软件设计文档

项目名称：RelaxBlog

参与人员：董建文 黄伟

目录

1. 引言

1.1 编写目的

本阶段完成系统的大致设计并明确系统的数据结构与软件结构。本概要设计说明书的目的就是进一步细化软件设计阶段得出的软件概貌，把它加工成在程序细节上非常接近与源程序开发的软件表示。
预期读者：软件测试员、程序开发员、软件分析员

1.2 项目背景

a. 项目的委托单位：无
b. 开发团队：SUNDAY 小组
c. 主管部门：SUNDAY 小组
d. 该软件系统与其他系统的关系：系统相对独立

1.3 定义

a. Mysql:系统服务器所使用的数据库关系系统（DBMS）。
b. SQL：一种用于访问查询数据库的语言
c. 事务流：数据进入模块后可能有多种路径进行处理。
d. 主键：数据库表中的关键域。值互不相同。
e. 外部主键：数据库表中与其他表主键关联的域。
f. ROLLBACK:数据库的错误恢复机制。
g. 缩写：
h. SQL：Structured Query Language(结构化查询语言)。
i. ATM：Asynchronous Transfer Mode (异步传输模式)。
j. UML：统一建模语言、是一套用来设计软件蓝图的标准建模语言，是一种从软件分析、设计到编写程序规范的标准化建模语言。

1.4 参考资料

a. 项目开发计划；
b. RelaxBlog 需求规格说明书；
c. RelaxBlog 需求清单
d. 《软件工程概论》 李存珠编著 南京大学计算机系出版 2001 年 8 月
e. 《软件工程——实践者的研究方法》Roger S Pressman 著

2．任务概述

2.1 目标

a. 由于本系统用于个人博客系统，使用频繁，因此可靠性要较高、安全性较高。
b. 系统的运行速度要快

2.2 运行环境

该系统为 B/S 三层结构，它的运行环境分客户端、应用服务器端和数据库服务器端三部分。以下是系统的软件环境。
（1）客户端
操作系统：Windows8 或更新版本。
浏览器：IE9 以上，其它常见浏览器如 FireFox，Chrome。
（2）应用服务器端
操作系统：Windows7 或更新版本。
应用服务器：Tomcat 5.5 或更新版本。
数据库访问：JDBC。
（3）数据库服务器端
操作系统：Windows7 或更新版本。
数据库系统：MYSQL5.7 或更新版本。

2.3 需求概述

现今很多技术开发人员都有撰写个人博客的习惯，一方面进行技术的交流学习，另一方面也可以进行自己开发心得以及经验的记录。

本项目旨在开发一个新型博客，主要功能是实现一个由个人管理的，能够提供发布博客，更新博客，播放（云）音乐等功能，结合了文字、图像、其他博客或网站的链接及其它与主题相关的媒体，能够让读者以互动的方式留下意见为用户，旨在开发出简洁易用，界面清爽的个人博客系统，提供一个深度交流沟通的网络平台。

2.4 条件与限制

开发时需要的支持条件：
硬件：
服务器：Pentium III 500 以上或更高，
内存：512M 以上；
硬盘：至少 80G 以上；

CD－ROM：32 倍速以上；
网络适配器：10MB/100MB 自适应；
打印机一台
UPS(选配)
工作站：Pentium 4 以上微机；
内存：512MB
硬盘：至少 80 以上；
CD－ROM：32 倍速以上；
网络适配器：10MＢ/100MＢ自适应
网络： 至少一台服务器
至少一台工作站
使用 TCP/IP 协议的局域网

软件：

操作系统为 Window XP，使用集成开发工具 Eclipse5.5.1,数据库采用 SQL Server2000，项目运行环境为 JDK6.0.
其他开发工具包括：Dreamweaver, Microsoft Visio, Rational Rose, Power Designer Trial 11, TomCat6.0 CVSNT2.5.03

运行时需要的支持条件：

服务器的要求

1. 服务器的中央处理部件（CPU）建议使用 PIII 1G（以上） Xeon 处理器芯片。
2. 服务器内存必须使用服务器专用 ECC 内存
3. 为了保证数据存储的绝对可靠，硬盘应使用磁盘冗余阵列（RAID 01）
4. 为了防止服务器不可预测的故障，或者服务器的定期维护对公司整个业务造成的影响，所有建议使用两台服务器。两台服务器应构成双机热备份。中间使用 Watchdog 电路。这样的结构可以保证整个系统的长时间不间断工作，即使在服务器定期维护的时候也可以使用后备另一台服务器工作。
5. 服务器应支持热插拔电源
6. 服务器必须配备 UPS（不间断电源）。
7. 服务器应该放在学校内部。不然无法进行程序调试。
8. 服务器应该必须有固定 IP 地址。
9. 其他性能在经济条件允许的情况下，应该尽量使用高速稳定的配件。

服务器上应该配备的软件

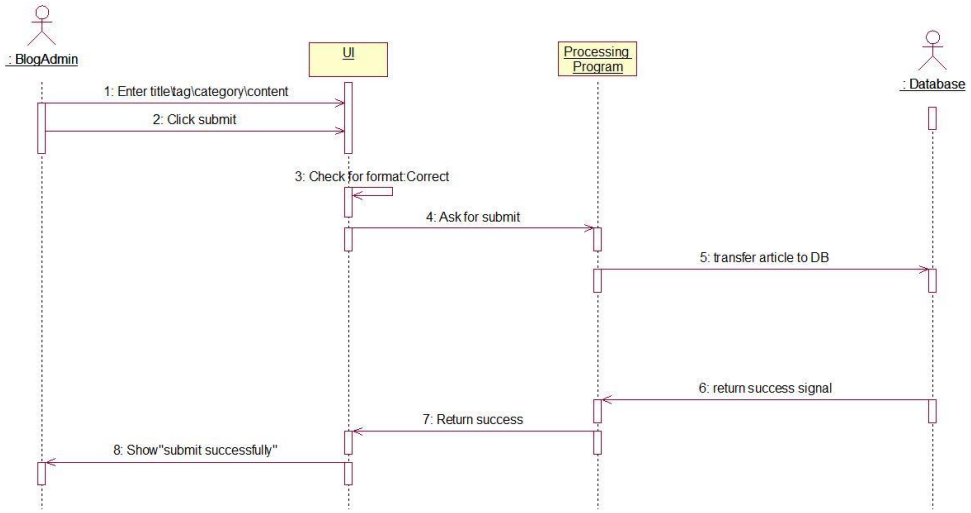
1. 操作系统：Microsoft Windows 2000 server 或者 Microsoft Windows 2000 Advanced server
2. 数据库：Microsoft SQL Server 2000 (简体中文版)
3. 服务器必须使用专业的防火墙和反病毒软件。
4. 除了为了运行必须配备的程序以外，服务器上建议尽量不要安装其他无关程序，以减少程序的混乱或者程序的意外冲突。
5. 各系的操作系统尽量统一。（Windows 9x 系列或者 Windows 2000 系列）。这样可以避免管理软件因为操作系统版本不一致造成的过多的开销。
6. 各系的机器必须也安装反病毒软件和防火墙。以防止网络上的蠕虫病毒在整个网络范围内的蔓延。
7. 如果要打印涉及字段比较多的报表，应该配备针式打印机。

