1 概述

1.1 系统简述

本系统旨在为程序员提供一个简洁的代码管理平台。“CodeHub”是一个基于分布式版本控制系统，供用户进行代码托管的管理网站。在该项目中，用户打开相应网页，注册登录后便可获得一个可供上传代码的独立空间。使用者在上传相应代码后，可以创建分支并进行管理，以开展协同工作。

1.2 软件设计目标

软件希望能为用户提供快捷方便的文件版本管理。用户可在登入模块中实现注册和登录，在仓库管理模块中实现项目创建，文件上传，文件修改，创建分支，合并分支，下载文件，添加成员，删除文件的功能。项目要求网站简单易用，支持50位用户并发操作，系统处理不超过4秒，并能在日后简易快速对服务器进行升级。

1.3 参考资料

github官方文档

码云

django官方文档

python教程

pygit官方文档

1.4 修订版本记录

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 17.11.1 | v1.0 | 详细设计初稿 | 黄子威、周昭育 |
| 17.11.10 | v1.1 | 细化详细设计，添加了用例 | 黄子威、周昭育 |
| 17.11.20 | v1.2 | 完善了术语表 | 黄子威 |
| 17.12.3 | v2.0 | 加入了非功能性需求 |  |

2 术语表

对本文档中所使用的各种术语进行说明。如果一些术语在需求规格说明书中已经说明过了，此处不用再重复，可以指引读者参考需求说明。

• 3 设计概述

3.1 系统的复用计划：库、框架、模式、构件等方面的复用

1.文件操作的复用

预先封装好针对一般仓库的文件操作，包括创建文件、上传文件、修改文件、删除文件和版本回退，系统可以直接调用文件模块进行直接快速的文件操作。文件模块和系统解耦，加强了系统的内聚性，提高了系统的可修改性，为以后的系统维护和修改提供便捷。

