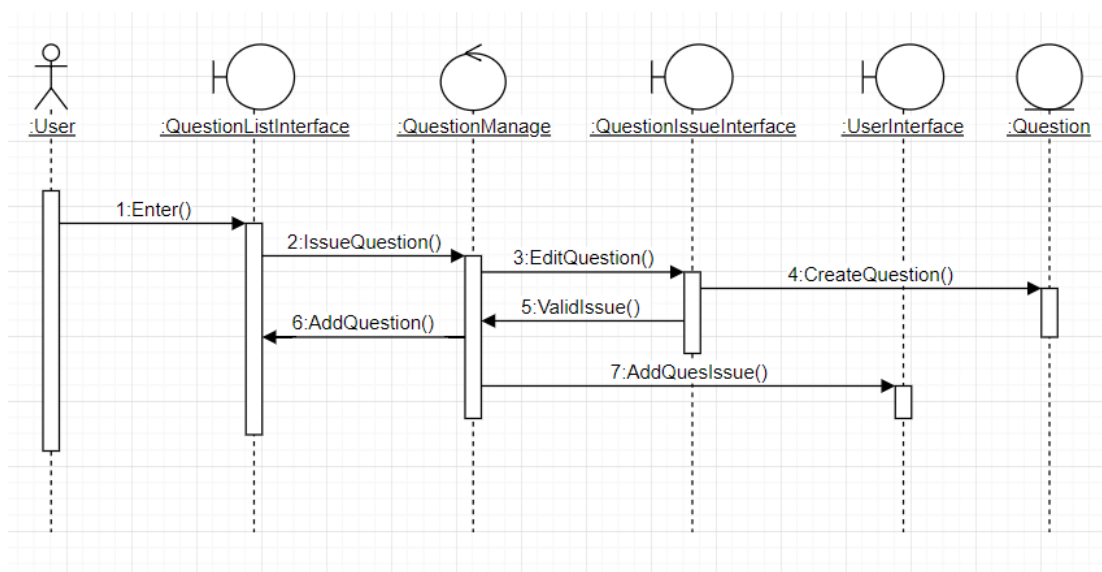


图 2-8 发布问题时序图

用户进入问题列表页面，点击发布问题，之后输入问题的具体信息，包括问题名称和问题内容，点击提交后，问题发布成功，之后会在问题列表页面中加上刚刚发布的问题，也会在用户的个人信息页面中加上自己发布的问题。

