

实验八

实验目的：

1. 学习Petri网基本知识及如何应用Petri网建模
2. 练习用各种动态建模工具 (状态图、Petri网、数据流图、OCL逻辑等) 对所负责的项目进行建模
3. 完善自己项目的SRS

实验内容：

1. 阅读“SYSTEM MODELLING WITH PETRI NETS”，进一步学习Petri网知识，了解如何应用Petri网对系统进行建模

Petri网是一种图形化建模工具，用于描述并发系统的行为和状态转换。它由两种基本元素组成：库所和变迁，它们之间通过弧相连。库所表示系统中的状态，变迁表示状态之间的转换，弧表示状态之间的依赖关系。下面是应用Petri网对系统进行建模的一般步骤：

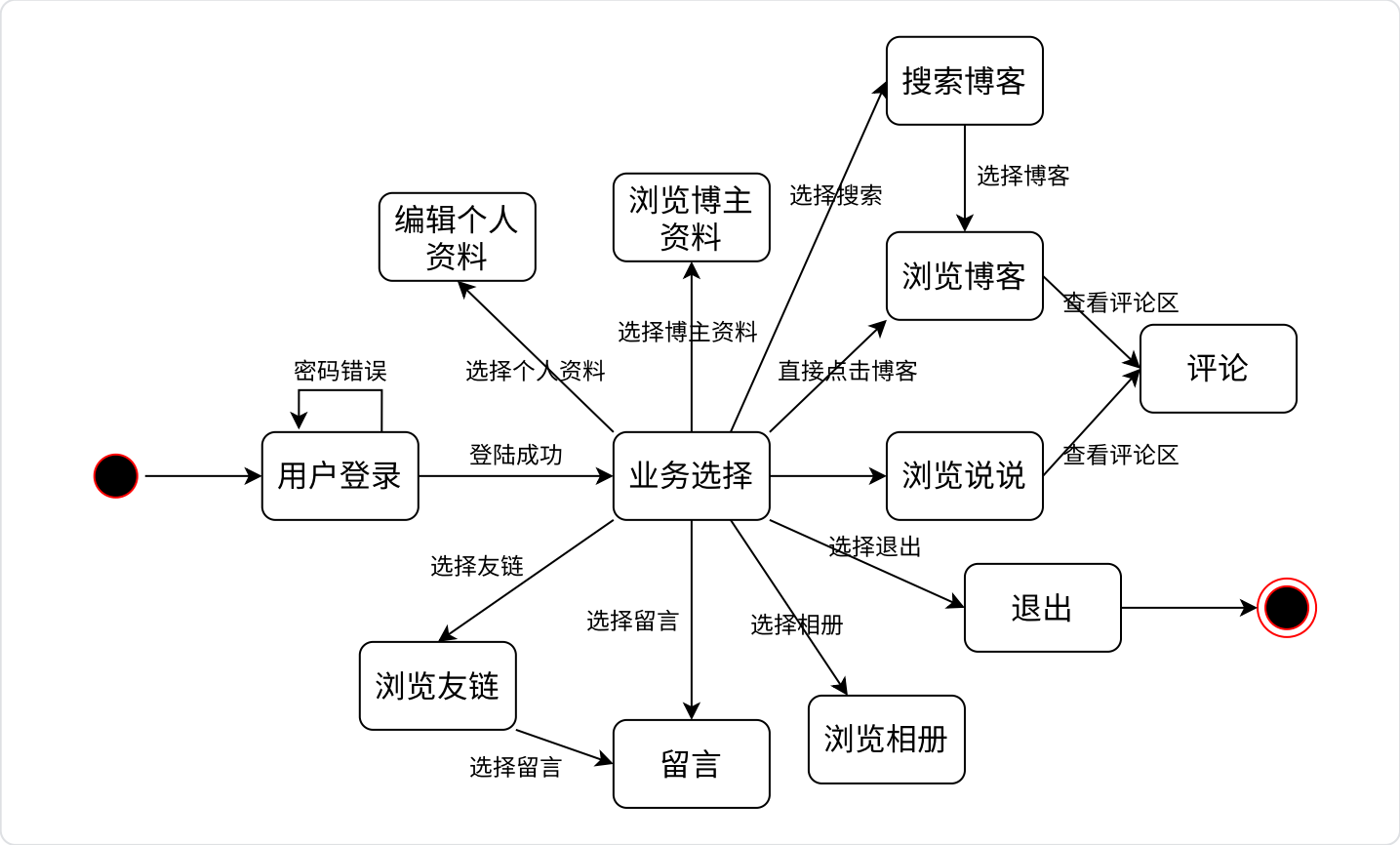
- 1、确定系统的目标和需求：明确系统的目标和需求，了解要建模的系统是什么，并确定建模的目的。
- 2、识别系统的关键组成部分：将系统拆分为不同的组成部分，例如模块、子系统或业务流程。
- 3、定义库所和变迁：为每个组成部分创建库所和变迁。库所表示系统的状态，变迁表示状态之间的转换。
- 4、连接库所和变迁：使用弧连接库所和变迁，建立状态之间的依赖关系。弧可以有向弧或无向弧，用于表示状态之间的数据流、控制流或其他类型的依赖关系。
- 5、添加初始标记和触发条件：确定系统的初始状态，并为库所添加初始标记。此外，为变迁定义触发条件，即变迁发生的条件或触发事件。
- 6、分析和验证模型：使用Petri网分析工具，例如Petri网模拟器或验证器，对模型进行分析和验证，通过模拟系统的行为，检查是否满足系统的目标和需求，发现潜在的问题和瓶颈，并进行性能评估。
- 7、迭代和改进：根据分析结果和反馈信息，对模型进行迭代和改进。
- 8、文档和沟通：将建立的Petri网模型文档化，并与团队成员或利益相关者进行沟通和共享，模型文档应清晰描述系统的结构和行为，以便他人理解和使用。

