

**基于Flask框架的实验报告提交系统需求分析报告**

|  |  |
| --- | --- |
| **名称：** | **基于Flask框架的实验报告提交系统** |
| **学号：** | **2014211483** |
| **学院名称：** | **计算机科学与技术学院** |
| **班 级：** | **0491401** |
| **姓名：** | **张宝宇** |
| **指导教师：** | **刘锐** |

**目录**

[1.Introduction 简介 1](#_Toc508976923)

[1.1 Purpose 目的 1](#_Toc508976924)

[1.2 Scope 范围 1](#_Toc508976925)

[2.general statement总体概述 2](#_Toc508976926)

[2.1 Software Overview软件概述 2](#_Toc508976927)

[2.2 Software features 软件功能 3](#_Toc508976928)

[2.3 User characteristics 用户特征 3](#_Toc508976929)

[2.4 Assumptions and dependencies 假设和依赖关系 3](#_Toc508976930)

[3.Requirements Modeling需求建模 3](#_Toc508976931)

[4.Specific needs具体需求 7](#_Toc508976932)

[4.1 Functional requirements功能需求 7](#_Toc508976933)

[4.1.1 注册/登录 7](#_Toc508976934)

[4.1.2 信息添加功能 7](#_Toc508976935)

[4.1.3报告管理 8](#_Toc508976936)

[4.1.4 课程添加 8](#_Toc508976937)

[4.1.5 报告提交 8](#_Toc508976938)

[4.1.6 用户管理 9](#_Toc508976939)

[4.2 Performance requirements性能需求 9](#_Toc508976940)

[4.2.1 Static quantitative requirements静态的量化需求 9](#_Toc508976941)

[4.2.2 Dynamic quantitative requirements动态的量化需求 9](#_Toc508976942)

[5. Software quality attributes软件质量属性 10](#_Toc508976943)

[5.1 Maintainability可维护性 10](#_Toc508976944)

[5.2 Reliability可靠性 10](#_Toc508976945)

[5.3 Security安全性 10](#_Toc508976946)

[5.4 Portability可移植性 10](#_Toc508976947)

[5.5 Usability易用性 10](#_Toc508976948)

[~~6. Other requirements其他需求~~ 10](#_Toc508976949)

[~~7. Summary 需求分析小结~~ 10](#_Toc508976950)

# 1.Introduction 简介

## 1.1 Purpose 目的

当今高校为发展学生的动手能力会开展各式各样的实验课，老师在是依然采用学习委员收集实验报告，手动评分等传统上课方式。这往往会造成收交混乱，不及时，报告资料保存不方便等问题的发生，为老师，学校，学生都带来了不便。所以一个较为成熟的实验报告提交系统就十分有开发价值与使用价值。

本需求分析报告是基于Flask框架的实验报告提交系统而编写的需求方面的分析说明性文档，旨在方便研究基于Flask框架的实验报告提交系统的开发途径和应用方法。同时它也是进行项目策划、概要设计和详细设计的基础，是维护人员进行内部维护，信息更新，验收和测试的依据。