2.2.5 点赞问题用例析取

画出用例析取图：

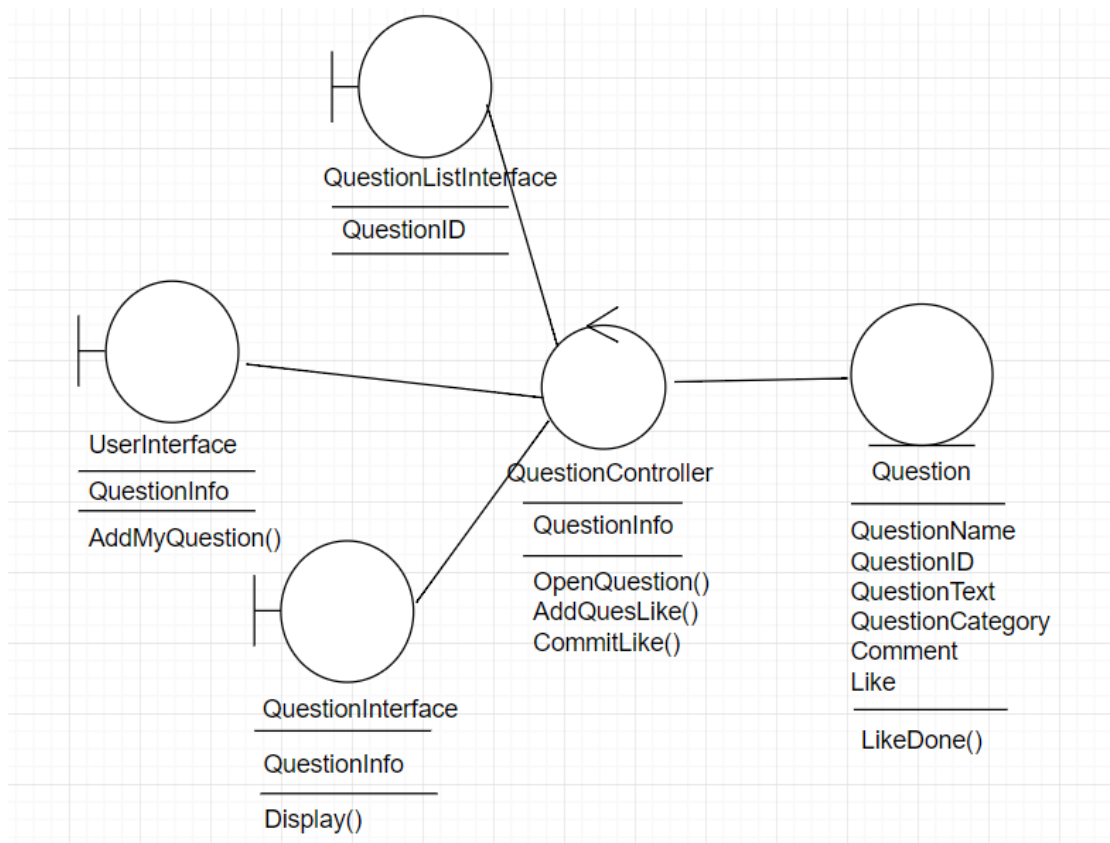


图 2-9 点赞问题用例析取图

画出时序图：

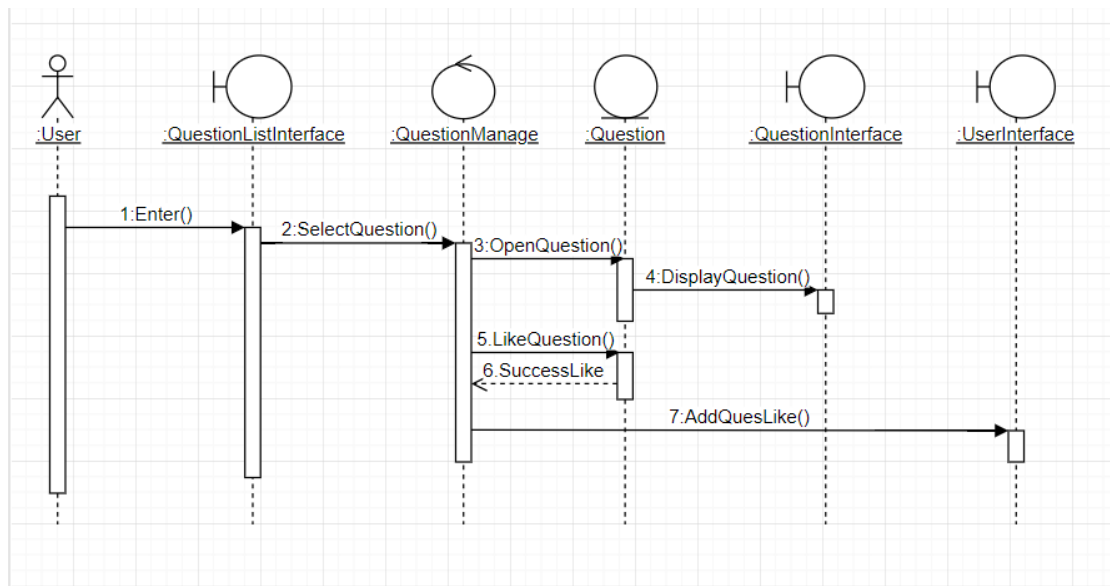


图 2-10 点赞问题时序图

用户先进入问题列表界面，选择一个问题并进入，之后在问题下方进行点赞操作，成功提交后，该问题的总点赞个数会增加，同时会在用户个人信息界面中添加此被点赞过的问题。

2.2.6 用户修改昵称用例析取