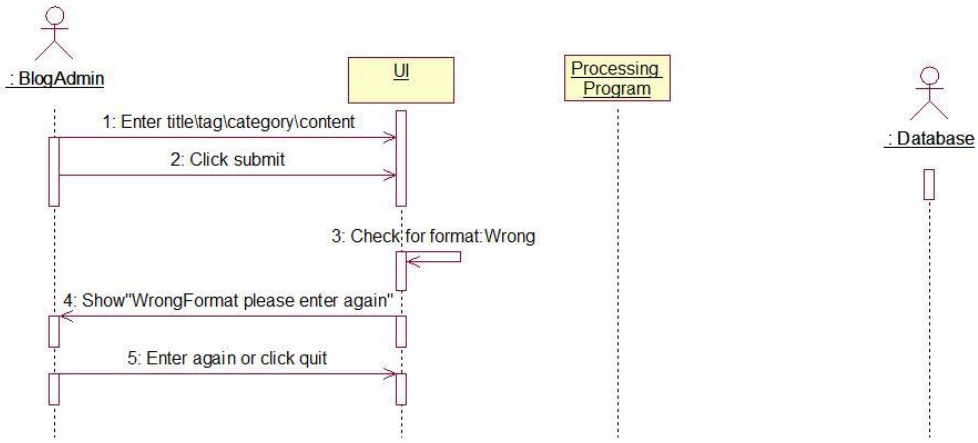
3. 总体设计

3.1 处理流程

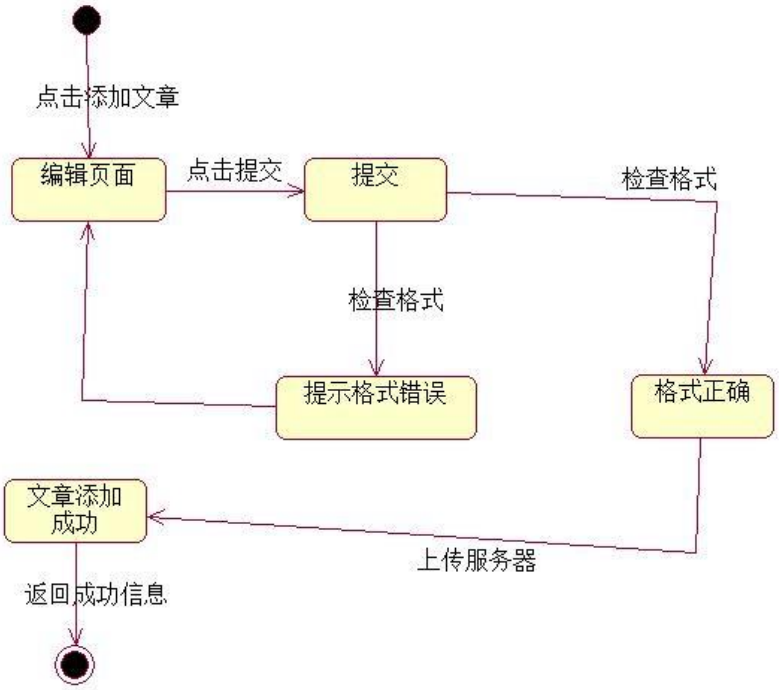
3.1.1 添加文章

时序图：



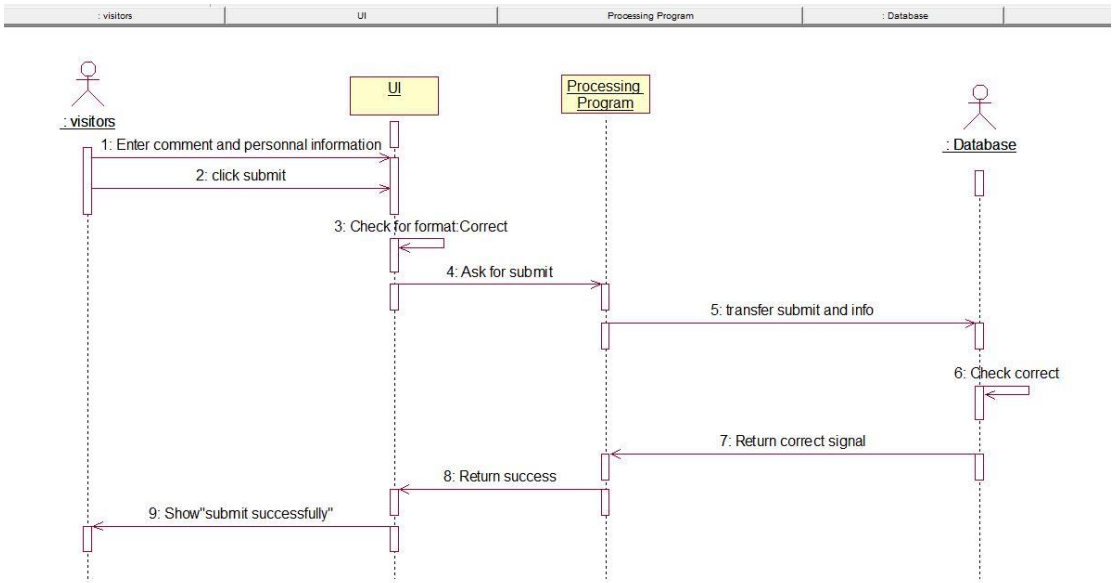


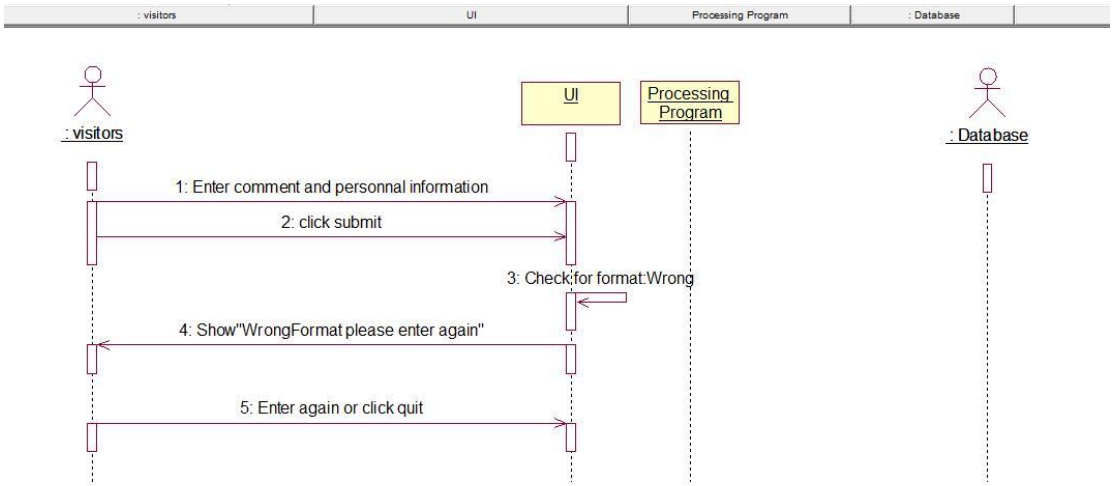
状态图：


点击添加文章
编辑页面
点击提交
提交
检查格式
检查格式
提示格式错误
格式正确
文章添加成功
上传服务器
返回成功信息

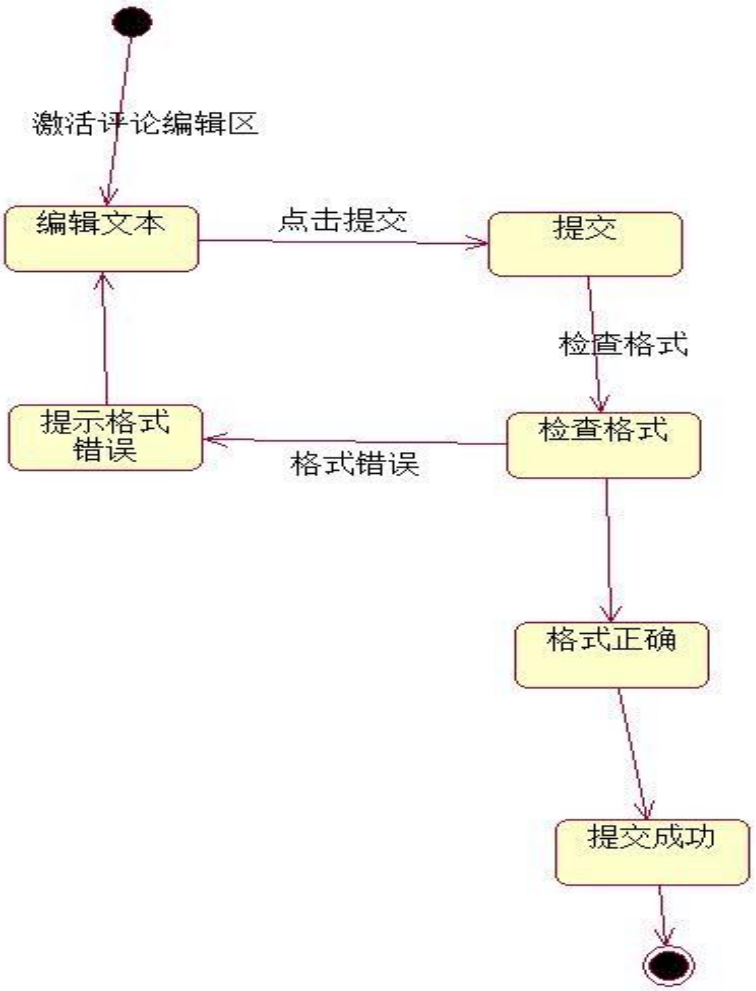
3.1.2 评论

时序图：


: visitors
UI
Processing Program
: Database
1: Enter comment and personnal information
