Whisper悄悄话匿名提问箱

项目开发文档



#### 课程名称：软件工程

#### 小组成员：

#### 张嘉伟 2019302141095

#### 马薪宇 2019302080309

#### 谢海洋 2018302091104

#### 项目时间：2022.4

1. 软件计划文档
2. 系统概述

新媒体背景下,人们的社会交往呈现出虚拟化加剧的现象,例如微博、微信、QQ等社交软件在人们日常生活中的运用,相比于这种熟人社交平台,匿名社交软件在人际传播过程中产生了新的特点社交网络的高速发展衍生出众多细分需求的空白，匿名社交应运而生。

以综合社交为主的社交媒体市场已经进入成熟阶段，许多较小的细分需求衍生出来，社交行业不同细分领域的需求一直存在，有已经被发掘的职业、婚恋和匿名，每个行业的发展速度都非常快.而匿名社交发展是顺应用户心智而为，当用户心里的压力在微博、微信等圈子里得不到释放时，基于通讯录的熟人匿名社交粉墨登场，作为一个树洞，满足了用户的猎奇心理;而在移动应用爆炸式发展的年代，仅仅满足用户需求是远远不够的，只有占领用户的心智，真正触到用户的痛点才能成功。

网络聊天的出现是为了方便人们沟通，但是渐渐的，这个“方便”也只是缩短了时间和距离，而并没有实现更加坦白的沟通，社交媒体依然没有真正实现“人人皆可说“的幻境，它只是给了每个人一支关掉的话筒，怎么能打开，还得靠自己。

Whisper匿名提问信箱提供了一种简便真诚的社交方式，就是为了能够打破这些束缚，实现提问双方的轻松社交。

1. 软件开发过程模型。例如是瀑布型、原型、敏捷等。

|  |  |  |  |
| --- | --- | --- | --- |
| **软件生命周期** | **是否首先定义好所有的需求** | **是否有多个开发周期** | **是否有中间软件** |
| 瀑布型 | 是 | 没有 | 没有 |

1. 项目估算
   1. 规模估算

模块复杂度估算：各种类别模块的个数。例如：简单模块（例如业务逻辑简单，数据库表的访问不超过3个表，较少的数据展示等）、中等模块（业务逻辑复杂度一般，表的访问个数不超过4个，数据展示量中等）、复杂模块（业务逻辑较复杂，表的访问个数超过4个，较多的数据展示）。预估的代码行数（KLOC）

|  |  |  |
| --- | --- | --- |
| **模块类型** | **模块数量** | **预估代码行数** |
| 简单模块 | 3 | 2000 |
| 中等模块 | 2 | 3000 |
| 复杂模块 | 0 | 0 |

* 1. 进度估算

|  |  |  |  |
| --- | --- | --- | --- |
| **阶段** | **开始日期** | **结束日期** | **工作日** |
| 软件计划 | 2022/03/25 | 2022/03/25 | 1 |
| 需求调研 | 2022/03/26 | 2022/03/26 | 1 |
| 需求分析 | 2022/03/27 | 2022/03/28 | 2 |
| 概要设计 | 2022/03/29 | 2022/03/30 | 2 |
| 详细设计 | 2022/03/31 | 2022/04/01 | 2 |
| 编码 | 2022/04/06 | 2022/04/10 | 5 |
| 测试 | 2022/04/11 | 2022/04/15 | 5 |
| 部署及验收 | 2022/04/16 | 2022/04/17 | 2 |
| **总计** |  |  | 20 |



* 1. 工作量估算

|  |  |  |
| --- | --- | --- |
| **阶段** | **阶段工作量占总工作量百分比** | **工作量（人.天）** |
| 软件计划 | 5 | 3 |
| 需求调研 | 5 | 3 |
| 需求分析 | 10 | 6 |
| 概要设计 | 10 | 6 |
| 详细设计 | 10 | 6 |
| 编码 | 25 | 15 |
| 测试 | 25 | 15 |
| 部署及验收 | 10 | 6 |
| **总计** | 100 | 60 |

* 1. 成本估算

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 功能 | 最佳值 | 可能值 | 悲观值 | 期望值 | 行平均成本 | 平均生产率 | 成本 | 工作量 |
| 调研、分析和设计 | 500 | 900 | 1200 | 883 | 48 | 100 | 42400 | 9 |
| 用户登录 | 400 | 800 | 1000 | 767 | 50 | 300 | 38333 | 3 |
| 用户提问 | 600 | 700 | 900 | 717 | 60 | 300 | 43000 | 2 |
| 用户回答 | 500 | 600 | 1000 | 650 | 60 | 300 | 39000 | 2 |
| 管理员登录 | 400 | 800 | 1000 | 767 | 50 | 300 | 38333 | 3 |
| 管理员管理用户 | 600 | 900 | 1100 | 883 | 60 | 300 | 53000 | 3 |
| 管理员管理提问和回答 | 800 | 1200 | 1500 | 1183 | 60 | 300 | 71000 | 4 |
| 部署 | 100 | 200 | 400 | 217 | 30 | 400 | 6500 | 1 |
| 测试 | 700 | 800 | 900 | 800 | 20 | 400 | 16000 | 2 |
| **总计** |  |  |  | 6867 |  |  | 347567 | 28 |

1. 人力资源安排

按照项目开发阶段或者模块进行人力资源的分配。例如

|  |  |
| --- | --- |
| **阶段** | **成员** |
| 软件计划 | 谢海洋 |
| 需求调研 | 马薪宇 |
| 需求分析 | 张嘉伟 |
| 概要设计 | 谢海洋 马薪宇 张嘉伟 |
| 详细设计 | 谢海洋 马薪宇 张嘉伟 |
| 编码 | 谢海洋 马薪宇 张嘉伟 |
| 测试 | 谢海洋 马薪宇 张嘉伟 |
| 部署及验收 | 谢海洋 马薪宇 张嘉伟 |

1. 开发环境：

硬件环境

|  |  |  |  |
| --- | --- | --- | --- |
| **设备名称** | **用途** | **配置信息** | **备注** |
| 个人笔记本 | 撰写文档、编码、部署、调试 | windows 11 | 每个人都有自己的笔记本电脑 |

软件环境

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **软件名称** | **软件类别** | **版本号** | **补丁程序** | **备注** |
| IntelliJ IDEA | IDLE | 2021.3.2 | 无 | 无 |
| Tomcat | 服务器软件 | 9 | 无 | 无 |

1. 测试环境：

硬件环境

|  |  |  |  |
| --- | --- | --- | --- |
| **设备名称** | **用途** | **配置信息** | **备注** |
| 个人笔记本 | 测试 | windows 11 | intel/arm 64位处理器 |

软件环境

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **软件名称** | **软件类别** | **版本号** | **补丁程序** | **备注** |
| IntelliJ IDEA | IDLE | 2021.3.2 | 无 | 无 |
| Chrome | 浏览器 | 100.0.4896.75 | 无 | 无 |

1. 风险估计和控制

|  |  |
| --- | --- |
| **风险描述** | **降低风险策略** |
| 出现存在技术空白或者未知领域 | 尽量采用成熟已有的技术，必要时请求老师的帮助 |
| 开发人员对信息理解出现偏差 | 固定一个时间段集中进行开发工作，便于即使沟通和交流。同时每个开发工作日最开始的时候开小组会议，总结问题以及指定本日计划。 |
| 代码风格不一致导致合作开发效率下降 | 统一小组成员的编码规范比如采用阿里命名规范。在关键模块上，采用结对编程的方式进行开发。 |

1. 测试计划

列出本项目内部进行单元和集成测试活动的安排计划，如：测试活动的类型、每类测试活动的主要内容（测试项、指标）、每类测试活动的开始与延续的大概时间、测试工作接口人员。

测试阶段分解：

| **测试** | **测试类型** | **负责人及成员** | **时间** |
| --- | --- | --- | --- |
| 1 | 单元 | 马薪宇 | 2022/04/11-2022/04/12 |
| 2 | 集成 | 张嘉伟 | 2022/04/12-2022/04/13 |
| 3 | 系统 | 谢海洋 | 2022/04/13-2022/04/14 |
| 4 | 验收 | 谢海洋 马薪宇 张嘉伟 | 2022/04/15-2022/04/16 |

详细测试活动写入《测试计划》

1. 评审计划

说明为保证开发质量,在项目开发过程中拟进行的各项评审活动、每次评审活动的评审对象、参加每次评审活动的评审人姓名、每次评审活动的大致时间。

|  |  |  |  |
| --- | --- | --- | --- |
| **评审活动名称** | **评审对象** | **评审人员** | **评审活动时间** |
| 代码互查 | 代码规范与质量 | 谢海洋 马薪宇 张嘉伟 | 每个开发工作日 |
| 文档检查 | 文档 | 谢海洋 马薪宇 张嘉伟 | 每个开发工作日 |

