**投标文件制作工具**

**技术方案**

作者：杨飞龙

时间：2017-7-29

版本控制：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 版本 | 时间 | 作者 | 备注 |
| 1 | V0.1 | 2017-7-29 | 杨飞龙 | 创建文件 |

**目录**

[1. 需求分析 5](#_Toc489177296)

[1.1. 现状 5](#_Toc489177297)

[1.2. 技术方案组成 5](#_Toc489177298)

[1.2.1. 前言 5](#_Toc489177299)

[1.2.2. 需求分析 5](#_Toc489177300)

[1.2.3. 方案设计 6](#_Toc489177301)

[1.2.4. 项目实施 6](#_Toc489177302)

[1.2.5. 售后服务 7](#_Toc489177303)

[1.2.6. 技术培训 7](#_Toc489177304)

[1.2.7. 附件 7](#_Toc489177305)

[1.3. 需求分析 7](#_Toc489177306)

[1.3.1. 功能性需求 7](#_Toc489177307)

[1.3.2. 非功能性需求 8](#_Toc489177308)

[1.4. 关键技术和难点 8](#_Toc489177309)

[1.5. 需求总结 8](#_Toc489177310)

[2. 整体设计 8](#_Toc489177311)

[2.1. 总体说明 8](#_Toc489177312)

[2.2. 系统架构 9](#_Toc489177313)

[2.3. 功能结构 10](#_Toc489177314)

[2.4. 数据库设计 10](#_Toc489177315)

[2.5. 部署设计 12](#_Toc489177316)

[3. 详细设计 12](#_Toc489177317)

[3.1. 技术文件生成模块 12](#_Toc489177318)

[3.2. 商务文件生成模块 13](#_Toc489177319)

[4. 测试 13](#_Toc489177320)

1. 需求分析

鉴于制作投标文件的时候总是需要花很多的时间做很多重复性工作，而且没有什么技术含量，为了能够提高大家的工作效率，制作这个工具。

本工具只针对技术部分内容，商务部分内容后续再说。

* 1. 现状

目前项目的主要招标机构为：中航技，本项目主要针对他们的招标文件来制作。当然，技术部分的要求和招标机构的关系不是很大。

技术部分编写一般包含四个内容：

1. 技术方案
2. 技术偏离表
3. 货物一览表
4. 分项报价表 （不含价格）

技术偏离表，暂时不自动生成，感觉比较复杂。一览表和分项报价表比较简单。主要说说技术方案。

* 1. 技术方案组成
     1. 前言

主要感谢甲方给予项目投标的机会，都是套话，主要涉及的工作是：

1. 说明双方的全名，投标文件中再次出现双方单位的简称；
2. 说明本项目的名称
   * 1. 需求分析

主要分成几个部分，

1. 分析用户的现状，如果不知道的话就不写此部分
2. 从招标文件摘取技术需求的主要部分，分成配置整体需求、配置需求和实施需求、售后服务需求等
3. 需求总结，在此说明本次投标的产品。
   * 1. 方案设计

一般包括下面几个内容：

1. 设计依据：一般就两个，招标文件和我公司的理解
2. 总体目标：建设一套适合用户的xx系统之类的。
3. 建设原则：一般不需要修改
4. 设备选型：方案的主要设计部分，对于简单的设备投标，单独一章即可，如果比较复杂的，自己调整。
5. 详细配置清单：一般放置原厂给的BOM清单，也可以放简单的清单。
6. 系统拓扑：画一个图，并进行说明，对于简单的设备招标可以没有这个部分；
7. 如果技术要求还有特殊要求的就添加在后面，如数据迁移，特殊系统部署等，主要注意技术要求中的星号部分。
8. 实施案例：贴上基本案例即可

一般就前面5个部分就行了。

* + 1. 项目实施

项目实施主要都是套路。

1. 实施总体要求：到货周期，实施周期（如果有），实施内容
2. 项目责任分工：双方，一般不用改
3. 项目人员组成：一般就一段话，明确要求再添加具体实施人员。
4. 项目实施内容：比较详细说明实施内容，一般只修改安装调试内容和培训内容
5. 项目实施计划：说明实施计划，和简单的实施步骤
6. 项目实施过程：讲述整个过程，包括机房环境到验收，之间需要修改的部分主要是安装调试、系统验收两个部分的内容
7. 质量保证措施：主要修改的部分是项目提供的文档，看是否招标文件中明确的要求
   * 1. 售后服务