2: click submit
3: Check for format:Correct
4: Ask for submit
5: transfer submit and info
6: Check correct
7: Return correct signal
8: Return success
9: Show"submit successfully"

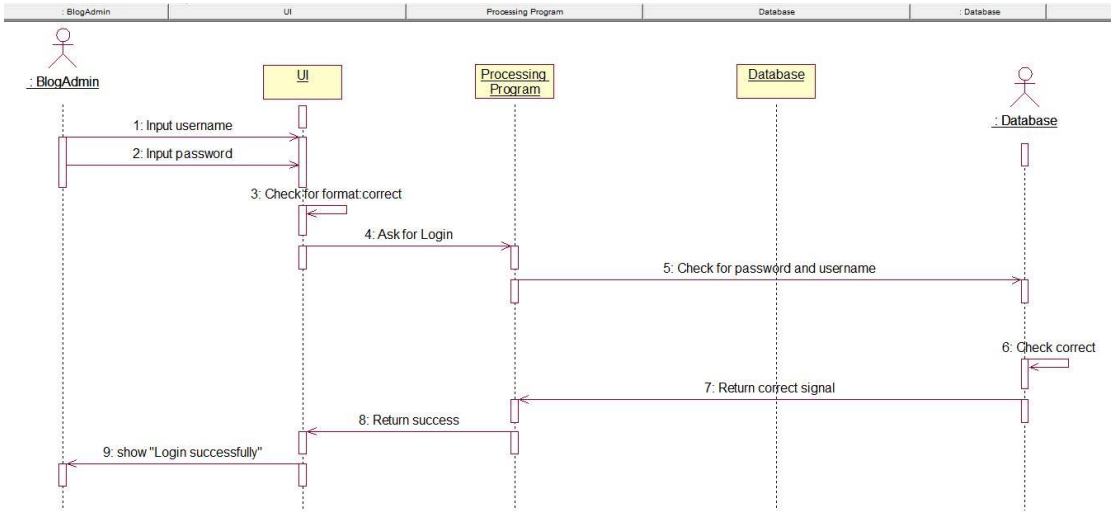

: visitors
UI
Processing Program
: Database
1: Enter comment and personnal information
2: click submit
3: Check for format:Wrong
4: Show"WrongFormat please enter again"
5: Enter again or click quit

