目录

引言

在软件生存期中，软件设计处于需求分析阶段及软件构造（编码） 阶段的中间位置。需求分析的主要任务是明确“做什么”，在完成了需求分析之后，就进入了软件设计阶段，它的任务是回答“怎么做”。结构化设计方法是在模块化、自顶向下逐步细化及结构化程序设计技术基础之上发展起来的。结构化设计方法可以分为两类：一类是根据系统的数据流进行设计，称为面向数据流的设计，或称过程驱动的设计；另一类是根据系统的数据结构进行设计，称为面向数据结构的设计，或称数据驱动的设计。

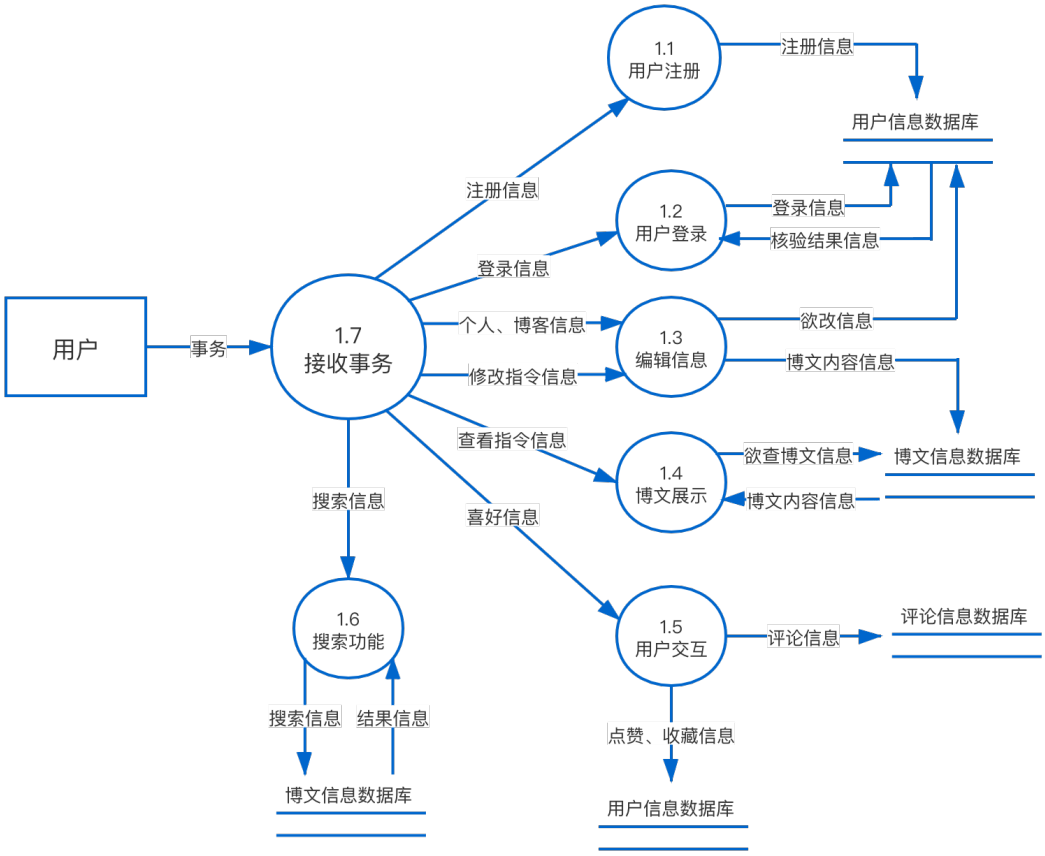
面向数据流的设计工作与软件需求分析阶段的结构化分析方法相衔接，可以很方便地将用数据流图表示的信息转换成程序结构的设计描述，这一方法还能和编码阶段的“结构化程序设计方法”相适应， 成为常用的结构化设计方法。三周前，我团队顺利完成了结构化需求分析文档的编写。此次我们会优化、润色已完成的结构化需求分析文档，并在此基础上努力完成此次文档的编写工作。

二. 体系结构设计

2.1 改善后的数据流图

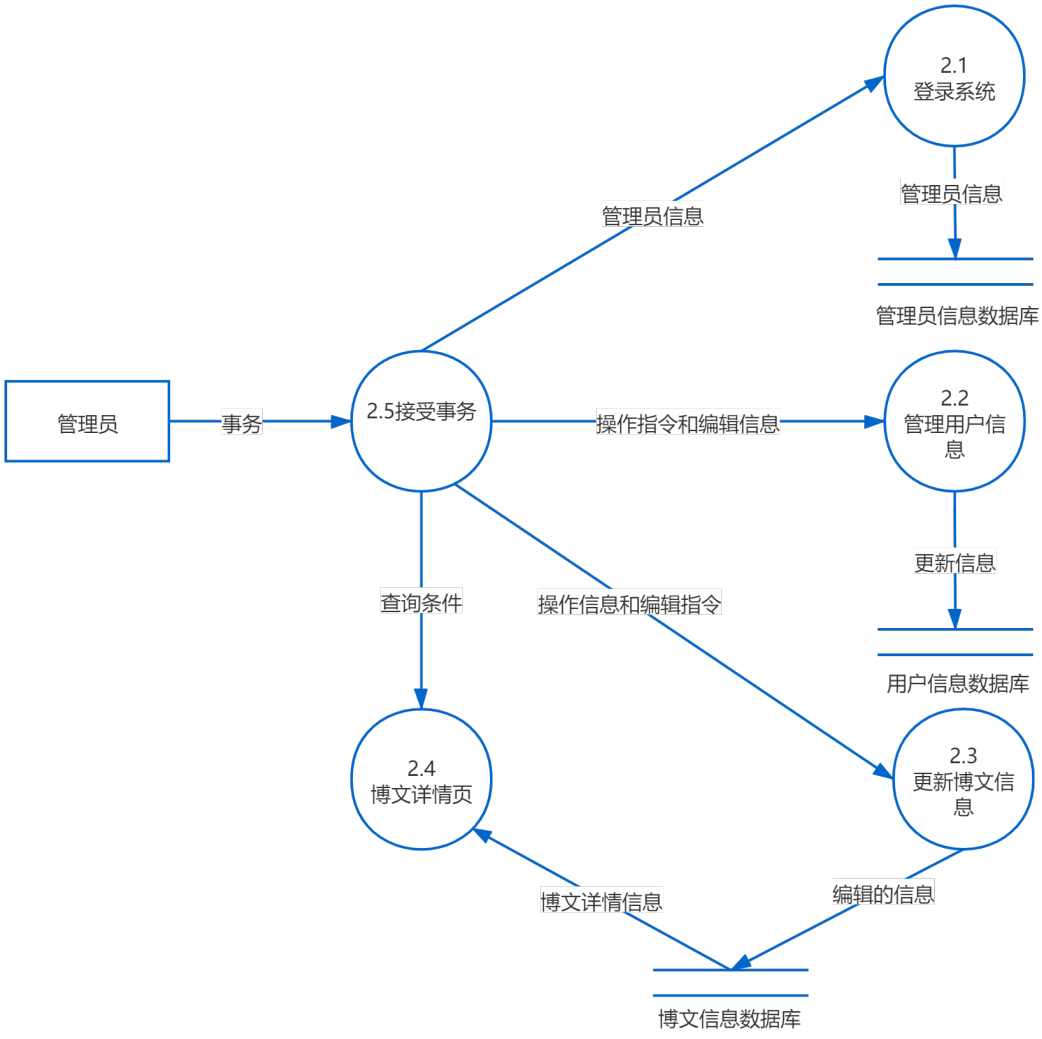
在需求分析阶段，共产生了顶层数据流图、一层数据流图、二层数据流图和三层数据流图，共四层。因此，在将其转化为系统结构图时，我们先重新绘制了二层数据流图，对部分功能进行了增添和删改，让结构更加完整，使其适应了事务型的数据流。