1. 质量保证计划

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 项目QA人员组成：马薪宇 张嘉伟 谢海洋 | | | | |
| QA活动安排 | | | | |
| QA任务 | 计划日期 | 活动形式 | QA人员 | 项目组参与成员 |
| 代码评审 | 2022/04/11 | 走查 | 谢海洋 | 马薪宇 张嘉伟 |
| 文档评审 | 2022/04/16 | 评审会 | 张嘉伟 | 马薪宇 谢海洋 |

二、需求分析

1. 系统概述

随着社会生活压力增加，匿名社交成为了新兴的社交形式，传统的社交软件也相继推出匿名聊天功能。而类似于tape的社交app，则提供了针对某个人的匿名沟通工具，深受追捧。

然而这些软件所提供的匿名功能大多是附属在其主干功能上，市面上缺少纯粹的匿名社交软件。所以，我们希望开发出一款足以令用户完全放松的、释放压力的、纯粹的匿名社交网站。

1. 需求概述

**目标**

本项目计划开发一个匿名社交网站。社交形式暂定为tape的提问箱模式，即：每位用户可以发起提问箱，其他用户可以以匿名的身份在提问箱中提问，该提问被用户收集后回答。

项目计划开发一个B/S架构的前后端分离的项目。暂定后端使用SSM框架，前端使用Vue框架，数据库使用mySql。

**运行环境**

软件环境：Windows10/11

硬件环境：intel/arm 64位处理器

开发环境：IntelliJ IDEA，Tomcat，MySQL，Chrome

**用户的特点**

普通用户：普通用户是系统的主要使用者，会用到系统的核心功能。用户的注册与登录，问题的发布与删除，回答的发布与删除，附加社交功能

系统管理员：系统管理员需要对数据库中的全部信息进行管理，主要行使信息的增删功能。用户信息的管理，问题的管理，回答的管理。

**关键点**

关键功能：业务逻辑：只有被回答的匿名提问才会被展示。社交功能：消息的实时提示（JS轮询）、私聊等。附加关键功能：敏感词过滤。

关键技术：前端使用Vue的现代前后端分离项目。radis缓存技术。docker部署等。

1. 运行环境

软件环境

|  |  |  |
| --- | --- | --- |
|  | **名称** | **版本** |
| 操作系统 | Windows | Win10 |
| 数据库平台 | Mysql | 8.0.26 |
| 应用平台 | 浏览器 | 无 |

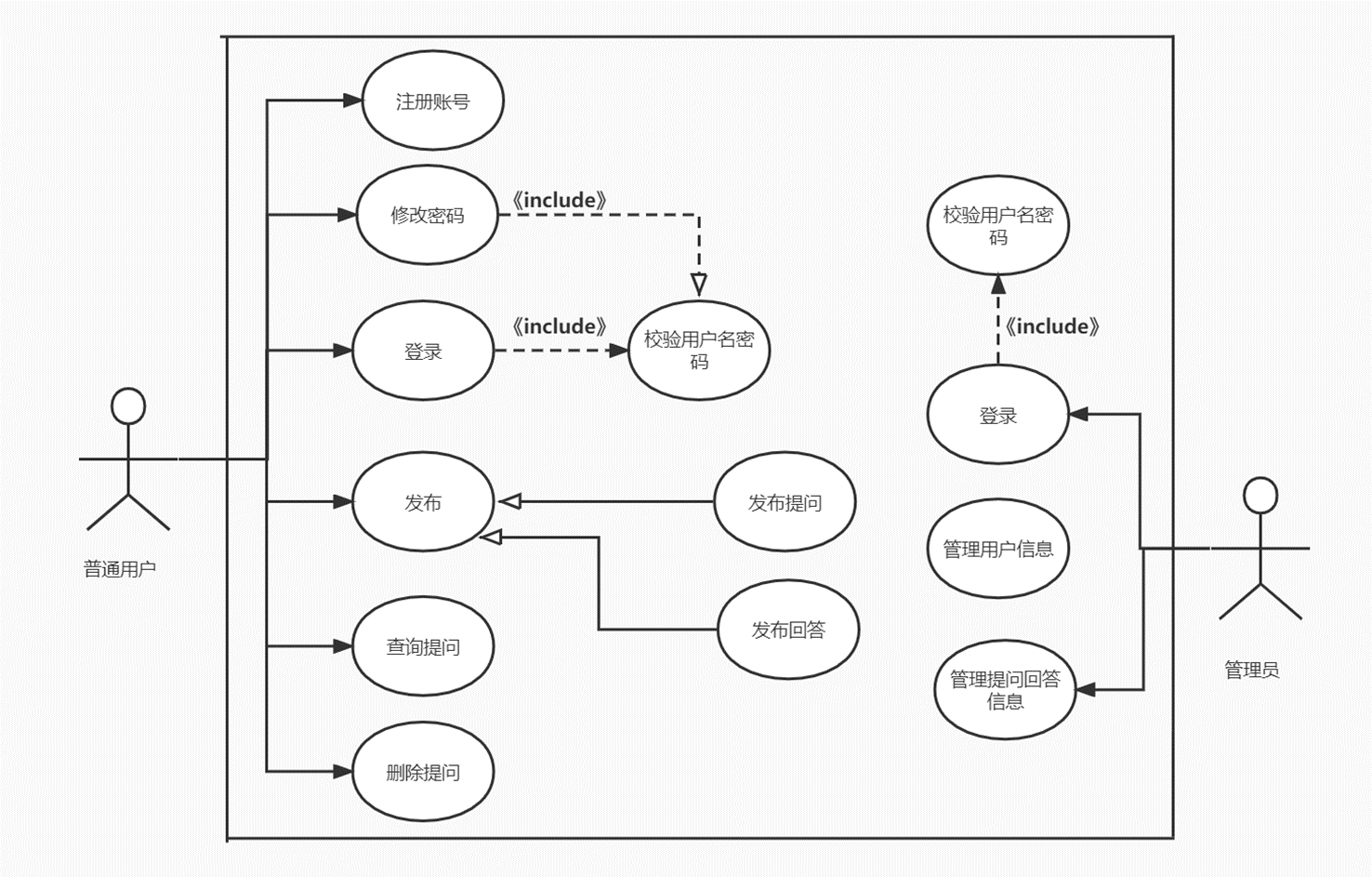
硬件环境

|  |  |  |
| --- | --- | --- |
| **服务器** | **最低配置** | **推荐配置** |
| 应用服务器 | intel/arm 64位处理器  RAM 4GB | intel i7-10875H或等效更高性能的处理器  RAM 16GB及以上  存储器可用容量大于等于20GB |
| 数据库服务器 | 同上 | 同上 |

