Flask框架研究扩展

软件项目计划书

**编写人：G组**

**2020年3月17日**

目录

[1. 引言 1](#_Toc35422264)

[1.1 背景 1](#_Toc35422265)

[1.2 定义 2](#_Toc35422266)

[1.3 参考资料 2](#_Toc35422267)

[1.4 标准、条约和约定 2](#_Toc35422268)

[2. 项目概述 3](#_Toc35422269)

[2.1 Flask微框架 3](#_Toc35422270)

[2.2 项目目标 4](#_Toc35422271)

[2.3 产品目标与范围 4](#_Toc35422272)

[2.4 工作内容 4](#_Toc35422273)

[2.5 应交付成果 5](#_Toc35422274)

[2.6 项目验收方式和依据 5](#_Toc35422275)

[3. 项目团队组织 5](#_Toc35422276)

[3.1 组织结构 5](#_Toc35422277)

[3.2 人员分工 6](#_Toc35422278)

[3.3 协作与沟通 6](#_Toc35422279)

[4. 实施计划 7](#_Toc35422280)

[4.1 总体进度计划 7](#_Toc35422281)

[4.2 项目控制计划 8](#_Toc35422282)

[5. 关键问题 8](#_Toc35422283)

[6. 专题计划要点 8](#_Toc35422284)

# 引言

## 背景

Flask是一个轻量级的可定制框架，使用Python语言编写，较其他同类型框架更为灵活、轻便、安全且容易上手。它可以很好地结合MVC模式进行开发，开发人员分工合作，小型团队在短时间内就可以完成功能丰富的中小型网站或Web服务的实现。另外，Flask还有很强的定制性，用户可以根据自己的需求来添加相应的功能，在保持核心功能简单的同时实现功能的丰富与扩展，其强大的插件库可以让用户实现个性化的网站定制，开发出功能强大的网站。

Flask是目前十分流行的web框架，采用Python编程语言来实现相关功能。它被称为微框架(microframework)，“微”并不是意味着把整个Web应用放入到一个Python文件，微框架中的“微”是指Flask旨在保持代码简洁且易于扩展，Flask框架的主要特征是核心构成比较简单，但具有很强的扩展性和兼容性，程序员可以使用Python语言快速实现一个网站或Web服务。一般情况下，它不会指定数据库和模板引擎等对象，用户可以根据需要自己选择各种数据库。Flask自身不会提供表单验证功能，在项目实施过程中可以自由配置，从而为应用程序开发提供数据库抽象层基础组件，支持进行表单数据合法性验证、文件上传处理、用户身份认证和数据库集成等功能。Flask主要包括Werkzeug和Jinja2两个核心函数库，它们分别负责业务处理和安全方面的功能，这些基础函数为web项目开发过程提供了丰富的基础组件。Werkzeug库十分强大，功能比较完善，支持URL路由请求集成，一次可以响应多个用户的访问请求；支持Cookie和会话管理，通过身份缓存数据建立长久连接关系，并提高用户访问速度；支持交互式Javascript调试，提高用户体验；可以处理HTTP基本事务，快速响应客户端推送过来的访问请求。Jinja2库支持自动HTML转移功能，能够很好控制外部黑客的脚本攻击。系统运行速度很快，页面加载过程会将源码进行编译形成python字节码，从而实现模板的高效运行；模板继承机制可以对模板内容进行修改和维护，为不同需求的用户提供相应的模板。

## 定义

关于Flask项目中的一些专业术语、缩略语如下表1所示。

|  |  |
| --- | --- |
| 术语/缩略语 | 描述 |
| Werkzeug | Werkzeug是Python的WSGI规范的实用函数库。 |
| Jinja2 | Jinja2是基于python的模板引擎。 |
| click | Flask的置命令行工具。 |
| WGSI | Web服务器网关接口是为Python语言定义的Web服务器和Web应用程序或框架之间的一种简单而通用的接口。 |

表 1 专业术语及缩略语

## 参考资料

[1] Flask官方文档 <https://flask.palletsprojects.com/en/1.1.x/>

[2] w3school flask学习教程：<https://www.w3cschool.cn/flask/>

[3]《Web安全攻防：渗透测试实战指南》

[4]《SQL注入攻击与防御(第2版)》

[5]《flask入门教程》：<http://helloflask.com/tutorial/>

[6] HelloFlask 论坛：<https://discuss.helloflask.com/>

## 标准、条约和约定

参照如下标准：

GB/T 13702-1992 计算机软件分类与代码

GB/T 19003-2008 软件工程

GB/T 5538-1995 软件工程标准分类法

GB/T 9385-2008 计算机软件需求规格说明

GB/T 5532-2008 计算机软件测试规范

GB/T 18221-2000 信息技术程序设计语言

GB/T 8567-2006 计算机软件文档编制规范

# 项目概述

## Flask微框架

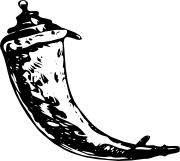


图2.1 Flask Logo

Flask是一个微框架。“微”并不代表整个应用只能塞在一个 Python 文件内， 当然塞在单一文件内也没有问题。 “微”也不代表 Flask 功能不强。 微框架中的“微”字表示 Flask 的目标是保持核心简单而又可扩展。 Flask 不会替你做出许多决定，比如选用何种数据库。 类似的决定，如使用何种模板引擎，是非常容易改变的。 Flask 可以变成你任何想要的东西，一切恰到好处，由你做主。