① 用户子系统



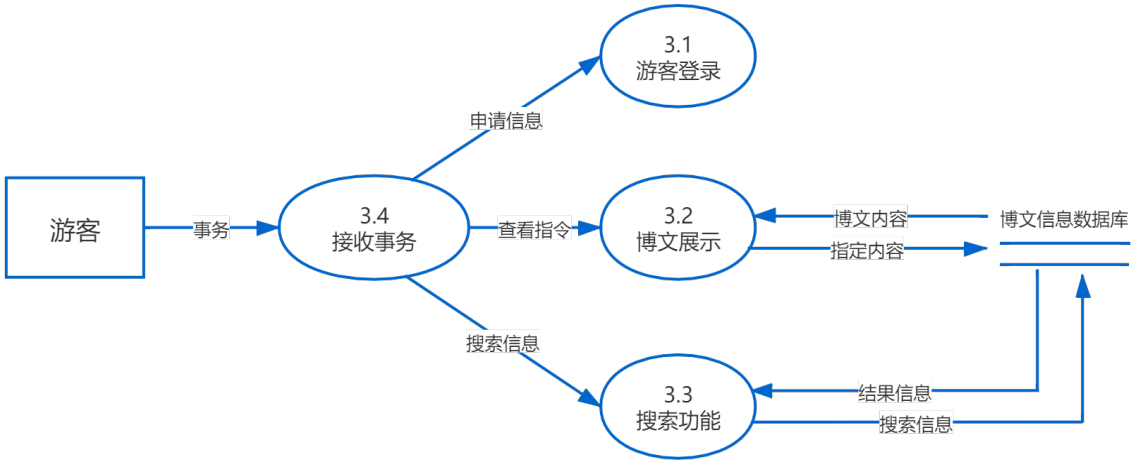
用户子系统数据流图经过了一定的改动后，被定义为事务型数据流。在数据流图中，共分为注册事务，登录事务，编辑信息事务，用户交互事务，查看博文事务和搜索事务。

② 管理员子系统



管理员子系统数据流图经过了一定的改动后，被定义为事务型数据流。在其数据流图中，共分为登录事务，管理用户信息事务，更新博文事务和查看博文信息事务。

③ 游客子系统



游客子系统数据流经过了一定改动后，被定义为事务型数据流。在其数据流图中，共分为登录事务，查看博文事务和搜索博文事务。

2.2 系统功能模块划分

经过需求分析，我团队最终确定了每个模块所具备的功能。整个项目含有六个子系统，每个子系统功能如下：

1）用户子系统功能需求

①用户注册与登录：

使用者在打开网页时，若已有账号，可通过电子邮箱、密码登录“CSDU”博客系统。若没有账号，可以输入用户名、密码以及电子邮箱注册账号，每个用户的电子邮箱都是唯一的。注册成功后，即可通过电子邮箱、密码登录“CSDU”博客系统。

②编辑个人信息：

用户可以在个人中心修改自己的头像、昵称、性别、密码。其中修改密码需要正确输入原密码方可完成。

③展示首页与博文详情：

用户可以浏览“CSDU”博客系统的首页，包括推荐博客、浏览热榜等信息；也可以浏览博文详情，包括博文的分类、标签、作者、标题与正文、评论区等信息。

④进行用户间交互：

用户可以对博文进行点赞、收藏、评论操作，也可以关注博主。用户可以向自己关注的博主发送聊天信息，收到消息的一方无需关注另一方即可回复消息。

2）管理员子系统功能需求

①管理员登录：

管理者在打开网页时，可通过电子邮箱、密码以管理员身份登录“CSDU”博客系统。

②管理用户信息：

管理者可以在管理中心进行用户信息的添加（包括电子邮箱、头像、昵称、性别、密码）、修改（包括上述可添加的信息与封禁操作等）、删除操作。

③管理博文信息：

管理者可以在管理中心进行博文分区与评论的添加、修改、删除操作，也可以对博文执行删除操作。

④展示首页与博文详情：

管理者可以浏览“CSDU”博客系统的首页，包括推荐博客、浏览热榜等信息；也可以浏览博文详情，包括博文的分类、标签、作者、标题与正文、评论区等信息。

3）游客子系统功能需求

①游客登录：

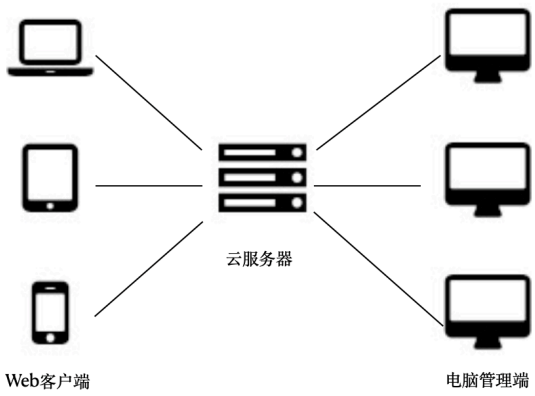
使用者在打开网页时，若没有账号可以选择游客登录。从而利用系统分配的游客 ID，无需注册账号就能体验“CSDU”博客系统。