1. 需求规定
2. 软件系统总体功能结构

对软件系统总体功能结构进行描述，包括系统功能结构图等。

1. 功能需求



用例图

用例描述

用户模块：

注册账户：新用户注册账户，并将账户信息添加至服务器数据库

修改密码：用户可以对账户密码进行修改

登录：核验账户和密码后，跳转至用户个人的系统首页

发布提问：用户可在某提问箱下以**匿名状态**发布提问

发布回答：被提问的用户可以对问题进行回答，回答后的问题会被展示

删除提问：用户可以对已发布的问题进行删除，同时清除数据库中的数据

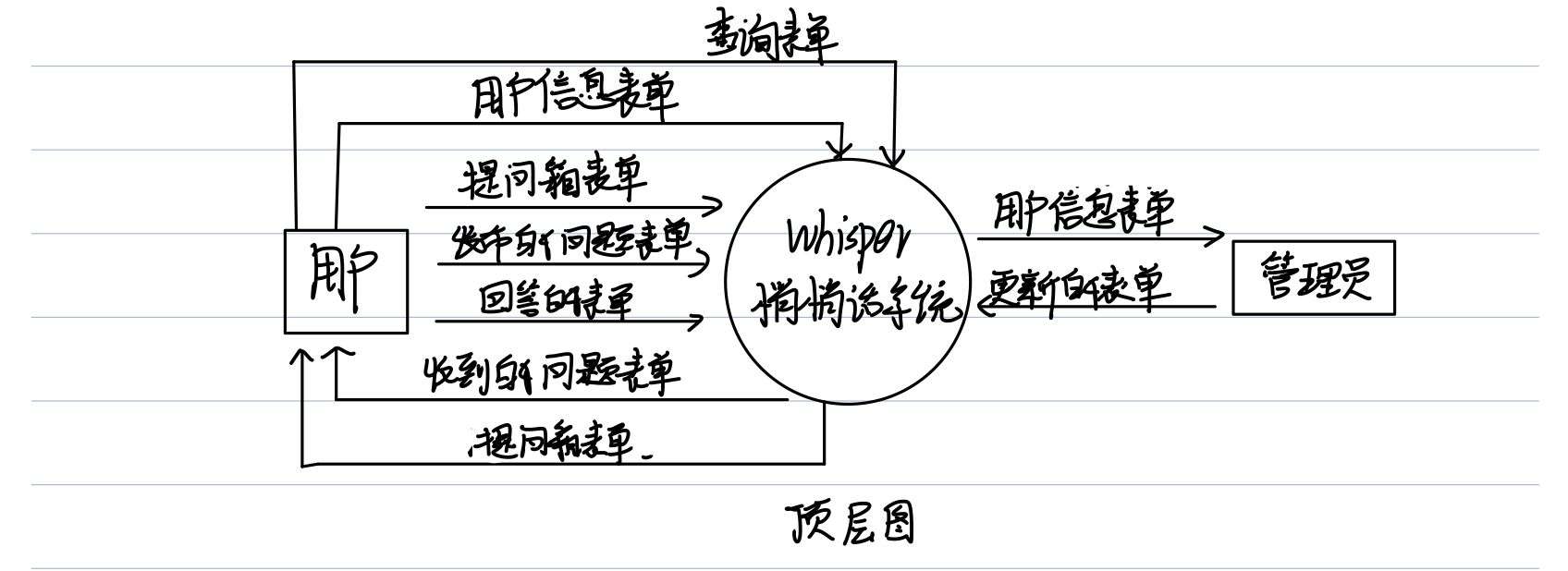
查询提问：用字符串匹配的方式对数据库中的信息进行模糊查询

管理员模块：

登录：检验账户和密码后，跳转至管理员管理页面

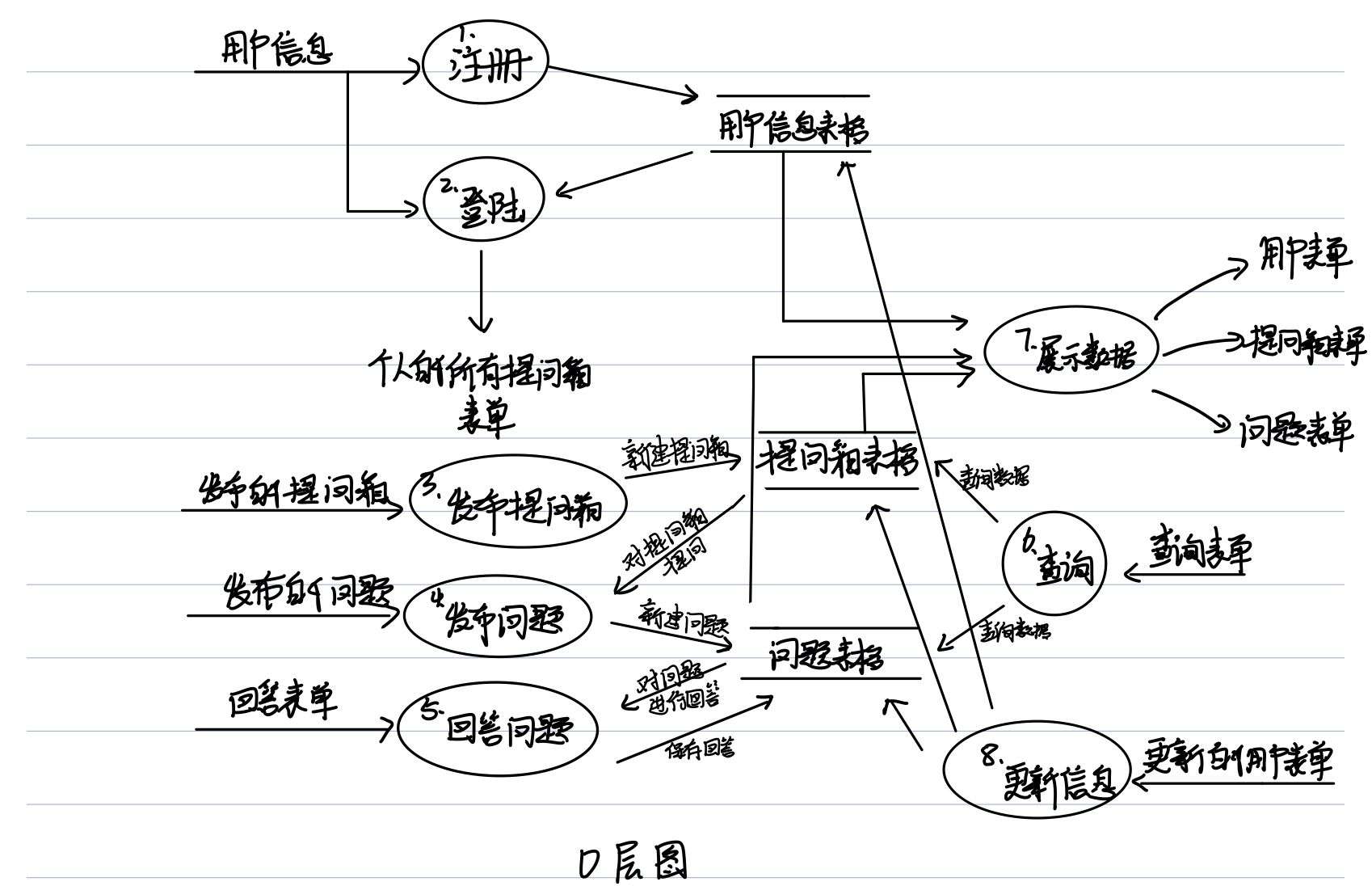
管理用户信息：对用户的账号进行增删改（例如注销不符合规定的账户、手动修改账户密码等）

管理提问回答信息：管理员可以对发布的提问或回答进行手动删除（例如删除引战、不文明的信息）



系统顶层数据流图

根据系统顶层数据顶图，在此系统中，主要的数据来源及去向包括：用户和管理员。其中用户通过注册账号、创建新建提问箱、发布问题和回答问题等操作向系统提交相应的输入数据，管理员则通过更改用户账号信息和修改系统中发布的内容向系统提交输入数据。同时根据不同的需求，系统也会向用户和管理员输出所需的数据。