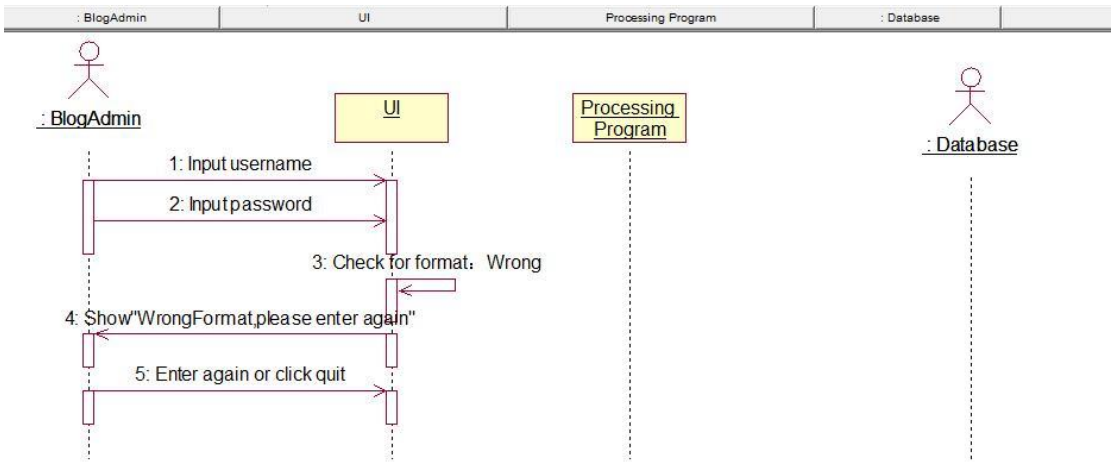
状态图：

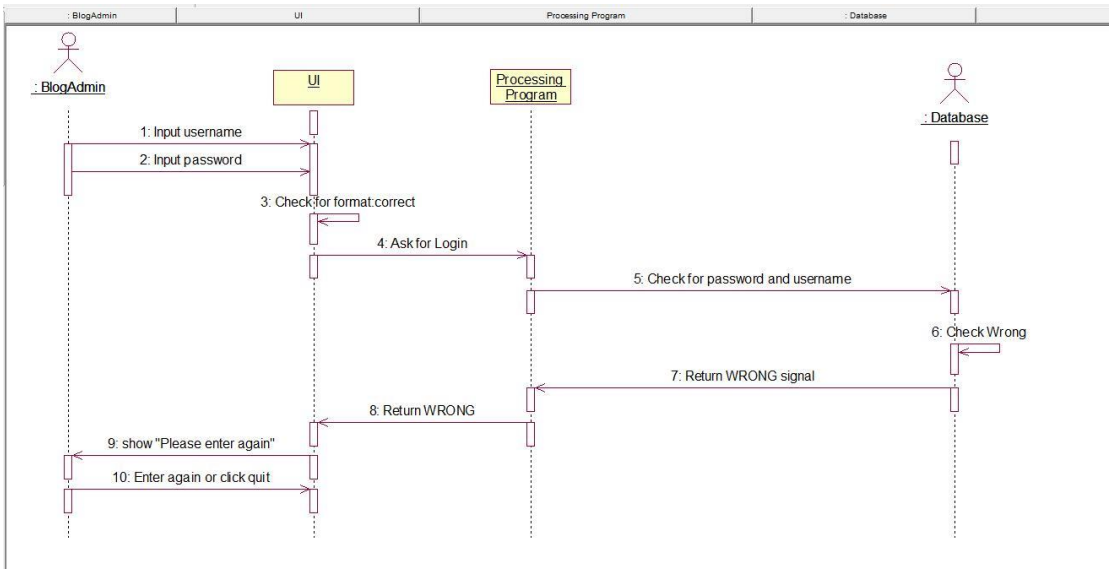

激活评论编辑区
编辑文本
点击提交
提交
检查格式
提示格式
错误
格式错误
检查格式
格式正确
提交成功

3.1.3 登录

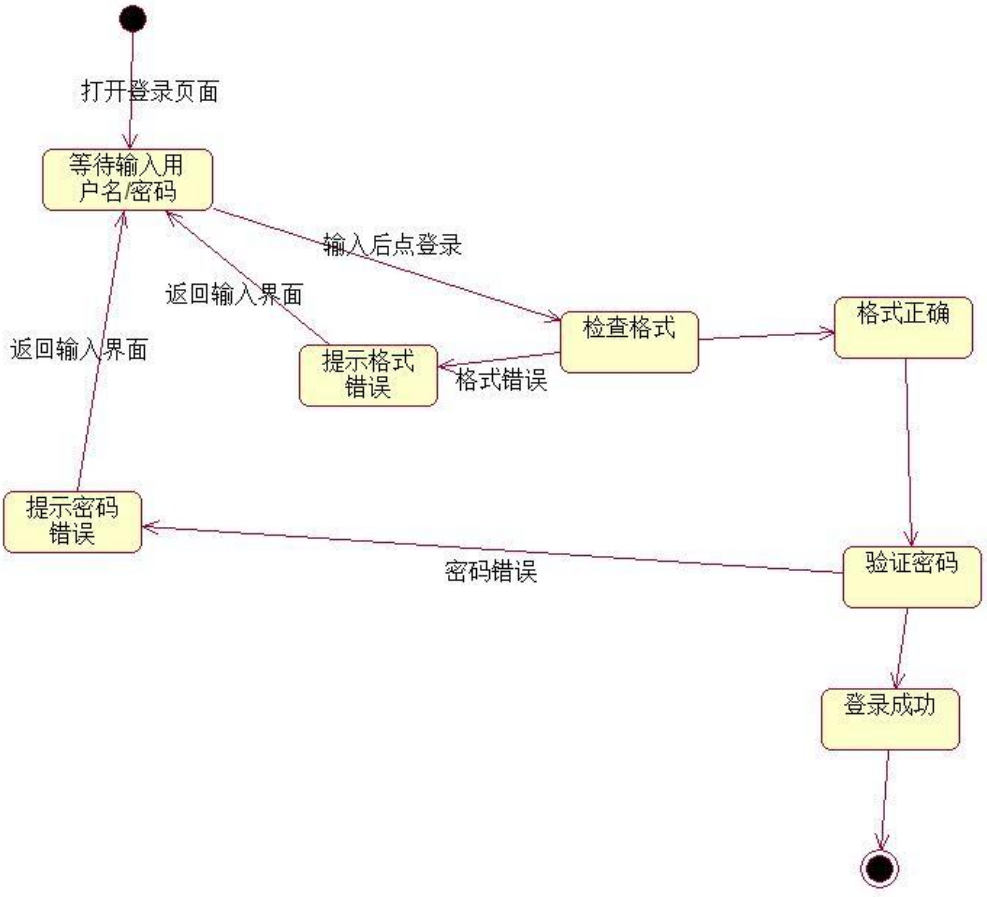
时序图：


: BlogAdmin
UI
Processing Program
Database
: Database
1: Input username
2: Input password
3: Check for format correct
4: Ask for Login
5: Check for password and username
6: Check correct
7: Return correct signal
8: Return success
9: show "Login successfully"


: BlogAdmin
UI
Processing Program
: Database
1: Input username
2: Input password
3: Check for format：Wrong
4: Show"WrongFormat,please enter again"
5: Enter again or click quit