3.2系统接口设计

• 各种提供给用户的界面

1.登录界面

注册按钮：点击后跳转到注册界面

忘记密码按钮：点击后将会跳转到账号管理界面

登录界面按钮：将输入的用户名和密码发送到服务器，并等待服务器的反馈信息

2.注册界面

登录按钮：跳转到登录按钮

注册按钮：首先核对输入信息的合法性，如果合法，则把信息发送到服务器，服务器端成功写入该账号信息后，会跳转到登录界面，如果注册失败，则提示错误信息。

3.项目界面

退出按钮：跳转到登录界面

Usernam按钮：跳转到用户界面

Project按钮：跳转到对应目录

代码按钮：显示当前目录下的可见文件

Master下拉框：选择分支

项目的文件按钮：点击会打开文件

Newfile按钮：创建新文件，点击会跳转到创建文件按钮

新建按钮：跳转到新建文件按钮

上传按钮：上传文件到当前仓库目录

下载按钮：下载本仓库到本地

4.提交记录查看界面

提交记录按钮：查看该项目目前为止的所有提交记录

5.分支查看界面

新建分支按钮：新建分支

下载按钮：点击后下载对应分支

合并分支按钮：将两个分支合并，并刷新界面显示合并后的仓库

6.项目成员查看界面

项目成员按钮：查看当前项目的项目成员

添加成员按钮：点击后将会查找该用户，查找成功则添加到项目里，失败则返回错误信息

•系统内部接口设计：各子系统、各模块间的接口设计

1. 文件操作模块的接口

