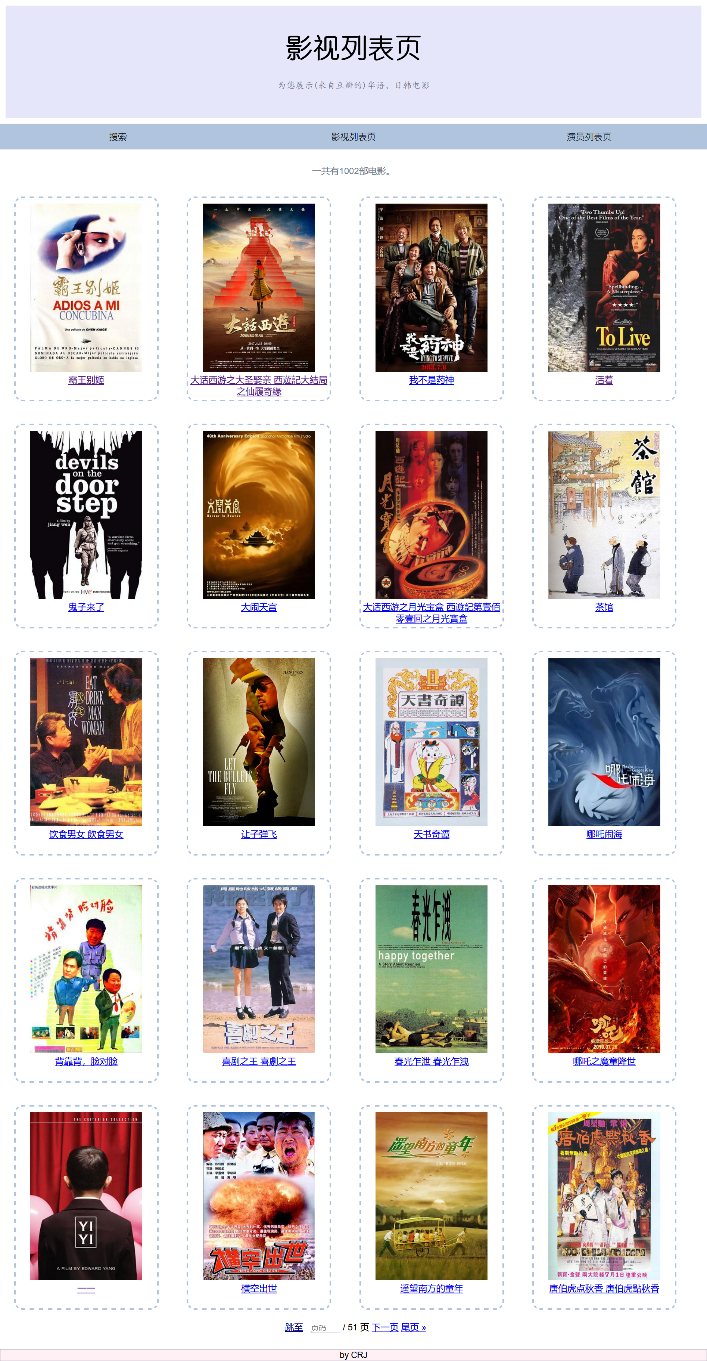
**影视爬虫与检索系统大作业 设计文档**

1. **功能介绍展示**

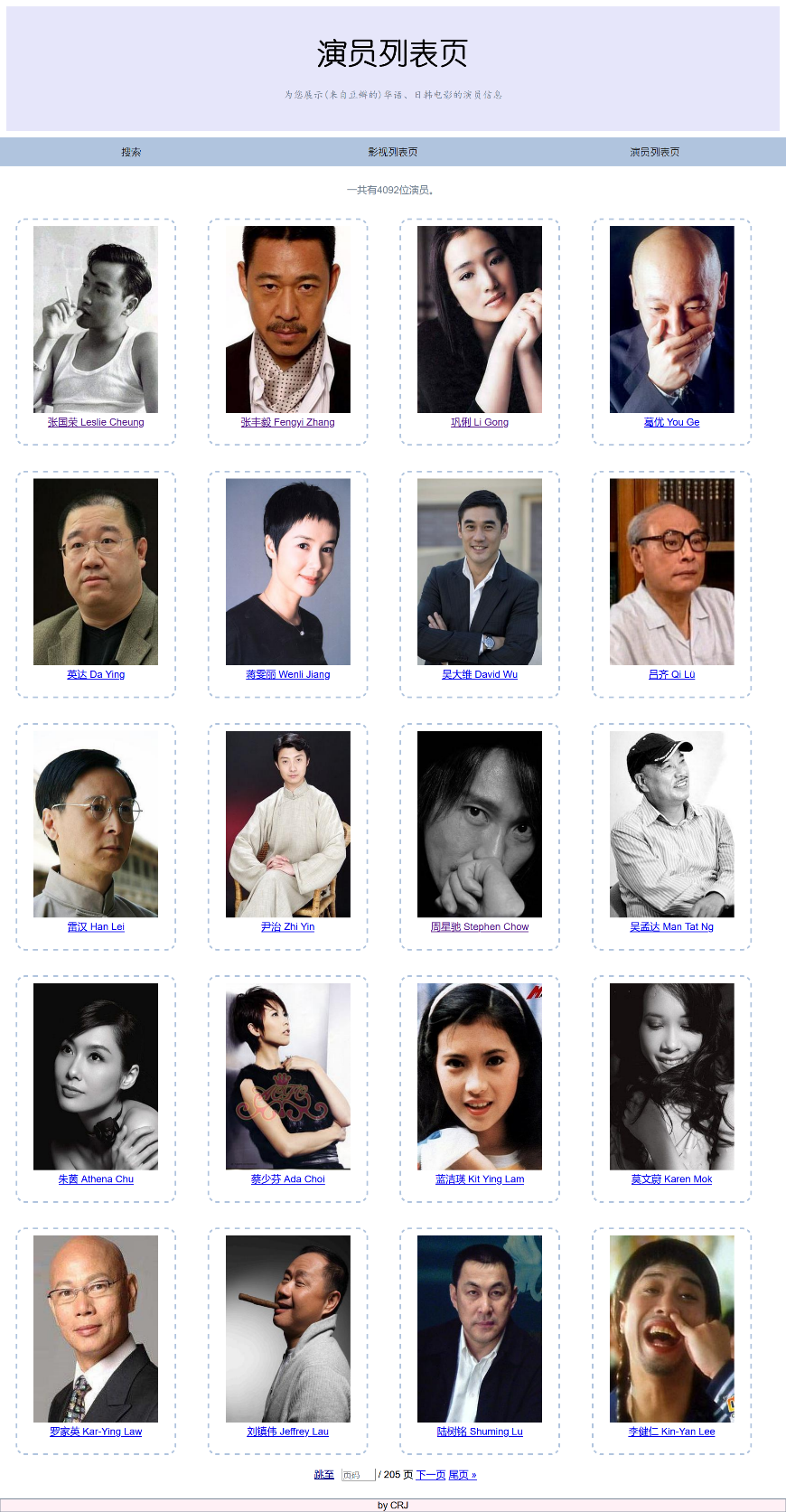
**1、影视列表页**

为系统主页，通过分页方式列出系统中的所有影视，并显示影视的总数量（1002部）。列表页中的每个条目展示电影名称和海报，点击列表页中的条目可以跳转到对应的影视信息页。页面底部有脚注。顶层有导航栏可跳转搜索等页面。



**2、演员列表页**

通过分页方式列出系统中的所有演员，并显示演员的总数量（4092位）。列表页中的每个条目展示演员名字和照片，点击列表页中的条目可以跳转到对应的演员信息页。页面底部有脚注，顶层有导航栏。



**3、影视信息页**

展示电影名称、海报、简介、导演、编剧、类型和五条短评。

展示参演演员列表，以条目形式展示，显示演员姓名和照片，点击姓名或照片可跳转至演员信息页。页面底部有脚注，顶层有导航栏。



**4、演员信息页**

显示演员姓名、简介、照片、性别等信息。列出该演员参演影视，以条目形式展示，显示影视名称和图片，点击名称或图片可跳转至影视信息页。列出该演员合作演员次数最多的十位，以条目形式展示，显示演员名、图片和合作作品数量，点击演员姓名或图片可跳转至其信息页。页面顶层有导航栏。