②展示首页与博文详情：

游客可以浏览“CSDU”博客系统的首页，包括推荐博客、浏览热榜等信息；也可以浏览博文详情，包括博文的分类、标签、作者、标题与正文、评论区等信息。

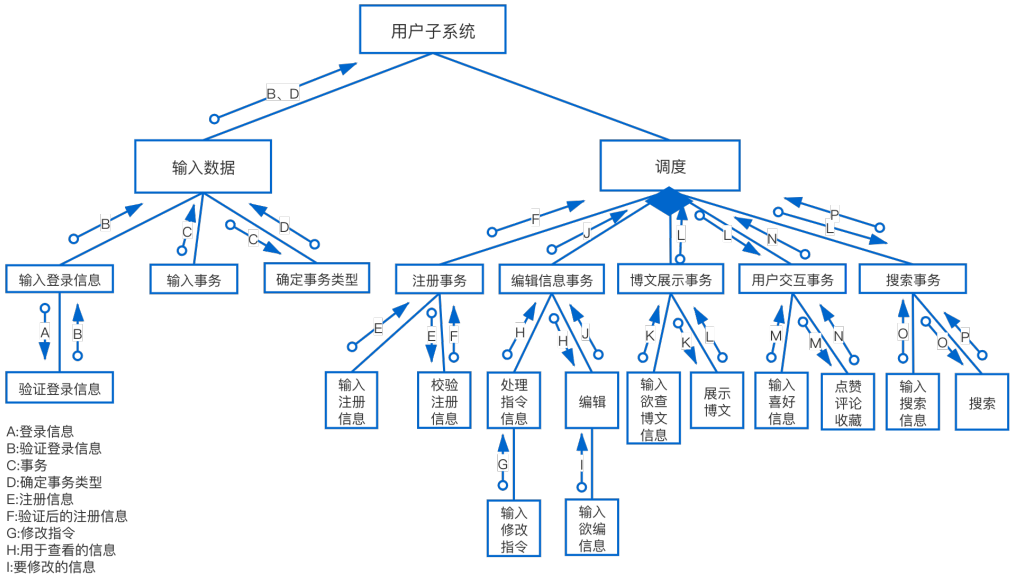
2.3 系统架构图

本系统采用 Spring Boot+MySQL 两种环境进行开发。Spring Boot 用于网页用户端、管理端的开发，MySQL 是本系统的云端数据库引擎。系统架构图如下：



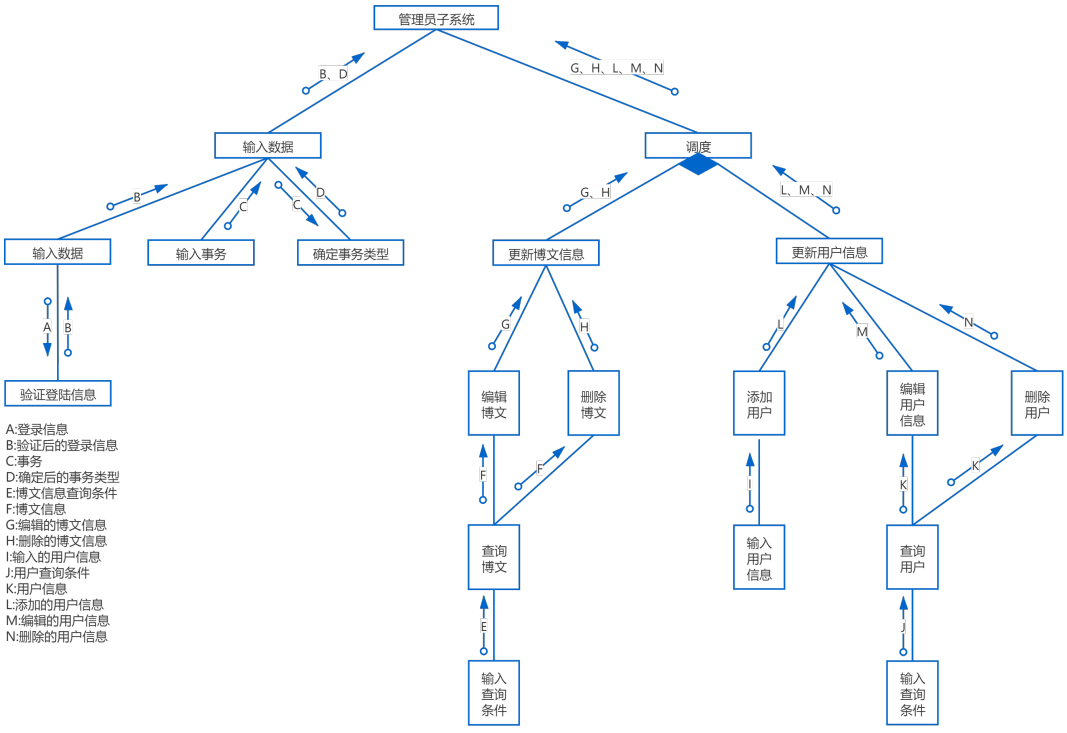
2.4 系统结构图

1）用户子系统的系统结构图



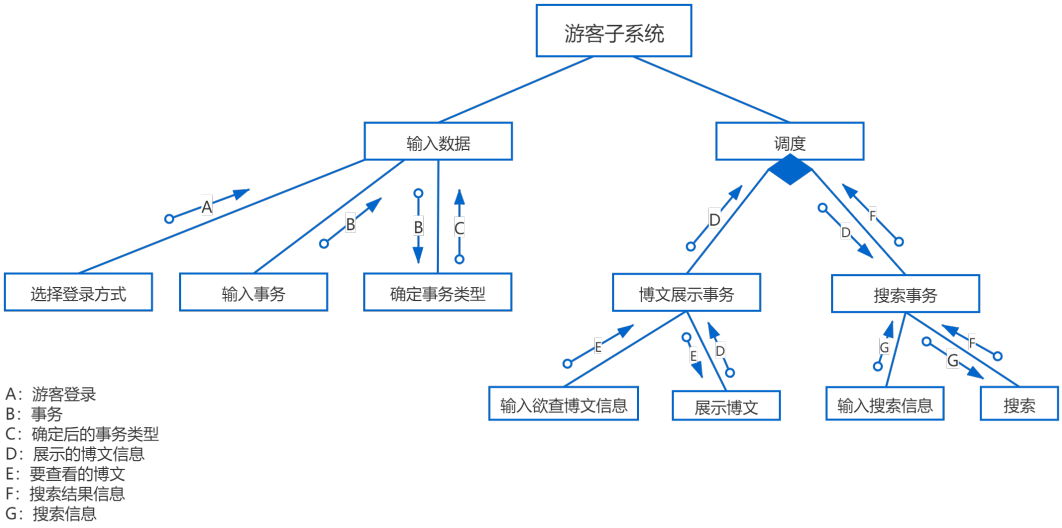
A:登录信息
B:验证登录信息
C:事务
D:确定事务类型
E:注册信息
F:验证后的注册信息
G:修改指令
H:用于查看的信息
I:要修改的信息
J:编辑后的信息
K:要查看的博文
L:展示的博文信息
M:喜好信息
N:交互结果信息
O:搜索信息
P:搜索结果信息