缺省情况下， Flask 不包含数据库抽象层、表单验证或者其他已有的库可以处理的东西。 然而， Flask 通过扩展为你的应用添加这些功能，就如同这些功能是 Flask 生的一样。 大量的扩展用以支持数据库整合、表单验证、上传处理和各种开放验证等等。Flask 可能是 “微小”的，但它已经为满足您的各种生产需要做出了充足的准备。

Flask本身相当于一个内核，其他几乎所有的功能都要用到扩展，都需要用第三方的扩展来实现。用 extension 增加其他功能。Flask没有默认使用的数据库、窗体验证工具。你可以选择MySQL，也可以用NoSQL。

## 项目目标

从Flask平台出发，重新编写起需求规格说明书、测试需求规格说明书，并根据需求规格说明书与测试需求规格说明书对Flask进行模块化测试，并在此基础上拓展应用包含改进算法、基于现有功能组合实现新功能、优化性能等。

## 产品目标与范围

编写Flask的需求规格说明书，为Flask的各个模块编写测试需求规格说明书、测试用例，并以此对Flask进行测试。同时深入研究Flask对其进行功能扩展。

## 工作内容

以Flask v1.1.1版本为标准，该版本也是目前的最新发行版本。学习了解Flask框架各个模块，熟悉各种用法，对其进行改进。具体内容如下：

1. 项目组成员均需学习Flask框架，深入了解其使用以及特性等，在对Flak有了初步的认识并能够完成简单程序的编写的基础上，分别着重对某一块的内容进行深入的研究学习，分享学习资料，笔记，并组织例会进行交流探讨。
2. 在对Flask有了较为深入的了解后，对其进行改进，可以考虑的改进点：改进算法、基于现有功能组合实现新功能、优化性能等。
3. 根据课程的整体安排和每个实验的具体要求，依照项目时间计划完成其他各类文档和报告。

## 应交付成果

项目应交付的产品如表2所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 名称 | 交付时间 | 要求 | 备注 |
| No.1 | 项目计划书 | 2020/3/18 | 完整、明确 | 暂无 |
| No.2 | 项目进度表 | 2020/3/18 | 完整、详细 | 暂无 |
| No.3 | 项目工作日志 | 待定 | 暂无 | 暂无 |
| No.4 | 软件需求规格说明书 | 待定 | 暂无 | 暂无 |
| No.5 | 需求检查单 | 待定 | 暂无 | 暂无 |
| No.6 | 测试需求规格说明书 | 待定 | 暂无 | 暂无 |
| No.7 | 软件测试分析报告 | 待定 | 暂无 | 暂无 |
| No.8 | 软件测试评审报告 | 待定 | 暂无 | 暂无 |
| No.9 | 项目工作量统计报告 | 待定 | 暂无 | 暂无 |
| No.10 | 项目方法总结报告 | 待定 | 暂无 | 暂无 |
| No.11 | 项目个人总结 | 待定 | 暂无 | 暂无 |

表 2 应交付成果列表

## 项目验收方式和依据

1. 验收方式

项目验收方式采用中期课堂评审和终期交付验收。

1. 验收依据

验收依据“应交付成果”清单进行验收。

# 项目团队组织

## 组织结构

此次项目有两个角色：组员和组长。组长与组员沟通协调，安排具体事宜、分配相应的任务。而组员则需要根据组长的安排要求，及时完成任务。

## 人员分工

人员组成如下表3所示。

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **姓名** | **角色** | **介绍** |
| 1 | 马广洲 | 组员  组长 | 智能信息处理研究所  熟悉C++, Python语言  Flask框架,TensorFlow框架 |
| 2 | 潘安佶 | 组员 | 系统结构实验室  熟悉java，Python语言  有机器人和后端开发经验 |
| 3 | 常佳辉 | 组员 | 系统结构研究所  熟悉C/C++，PHP语言  有Web后端开发经验 |
| 4 | 刘佳恒 | 组员 | （软件学院）视觉计算组  熟悉Python语言  熟悉PyTorch框架，方向计算机视觉 |
| 5 | 张祥国 | 组员 | 软国重实验室  熟悉C，python语言  pytorch框架、后端开发经验 |
| 6 | 王康明 | 组员 | 软国重实验室  熟悉C++，Python语言  喜欢探索底层，有网络爬虫和Web开发经验 |
| 7 | 牟秋宇 | 组员 | ACT实验室  熟悉C，JAVA语言  有web后端经验 |