本文档预期读者是是基于Flask框架的实验报告提交系统开发人员，支持本项目的老师。

## 1.2 Scope 范围

本项目的名称是基于Flask框架的实验报告提交系统，要实现的就是更加方便用户在实验报告的收发，增强实用性和和便捷性，力求为用户达到其预期的效果。

网站主要应用于上实验课的老师和同学的pc电脑上。

# 2.general statement总体概述

## 2.1 Software Overview软件概述

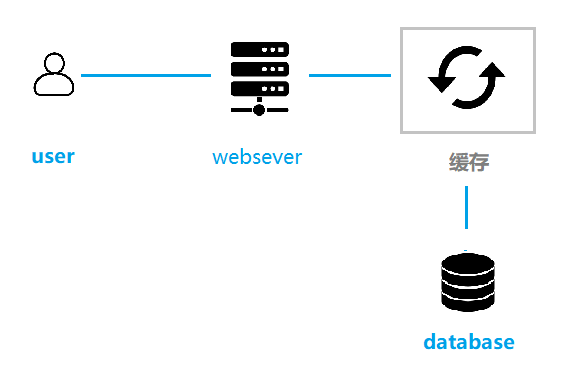


图1.1 系统总体设计图

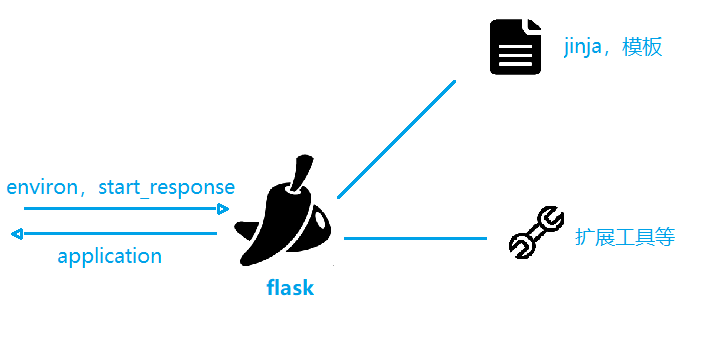


图1.2 Webserver设计图

本系统从两个层面来看，首先从整体来看，用户访问服务器提交请求，服务器更具请求到数据库中读取，计划在数据库之上设计一个缓存层，服务器先访问缓存中是否有需要的数据，如果有并且不需要更新就将响应（一般为html文件）返回给用户，如果没有再去数据库查找。

缓存层使用Redis，数据库使用MySQL或者PostgreSQL。

接下来从Webserver的层面看，使用Web框架flask将不同Web应用程序中的共性部分给抽象出来，提供一系列通用的接口，从而避免开发者做重复性工作，让其将精力放在业务相关的实现上，使用jinja插件作为模板/渲染工具，其他的使用Bootstrap作为页面设计工具等。

## 2.2 Software features 软件功能

注册与登录功能，老师可以教师编号注册，从excel文件导入学生名单，学生用户由管理员导入名单注册，直接登陆使用。

* 教师
  1. 信息添加功能，教师用户可以添加自己所教授的课程信息与学生名单。
  2. 报告管理功能，教师用户可以发布报告任务，查看报告的提交情况。
* 学生
  1. 课程添加功能，添加自己所修的课程。
  2. 报告提交功能，提交自己的实验报告。
* 系统管理员
  1. 用户管理功能，添加或者删除用户。

## 2.3 User characteristics 用户特征

本软件的使用对象是上实验课的老师和同学。

## 2.4 Assumptions and dependencies 假设和依赖关系

* 本项目的服务器依赖Python运行环境。

# 3.Requirements Modeling需求建模

* 实体-联系图

（1）教师实体E-R模型设计



图2.1 教师E-R图

（2）学生实体E-R模型设计



图2.2 学生E-R图

（3）报告实体E-R模型设计

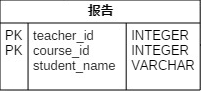


图2.3 报告E-R图

（4）课程实体E-R模型设计



图2.4 课程E-R图

（5）学生与课程关系-上课E-R模型设计



图2.5 上课E-R图

（6）教师与课程关系-授课E-R模型设计



图2.6 授课E-R图

（5）总体E-R模型设计

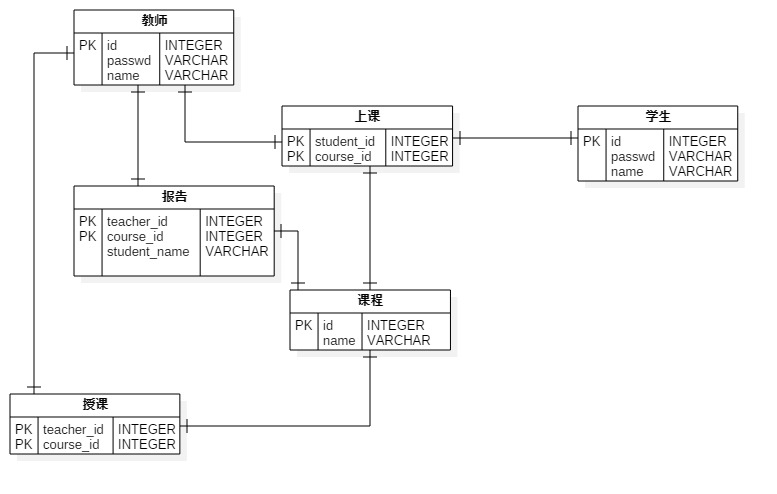


图2.7 总体E-R图

* 数据流图

图3.1至图3.4所示的数据流图主要反映的是基于Flask框架的实验报告提交系统中各功能模块之间的数据流程和简单的数据处理、加工过程和方法，也为系统的后续设计打下了基础。