2）管理员子系统的系统结构图



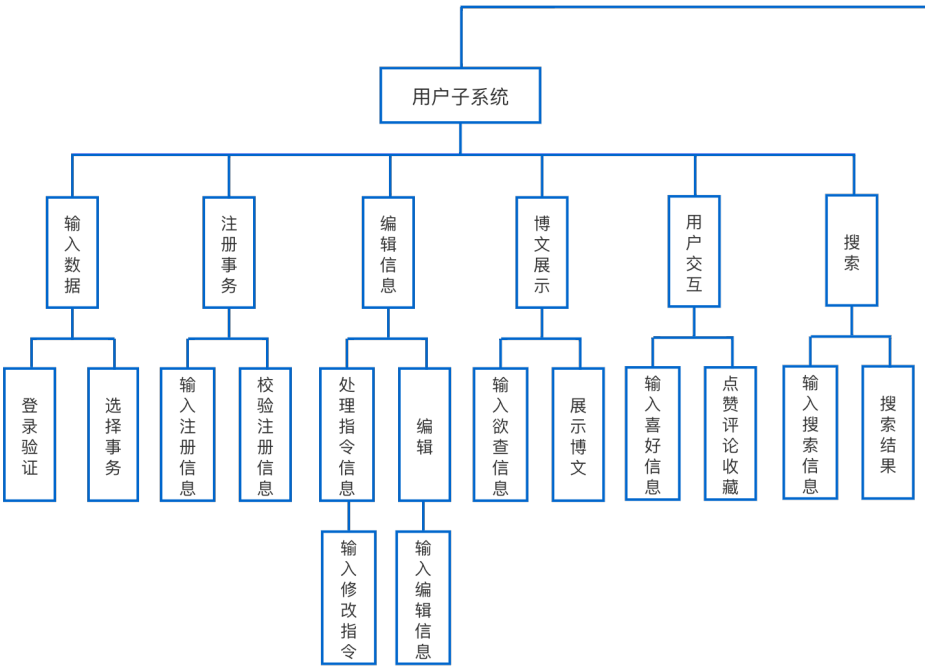
A:登录信息
B:验证后的登录信息
C:事务
D:确定后的事务类型
E:博文信息查询条件
F:博文信息
G:编辑的博文信息
H:删除的博文信息
I:输入的用户信息
J:用户查询条件
K:用户信息
L:添加的用户信息
M:编辑的用户信息
N:删除的用户信息

3）游客子系统的系统结构图

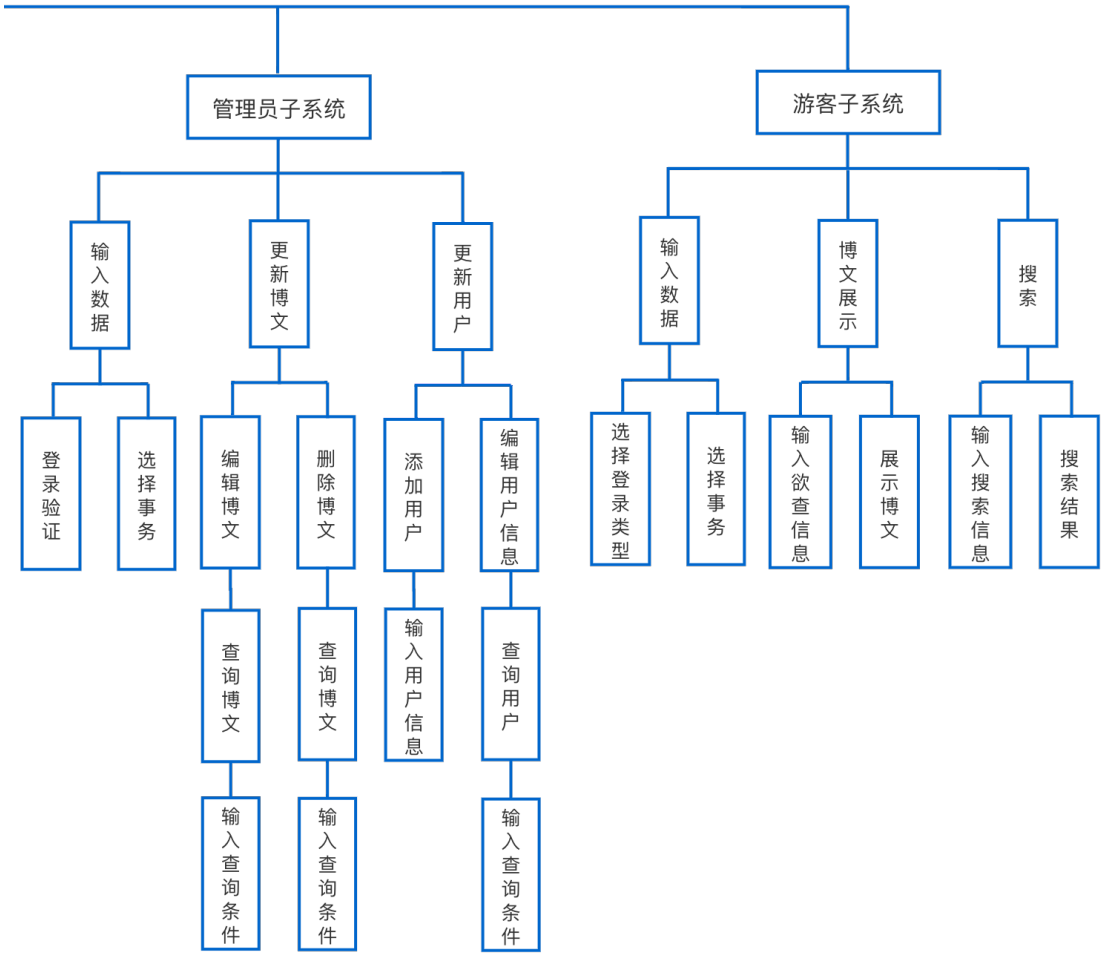


A：游客登录
B：事务
C：确定后的事务类型
D：展示的博文信息
E：要查看的博文
F：搜索结果信息
G：搜索信息

4）总体系统结构图



part1

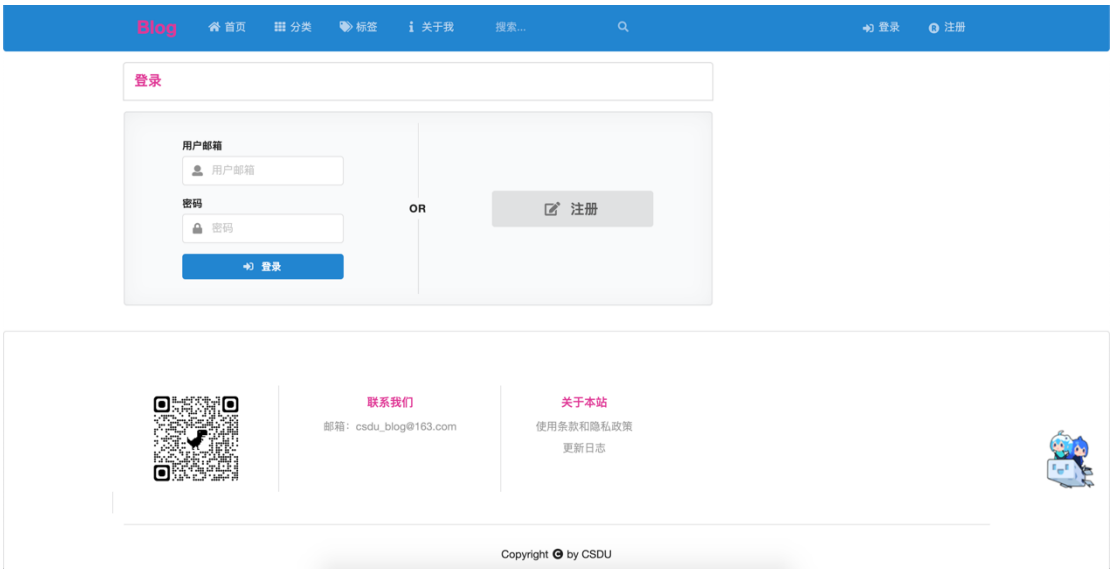

管理员子系统
游客子系统
输入数据
更新博文
更新用户
输入数据
博文展示
搜索
登录验证
选择事务
编辑博文
删除博文
添加用户
编辑用户信息
选择登录类型
选择事务
输入欲查信息
展示博文
输入搜索信息
搜索结果
查询博文
查询博文
输入用户信息
查询用户
输入查询条件
输入查询条件
输入查询条件