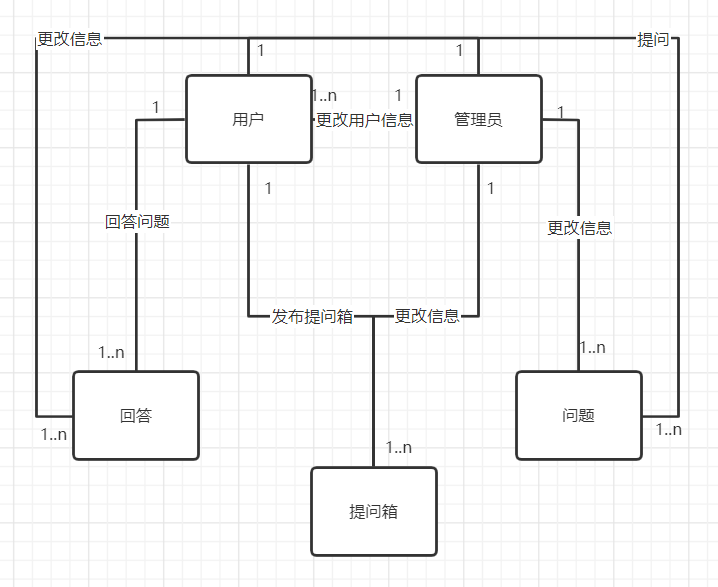
用户模块子数据流图

如上用户模块子数据流图所示，不同的加工过程对应不同的数据输入输出。

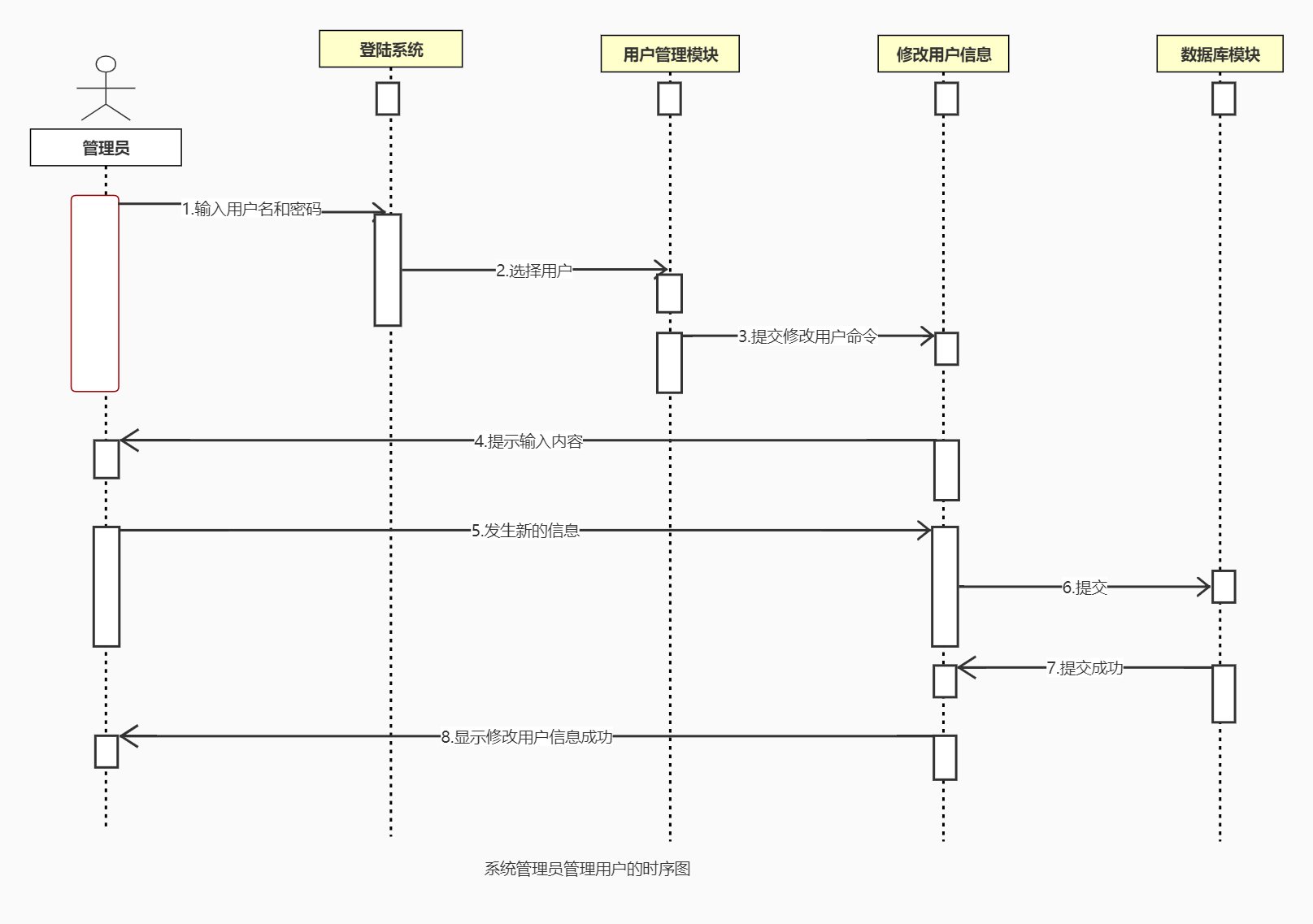
当用户第一次使用软件时，我们需要用户注册一个自己的账号才能使用软件，因此用户需要提交自己的账号信息到数据库中的用户表文件中；而用户登录则需要对输入的账号信息与数据库中的数据信息进行对照，确认无误后进入软件的主界面。

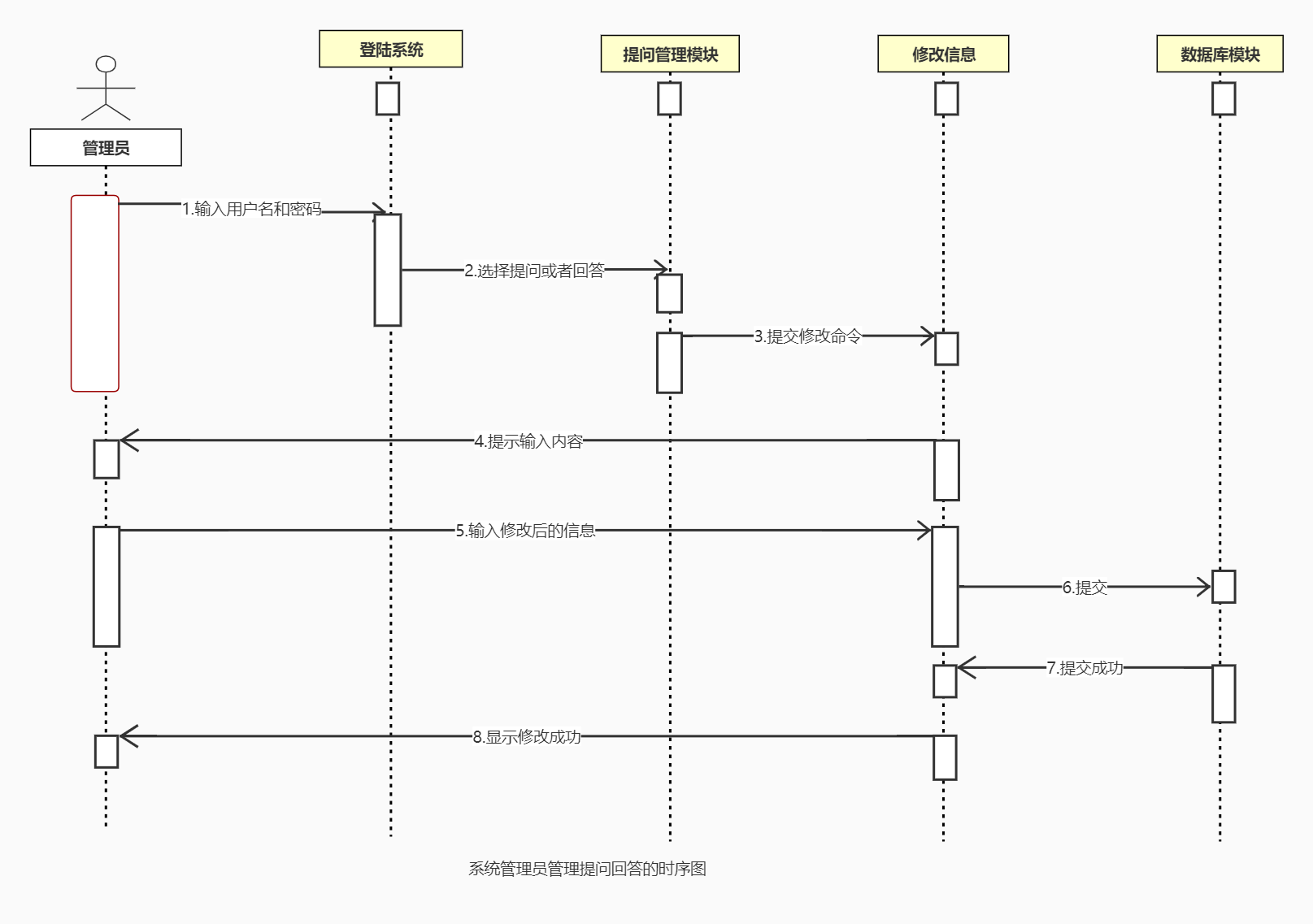
当用户进行发布提问箱、发布问题、回答问题、模糊查询等加工过程时，需要向系统的数据库表格文件提交对应的提问箱、问题、回答、查询信息等输入数据，而系统进行完加工后输出用户所需的文件。