2. 针对各自负责项目的不同场景，练习用各种动态建模工具 (状态图、Petri网、数据流图、OCL逻辑等) 建模需求，与用户沟通。

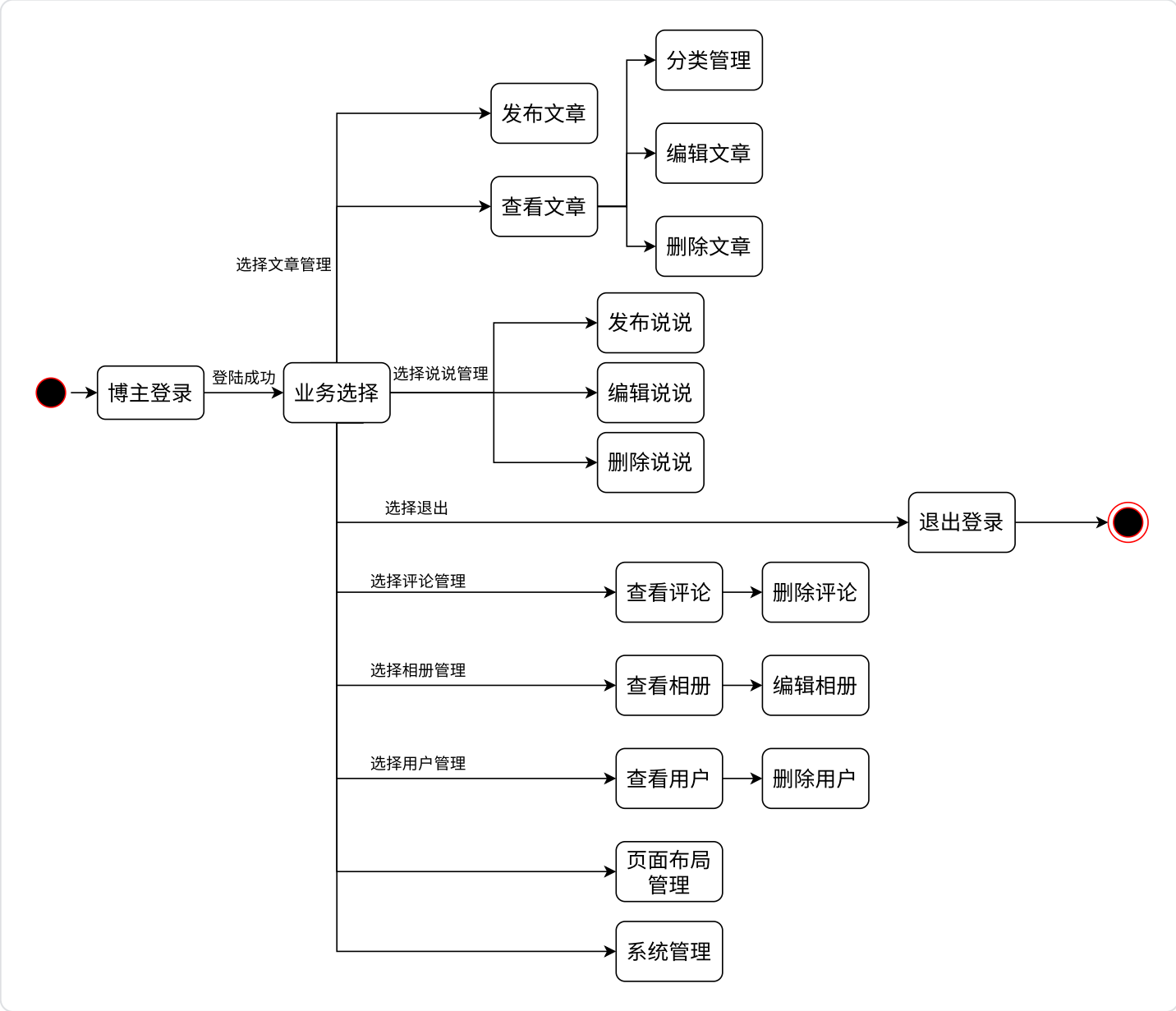
1.状态图

负责人：胡峻岩

前台用户状态图：