售后服务一般会包含两个部分

1. 我公司售后服务和保修体系：服务承诺+套话
2. 原厂售后服务和保修体系：服务承诺+套话
   * 1. 技术培训

主要两个部分

现场培训：一般都是提供的，在实施过程中进行培训

原厂培训：也是机构培训，看招标文件是否要求

* + 1. 附件

放产品彩页或或者说明书。

* 1. 需求分析

目标很简单，就是要生成投标书的技术部分，分成下面几个部分吧

1. 生成技术方案
2. 生成技术偏离表（TODO）
3. 生成投标货物说明一览表
4. 生成分项报价表

四个word文档。

* + 1. 功能性需求

分成下面几个功能

1. 从上传的配置清单文档中获取配置等生成一览表
   * 能够读取上传的配置清单
   * 能够将配置清单中的内容填写到一览表模版中
2. 从上传的配置清单文档中获取配置等生成分项报价表
   * 能够读取上传的配置清单
   * 能够将配置清单中的内容填写到分项报价表中
3. 生成技术方案
   * 读取上传配置清单中的招标人等信息，配置等
   * 读取上传的原厂售后服务、产品介绍；
   * 使用系统中的技术方案相应部分的模版，替换相关词汇
   * 生成整体的技术方案
4. 生成技术偏离表（TODO）
5. 可以上传自己的模版（TODO）
6. 技术方案使用的模版可以自定义（TODO）
   * 1. 非功能性需求

（1）并发需求

满足50个人使用即可；并发基本没有要求。

（2）安全性需求

模版可以共享，

* 1. 关键技术和难点

模版的准备

* 1. 需求总结

1. 整体设计
   1. 总体说明

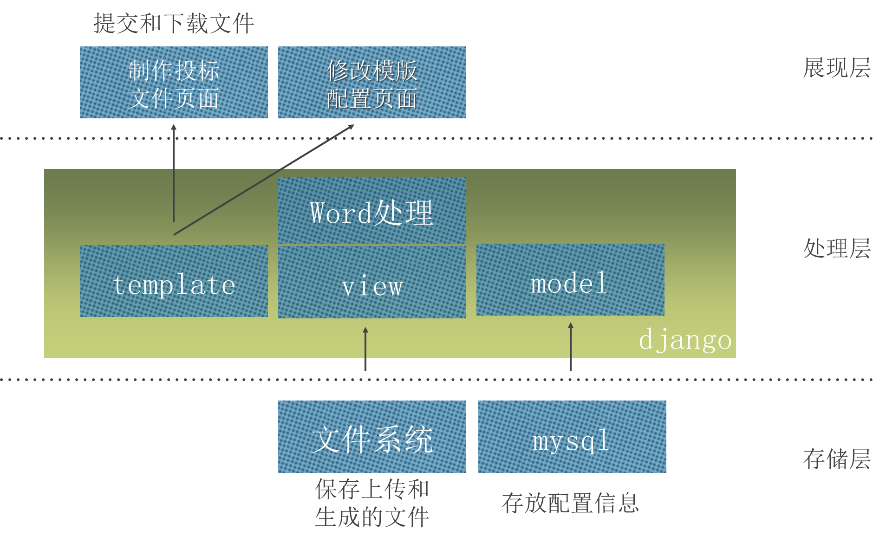
使用的开发工具有：

* 开发语言：python3
* Web框架：django
* 处理word文档：python-docx
* 前端：bootstrap

工程师获得招标文件后，通过填写一个配置表格（excel 表），完成后同其他一些产品说明等文档上传到系统中，在必要的情况下进行模版的定制，系统自动完成投标文件技术部分的编写，并提供下载链接。工程师下载系统自动生成的投标文件，进行适当修改就可以完成投标文件。

尽量少的需要工程师进行重复性工作是本工具的主要目标。

* 1. 系统架构



存储层：使用数据库mysql或者其他数据库，保存的主要是系统的配置文件，生成记录等；

处理层：几乎所有工作都由django完成，单独一个模块处理投标文件的生成；

表现层：html页面，主要制作页面和模版修改的页面。

* 1. 功能结构

管理员：（TODO，使用django-admin）

* 查看生成记录
* 创建、删除、修改、查看 系统用户（工程师）

工程师：

* 登陆系统和退出系统
* 提交配置文件，填写表单，获得投标文件
* 自定义投标方案的目录，和目录对应的模板

整体结构如下：



* 1. 数据库设计

ER图就不画了，主要是用表格来编写吧，django中的表会自动生成ID，因此在数据库设计时不单独增加ID列。

用户表：

|  |  |  |  |
| --- | --- | --- | --- |
| **表名：** | **UserProfile** |  |  |
| 序号 | 列名 | 类型 | 说明 |
|  | User | oneToOneField | 1:1 django-user |
| 1 | name | Charfield 16 | 姓名 |
| 2 | Role | Charfield 16 | 角色 |
| 3 | Memo | Textfield | 说明 |