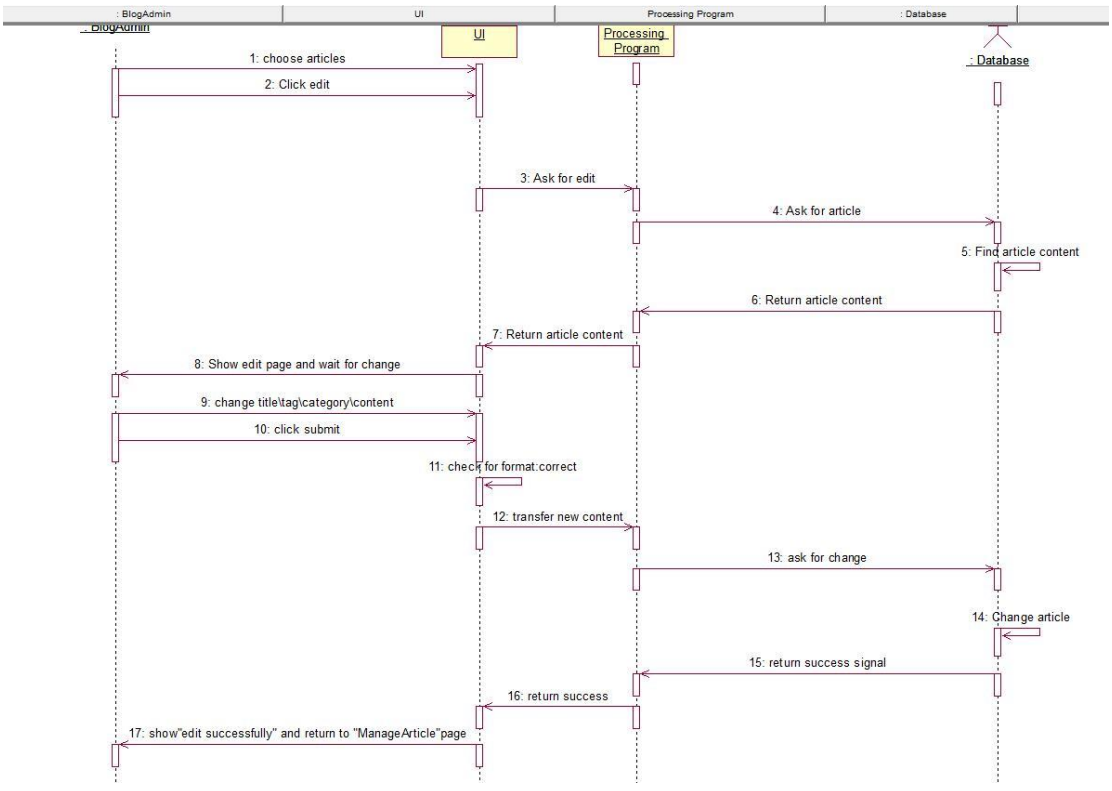


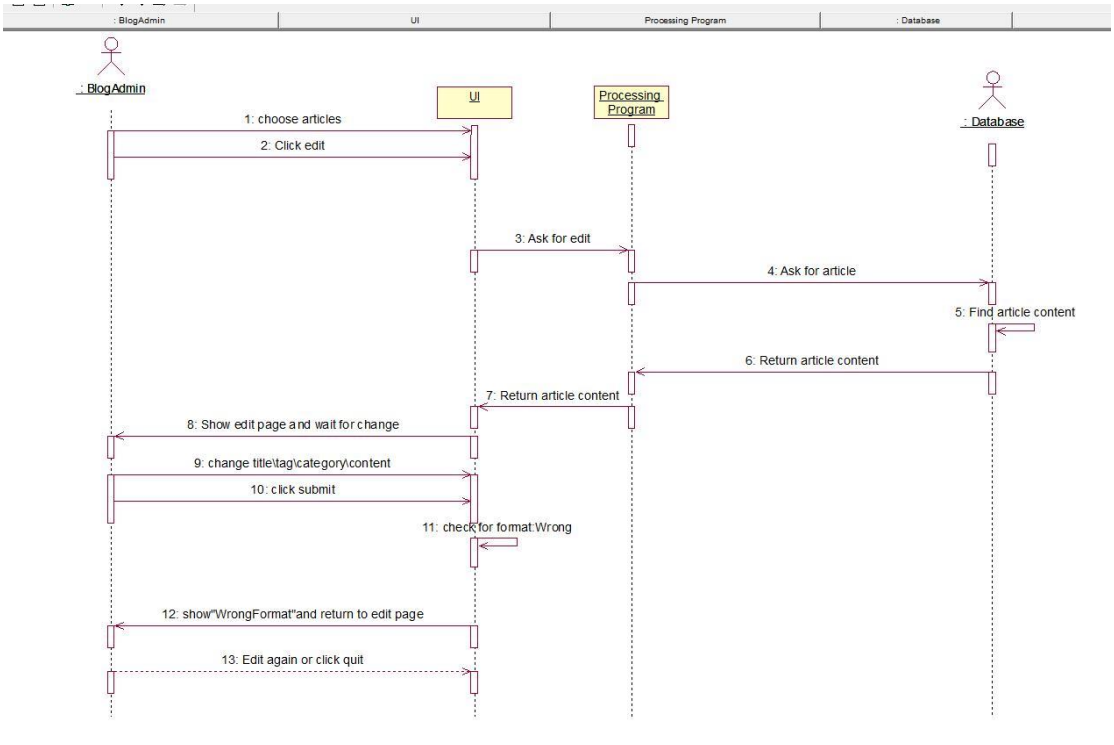
状态图：



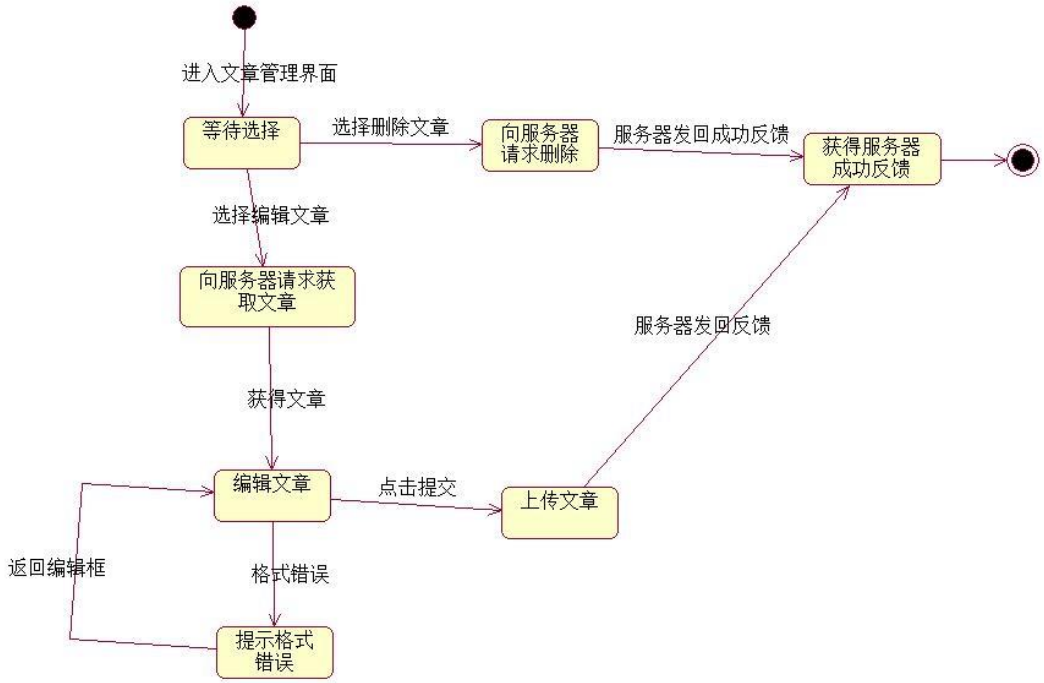
3.1.4 文章管理

时序图：


: BlogAdmin
UI
Processing Program
: Database
1: choose articles
2: Click edit
3: Ask for edit
4: Ask for article
5: Find article content
6: Return article content
7: Return article content
8: Show edit page and wait for change
9: change title\tag\category\content
10: click submit
11: check for format:correct
12: transfer new content
13: ask for change
14: Change article
15: return success signal
16: return success
17: show"edit successfully" and return to "ManageArticle"page

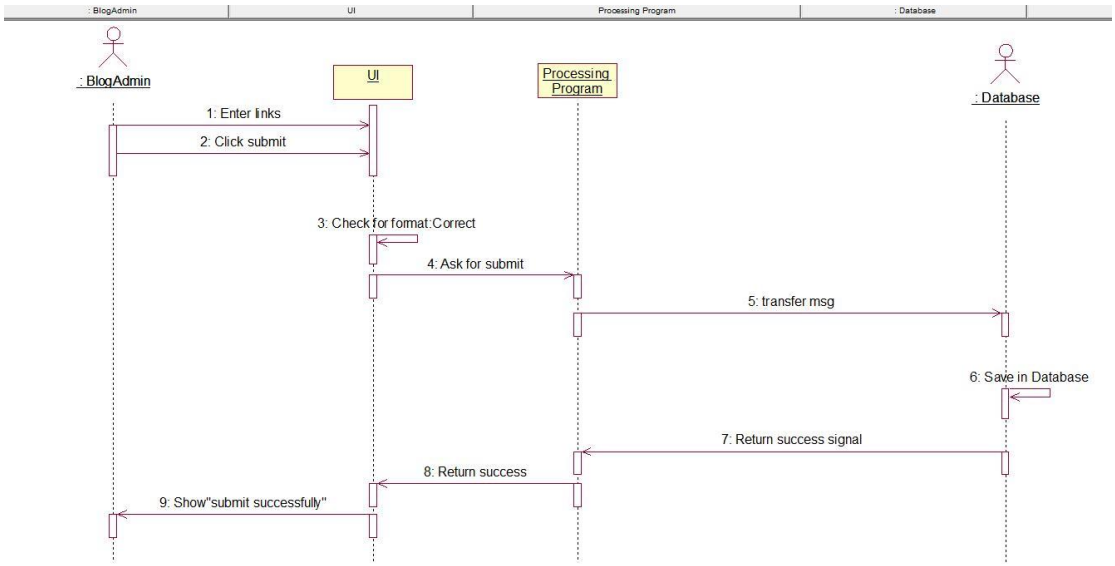

: BlogAdmin
UI
Processing Program
: Database
1: choose articles
2: Click edit
3: Ask for edit
4: Ask for article
5: Find article content
6: Return article content
7: Return article content
8: Show edit page and wait for change
9: change title\tag\category\content
10: click submit
11: check for format:Wrong
12: show"WrongFormat"and return to edit page
13: Edit again or click quit