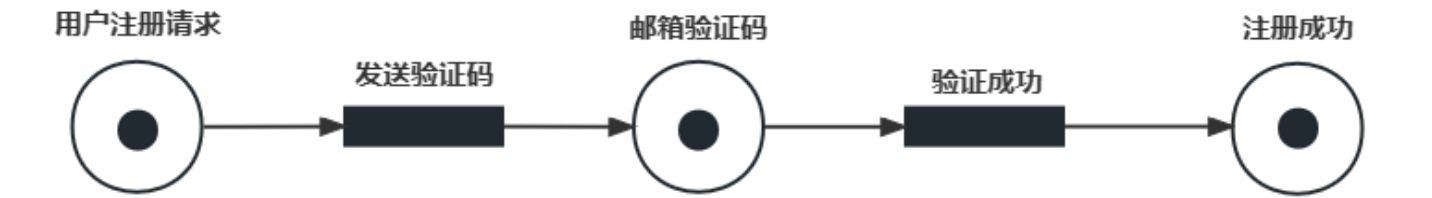
后台博主状态图：



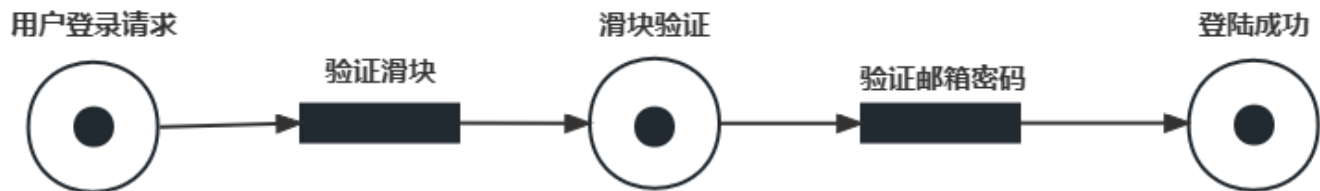
2.Petri网

负责人：邢乐凯