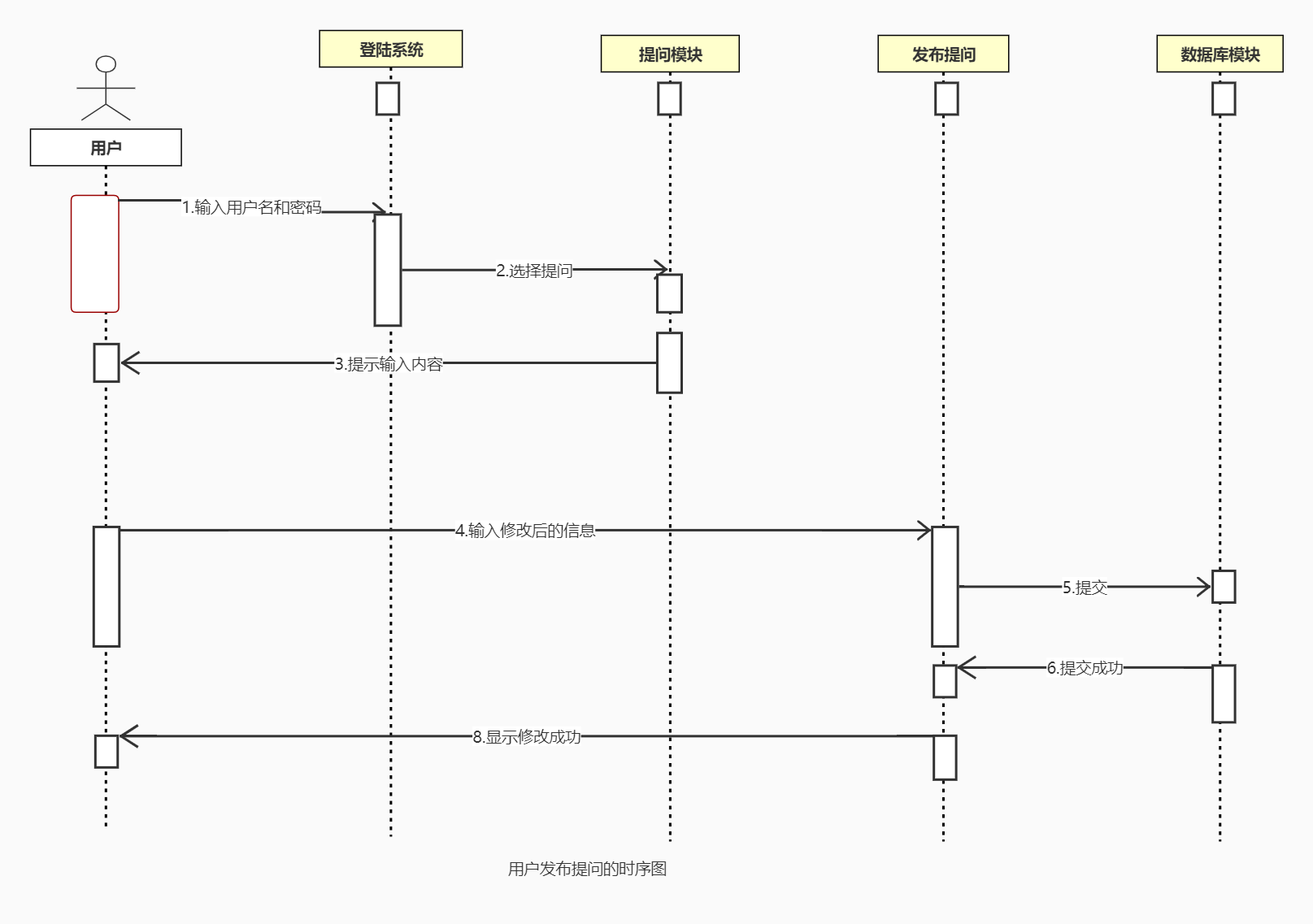
当管理员希望对数据库的文件信息进行修改或者查询时，会提交更新的数据和想要查找的数据到系统中，数据库的数据库文件会对应进行更新或者输出管理员希望的数据。

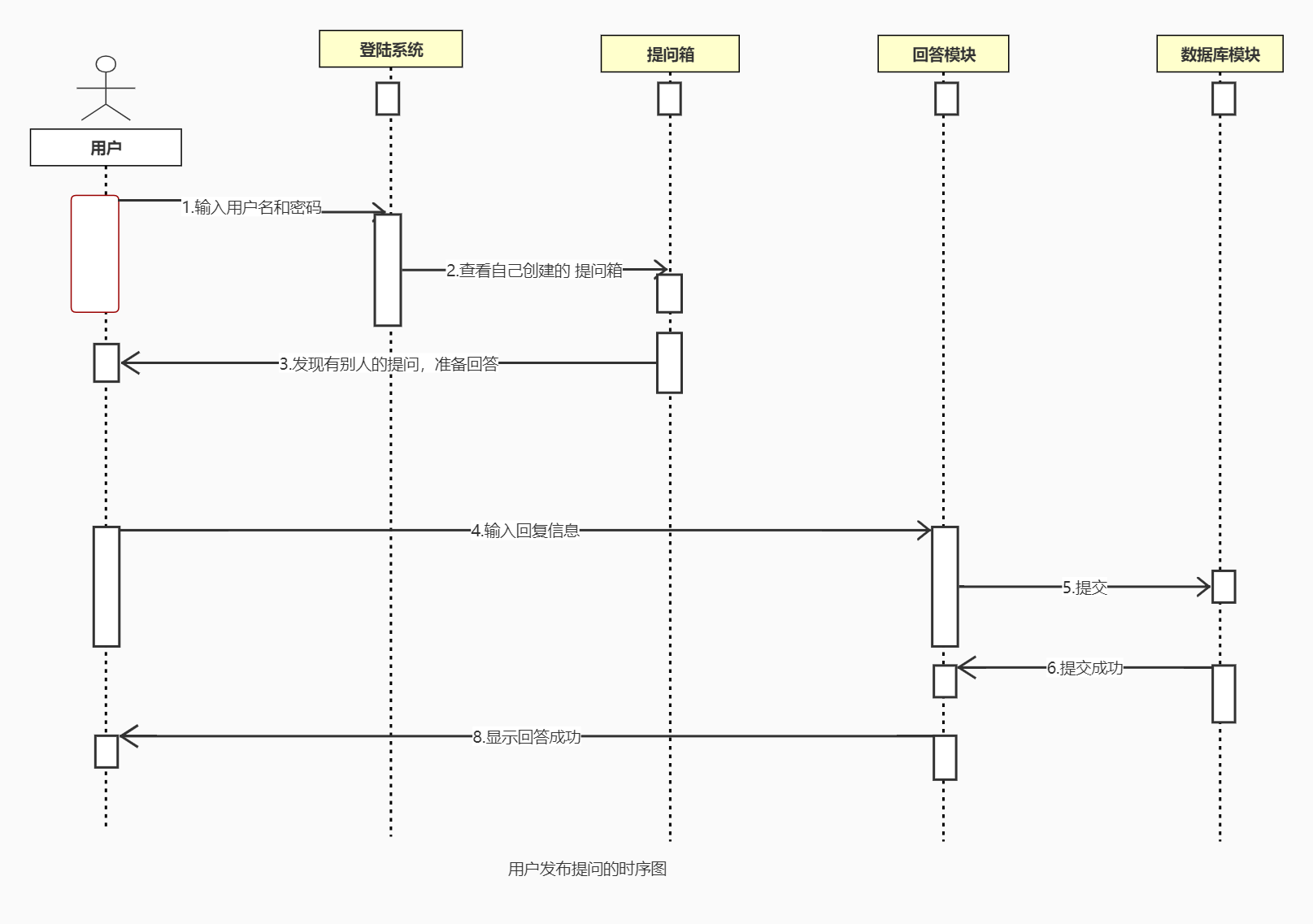


领域模型图









1. 非功能性需求

软件需要满足多人在线使用，需要具有良好的并发性；

软件需要保障用户的信息安全，在同一时间，每一个账号只允许在一台设备上登录；

软件需要能够进行异地登陆提醒，保障账号安全；

管理员应有最高权限，可以对软件中的信息进行修改；

数据库应具有良好的可迁移性，方便日后软件的更新迭代。

三、概要设计

1. 系统概述

新媒体背景下,人们的社会交往呈现出虚拟化加剧的现象,例如微博、微信、QQ等社交软件在人们日常生活中的运用,相比于这种熟人社交平台,匿名社交软件在人际传播过程中产生了新的特点社交网络的高速发展衍生出众多细分需求的空白，匿名社交应运而生。

以综合社交为主的社交媒体市场已经进入成熟阶段，许多较小的细分需求衍生出来，社交行业不同细分领域的需求一直存在，有已经被发掘的职业、婚恋和匿名，每个行业的发展速度都非常快.而匿名社交发展是顺应用户心智而为，当用户心里的压力在微博、微信等圈子里得不到释放时，基于通讯录的熟人匿名社交粉墨登场，作为一个树洞，满足了用户的猎奇心理;而在移动应用爆炸式发展的年代，仅仅满足用户需求是远远不够的，只有占领用户的心智，真正触到用户的痛点才能成功。

网络聊天的出现是为了方便人们沟通，但是渐渐的，这个“方便”也只是缩短了时间和距离，而并没有实现更加坦白的沟通，社交媒体依然没有真正实现“人人皆可说“的幻境，它只是给了每个人一支关掉的话筒，怎么能打开，还得靠自己。

Whisper匿名提问信箱提供了一种简便真诚的社交方式，就是为了能够打破这些束缚，实现提问双方的轻松社交。

1. 运行环境

软件环境

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **软件名称** | **软件类别** | **版本号** | **补丁程序** | **备注** |
| Tomcat | 服务器软件 | 9 | 无 | 无 |
| Chrome | 浏览器 | 100.0.4896.75 | 无 | 无 |

硬件环境

|  |  |  |  |
| --- | --- | --- | --- |
| **设备名称** | **用途** | **配置信息** | **备注** |
| 个人笔记本 | 撰写文档、编码、部署、调试 | windows 11 | 每个人都有自己的笔记本电脑 |