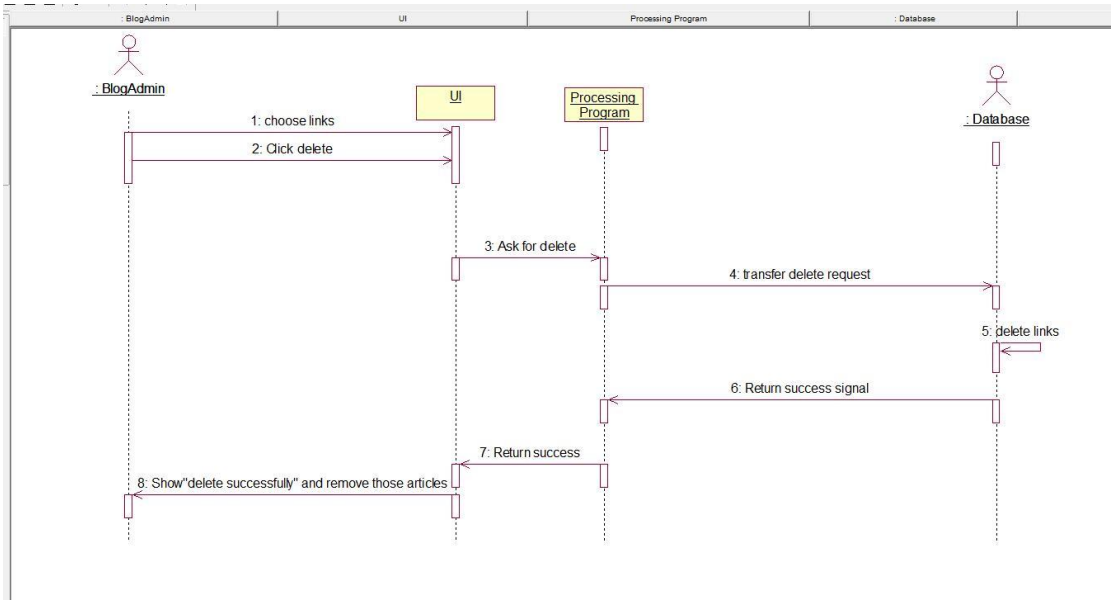
状态图：

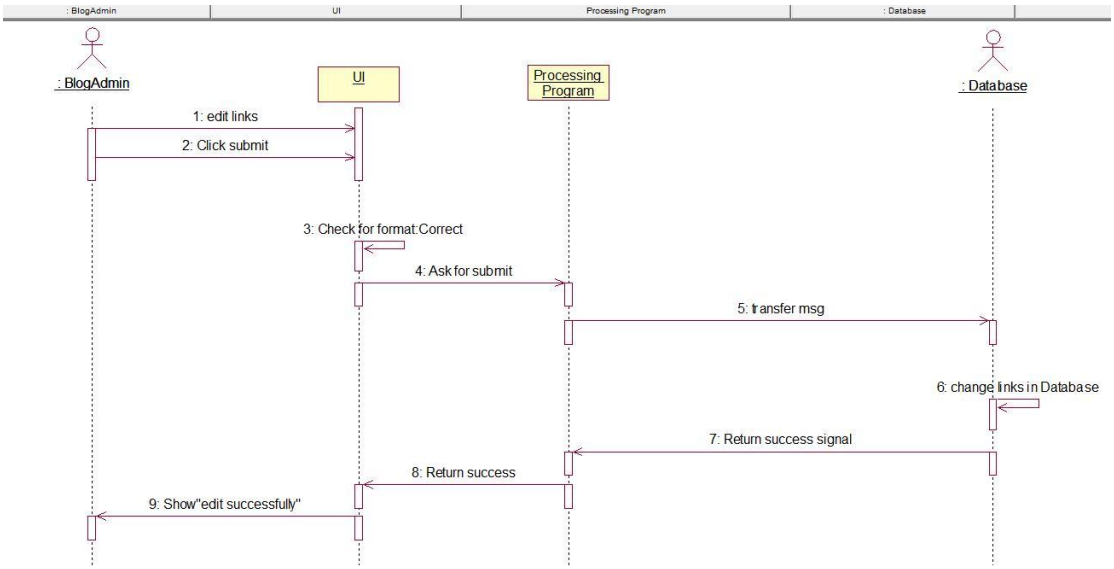

进入文章管理界面
等待选择
选择删除文章
向服务器
请求删除
服务器发回成功反馈
获得服务器
成功反馈
选择编辑文章
向服务器请求获
取文章
服务器发回反馈
获得文章
编辑文章
点击提交
上传文章
返回编辑框
格式错误
提示格式
错误

3.1.5 链接管理

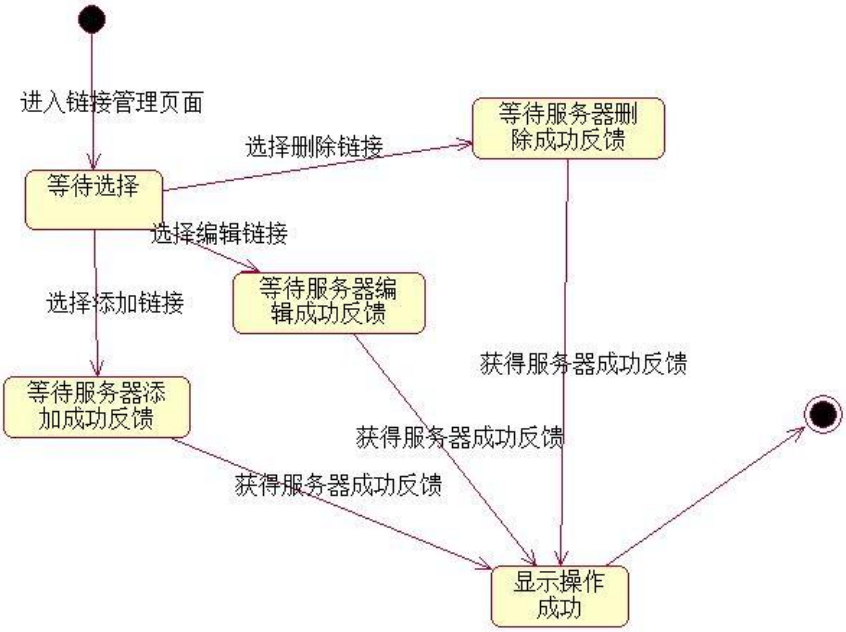
时序图：


: BlogAdmin
UI
Processing Program
: Database
: BlogAdmin
UI
Processing
Program
: Database
1: Enter links
2: Click submit
3: Check for format:Correct
4: Ask for submit
5: transfer msg
6: Save in Database
7: Return success signal
8: Return success
9: Show"submit successfully"


: BlogAdmin
UI
Processing Program
: Database
1: choose links
2: Click delete
3: Ask for delete
4: transfer delete request
5: delete links
6: Return success signal
7: Return success
8: Show"delete successfully" and remove those articles