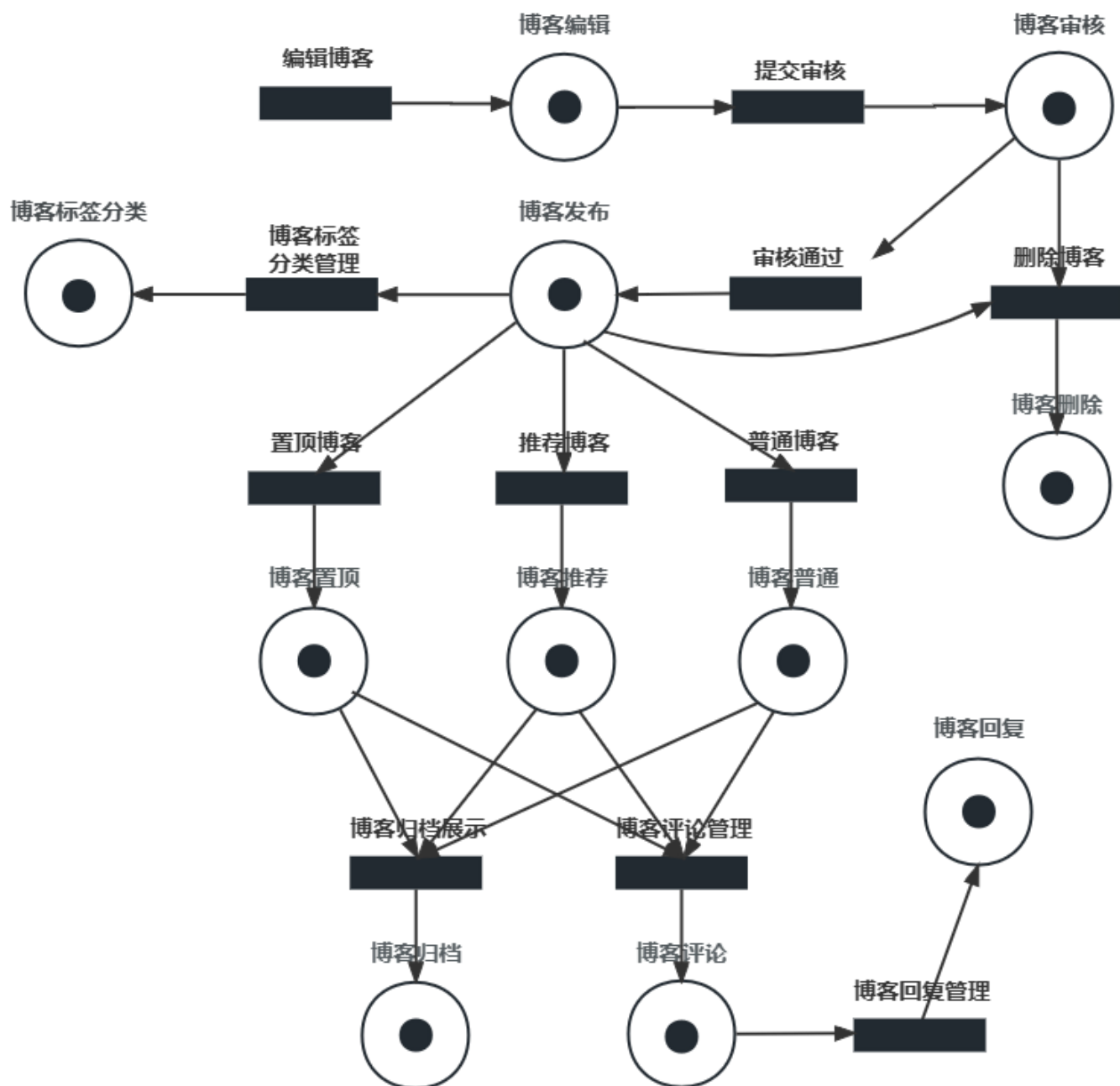
用户注册Petri网



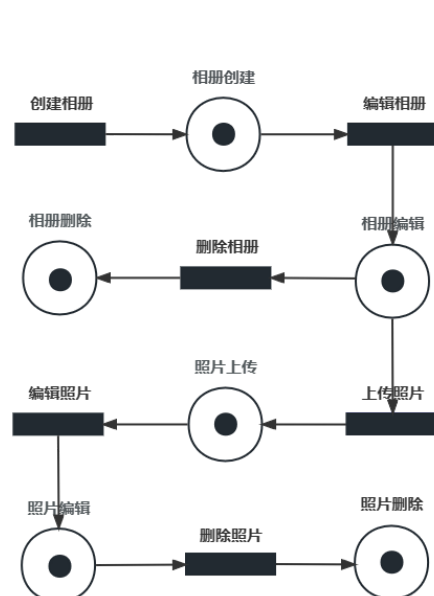
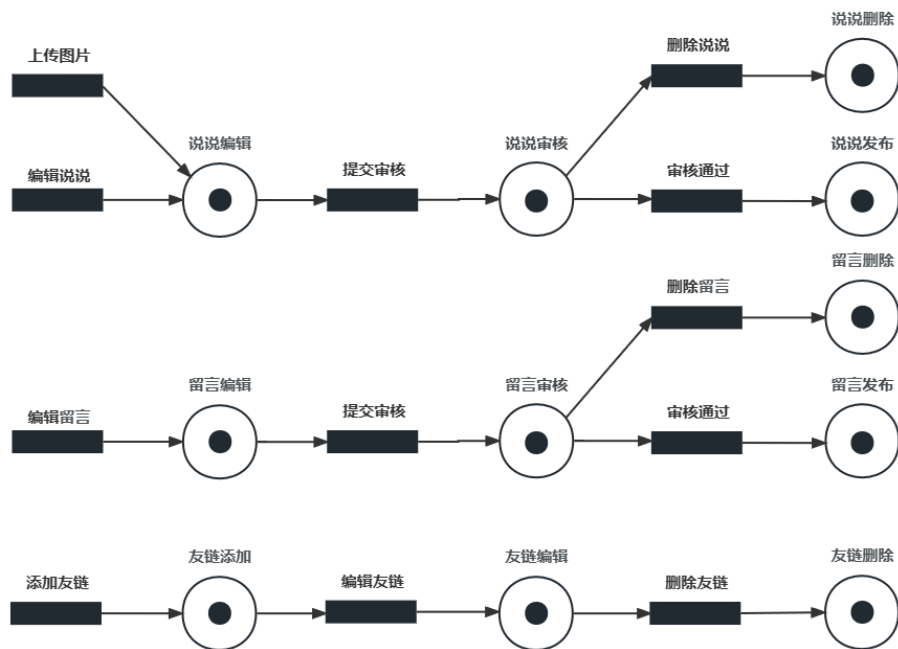
用户登录Petri网