part2

三. 接口设计

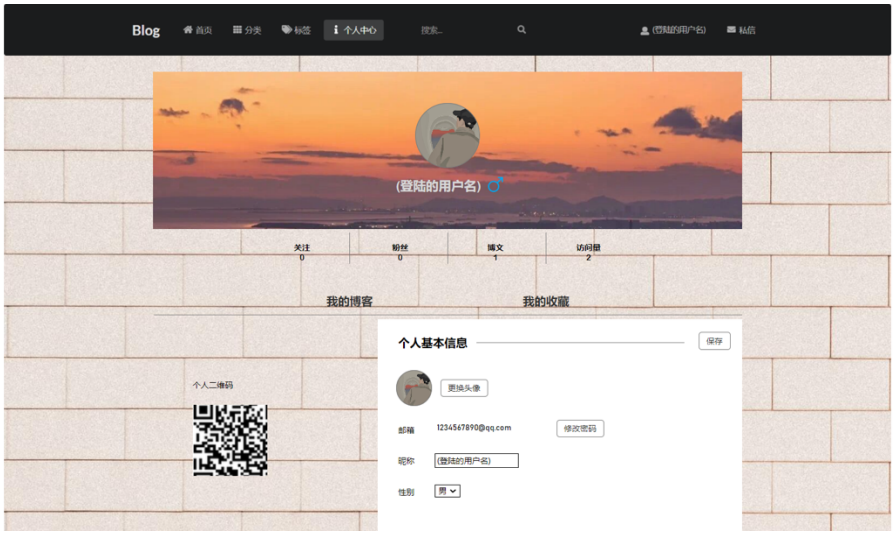
3.1 人机交互接口设计

①用户登录界面：



②博客首页：

③个人中心：



3.2 模块间接口设计

①登陆验证与其他子系统页面之间的接口

描述：由于管理员和用户登录后展示的页面稍有不同，需要在登陆时验证其身份并在相关页面启动时传递其身份信息。

例：用户登录:

```
@PostMapping("/login")
public String login(@RequestParam String username,@RequestParam
String password,HttpSession session,RedirectAttributes attributes){
    User user = userService.checkUser(username,password);
    if(user != null){
        user.setPassword(null);
        session.setAttribute( s: "user",user);
        return "admin/index";
    }
    else{
        attributes.addFlashAttribute( s: "message", o: "用户名和密
码错误");
        return "redirect:/admin";
    }
}
```

②管理员管理博客分区接口

描述：管理员需要对博文分区部分进行增删改，如可能一个用户写的博文类型不在原有的分区里，则需要接口先传递一个新的分区或者传递博文内容，然后管理员对新分区进行添加。

例：管理员编辑分区：

```
@Transactional
@Override
public Type updateType(Type type){//type 为要编辑的分区名
    Type t = typeRepository.findOne(type);
    if(t == null){
        throw new NotFoundException("不存在该类型");
    }
    BeanUtils.copyProperties(type,t);
    return typeRepository.save(t);
}
```

（管理员编辑，增加，删除用户信息与对分区所进行的操作类似，均可调用相关 JpaRepository，故不再赘述）

③全局搜索模块与用户提供搜索信息之间的接口

描述：页面显示结果时，必须根据用户输入的信息进行相关查询。

例：用户查询博文

```
public interface BlogRepository extends JpaRepository<Blog, Long>,
JpaSpecificationExecutor<Blog> {

    @Query("select b from Blog b where b.recommend = true")
    List<Blog> findTop(Pageable pageable);

    @Query("select b from Blog b where b.title like ?1 or b.content
like ?1")
    Page<Blog> findByQuery(String query,Pageable pageable);
}
```

（query 为用户输入的查询信息，之后管理员的查询操作也与这个类似，故不再赘述）

3.3 云服务器通信接口设计

①利用 Data Access Object，Springboot 自带的 JPA 中的…Repository 来与数据库进行交互。

②使用 maven 和 SpringBoot 自带的 Tomcat 打包文件，将 jar 文件部署到云服务器上。

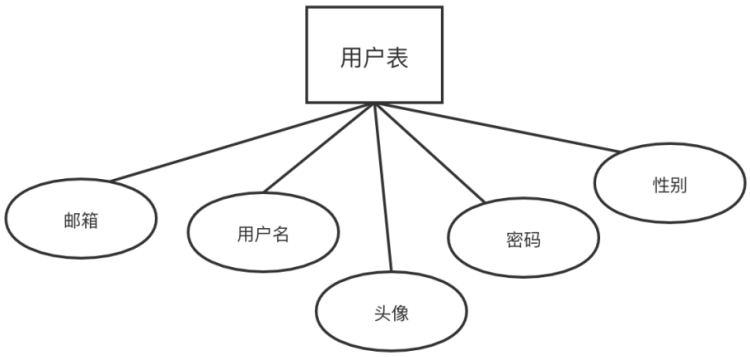
四. 数据设计

4.1 调整后的 E-R 图

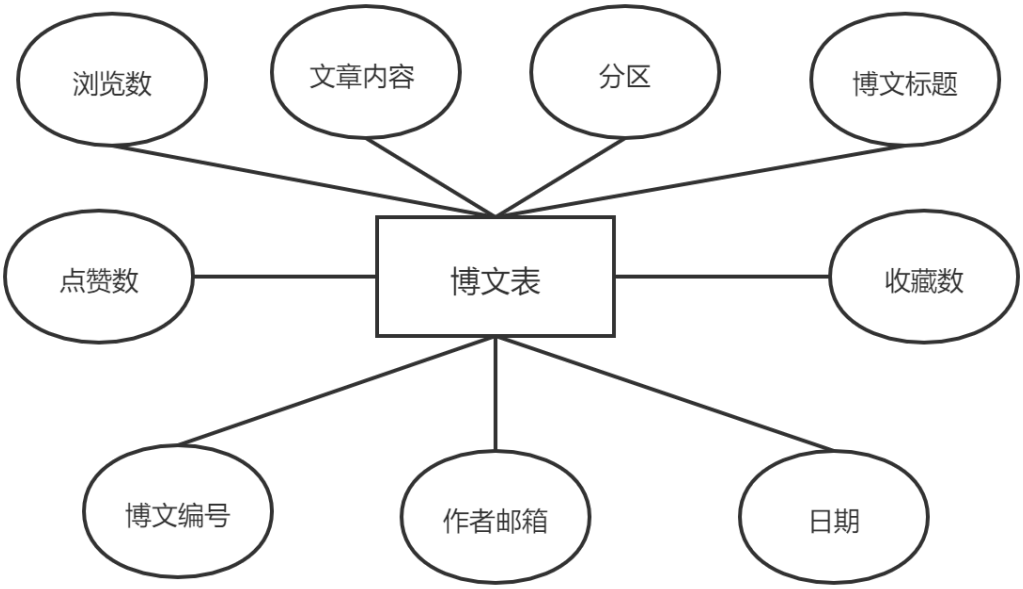
在需求分析阶段，共产生了顶层数据流图、一层数据流图、二层数据流图和三层数据流图，共四层。因此，在将其转化为系统结构图时，我们先重新绘制了二层数据流图，使其适应了事务型的数据流。