画出用例析取图：

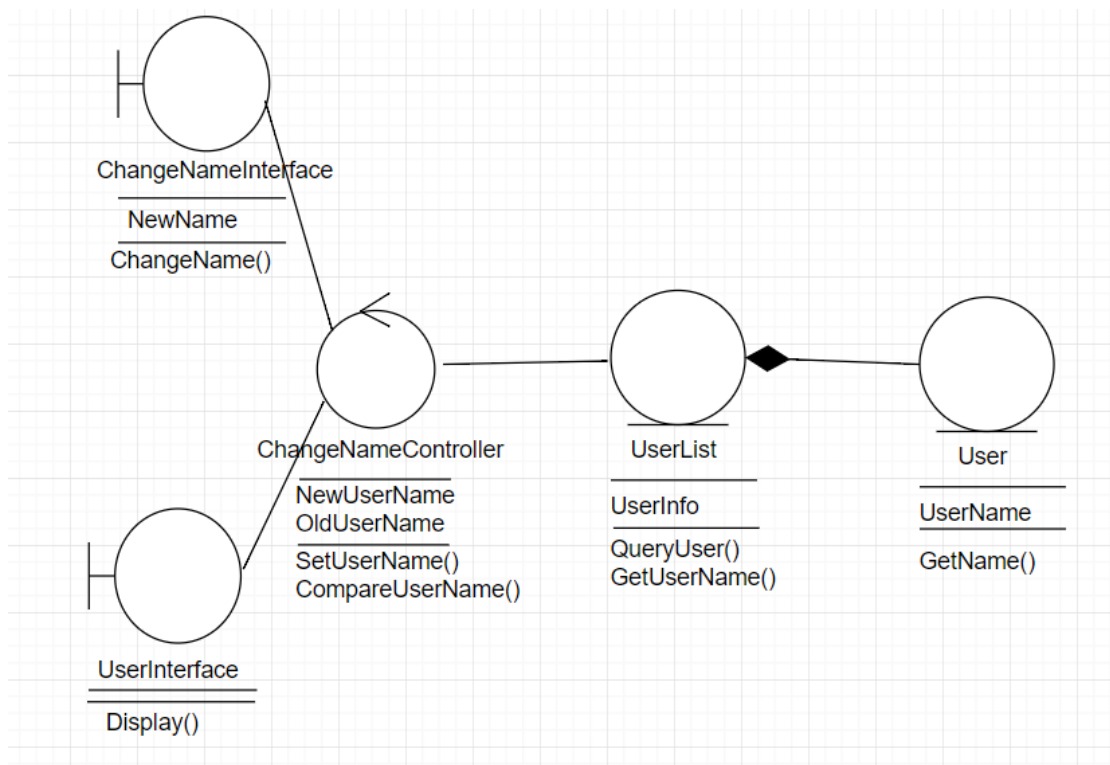


图 2-11 用户修改昵称用例析取图

画出时序图：

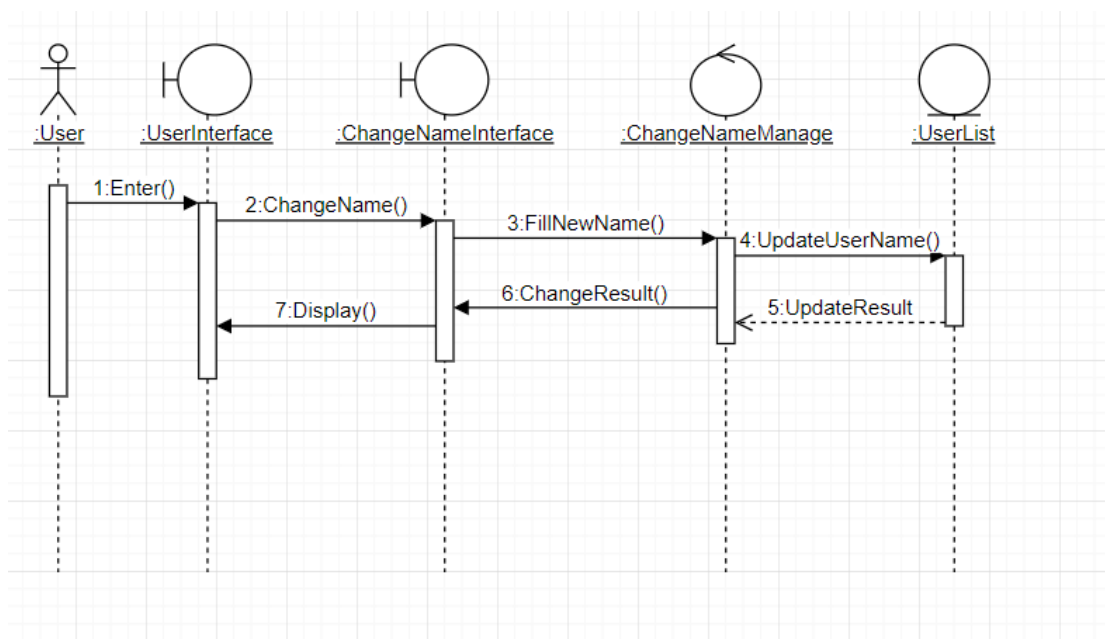


图 2-12 用户修改昵称时序图

用户先进入个人信息界面，选择更改昵称，之后在更改昵称的界面输入想改的新昵称，点击提交后，更新用户列表中的相应数据，之后在用户更改昵称界面显示更改昵称成功，并回到个人信息界面，显示出新的昵称。