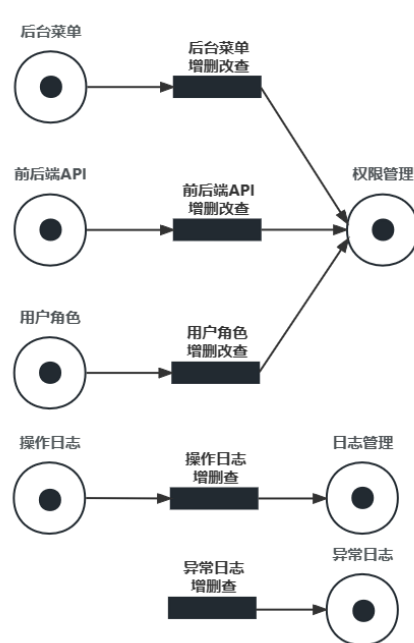
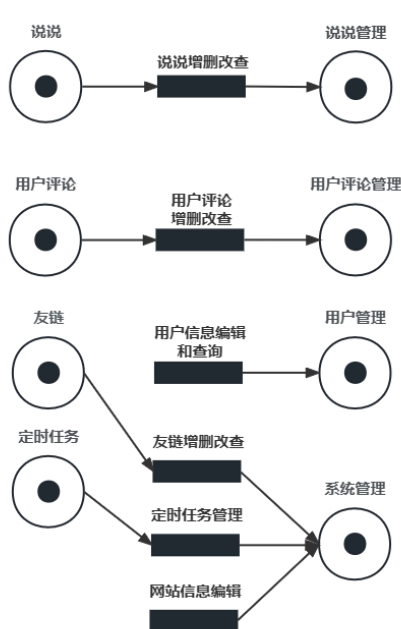
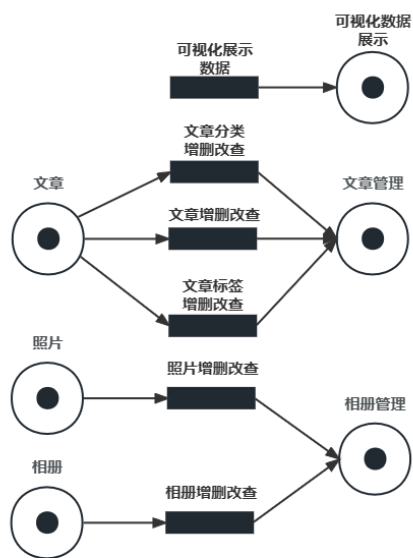
博客Petri网