文件模块以函数形式提供方便的接口给系统调用，调用成功将会自动完成服务器上的文件操作

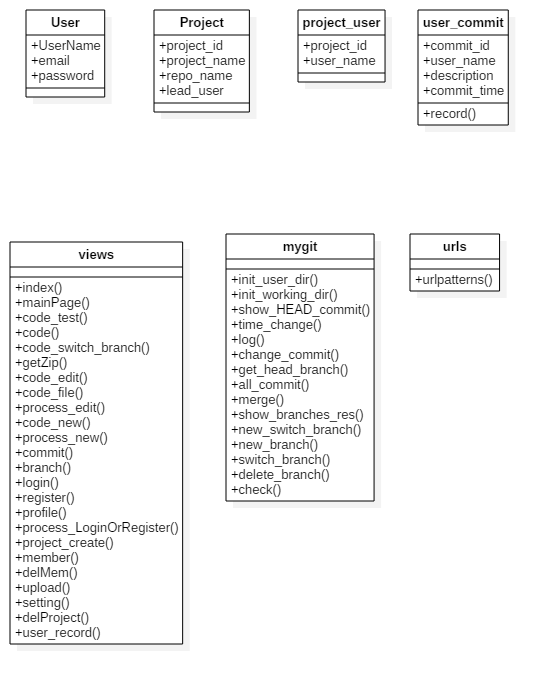
2. 权限模块的接口

所有初始化数据仓库的操作执行前，都会先通过权限模块的接口导入，权限模块会阻止超越权限的操作。

3.3 对象模型设计

• 对象模型（类图或对象图）：提供整个系统的对象模型，在其中应该包含

所有的系统对象。所有对象之间的关联必须被确定并且必 须指明联系的基数。



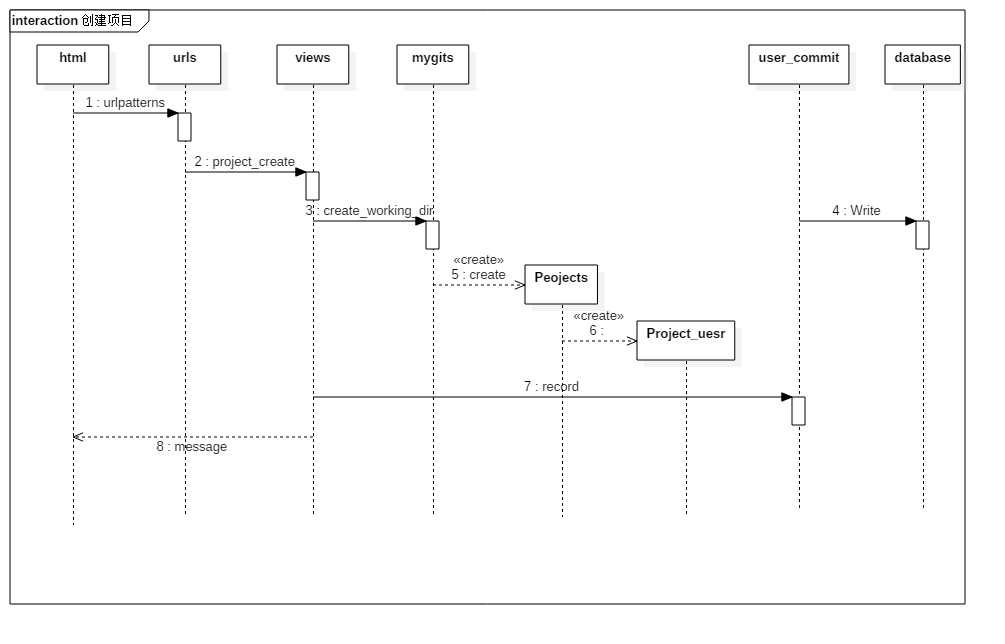
• 对象描述: 在这个部分叙述每个对象的细节，它的属性、

它的方法。对每个对象的每个属性详细说明：名字、类 型; 对每个对象的每个方法详细说明：方法名，返回类