2.2.7 查看问题用例析取

画出用例析取图：

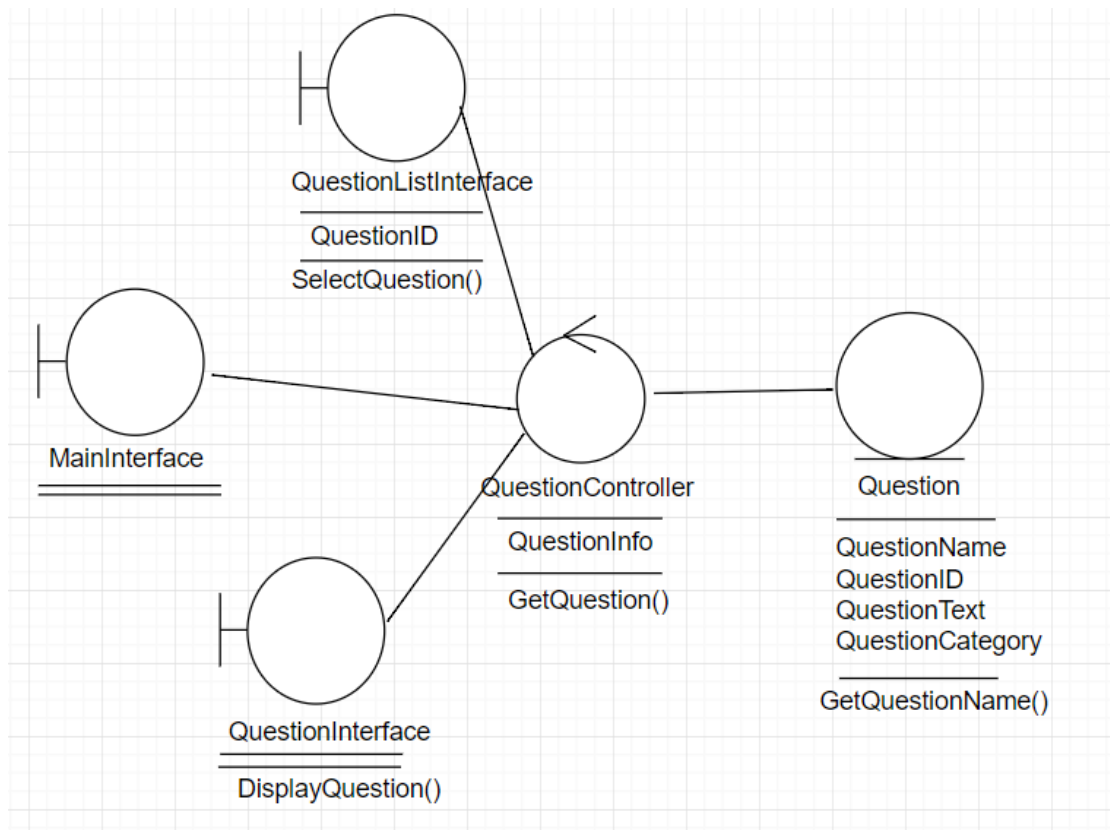


图 2-13 查看问题用例析取图

画出时序图：

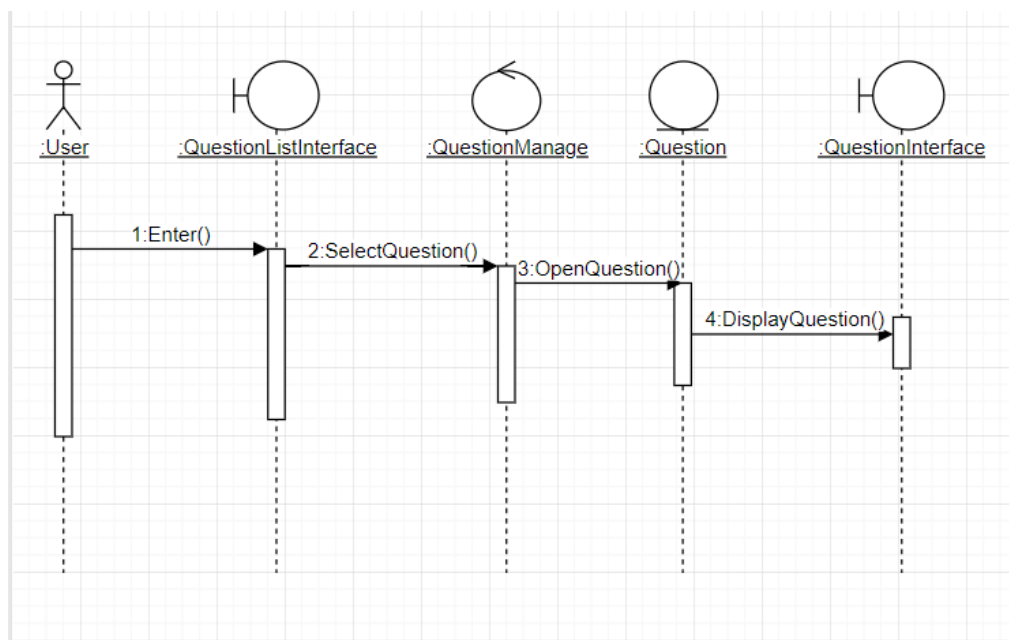


图 2-14 查看问题时序图

用户进入问题列表界面，选择想要查看的问题，并点击，进入后，会在问题页面中显示该问题的详细信息，包括问题名称，问题内容，问题的评论，以及问题的分类和发布时间等。

2.3 分析机制

根据 1.4 补充规约得到上述边界类、控制类、实体类需要满足的非功能性需求，列出系统的分析机制表，如下表所示：

表 2-1 中山大学师生教学平台分析机制表

分析类	分析机制
User	持久性、安全性
UserList	持久性、安全性
Question	持久性、安全性

2.4 合并分析类

将析取出来的边界类、控制类、实体类进行合并整理，得到的系统的合并类图如下图所示：

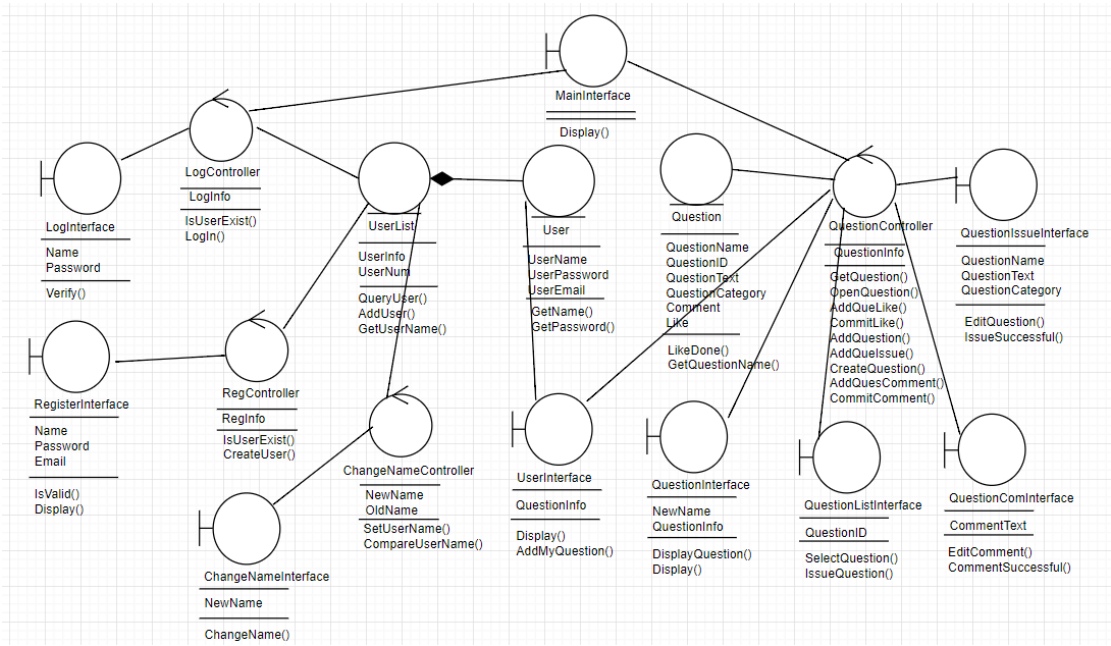