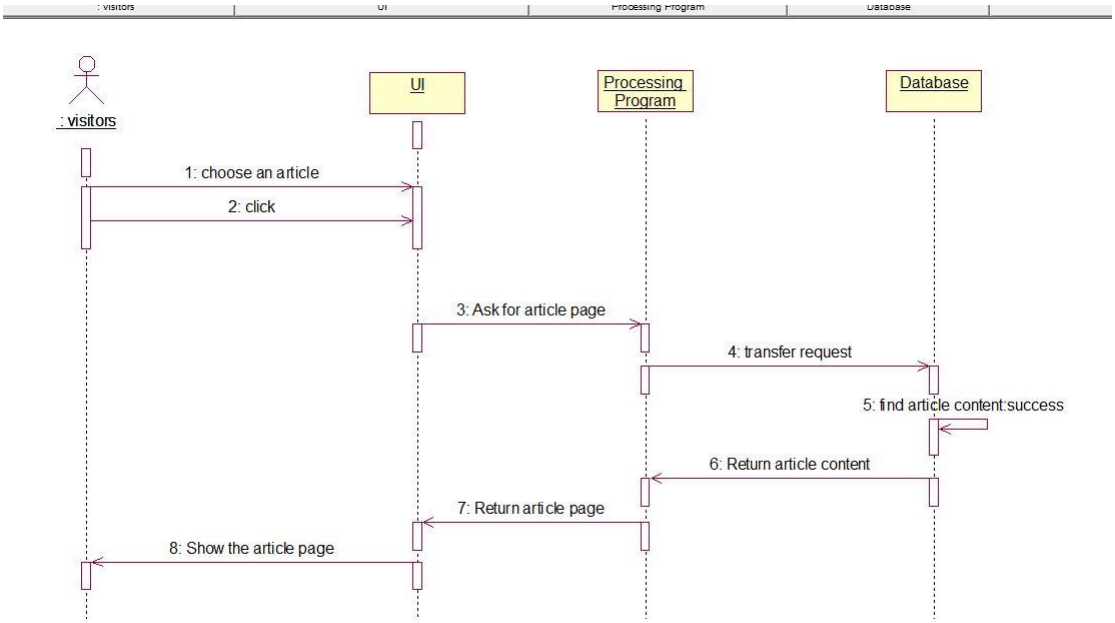

: BlogAdmin
UI
Processing Program
: Database
1: edit links
2: Click submit
3: Check for format:Correct
4: Ask for submit
5: transfer msg
6: change links in Database
7: Return success signal
8: Return success
9: Show"edit successfully"

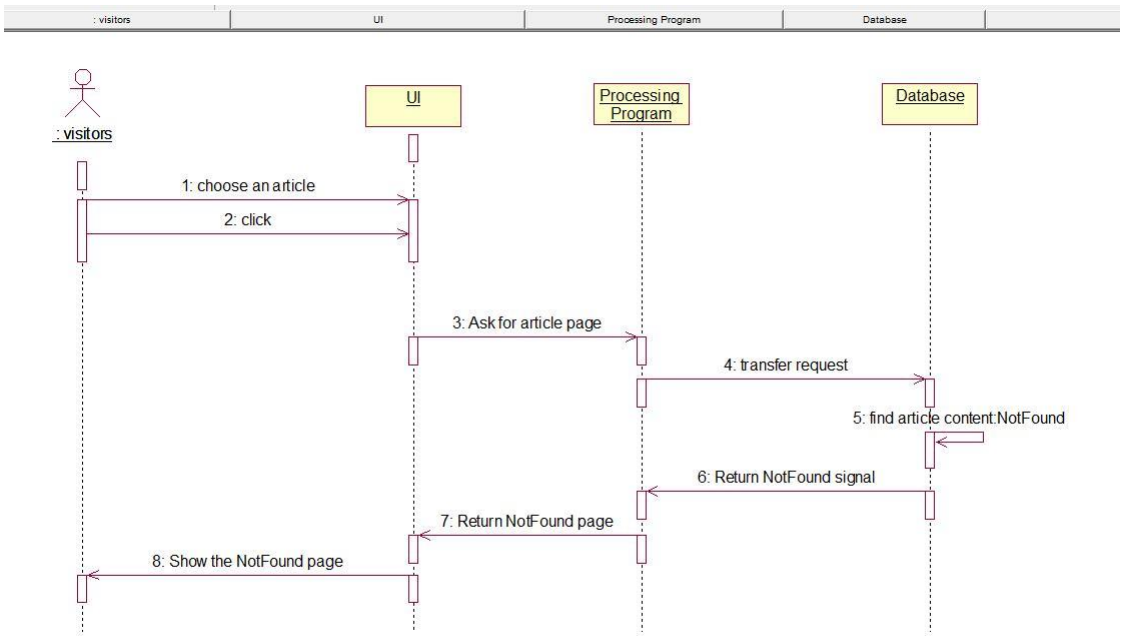
状态图：


进入链接管理页面
等待选择
选择删除链接
等待服务器删除成功反馈
选择编辑链接
等待服务器编辑成功反馈
选择添加链接
等待服务器添加成功反馈
获得服务器成功反馈
获得服务器成功反馈
获得服务器成功反馈
显示操作成功

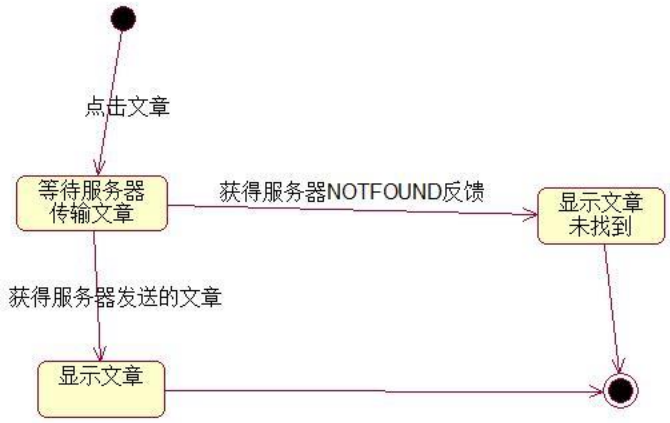
3.1.6 阅读文章

时序图


: visitors
UI
Processing Program
Database
1: choose an article
2: click
3: Ask for article page
4: transfer request
5: find article content:success
6: Return article content
7: Return article page
8: Show the article page

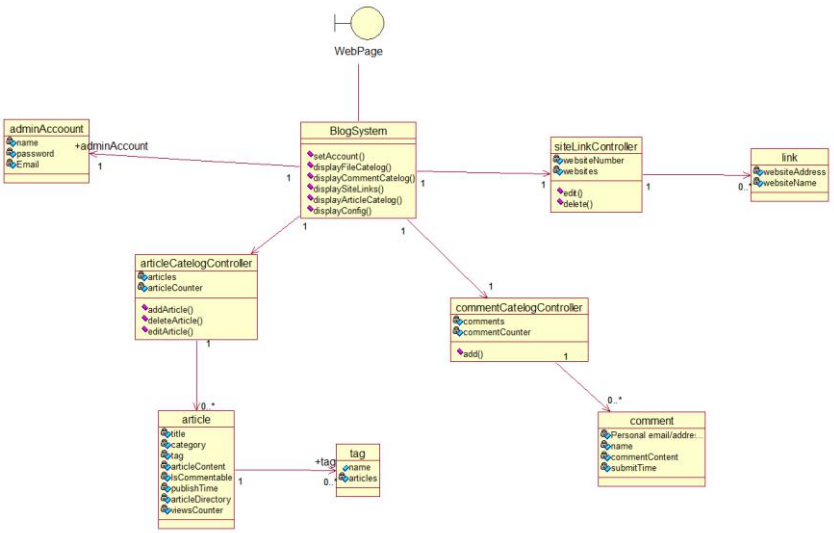


状态图：



3.2 总体结构

总体设计类图



4. 接口设计

4.1 外部接口

1. USB 接口，连接鼠标等设备。
2. DVI 接口，DVI(Digital Visual Interface，数字视频接口)，它是 D-SUB 接口的继承者，用来传输数字信号。
3. SCSI 接口，SCSI(Small Computer System Interface，支持包括磁盘驱动器、磁带机、光驱、扫描仪在内的多种设备。