表 3 人员组成

## 协作与沟通

根据老师的总体进度安排，确定本周具体任务。根据组员实现对系统了解程度等因素，对任务进行分工，大致要做到小组成员的工作量大致均等。

小组会议在腾讯会议中进行，并且随时可在在微信上提出问题，资料分享以及汇报各自进度。每周至少讨论一次，来交流最近工作、学习的内容和遇到的困难等。

沟通和协作都要记录具体内容。

# 实施计划

## 总体进度计划

|  |  |  |  |
| --- | --- | --- | --- |
| 任务名称 | 工期 | 开始时间 | 完成时间 |
| **项目确定以及介绍** | **5 days** | **2020年3月8日** | **2020年3月12日** |
| 项目调研 | 4 days | 2020年3月8日 | 2020年3月11日 |
| 选题会议 | 1 day | 2020年3月12日 | 2020年3月12日 |
| **项目计划书** | **4.5 days** | **2020年3月14日** | **2020年3月18日** |
| 撰写引言 | 4 h | 2020年3月14日 | 2020年3月14日 |
| 项目概述 | 4 h | 2020年3月15日 | 2020年3月15日 |
| 目的及成果 | 4 h | 2020年3月16日 | 2020年3月16日 |
| project学习与使用 | 8 h | 2020年3月17日 | 2020年3月17日 |
| **软件需求分析** | **6 days** | **2020年3月21日** | **2020年3月26日** |
| 软件模块分析 | 2 days | 2020年3月21日 | 2020年3月22日 |
| 测试需求分析 | 1 day | 2020年3月23日 | 2020年3月23日 |
| 软工规范分析 | 1 day | 2020年3月24日 | 2020年3月24日 |
| 扩展功能分析 | 1 day | 2020年3月25日 | 2020年3月25日 |
| 需求分析会议 | 1 day | 2020年3月26日 | 2020年3月26日 |
| **软件需求评审** | **7 days** | **2020年3月27日** | **2020年4月2日** |
| 修改软件需求模块 | 2 days | 2020年3月27日 | 2020年3月28日 |
| 修改测试需求 | 1 day | 2020年3月29日 | 2020年3月29日 |
| 修改软工规范需求 | 2 days | 2020年3月30日 | 2020年3月31日 |
| 修改扩展功能需求 | 2 days | 2020年3月31日 | 2020年4月1日 |
| 需求评审会议 | 2 days | 2020年4月1日 | 2020年4月2日 |
| **软件需求复评审** | **7 days** | **2020年4月3日** | **2020年4月9日** |
| 确定软件模块需求 | 2 days | 2020年4月3日 | 2020年4月4日 |
| 确定测试需求 | 2 days | 2020年4月5日 | 2020年4月6日 |
| 确定软工规范需求 | 1 day | 2020年4月6日 | 2020年4月6日 |
| 确定扩展功能需求 | 2 days | 2020年4月7日 | 2020年4月8日 |
| **软件产品改进与展示** | **28 days** | **2020年4月10日** | **2020年5月7日** |
| 软件改进 | 27 days | 2020年4月10日 | 2020年5月6日 |
| 软件展示 | 1 day | 2020年5月6日 | 2020年5月6日 |
| **软件测试分析** | **7 days** | **2020年5月8日** | **2020年5月14日** |
| 测试需求分析 | 7 days | 2020年5月8日 | 2020年5月14日 |
| **软件测试评审** | **7 days** | **2020年5月15日** | **2020年5月21日** |
| 修改测试需求 | 7 days | 2020年5月15日 | 2020年5月21日 |
| **软件测试复评审** | **7 days** | **2020年5月22日** | **2020年5月28日** |
| 确定测试需求 | 2 days | 2020年5月22日 | 2020年5月23日 |
| 软件演示与测评 | 1 day | 2020年5月24日 | 2020年5月24日 |
| 综合实验分析 | 2 days | 2020年5月25日 | 2020年5月26日 |
| 综合实验总结 | 1 day | 2020年5月27日 | 2020年5月27日 |
| **软件进度计划与控制** | **92 days** | **2020年3月12日** | **2020年6月11日** |
| 每周组会 | **92 days** | 2020年3月12日 | 2020年6月11日 |
| 每日更新github | **92 days** | 2020年3月12日 | 2020年6月11日 |
| 工作量估计与统计分析 | **92 days** | 2020年3月12日 | 2020年6月11日 |
| 配置管理 | **92 days** | 2020年3月12日 | 2020年6月11日 |

## 项目控制计划

暂无。

# 关键问题

暂无。

# 专题计划要点

暂无。