图 2-15 本系统合并类图

3 子系统及其接口设计

3.1 确定设计类

经过第三章分析可知，师生教学平台的设计类图如下所示：

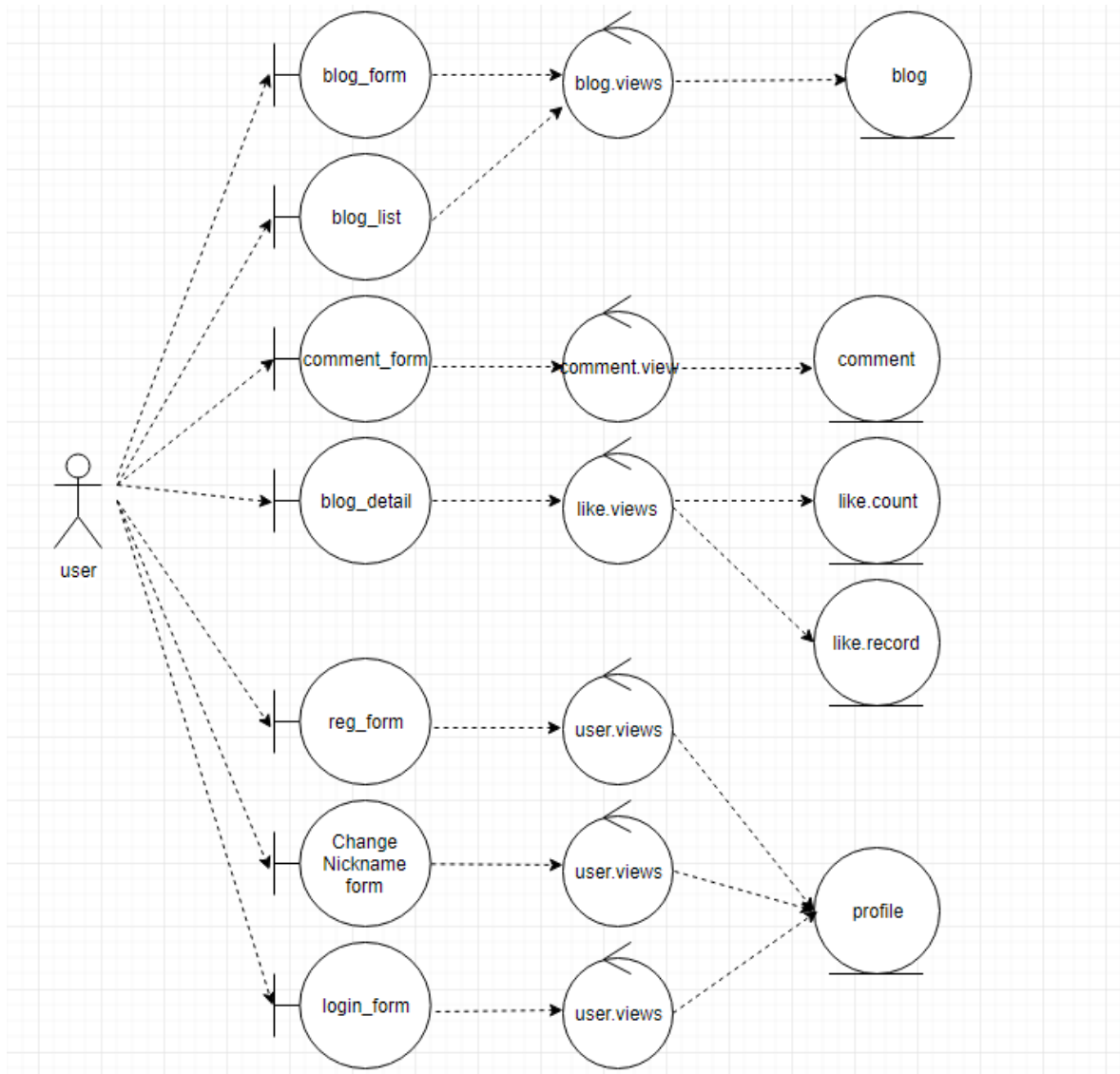


图 3-1 师生教学平台设计类图

3.2 定义子系统

本研究平台网站包括四个子系统：评论模块子系统、点赞模块子系统、发布问题子系统以及用户模块子系统及接口设计。四个子系统向主程序提供接口服务，主程序调用子系统提供的接口完成主要的流程。

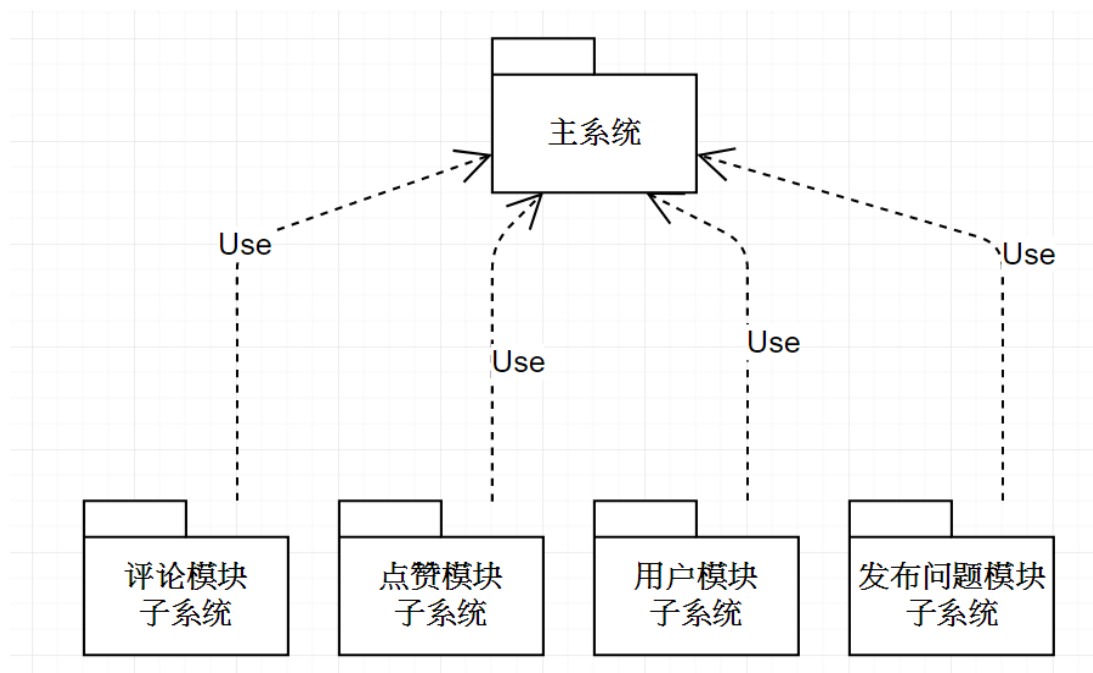


图 3-2 系统设计图

3.3 定义接口

3.3.1 评论模块子系统

评论模块子系统遵循总体架构的分层策略，分别分为表示层、业务逻辑层与数据访问层。

表示层以窗口的方式与用户进行交互，展示用户需要输入的信息，实现评论功能。

业务逻辑层负责提供完整的数据流控制服务，对每一个数据实体提供一个管理器类，管理器类负责处理数据的更改和转移，是整个系统的中枢。管理器向上层提供调用接口，并调用下层的数据访问服务来完成持久化的操作。

