**软件设计文档**

**版本号：1.2**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **修改内容** | **修改日期** | **修改人** |
| **1.0** | **完成文档模板，初步编写技术选型、架构设计** | **2016.7.2** | **何翔** |
| **1.1** | **进一步编写技术选型、架构设计** | **2016.7.2** | **苏逸民** |
| **1.2** | **编写服务器设计内容** | **2016.7.2** | **何翔** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Table 1版本管理

# 1.技术选型

系统前端采用html、css、javascript以及bootstrap框架等，服务器采用python的tornado框架，数据库使用mysql。这些都是组员熟悉的技术。

爬虫使用python，并利用beautifulsoup解析html。基于html的解析，健壮性更好。

推荐算法使用FP-Growth。这是基于关联规则的推荐。由于爬虫只获取了豆列（类似购物单），不适合使用协同过滤等方法。

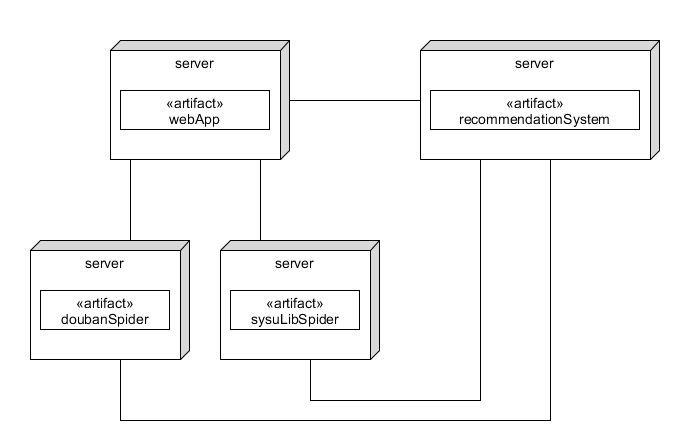
# 2.架构设计

系统由4部分组成，分别为web应用、豆瓣爬虫、图书馆爬虫、推荐算法组成。

爬虫为推荐系统提供训练数据，为web应用提供图书基本信息。

推荐系统为web应用提供推荐书籍列表。

Web应用提供访问服务。



# 3.模块划分

## 3.1.前端

（例如：根据对用例图的分析得出前端5个页面XXX…………………………）

## 3.2.服务器

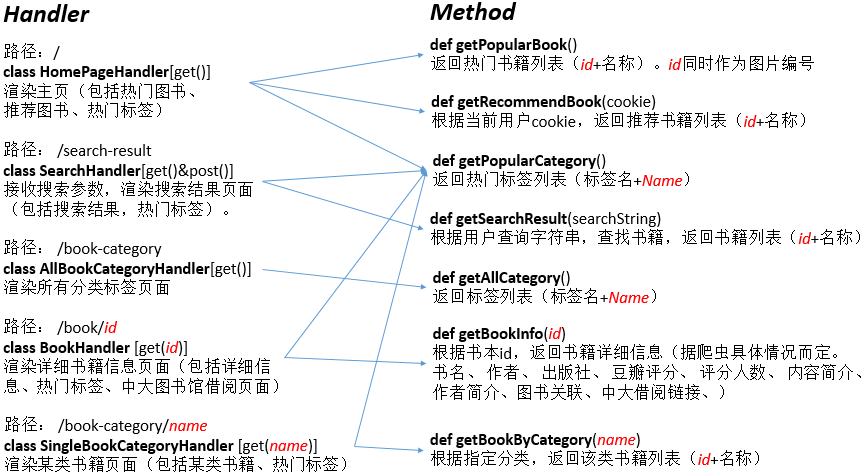


Figure 1服务器结构

服务器利用tornado框架搭建，并根据上述前端页面的设计，将服务器结构划分为五个主要Handler：HomePageHandler、SearchHandler、AllBookCategoryHandler、BookHandler、SingleBookCategoryHandler。每个Handler对应于一个Web页面，并根据其渲染的页面所需要的数据，调用相应Method获取。每个Method利用数据库API（具体设计下文”数据库”部分给出）从数据库中获取相应数据。

### 3.2.1.Handler

* **HomePageHandler**
* **SearchHandler**
* **AllBookCategoryHandler**
* **BookHandler**
* **SingleBookCategoryHandler**

### 3.2.1.Method

* **getPopularBook()**
* **getRecommendBook(cookie)**
* **getPopularCategory()**
* **getSearchResult(searchString)**
* **getAllCategory()**
* **getBookInfo(id)**
* **getBookByCategory(name)**

## 3.3数据库

## 3.4豆瓣爬虫

## 3.5图书馆爬虫