4.2 关系模式的建立

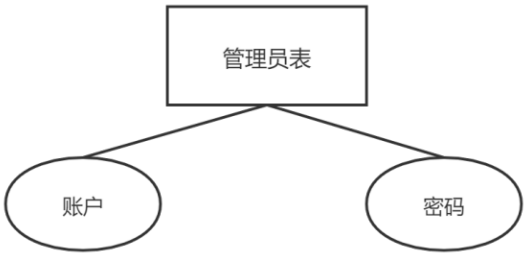
1. 用户信息表(邮箱，用户名，头像，密码，性别)



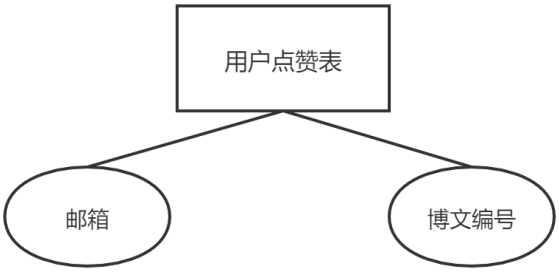
2. 博文信息表(博文编号，作者邮箱, 博文标题, 日期, 收藏数, 分区, 文章内容，浏览数，点赞数)



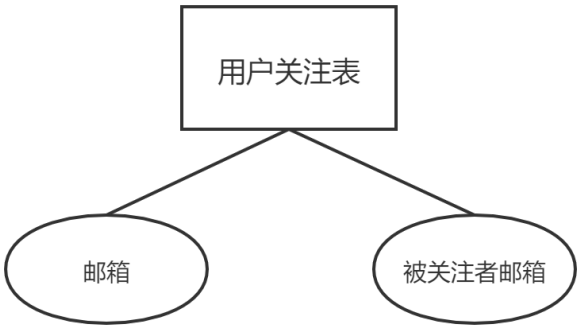
3.管理员信息表(账户，密码)



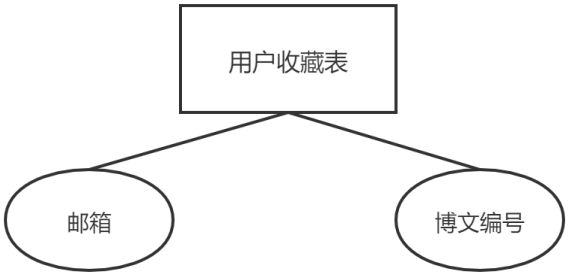
4.用户点赞表（邮箱，博文编号）



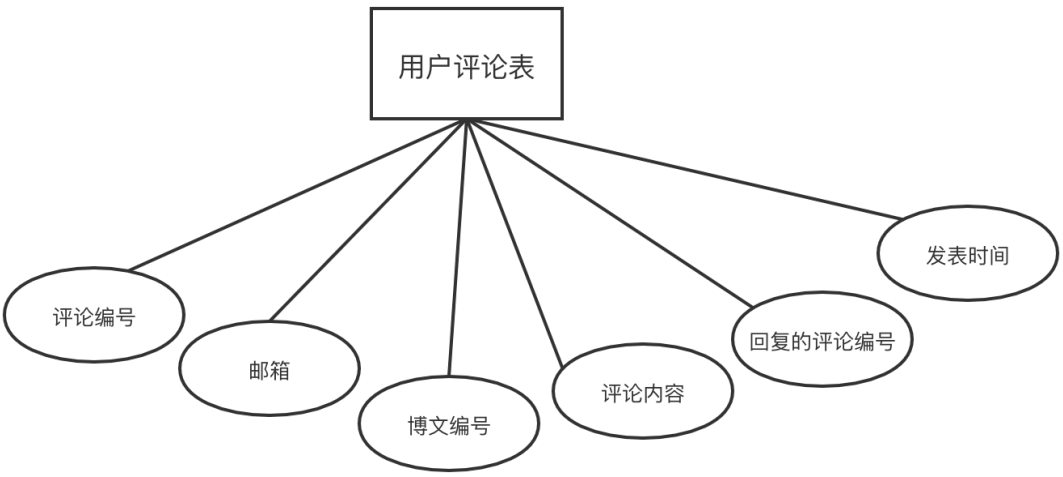
5.用户关注表(邮箱，被关注者邮箱)



6. 用户收藏表(邮箱，博文编号)



7. 用户评论表(评论编号，博文编号，邮箱，评论内容，回复的评论编号，发表时间)



4.3 数据库中表的实现

用户信息表		
属性名	数据类型	约束
邮箱	varchar(50)	primary key，长度大于等于 6 且小于等于50
用户名	varchar(64)	not null，长度大于等于 6 且小于等于 20
头像	varchar(255)	长度大于等于 6 且小于等于20
密码	varchar(64)	not null，长度大于等于 6 且小于等于20
性别	varchar(2)	not null，in('男','女')

博文信息表		
属性名	数据类型	约束
博文编号	varchar(20)	primary key，not null，长度大于等于 6 且小于等于20
作者邮箱	varchar(20)	FOREIGN KEY REFERENCES 用户信息表(邮箱)，not null，长度大于等于 6 且小于等于 20
博文标题	varchar(128)	not null
日期	date	not null
收藏数	int	not null
分区	varchar(20)	not null
文章内容	varchar(255)	not null
浏览数	int	not null
点赞数	int	not null

管理员信息表		
属性名	数据类型	约束
账户	varchar(20)	primary key，not null，长度大于等于 6 且小于等于20
密码	varchar(20)	not null，长度大于等于 6 且小于等于20

用户点赞表		
属性名	数据类型	约束
邮箱	varchar(20)	primary key，FOREIGN KEY REFERENCES 用户信息表(邮箱)，not null，长度大于等于 6 且小于等于20
博文编号	varchar(20)	primary key，FOREIGN KEY REFERENCES 博文信息表(博文编号)，not null，长度大于等于 6 且小于等于20