图3.1 顶层数据流图

* 状态图

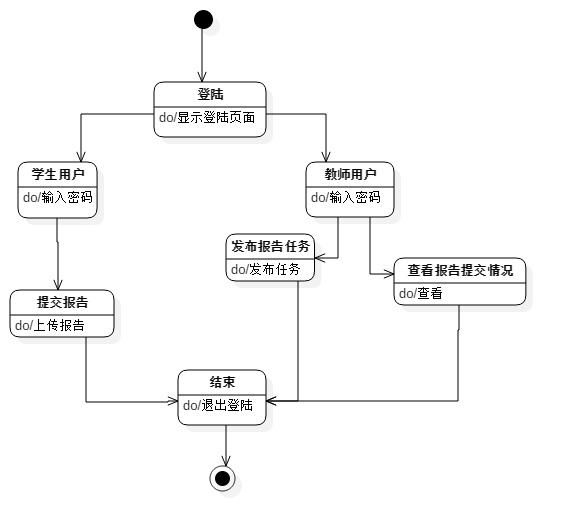


图4.1 状态图

* UML用例图

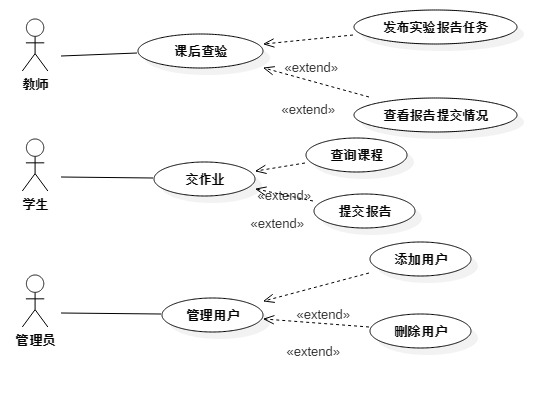


图5.1 UML用例图

# 4.Specific needs具体需求

## 4.1 Functional requirements功能需求

### 4.1.1 注册/登录

1.Introduction 介绍

教师可以使用教师编号注册，从excel文件导入学生名单，学生用户由管理员导入名单注册，直接登陆使用。

2.Inputs 输入

教师输入注册用教师编号与密码，学生输入学号与密码。

3.Process 处理

系统验证教师编号的真实性，学生账户的正确性，分别显示教师与学生用户界面。

4.Output 输出

教师使用界面与学生使用界面。

### 4.1.2 信息添加功能

1.Introduction 介绍

教师用户可以添加自己所教授的实验课程信息与学生名单。

2.Inputs 输入

实验课名称，上传附件上课学生名单（Excel文件）

3.Process 处理

服务器获取到上传文件后，读取学生名单，将信息添加近数据库中，将课程信息与教师信息添加近数据库中。

4.Output 输出

添加成功（上传成功）。

### 4.1.3报告管理

1.Introduction 介绍

教师用户可以发布报告任务，查看报告的提交情况。

2.Inputs 输入

输入报告任务。

3.Process 处理

将输入的报告任务存入数据库。

4.Output 输出

发布成功

### 4.1.4 课程添加

1.Introduction 介绍

学生用户添加自己所修的课程。

2.Inputs 输入

点击拖动歌曲以调整播放顺序。

3.Process 处理

客户端获取用户改动信息后，将改动信息发送给后台服务器，并刷新列表显示页面。

4.Output 输出

修改过的歌曲列表。

### 4.1.5 报告提交

1.Introduction 介绍

学生用户根据任务提交自己的实验报告。

2.Inputs 输入

上传附件。

3.Process 处理

系统将报告存入服务器中，并将成功的信息天机进数据库。

4.Output 输出

提交成功。

### 4.1.6 用户管理

1.Introduction 介绍

系统管理员添加或者删除用户。

2.Inputs 输入

删除/添加用户。

3.Process 处理

4.Output 输出

删除/添加成功。

## 4.2 Performance requirements性能需求

### 4.2.1 Static quantitative requirements静态的量化需求

a.支持的访问数为n台 (n>=3)

b.处理n个文件及n条记录(n>=100)

c.网站访问最大并发量10000。

### 4.2.2 Dynamic quantitative requirements动态的量化需求

a.网络环境较好时，并发量达到10000。

b.网络环境较差时，并发量至少达到3000。

## 5. Software quality attributes软件质量属性

## 5.1 Maintainability可维护性

该系统开放源代码，采用GPLv2许可证发布源码，可维护性健壮。

## 5.2 Reliability可靠性

低并发量下可稳定运行。

## 5.3 Security安全性

该系统获取权限低，不涉及用户隐私与财产，安全性高。

## 5.4 Portability可移植性

该系统利用跨平台语言Python开发。系统采用B/S结构。整个系统所依赖的（类库）均采用跨平台开源类库。因此系统具有较高的可移植性。

## 5.5 Usability易用性

a.易懂性：界面区域名称易懂并且易于区分，用词准确，能达到望文知意。

b.易学性：会网站基本操作即可使用。

c.易操作性：以用户为中心的设计理念，友好的用户交互界面，易于操作。

# ~~6. Other requirements其他需求~~

# ~~7. Summary 需求分析小结~~