一、构件设计细节

1.模型构件

**功能：**

接收用户上传的文件

对文件进行处理，提取关键字并进行排序

将处理结果返回给控制器

**属性和方法：**

文件路径：记录用户上传的文件路径

处理结果：存储处理后的关键字和排序结果

处理文件方法：处理上传的文件，并提取关键字并排序

获取处理结果方法：返回处理后的关键字和排序结果

2.视图构件

**功能：**

提供文件上传界面

显示文件处理进度和结果

**属性和方法：**

文件上传按钮：用于上传待处理文件

进度条：用于显示文件处理进度

结果列表：用于显示处理后的关键字和排序结果

3.控制器构件

**功能：**

接收用户上传的文件，并传递给模型进行处理

获取处理结果，并传递给视图进行显示

**属性和方法：**

模型对象：用于连接模型

视图对象：用于连接视图

文件上传方法：接收用户上传的文件，并传递给模型进行处理

二、连接件设计

**设计：**

采用响应式 Web 设计技术，即使得应用程序的用户界面能够自适应不同的设备屏幕尺寸和分辨率，使用户在手机端和PC端都能有良好体验，同时减少了重复开发的过程。

使用现有的Web前端框架，如：Bootstrap、Material Design。这些框架提供了一系列的 CSS 和 JavaScript 组件和模板，能够帮助我们快速地搭建具有响应式布局的 Web 应用程序

使用 CSS 媒体查询技术和 JavaScript 脚本来实现不同终端的自适应性

**保证稳定性：**

采用可靠的协议和技术，例如：HTTPS

延迟处理：如果用户上传的文件过大，可以让服务器将文件先存储到一个临时文件夹中，在后台，服务器可以将临时文件中的内容读取到内存中，进行 KWIC 处理。