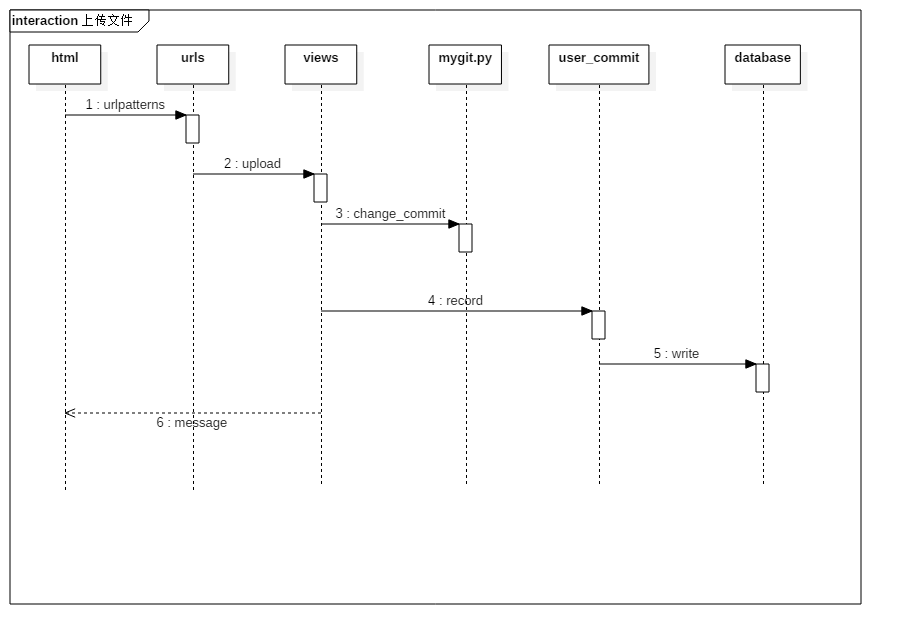
型，返回值，参数，用途以及使用的算法的简要说明。

3.4系统功能（用例）实现详细设计

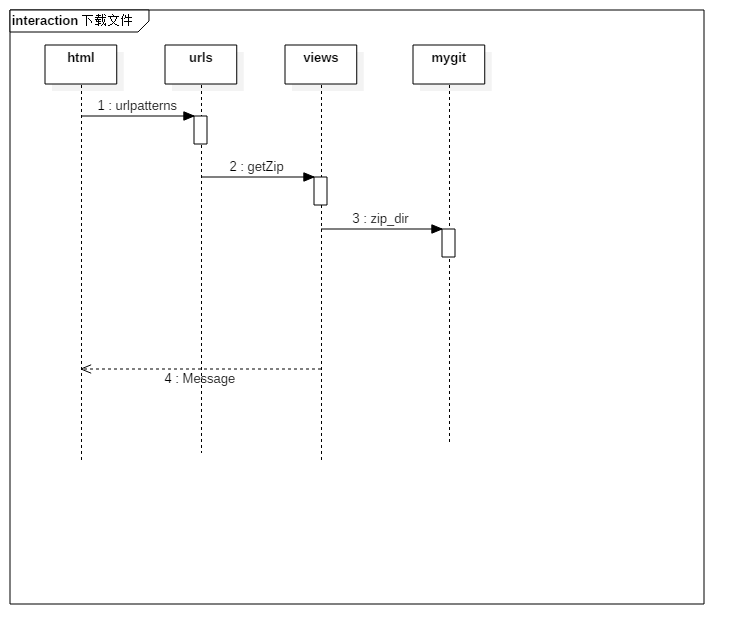
3.4.1创建项目



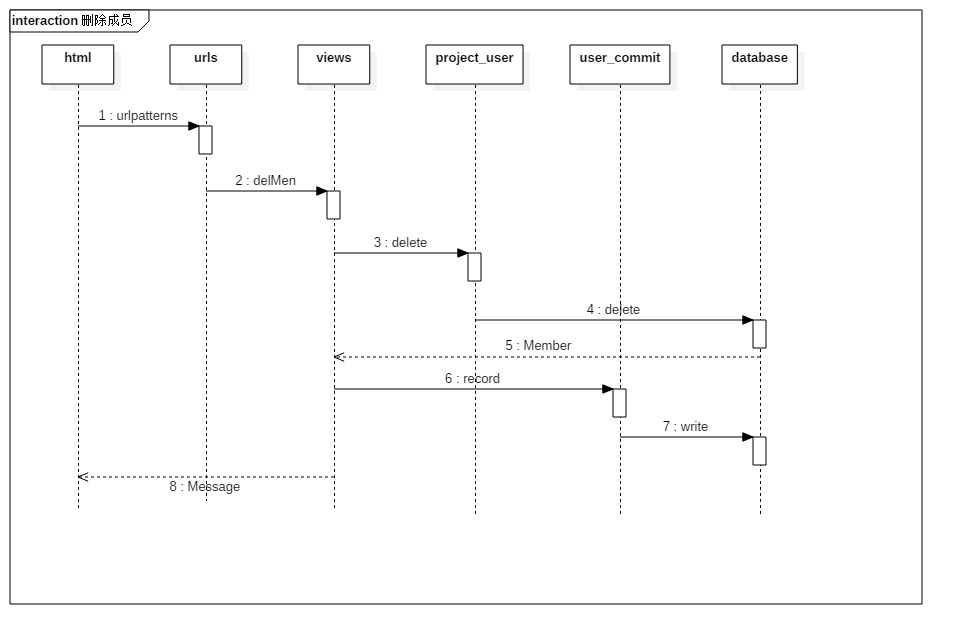
3.4.2 上传文件



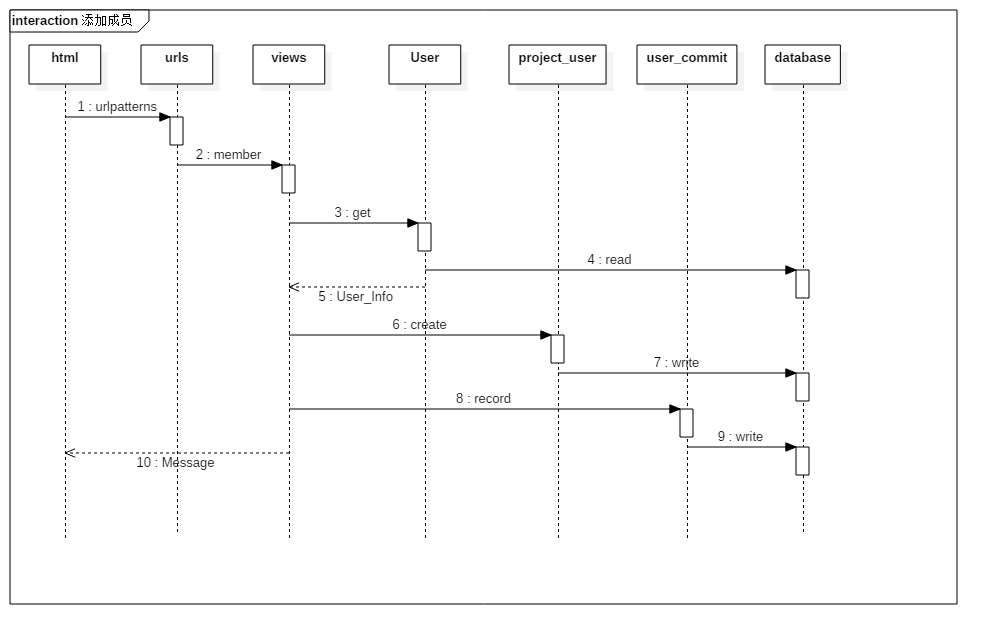
3.4.3下载文件



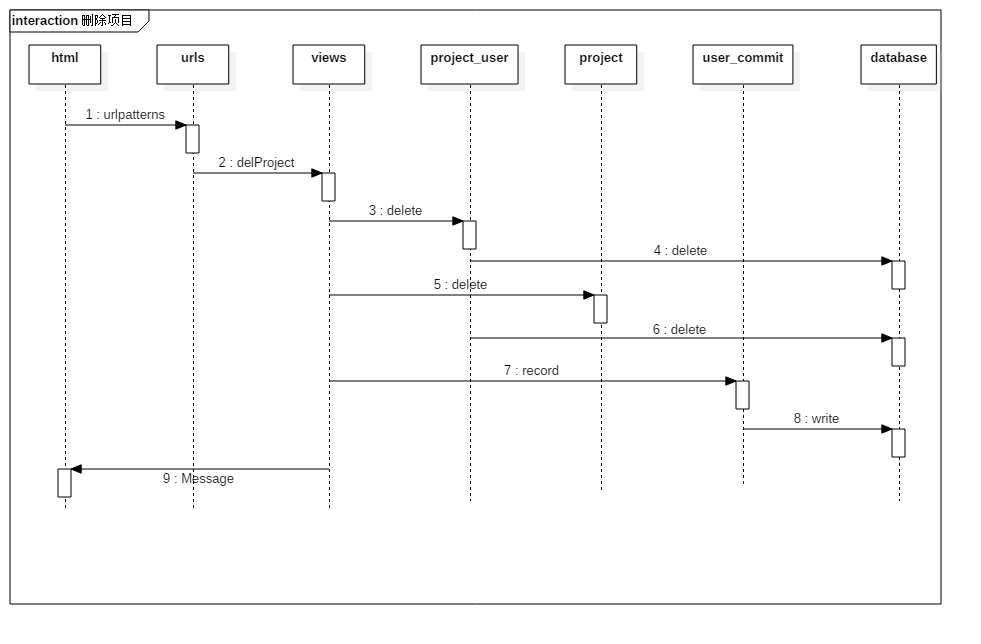