4.2 内部接口

通过面向对象语言设计类，在 public 类中实现调用；类间实现严格封装

5. 数据结构设计

5.1 逻辑结构设计

文章字典数据类型

属性名	存储代码	类型	备注
编号	Id	Int	数据库设置为自增
标题	Titile	Varchar	非空

创建日期	Create_date	Date	存储创建日期
修改日期	Modified_date	Date	储存最后修改日期
内容	Content	File	文章较大以文件形式存储
标签	Catefories	Varchar	储存文件的分类
评论数	Comments_num	Int	

评论表字典数据类型

属性名	存储代码	类型	备注
编号	Id	Int	自增
评论日期	Create_date	Date	非空
评论者	Author	Varchar	非空
评论者编号	Author_id	Int	
评论内容	Content	Text	

用户字典数据类型

属性名	存储代码	类型	备注
用户编号	User_id	Int	
用户名	Username	Varchar	
登录密码	Password	Varchar	
用户邮箱	Email	Varchar	
用户主页地址	Home_url	varchar	

5.2 物理结构设计

文章字典数据类型

```
public String getId() { return id; }
public void setId(String id) { this.id = id; }
public String getTitle() { return title; }
public void setTitle(String title) {...}
public Date getCreateDate() { return createDate; }
public void setCreateDate(Date createDate) { this.createDate = createDate; }
public Date getModifiedDate() { return modifiedDate; }
public void setModifiedDate(Date modifiedDate) { this.modifiedDate = modifiedDate; }
public File getContent() { return content; }
public void setContent(File content) { this.content = content; }
public String getAuthorId() { return authorId; }
public void setAuthorId(String authorId) { this.authorId = authorId; }
public String getCatefories() { return Catefories; }
public void setCatefories(String catefories) { Catefories = catefories; }
public int getCommentsNum() { return commentsNum; }
public void setCommentsNum(int commentsNum) { this.commentsNum = commentsNum; }
```

评论字典数据类型

```
public int getId() { return id; }

public void setId(int id) { this.id = id; }

public Date getCreateDate() { return createDate; }

public void setCreateDate(Date createDate) { this.createDate = createDate; }

public String getAuthor() { return author; }

public void setAuthor(String author) { this.author = author; }

public int getAuthorId() { return authorId; }

public void setAuthorId(int authorId) { this.authorId = authorId; }

public int getOwnerId() { return ownerId; }

public void setOwnerId(int ownerId) { this.ownerId = ownerId; }

public Text getContent() { return content; }

public void setContent(Text content) { this.content = content; }
```

关系字典数据结构

```
public String usrId;
public String relId;

public String getUsrId() {
    return usrId;
}

public void setUsrId(String usrId) {
    this.usrId = usrId;
}

public String getRelId() {
    return relId;
}

public void setRelId(String relId) {
    this.relId = relId;
}
```

用户字典数据结构

```
public String userId;
public String username;
public String password;
public URL emailURl;
public URL homeURL;

public String getUserId() { return userId; }

public void setUserId(String userId) { this.userId = userId; }

public String getUsername() { return username; }

public void setUsername(String username) { this.username = username; }

public String getPassword() { return password; }

public void setPassword(String password) { this.password = password; }

public URL getEmailURl() { return emailURl; }

public void setEmailURl(URL emailURl) { this.emailURl = emailURl; }

public URL getHomeURL() { return homeURL; }

public void setHomeURL(URL homeURL) { this.homeURL = homeURL; }
```

5.3 数据结构与程序的关系

数据结构与程序是软件的重要组成部分，程序的正确执行依赖于合理的数据结构

6. 运行设计

6.1 运行模块的组合

本程序主要是以一个窗口为模块，一般一个窗口完成一个特定的功能，主窗口通过打开另一个子窗口来实现个模块之间不同功能的连接和组合。各模块之间相对独立，程序的可移植性好。各模块之间主要以传递数据项的引用来实现模块之间的合作和数据共享，不同的窗口之间实现使用 session 以及 cookie 来进行数据的共享与传递。

6.2 运行控制

运行控制将严格按照各模块间的函数调用关系来实现。

在网络传输方面，客户机再发送数据后，将等待服务器的确认到信号，收到后，在此等待服务器发送数据，然后对数据进行确认，服务器再接收到数据后发送确认信号，在对数据处理、访问数据库后，将饭返回信息送回客户机，并等待确认。采用 http 协议进行数据的传输，后台采用 jdbc 对数据库进行使用

6.3 运行时间

各模块运行时间不定，这也跟用户的操作以及数据的大小有关

7. 出错处理设计

7.1 出错输出信息

错误类型	原因	解决办法
数据库连接错误	数据库设置不正确或数据库异常	取消操作，并重新尝试
数据库查询错误	没有按照要求进行操作	取消操作，并重新尝试
输入错误	输入格式不规范	重新按照标准输入
不可预知错误	未知异常	暂无
其他操作错误	用户的不当操作使程序发生错误	终止操作,并提醒用户正确的操作方式

7.2 出错处理对策

我们对于本程序的几种可能的错误进行了分析，分别进行了不同的处理。 主要的错误可能有：

数据库连接错误：这类错误主要是数据库设置不正确，或 SQL Server 异常引起的，我们只要取消本次操作，提醒用户检查数据库问题就可。

数据库查询错误：这类错误主要是用户没有按照，参数进行输入，我们中断该操作，并提示用户按照标准进行操作即可。

输入错误：这主要是用户输入不规范造成的，我们在尽量减少用户出错的条件的情况下，主要也是通过对话框，提醒用户，然后再次操作。

其他操作错误：对于用户的不正当操作，有可能使程序发生错误。我们主要是中止操作，并提醒用户中止的原因和操作的规范。

其他不可预知的错误：程序也会有一些我们无法预知或没考虑完全的错误，我们对此不可能作出安全的异常处理，这时我们主要要保证数据的安全，所以要经常的进行数据库备份，并能及时的和我们联系，以逐步的完善我们的程序

8. 安全保密设计

1. 数据库只为特定的用户开放，其他用户无法访问
2. 权限控制，根据不同用户角色，设置相应权限，用户的重要操作都做相应的日志记

录以备查看，没有权限的用户禁止使用系统。

3. 重要数据加密，本系统对一些重要的数据按一定的算法进行加密，如用户口令、重要参数等。
4. 由于个人博客系统是属于个人的一片领域因此该博客系统的安全性要达到一定的标准，服务器性能良好
5. 允许博主设置是否允许评论等操作。

9.维护设计

1. 采用 github 进行版本控制，可以进行版本的更新还原；
2. 利用好 github 平台进行，项目的管理开发，以及维护