开发环境

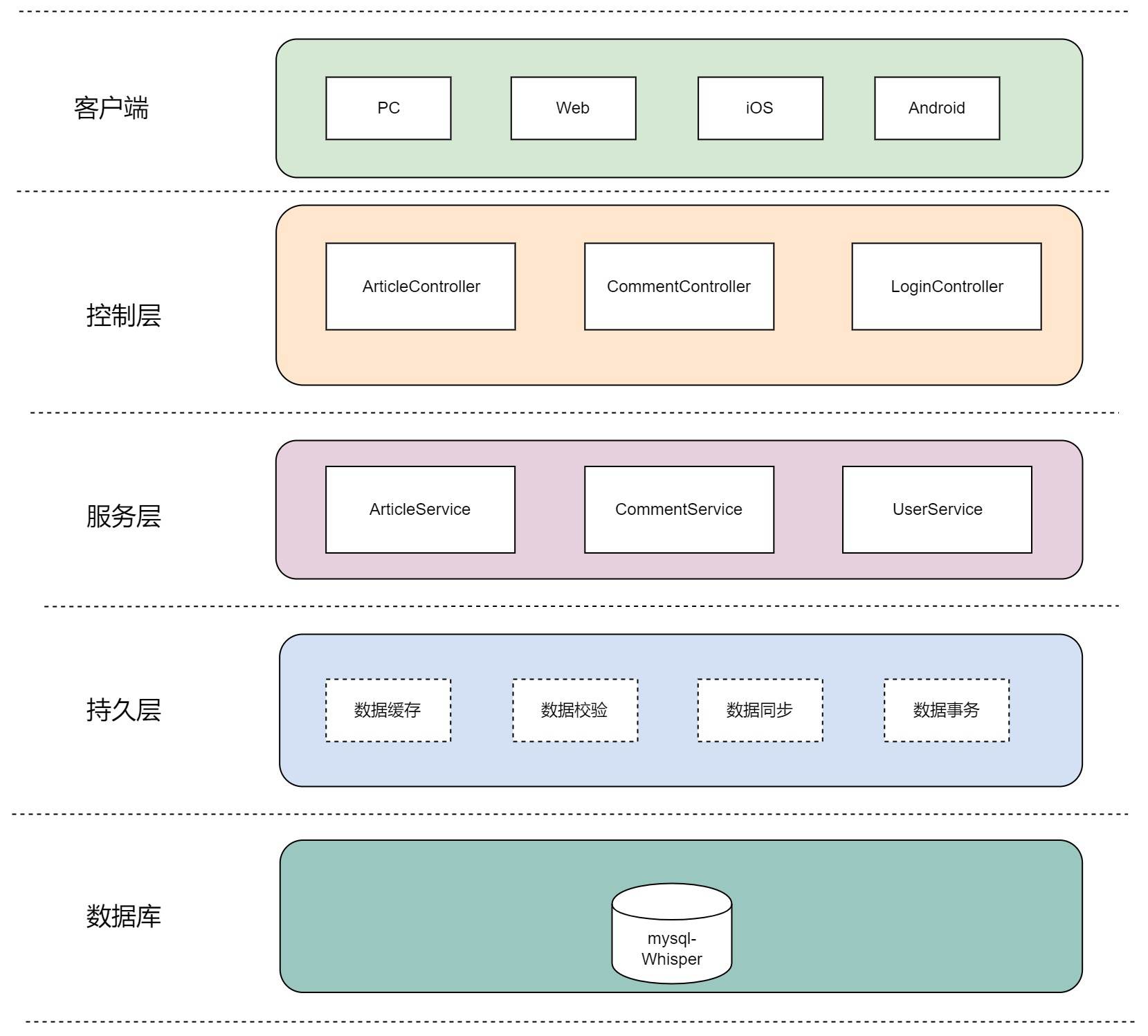
硬件环境

|  |  |  |  |
| --- | --- | --- | --- |
| **设备名称** | **用途** | **配置信息** | **备注** |
| 个人笔记本 | 测试 | windows 11 | intel/arm 64位处理器 |

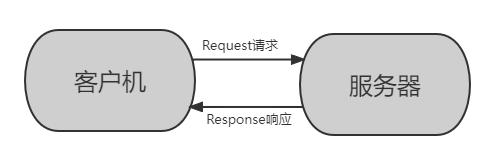
软件环境

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **软件名称** | **软件类别** | **版本号** | **补丁程序** | **备注** |
| IntelliJ IDEA | IDLE | 2021.3.2 | 无 | 无 |
| Chrome | 浏览器 | 100.0.4896.75 | 无 | 无 |
| IntelliJ IDEA | IDLE | 2021.3.2 | 无 | 无 |

1. 软件架构及说明（软件架构图）



1. 硬件架构及说明（网络拓扑图）



1. 关键技术与算法

**Spring**

pring框架是一个开放源代码的J2EE应用程序框架，由Rod Johnson发起，是针对bean的生命周期进行管理的轻量级容器（lightweight container）。 Spring解决了开发者在J2EE开发中遇到的许多常见的问题，提供了功能强大IOC、AOP及Web MVC等功能。Spring可以单独应用于构筑应用程序，也可以和Struts、Webwork、Tapestry等众多Web框架组合使用，并且可以与 Swing等桌面应用程序AP组合。因此， Spring不仅仅能应用于J2EE应用程序之中，也可以应用于桌面应用程序以及小应用程序之中。Spring框架主要由七部分组成，分别是 Spring Core、 Spring AOP、 Spring ORM、 Spring DAO、Spring Context、 Spring Web和 Spring Web MVC

**SpringMVC**

Sring MVC 是 Spring 提供的一个基于 MVC 设计模式的轻量级 Web 开发框架，本质上相当于 Servlet。Spring MVC 角色划分清晰，分工明细。由于 Spring MVC 本身就是 Spring 框架的一部分，可以说和 Spring 框架是无缝集成。性能方面具有先天的优越性，是当今业界最主流的 Web 开发框架，最热门的开发技能。一个好的框架要减轻开发者处理复杂问题的负担，内部有良好的扩展，并且有一个支持它的强大用户群体，恰恰 Spring MVC 都做到了。

SpringMVC在项目中拦截用户请求，它的核心Servlet即DispatcherServlet承担中介或是前台这样的职责，将用户请求通过HandlerMapping去匹配Controller，Controller就是具体对应请求所执行的操作。SpringMVC相当于SSH框架中struts。

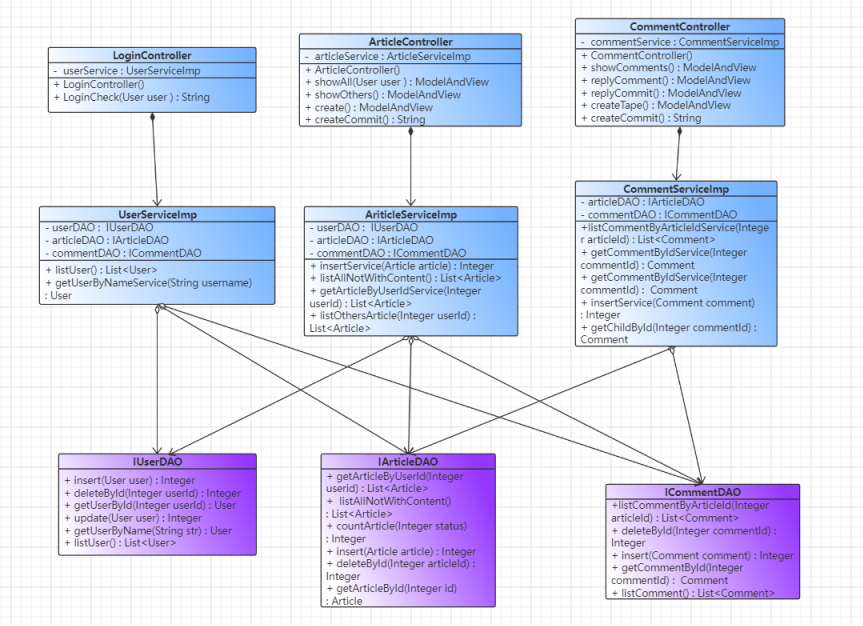
**mybatis**

mybatis是对jdbc的封装，它让数据库底层操作变的透明。mybatis的操作都是围绕一个sqlSessionFactory实例展开的。mybatis通过配置文件关联到各实体类的Mapper文件，Mapper文件中配置了每个类对数据库所需进行的sql语句映射。在每次与数据库交互时，通过sqlSessionFactory拿到一个sqlSession，再执行sql命令。

页面发送请求给控制器，控制器调用业务层处理逻辑，逻辑层向持久层发送请求，持久层与数据库交互，后将结果返回给业务层，业务层将处理逻辑发送给控制器，控制器再调用视图展现数据。