数据访问层封装了数据库的连接、读取、写入等操作，以提供数据库访问的服务。

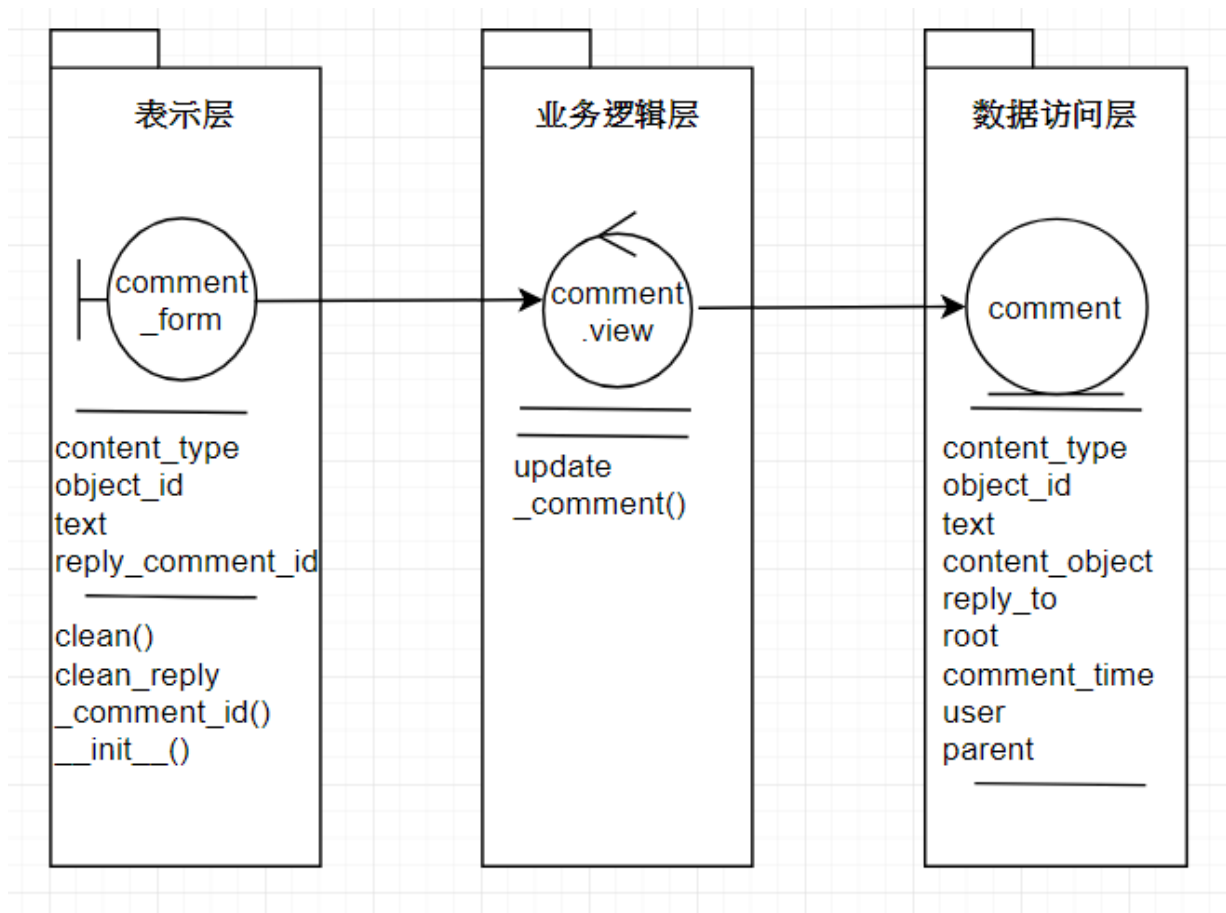


图 3-3 评论模块子系统系统架构

3.3.2 点赞模块子系统

点赞模块子系统遵循总体架构的分层策略，分别分为表示层、业务逻辑层与数据访问层。

表示层以窗口的方式与用户进行交互，展示用当前点赞数量，同时对点赞功能进行逻辑处理，增加与取消点赞的数量。

在交互过程中，表示层获取点赞数量的相关信息后，以消息的形式传到业务逻辑层的控制器，控制器根据消息的类型，转发到相应的业务逻辑层处理类进行处理。处理完毕后，业务逻辑层处理类根据需要将处理后的消息进一步转发到数据访问层的相应类进行处理。最后，通过数据访问层对信息的持久化处理。