其他模块Petri网

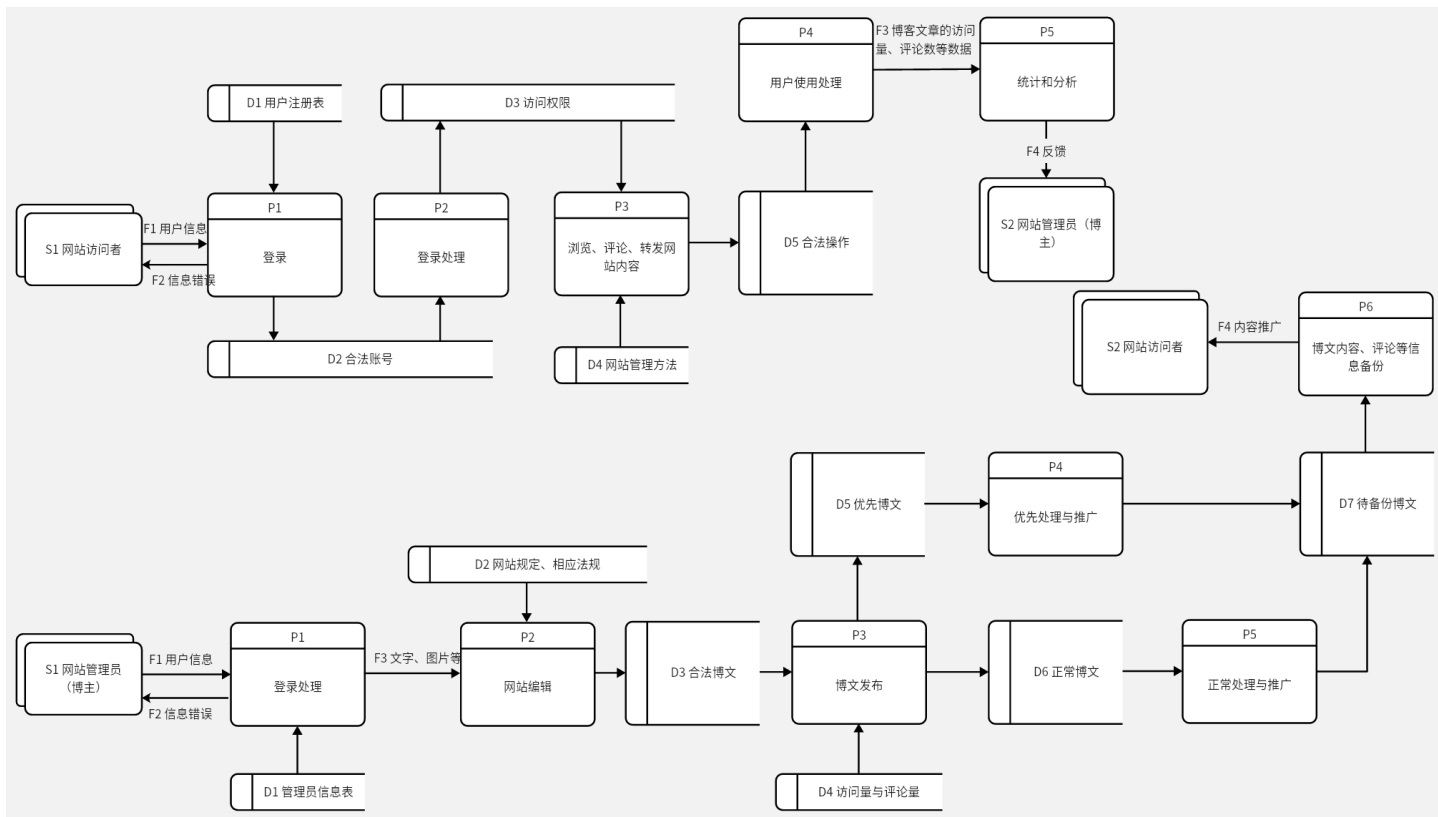


后台管理Petri网



3.数据流图

负责人：于一帆



4.OCL逻辑

负责人：付召帅

1.登陆处理，检查用户是否存在以及用户名和密码是否匹配

```

1 context User
2   def: authenticate(username: String, password: String) : Boolean =
3     self.username = username and self.password = password
4   enddef
5 endcontext
6
7 context LoginController
8   pre: User.allInstances()->exists(user | user.username = username)
9   pre: User.allInstances()->select(user | user.username = username)->one().authen
10  post: currentUser = User.allInstances()->select(user | user.username = usernam
11 endcontext
  
```

2.操作合法性验证，以验证输入内容字数为例

```

1 context BlogPost
2   inv: self.title.size() <= 100
3 endcontext
  
```

3.用户权限检测，检查当前用户为普通用户或管理员

```
1 context BlogPost
2   inv: self.owner = currentUser or currentUser.role = 'admin'
3 endcontext
4
5 context Comment
6   inv: self.author = currentUser or currentUser.role = 'admin'
7 endcontext
```

4.文章标签约束

```
1 context BlogPost
2   inv: self.tags->forAll(tag | allTags->includes(tag))
3 endcontext
4
5 context Tag
6   inv: self.posts->forAll(post | allPosts->includes(post))
7 endcontext
```

5.主楼和评论的状态

```
1 context BlogPost
2   pre: self.status = 'draft'
3   post: self.status = 'published'
4 endcontext
5
6 context Comment
7   pre: self.status = 'published'
8   post: self.status = 'edited'
9 endcontext
```

6.检查文章和用户的关系

```
1 context BlogPost
2   inv: self.owner = currentUser or self.status = 'published'
3 endcontext
4
5 context User
6   inv: self.posts->forAll(post | post.owner = self)
7 endcontext
```

7.非注册用户（游客）的访问权限

```
1 context BlogPost
2   inv: currentUser.role <> 'guest' or self.status = 'published'
3 endcontext
4
5 context Comment
6   inv: currentUser.role <> 'guest' or self.status = 'published'
7 endcontext
```

3. 分工协作，用上面的工作补充完善SRS和所负责的项目。