3.4.4删除成员



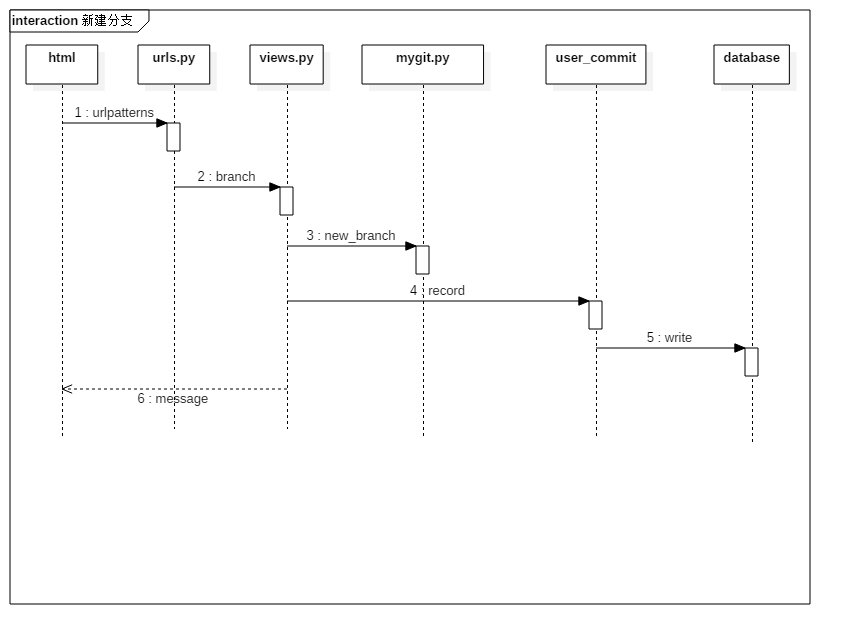
3.4.5添加成员



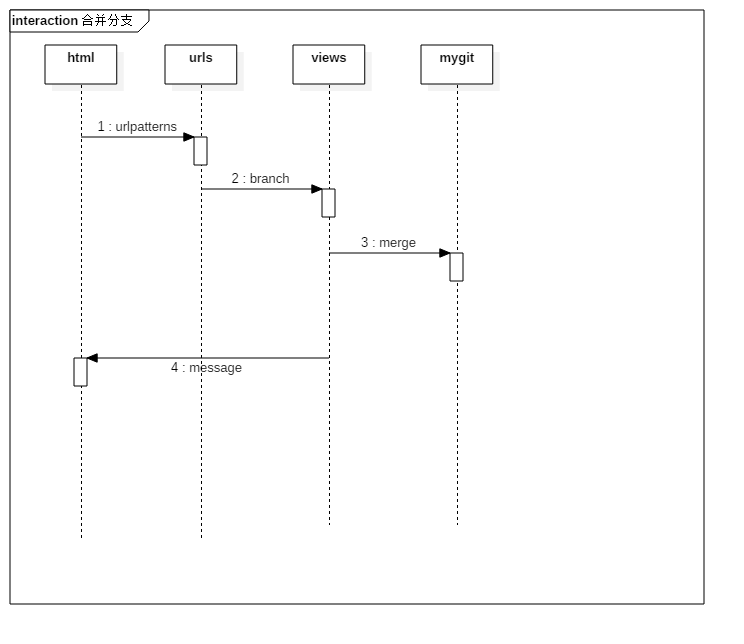
3.4.6删除项目



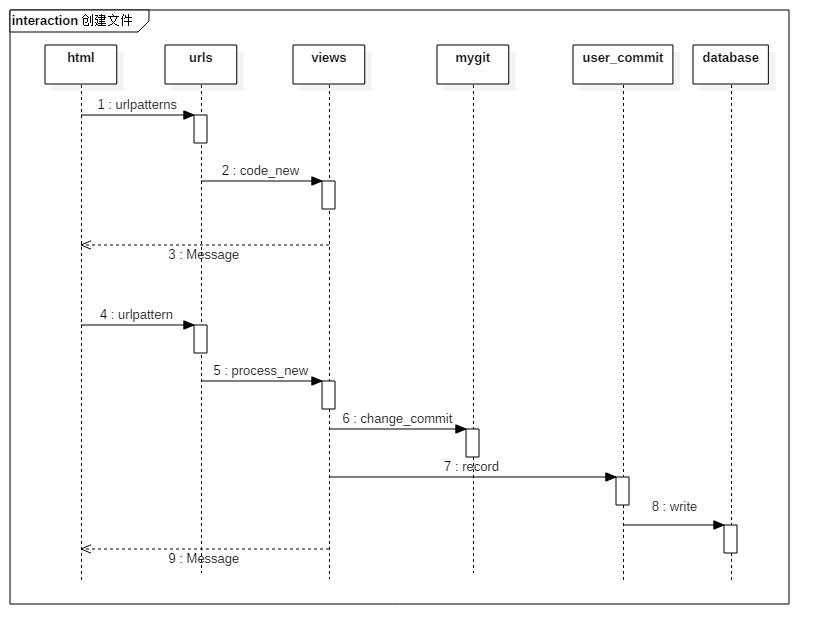
3.4.7切换分支



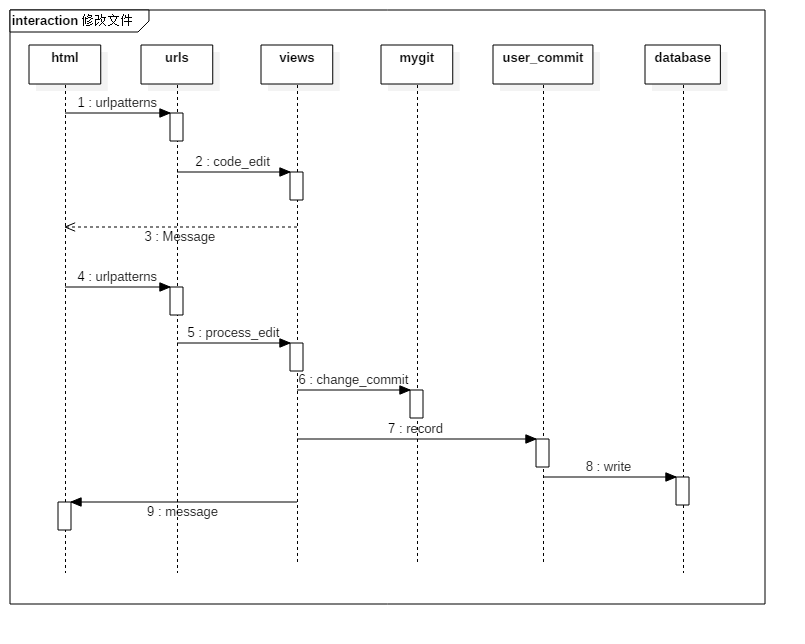
3.4.8 合并分支



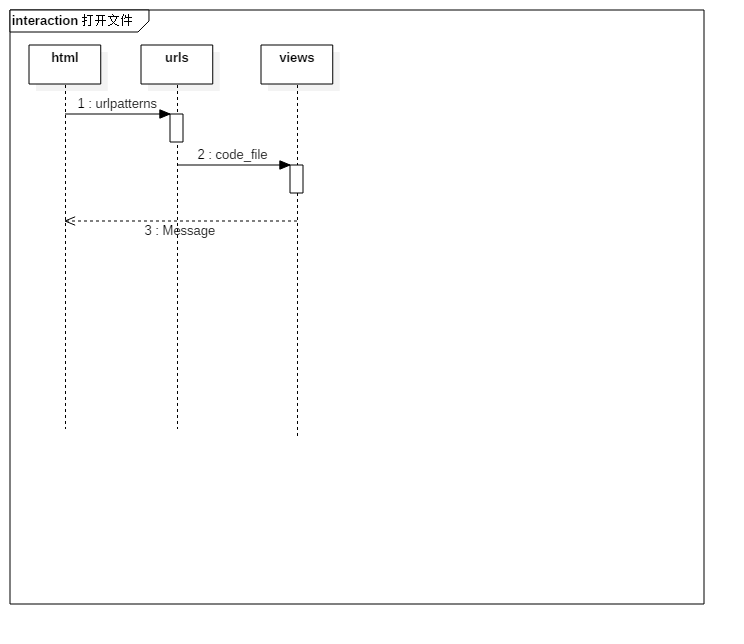
3.4.9创建文件



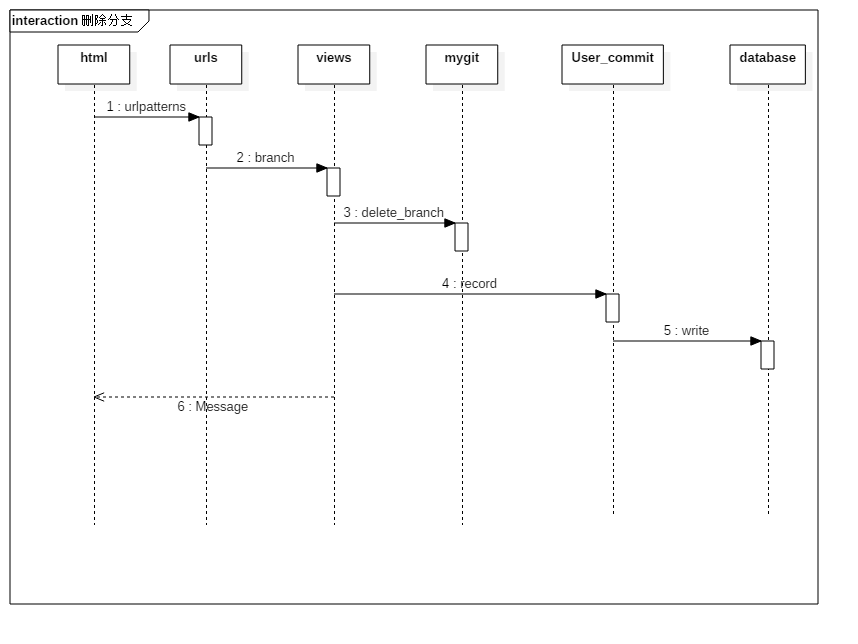