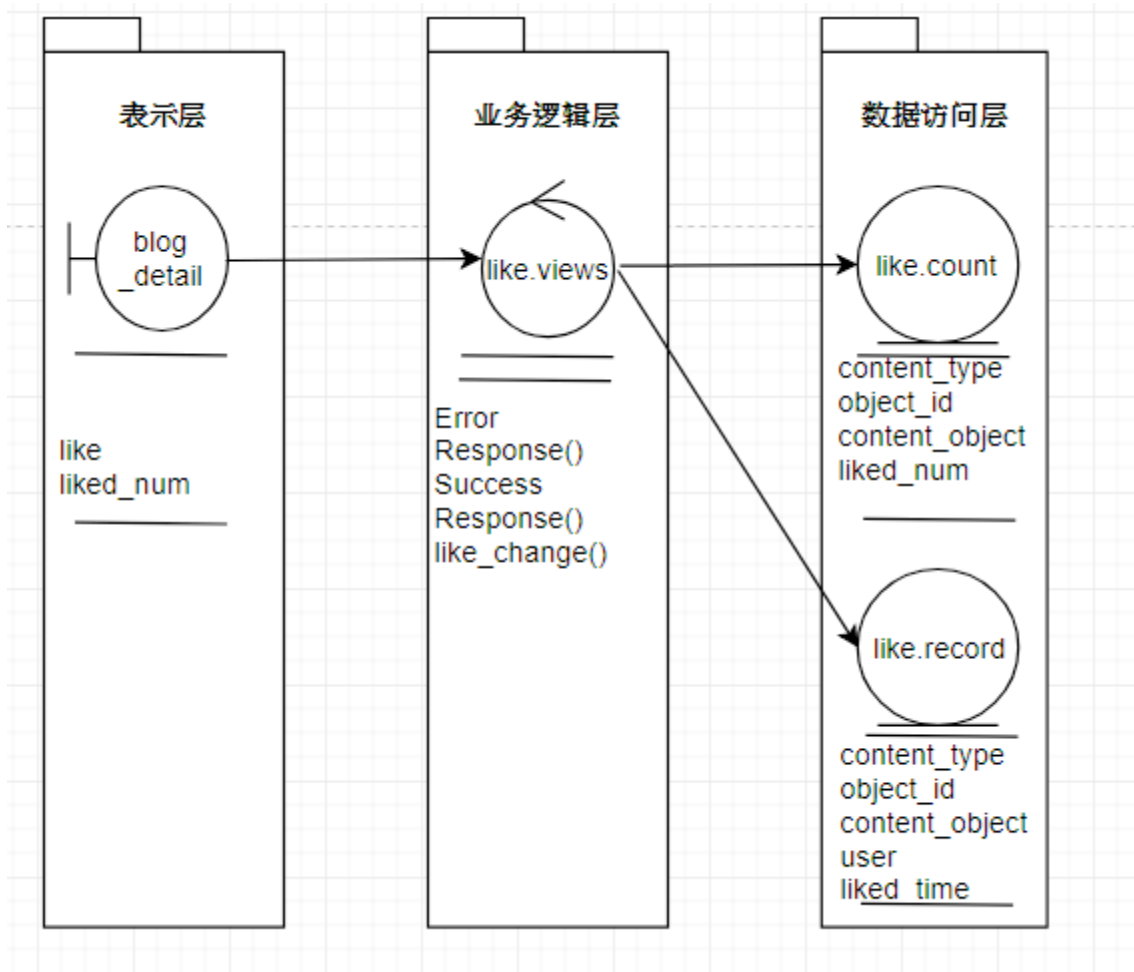


图 3-4 点赞模块子系统系统架构

3.3.3 用户模块子系统

用户模块子系统遵循总体架构的分层策略，分别分为表示层、业务逻辑层与数据访问层。

表示层以窗口的方式与用户进行交互，展示用户注册，登录时需要输入的信息，包括用户的用户名、密码、邮件等等。

业务逻辑层负责提供完整的数据流控制服务，对每一个数据实体提供一个管理器类，管理器类负责处理数据的更改和转移，是整个系统的中枢。管理器向上层提供调用接口，并调用下层的数据访问服务来完成持久化的操作。