提交表：

|  |  |  |  |
| --- | --- | --- | --- |
| **表名：** | **SubmitFile** |  |  |
| 序号 | 列名 | 类型 | 说明 |
| 1 | ProjectName |  | 项目名称 |
| 2 | Profile |  | 配置文件 |
| 3 | TechTemplate |  | 技术方案模版  外键 |
| 4 | SubmitTime |  | 时间 |
| 6 | DownloadFile | Filepath | 可下载文件位置 |
| 5 | User |  | 用户 |

技术方案模版表

|  |  |  |  |
| --- | --- | --- | --- |
| **表名：** | **TechTemplate** |  |  |
| 序号 | 列名 | 类型 | 说明 |
| 1 | Name |  | 模版名字 |
| 2 | TechContent | ManyToMany | 技术方案目录，外键 |
| 3 | Memo |  | 备注 |
| 4 | ChangeTime |  | 模版修改时间 |
| 5 | User |  | 属主，外键 |

技术方案目录标题表

|  |  |  |  |
| --- | --- | --- | --- |
| **表名：** | **TechContent** |  |  |
| 序号 | 列名 | 类型 | 说明 |
| 1 | Name |  | 标题名 |
| 2 | Level | int | 标题级别，可选1,2,3,4 |
| 3 | Memo |  | 级别说明 |
| 4 | UpTitle |  | 上级标题，外键 |
| 5 | User |  | 属主，外键 |

分块内容模版

|  |  |  |  |
| --- | --- | --- | --- |
| **表名：** | **PartTemplate** |  |  |
| 序号 | 列名 | 类型 | 说明 |
| 1 | Name |  | 模版名字 |
|  | Content | N:1 | 目录，外键 |
| 2 | TempFile | File | 模版文件 |
| 3 | Memo |  | 模版说明 |
| 4 | ChangeDate |  | 模版修改时间 |
| 5 | User |  | 属主，外键 |

* 1. 部署设计

应用服务器和数据库服务器采用同一服务器，部署情况：

|  |  |  |
| --- | --- | --- |
| 序号 | 项目 | 版本 |
| 1 | 操作系统 | CentOS 6 |
| 2 | 数据库 | Mysql 5 |
| 3 | 反向代理 | Nginx |
| 4 | Web框架 | Django |
| 5 | Web应用系统 | Uwsgi |

1. 详细设计

详细说明每个模块的设计。

* 1. Templates & views 总体说明

说明temp和view的整体情况，继承说明等。

* + 1. URL设计

主要部分在详细模块中说明。

* + 1. Temp设计

html模版。

* + 1. Views设计

Web处理函数。

* + 1. Backends

后台处理，主要是处理投标文件的自动生成。

* 1. Base.html的设计

说明base的设计。

* 1. 登陆登出模块
     1. 整体结构

如下表所示：

|  |  |  |  |
| --- | --- | --- | --- |
| **Views** | **相关Templates** | **说明** | **url** |
| Acc\_login | Login.html  Index.html | 登陆，返回主页 | /acc\_login |
| Acc\_logout | Index.html  Login.html | 登出，返回主页 |  |
|  | Index.html | 主页 | /index |

* + 1. Models

涉及到的表就一个用户表（UserProfile）

|  |  |  |  |
| --- | --- | --- | --- |
| **表名：** | **UserProfile** |  |  |
| 序号 | 列名 | 类型 | 说明 |
|  | User | oneToOneField | 1:1 django-user |
| 1 | name | Charfield 16 | 姓名 |
| 2 | Role | Charfield 16 | 角色 |
| 3 | Memo | Textfield | 说明 |

* + 1. Views

|  |  |  |  |
| --- | --- | --- | --- |
| Views函数 | 处理过程 | 返回 | 备注 |
| Acc\_login | 接受post信息，并验证用户和密码是否正确， |  |  |
| 如果正确： | /index |  |
| 不正确 | Login.html | 登陆页面，外加错误信息 |
| Acc\_logout | Logout登出用户 | Login.html |  |
| Index | 如果用户未登陆 | Login.html |  |
| 如果用户登陆 | Index.html |  |

* + 1. Templates

本部分就2个temp，分别为index和login，都是继承至base的模版。

* 1. 技术文件生成模块
     1. 整体结构
     2. Models
     3. Views
     4. Templates
     5. Backends
  2. 商务文件生成模块（TODO）
     1. 整体结构
     2. Models
     3. Views
     4. Templates
     5. 其他

1. 测试
   1. 测试环境
   2. 测试内容
   3. 测试用例
   4. 测试效果