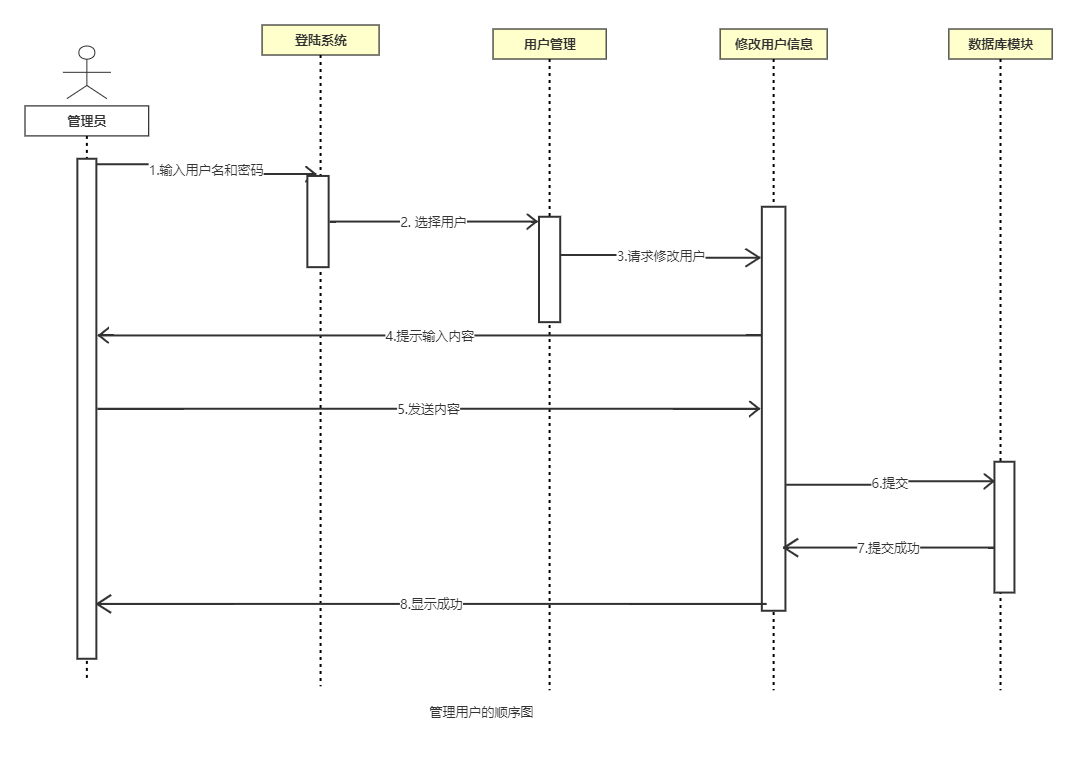
1. 系统功能设计

（1）系统功能类图



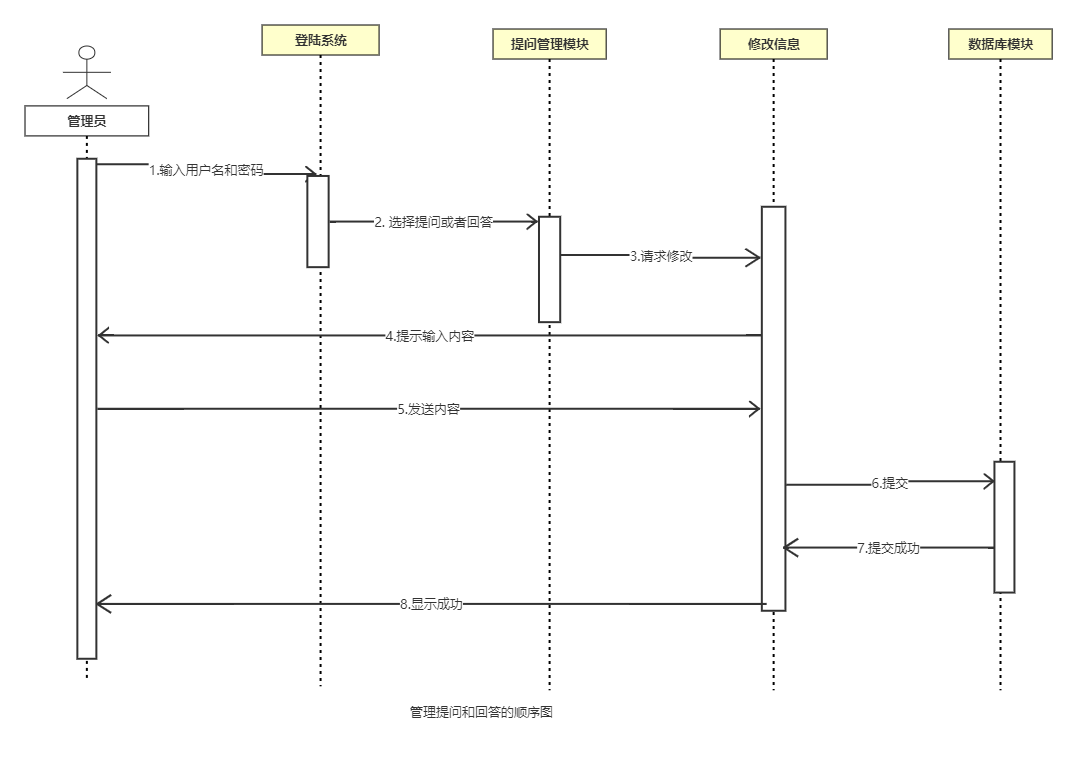
（2）系统功能详细设计

* 1. 功能一 系统管理员管理用户功能



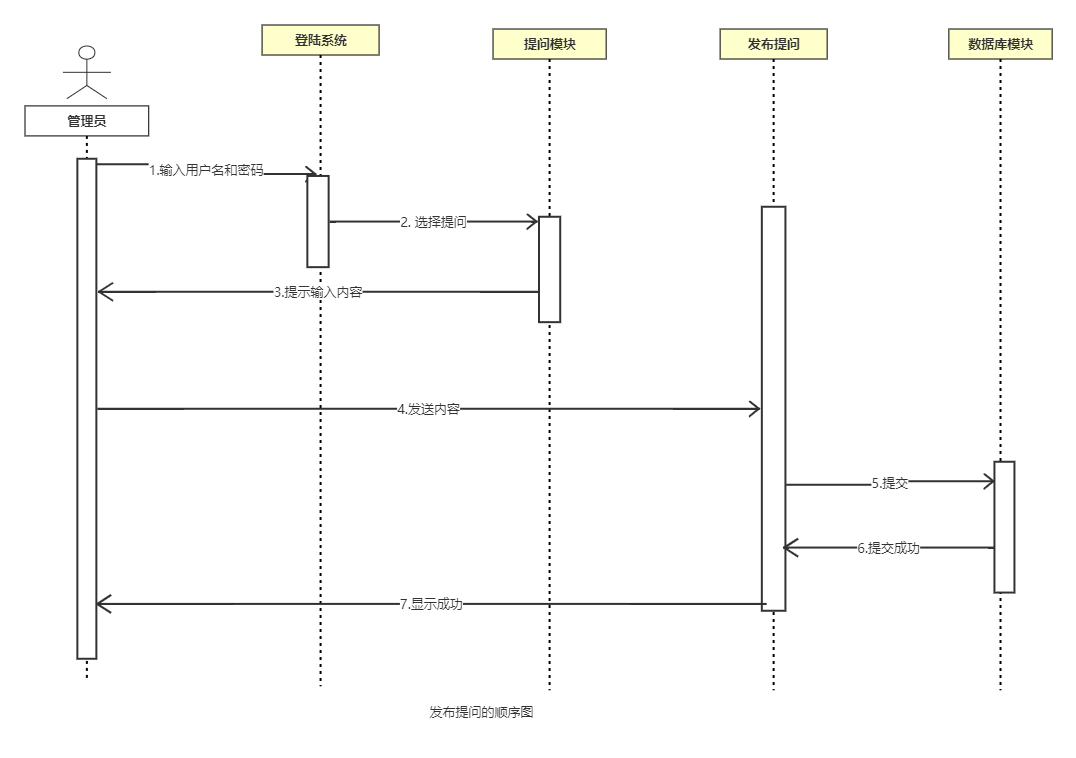
如图所示，管理员登录系统后，输入用户名和密码并发送至服务器，服务器检验通过后获得管理用户的权限。之后前端提交管理员选择的用户，后端将该用户信息返回给前端。之后管理员将要修改的内容发送至后端，后端将信息修改并保存至数据库后返回给前端显示。