数据访问层封装了数据库的连接、读取、写入等操作，以提供数据库访问的服务。

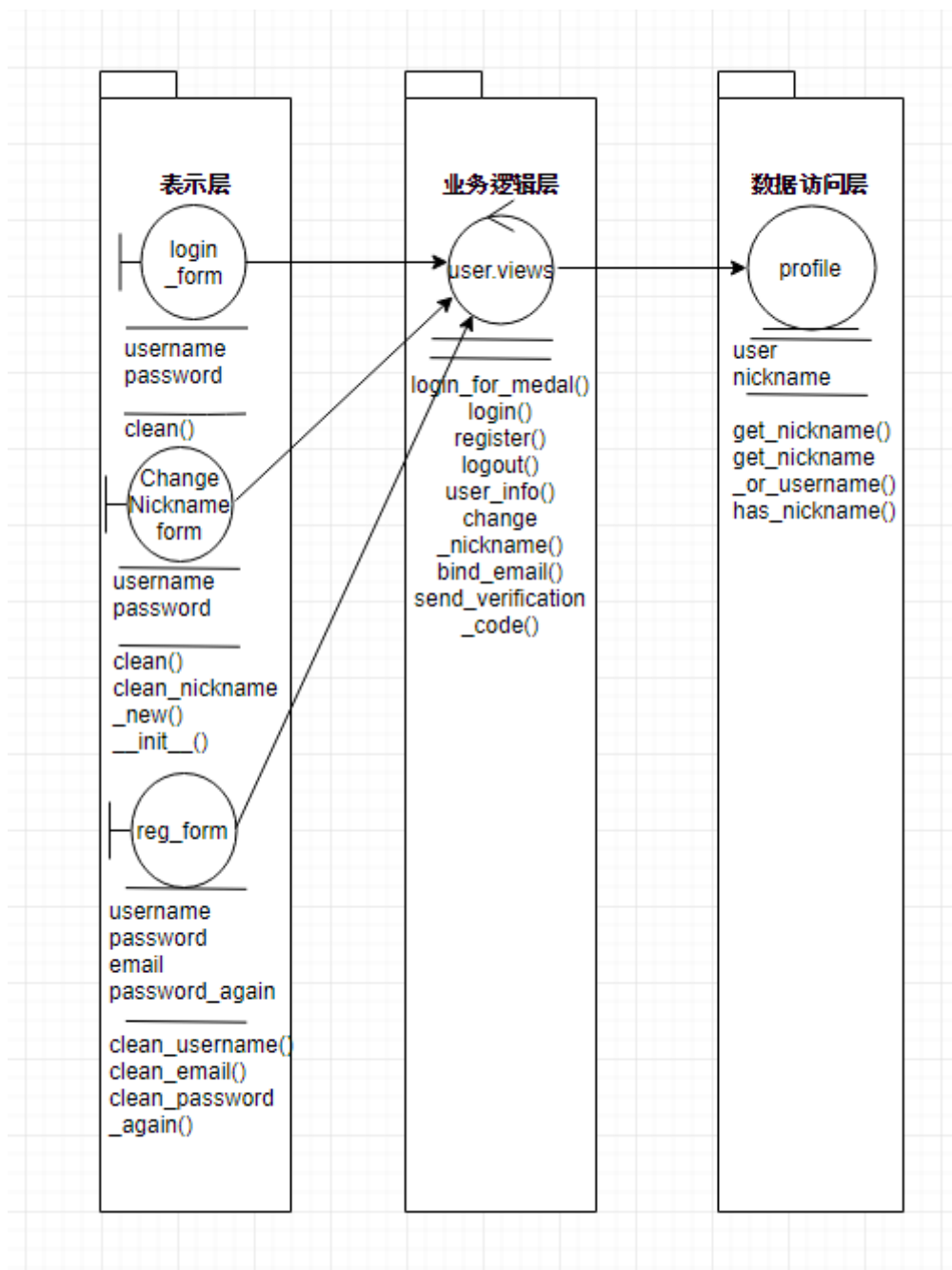


图 3-5 注册管理子系统系统架构

3.3.4 发布问题模块子系统

发布问题模块子系统遵循总体架构的分层策略，分别分为表示层、业务逻辑层与数据访问层。表示层以窗口的方式与用户进行交互，展示用户需要输入的信息，实现发布问题功能。

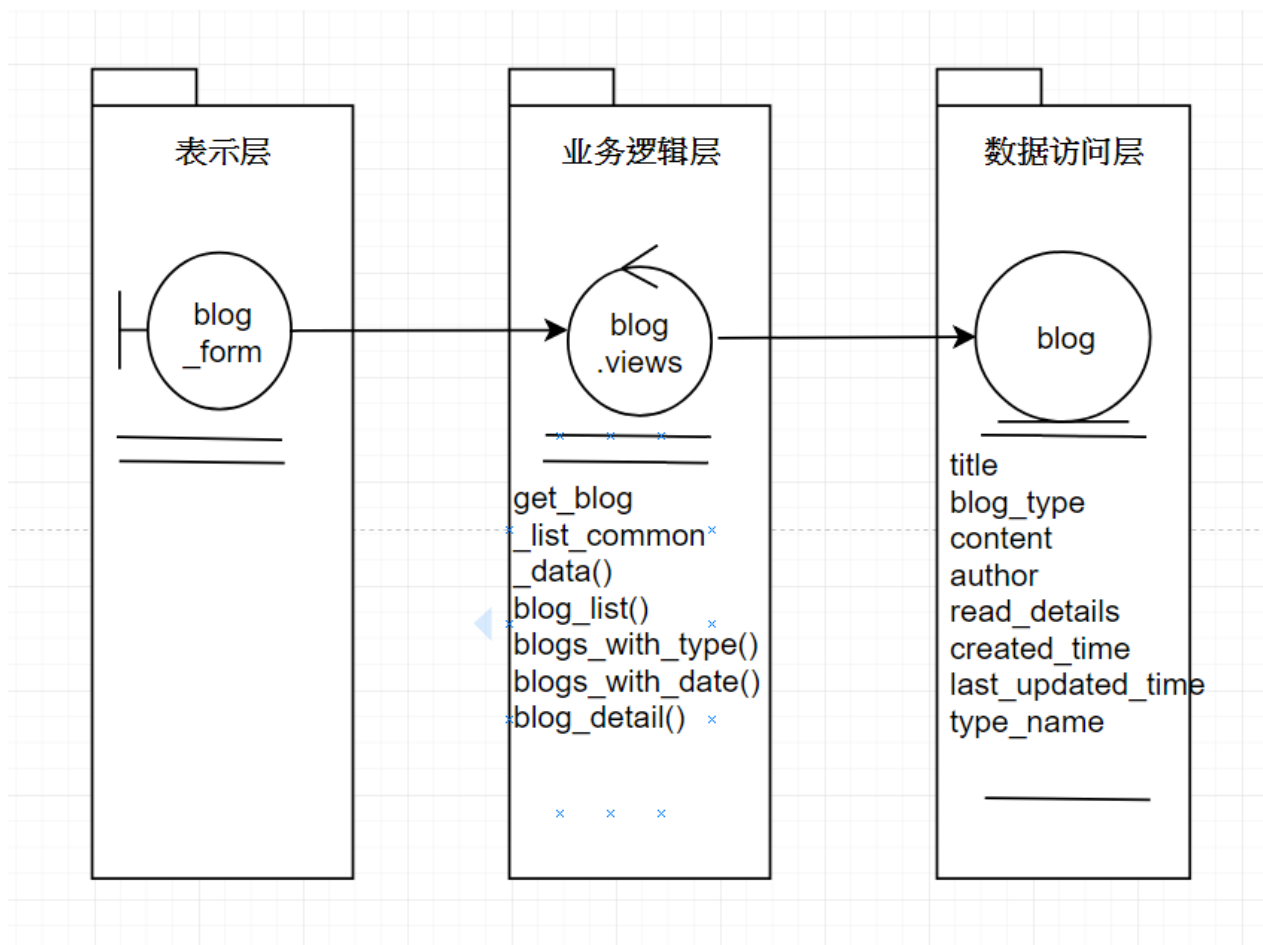


图 3-6 发布问题子系统系统架构

3.4 可重复使用子系统

是次子系统，我们除了发布问题模块子系统以外，评论模块子系统，点赞模块子系统和用户模块子系统都是可重复使用的子系统。

4 描述系统运行时架构

本系统作为一个网页系统，供给不同的用户登录使用，因此该系统需要满足多用户登录、并发请求加载网页的内容等。经分析，本系统主要包含六个模块：注册、登录、问题发布、问题评论、个人资料管理。假设该网站的注册用户数为 5w 人，同时在线最多 5k 人，这 5k 人中最多 1k 人会浏览课程评论、100 人会提出问题或发表评论，即使以最高的 1.1k 为标准计算并发量，可以发现原生的 Django 框架完全可以支持最高访问需求。