3.4.10修改文件



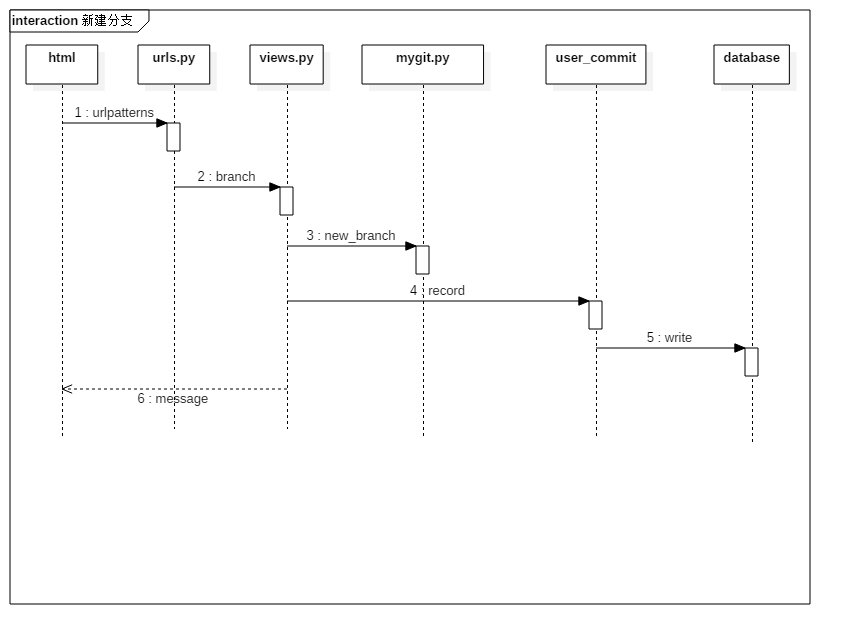
3.4.11 打开文件



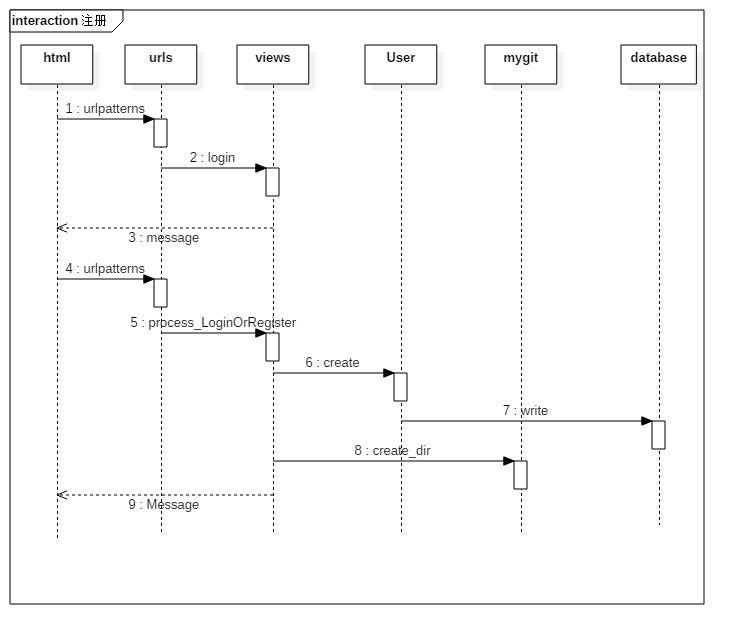
3.4.12删除分支



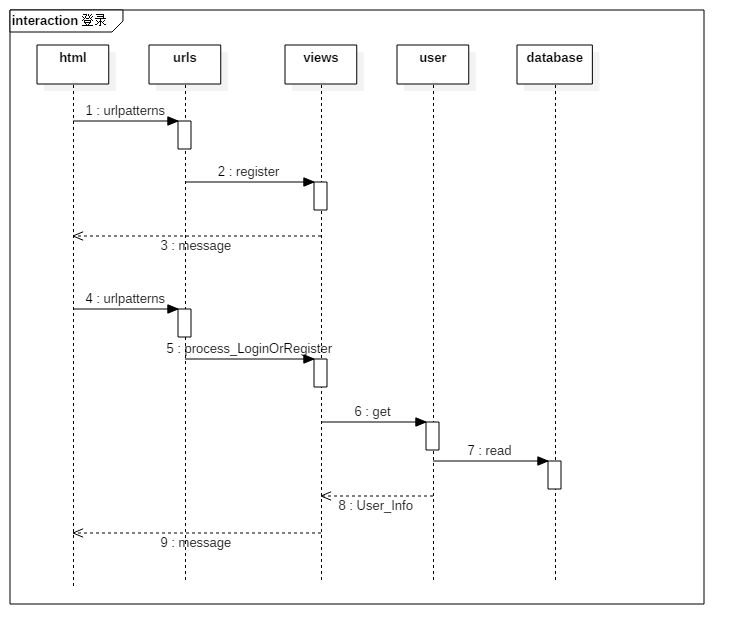
3.4.13新建分支



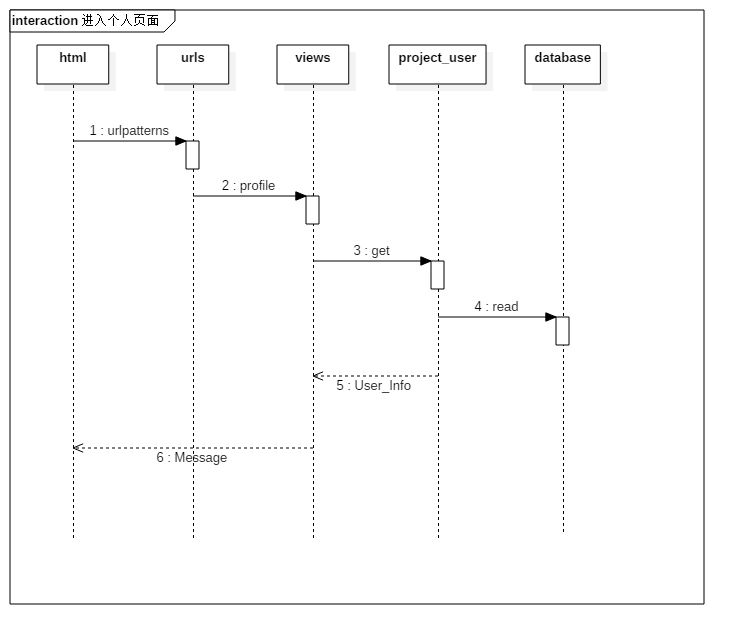
3.4.14 注册



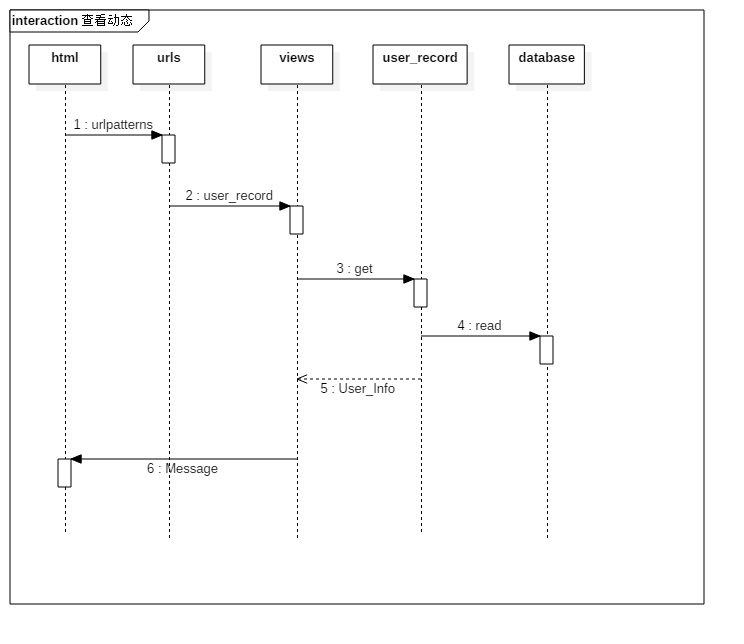
3.4.15登录



3.4.16进入个人页面



3.4.17查看动态



3.5系统非功能设计

•针对系统非功能需求进行的系统设计

1. 可用性

用户能够快速简易完成操作

用户能从错误中恢复

1. 性能需求

用户可以在能接受的时间内看到项目，进行修改操作时能够快速地提交结果

支持50个用户的并发操作

上传文件完成后系统处理不超过4秒

系统响应http网页请求不超过2秒

1. 可靠性

系统7\*24小时运行，全年持续运行故障时间不能超过10小时

传输文件失败提示用户错误信息并回滚

1. 可扩展性

当访问量增多，系统能够快速简易地添加设备（数据库、服务器）进行必要的扩展

5．易用性

接口清晰，操作简单，产品学习曲线低，用户容易上手

用户易用满意度超过90%