b. 功能二 管理员管理回答功能



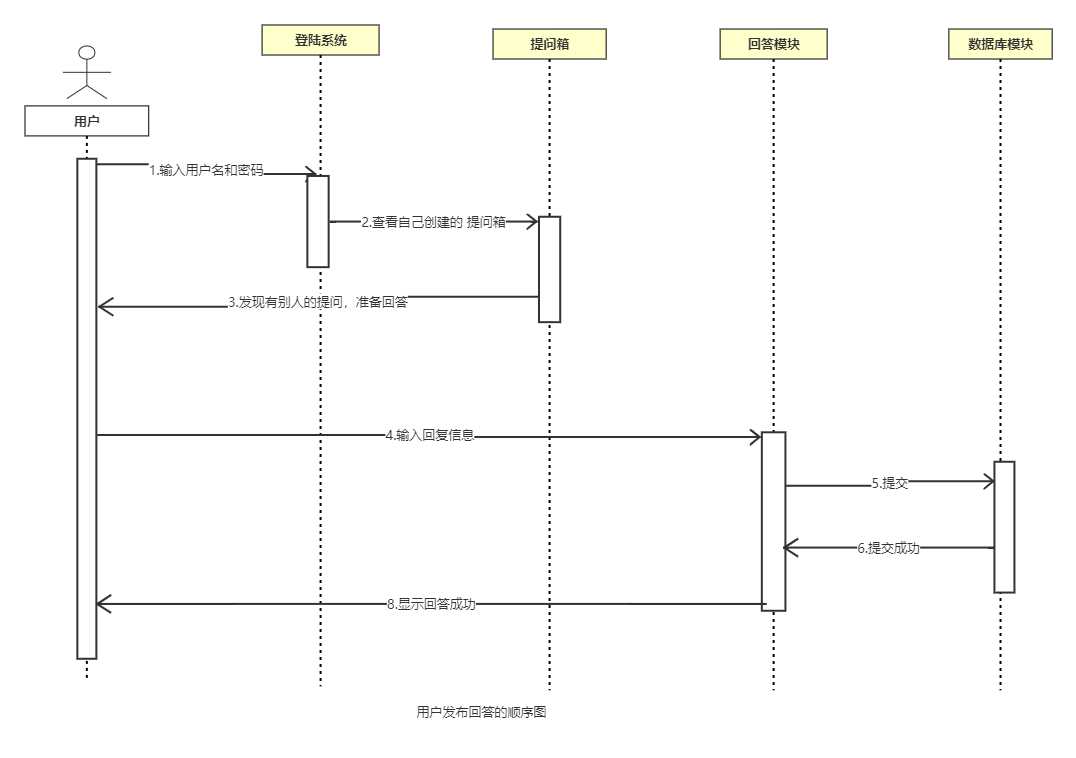
如图所示，管理员输入用户名和密码后发送至后端检查，检查通过后获得管理权限。前端选择回答后发送至后端，后端将被选择的回答详细内容返回给前端显示。之后前端可以对其进行修改，修改结果发送给后端。后端将其保存至数据库后发送回前端展示。

c. 功能三 用户发布提问功能



如图所示，用户输入用户名密码后发送至后端验证，验证通过后登录成功。登陆成功后可以选择发布提问，用户输入相应内容后发送至后端。后端根据用户输入信息建立新的提问并保存至数据库。后端将新发布的内容发送至前端显示。

d. 功能四 用户发布回答功能



如图所示，用户输入用户名密码后发送至后端验证，验证通过后登录成功。用户选择自己创建的提问箱后，将选择的信息发送给后端，后端将其详细内容读出发送给前端显示。用户在前端可对其输入回答，将回答发送至后端，后端将其更新后保存至数据库，并将更新后的内容发送到前端显示。

（3）非功能性设计

使用spring security来增强安全性

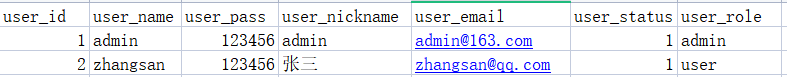
使用http缓存技术减少浏览器和服务器之间来回传输的数据量，从而节省带宽，提高新能和应对高并发的能力；

1. 数据库设计
2. 采用的数据库

项目使用数据库系统为Mysql，版本为8.0.26

1. 数据库表的设计

User（用户）表格：



其中

1) user\_id为主键，int型，自动增一

2）user\_name为用户名，varchar型，不可为空，唯一

3）user\_pass为用户密码，varchar型，不可为空

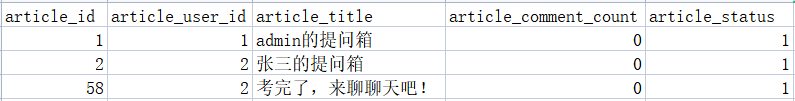
4) user\_nickname为用户昵称,varchar型

5）user\_email为用户邮箱，varchar型，唯一

6）user\_status为用户状态，表示是否被禁言的状态，默认为1，即否禁言状态

7）user\_role为用户权限，有管理员和普通用户两种权限

Article表格：



其中

1）article\_id为提问箱id，不可为空，int型，自动增一

2) article\_user\_id为提问箱所有者的id，int型

3) article\_title为提问箱的标题varchar型，默认为空

4）article\_comment\_count为提问箱的评论数量，int型，默认为空

5）article\_status表示提问箱是否可见，int型，默认为1，即可见

Comment表格：



其中

1）comment\_id为评论id，不可为空，int型，自动增一

2）comment\_pid为当前评论上一个评论（回复上一个评论）的id，默认为0

3）comment\_article\_id为评论所在的文章id，int型，不为空

4）comment\_content为评论的内容，varchar型，默认为空

5）comment\_role为评论的权限，表示是否是管理员，int型

6）comment\_user\_id为评论作者的用户id，int型

四、测试计划

1. 里程碑及进度人员安排

给出进行各项测试的日期和工作内容。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **里程碑任务** | **工作内容** | **人员安排** | **开始日期** | **结束日期** |
| 制定测试计划 |  |  |  |  |
| 设计测试 |  |  |  |  |
| 实施测试 |  |  |  |  |
| 对测试进行评估 |  |  |  |  |

1. 测试环境描述（包括软硬件环境）
2. 功能测试用例

例如

|  |  |  |  |
| --- | --- | --- | --- |
| 功能A描述 |  | | |
| 用例目的 |  | | |
| 前提条件 |  | | |
| 输入/动作 | | 期望的输出/相应 | 实际情况 |
| 示例：典型值… | |  |  |
| 示例：边界值… | |  |  |
| 示例：异常值… | |  |  |
| 功能B描述 |  | | |
| 用例目的 |  | | |
| 前提条件 |  | | |
| 输入/动作 | | 期望的输出/相应 | 实际情况 |
| …… | | | |

1. 健壮性测试用例

例如容错能力/恢复能力测试用例

|  |  |  |
| --- | --- | --- |
| 异常输入/动作 | 容错能力/恢复能力 | 造成的危害、损失 |
| 示例：错误的数据类型… |  |  |
| 示例：定义域外的值… |  |  |
| 示例：错误的操作顺序… |  |  |
| 示例：异常中断通信… |  |  |
| 示例：异常关闭某个功能… |  |  |
| 示例：负荷超出了极限… |  |  |

1. 性能测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 性能A描述 |  | | |
| 用例目的 |  | | |
| 前提条件 |  | | |
| 输入数据 | | 期望的性能（平均值） | 实际性能（平均值） |
|  | |  |  |
|  | |  |  |
|  | |  |  |
| 性能B描述 |  | | |
| 用例目的 |  | | |
| 前提条件 |  | | |
| 输入数据 | | 期望的性能（平均值） | 实际性能（平均值） |
| …… | | | |
|  | | | |

1. 用户界面测试用例

示例

|  |  |
| --- | --- |
| 检查项 | 测试人员的类别及其评价 |
| 窗口切换、移动、改变大小时正常吗？ |  |
| 各种界面元素的文字正确吗？（如标题、提示等） |  |
| 各种界面元素的状态正确吗？（如有效、无效、选中等状态） |  |
| 各种界面元素支持键盘操作吗？ |  |
| 各种界面元素支持鼠标操作吗？ |  |
| 对话框中的缺省焦点正确吗？ |  |
| 数据项能正确回显吗？ |  |
| 对于常用的功能，用户能否不必阅读手册就能使用？ |  |
| 执行有风险的操作时，有“确认”、“放弃”等提示吗？ |  |
| 操作顺序合理吗？ |  |
| 有联机帮助吗？ |  |
| 各种界面元素的布局合理吗？美观吗？ |  |
| 各种界面元素的颜色协调吗？ |  |
| 各种界面元素的形状美观吗？ |  |
| 字体美观吗？ |  |
| 图标直观吗？ |  |
|  |  |
| … |  |

1. 压力测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 极限名称A | *例如“最大并发用户数量”* | | |
| 前提条件 |  | | |
| 输入/动作 | | 输出/响应 | 是否能正常运行 |
| *例如10个用户并发操作* | |  |  |
| *例如20个用户并发操作* | |  |  |
| *…* | |  |  |
| 极限名称B |  | | |
| 前提条件 |  | | |
| 输入/动作 | | 输出/响应 | 是否能正常运行 |
|  | |  |  |
| *…* | |  |  |