用户关注表		
属性名	数据类型	约束
邮箱	varchar(20)	primary key，FOREIGN KEY REFERENCES 用户信息表(邮箱)，not null，长度大于等于 6 且小于等于20
被关注者邮箱	varchar(20)	primary key，FOREIGN KEY REFERENCES 用户信息表(邮箱)，not null，长度大于等于 6 且小于等于20

用户收藏表		
属性名	数据类型	约束
邮箱	varchar(20)	primary key，FOREIGN KEY REFERENCES 用户信息表(邮箱)，not null，长度大于等于 6 且小于等于20
博文编号	varchar(20)	primary key，FOREIGN KEY REFERENCES 博文信息表(博文编号)，not null，长度大于等于 6 且小于等于20

用户评论表		
属性名	数据类型	约束
评论编号	varchar(20)	primary key，not null，长度大于等于 6 且小于等于20
博文编号	varchar(20)	primary key，FOREIGN KEY REFERENCES 博文信息表(博文编号)，not null，长度大于等于 6 且小于等于20

评论内容	varchar(20)	not null
邮箱	varchar(20)	FOREIGN KEY REFERENCES 用户信息表(邮箱)，not null，长度大于等于 6 且小于等于20
发表时间	datetime	not null
回复的评论编号	varchar(20)	FOREIGN KEY REFERENCES 用户评论表(评论编号)，not null，长度大于等于 6 且小于等于20

4.4 建表语句（MySQL）

在需求分析阶段，共产生了顶层数据流图、一层数据流图、二层数据流图和三层数据流图，共四层。因此，在将其转化为系统结构图时，我们先重新绘制了二层数据流图，使其适应了事务型的数据流。

以下为建表语句：

```
--用户信息表：
CREATE TABLE 用户信息表
(
        邮箱 VARCHAR(50) PRIMARY KEY,
        用户名 VARCHAR(64) not null,
        头像 VARCHAR(255),     --url
        密码 VARCHAR(64) NOT NULL,
        性别 VARCHAR(2) CHECK(性别='男' OR 性别='女')
);
```

```
--博文信息表
CREATE TABLE  博文信息表
(
        博文编号  VARCHAR(20) PRIMARY KEY,
        作者邮箱  VARCHAR(50) NOT NULL,
        博文标题  VARCHAR(128) NOT NULL,
        日期  DATE NOT NULL,
        收藏数  INT DEFAULT 0,
        分区  VARCHAR(20) NOT NULL,
        文章内容  VARCHAR(255) NOT NULL,      --url
        浏览数  INT DEFAULT 0,
        点赞数  INT DEFAULT 0,
        FOREIGN KEY(作者邮箱) REFERENCES  用户信息表(邮箱)
);

--管理员信息表
CREATE TABLE  管理员信息表
(
        账户  VARCHAR(20) PRIMARY KEY,
        密码  VARCHAR(64) NOT NULL
);
```

```
--用户点赞表
CREATE TABLE 用户点赞表
(
    邮箱 VARCHAR(50),
  博文编号 VARCHAR(20),
    PRIMARY KEY(邮箱,博文编号),
    FOREIGN KEY(邮箱) REFERENCES 用户信息表(邮箱),
    FOREIGN KEY(博文编号) REFERENCES 博文信息表(博文编号)
);

--用户关注表
CREATE TABLE 用户关注表
(
    邮箱 VARCHAR(50),
  被关注者邮箱 VARCHAR(50),
    PRIMARY KEY(邮箱,被关注者邮箱),
    FOREIGN KEY(邮箱) REFERENCES 用户信息表(邮箱),
    FOREIGN KEY(被关注者邮箱) REFERENCES 用户信息表(邮箱)
);
```

```
--用户收藏表
CREATE TABLE 用户收藏表
(
        邮箱 VARCHAR(50),
        博文编号 VARCHAR(20),
        PRIMARY KEY(邮箱,博文编号),
        FOREIGN KEY(邮箱) REFERENCES 用户信息表(邮箱),
        FOREIGN KEY(博文编号) REFERENCES 博文信息表(博文编号)
);

--用户评论表
CREATE TABLE 用户评论表
(
    评论编号 VARCHAR(20),
    博文编号 VARCHAR(20),
    评论内容 VARCHAR(20) NOT NULL,
      邮箱 VARCHAR(50),
    发表时间 DATETIME NOT NULL,
      回复的评论编号 VARCHAR(20),
      PRIMARY KEY(评论编号,博文编号),
```

FOREIGN KEY(博文编号) REFERENCES 博文信息表(博文编号),

FOREIGN KEY(邮箱) REFERENCES 用户信息表(邮箱),

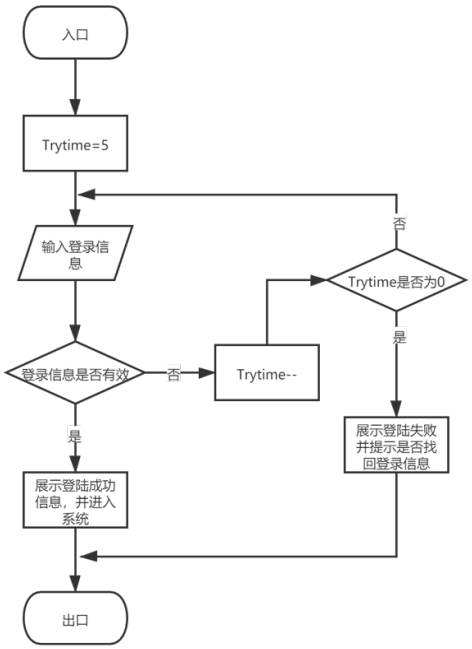
FOREIGN KEY(回复的评论编号) REFERENCES 用户评论表(评论编号)

);

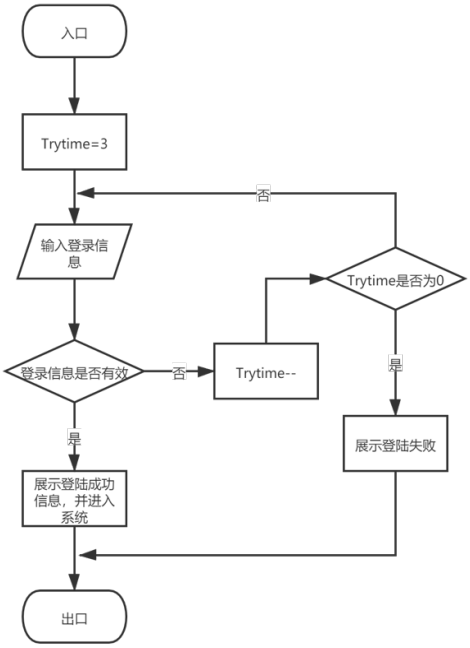
五．过程设计

5.1 主要功能模块的程序流程图

①登录功能：

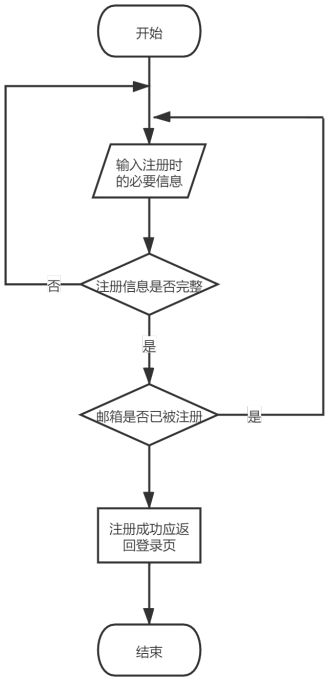


用户登陆验证

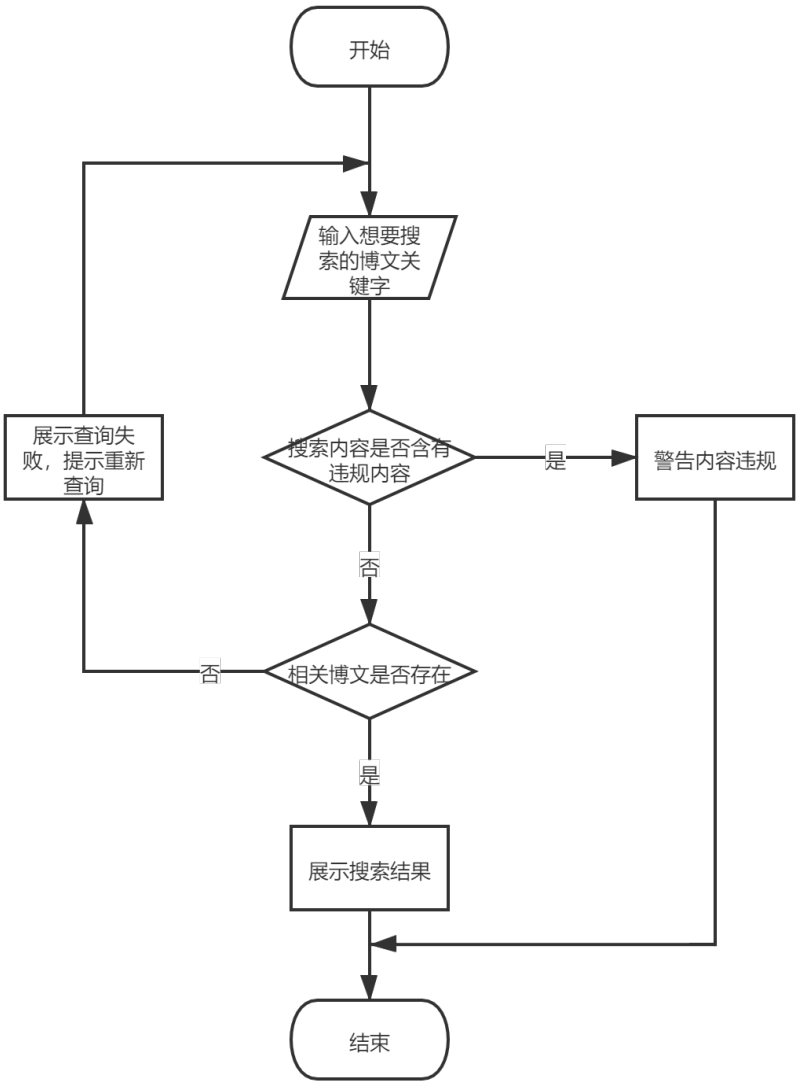


管理员登陆验证

②用户注册功能：

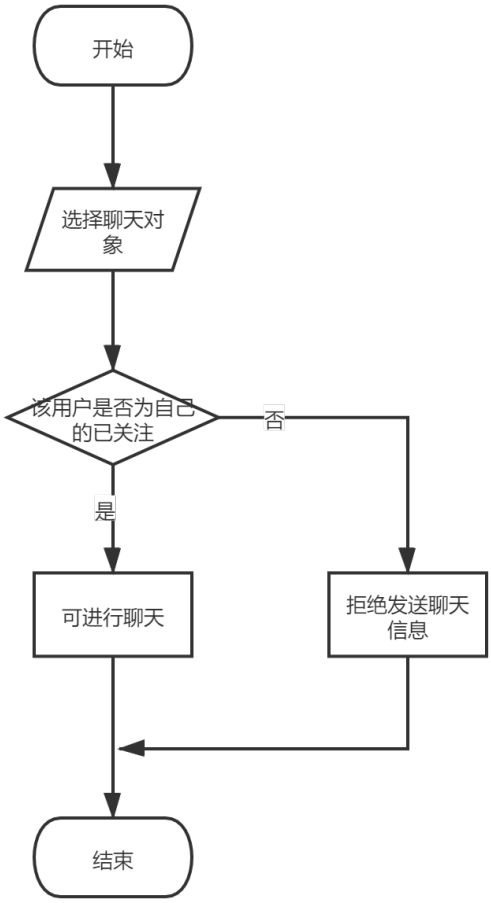


③用户搜索模块：

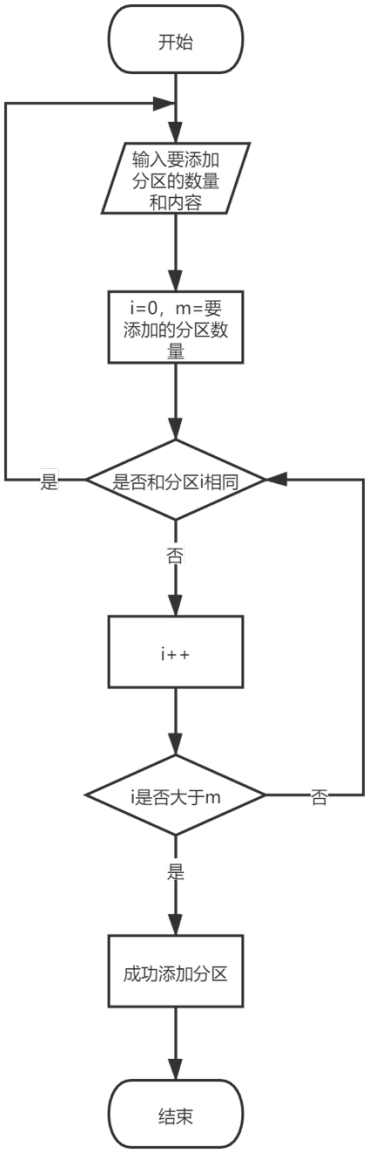


用户搜索功能

④用户间的聊天交互模块：



⑤管理员添加分区模块：


开始
输入要添加分区的数量和内容
i=0，m=要添加的分区数量
是否和分区i相同
是
否
i++
i是否大于m
否
是
成功添加分区
结束

5. 2 重要算法的伪代码表述

用户间的聊天；

```
chat_A_to_B(String A, String B, String message){
        if(A_follows_B(A, B) == true){
            show_messages(B, A, message);
        }else{
            show_messages(B, A, message = "发送失败");
        }
    }
```