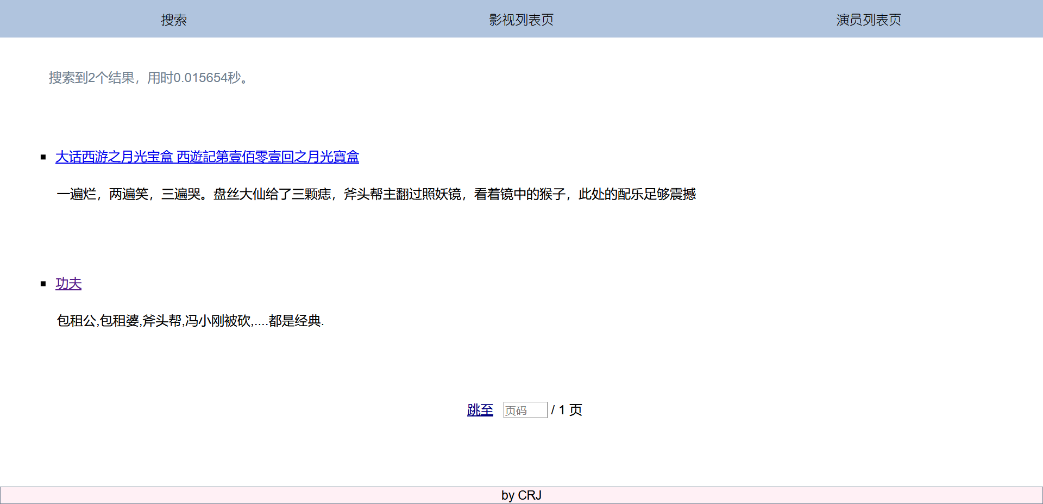
**4、搜索**

搜索是一个单独的页面。搜索栏搜索功能包括一个搜索栏、一个单选组（影视、演员、影评），回车即可搜索。输入一个关键词或一段文本（20字以内），根据单选组选项，搜索系统中存在的影视、演员和影评，搜索后跳转到搜索结果页。页面底部有脚注，顶层有导航栏。

**5、搜索结果页**

搜索结果页包含所有的搜索结果列表，以分页形式展示。搜索结果页中的每个条目设计为超链接格式，点击可跳转到相应信息页。点击影评的搜索结果跳转到该影评所在的影视信息页。页面底部有脚注，顶层有导航栏。

搜索结果页最上方显示搜索条目数量和搜索花费的时间。



1. **性能统计信息**

**1、爬虫数据量统计**

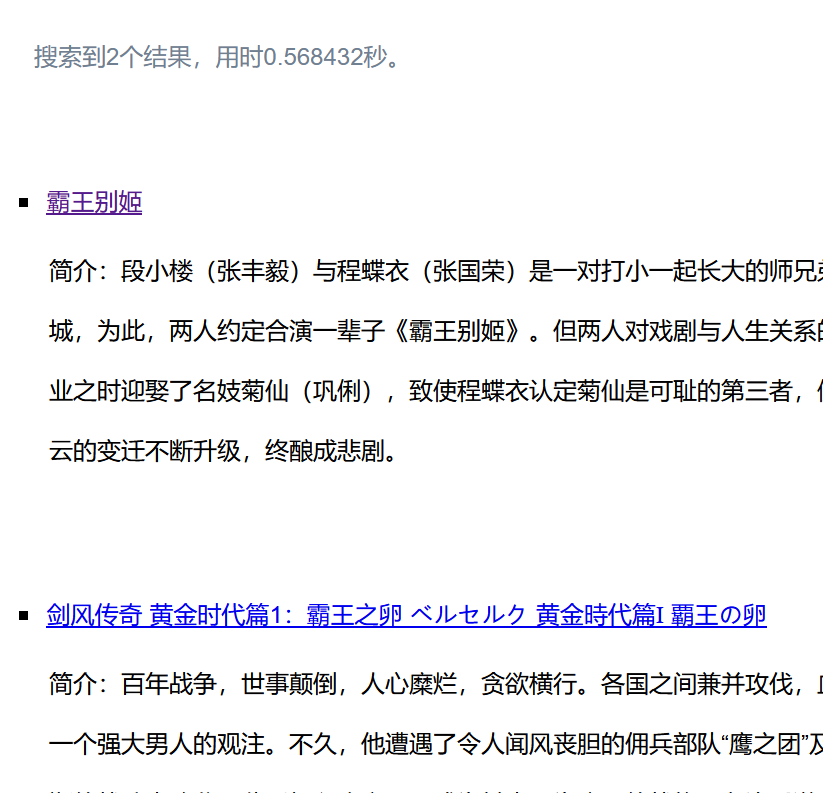
共爬取豆瓣华语电影500部，日韩电影502部，总计1002部。

电影信息包括名称、演员、简介、图片、五条短影评和导演、编剧、类型。

共爬取电影相关演员4092位。

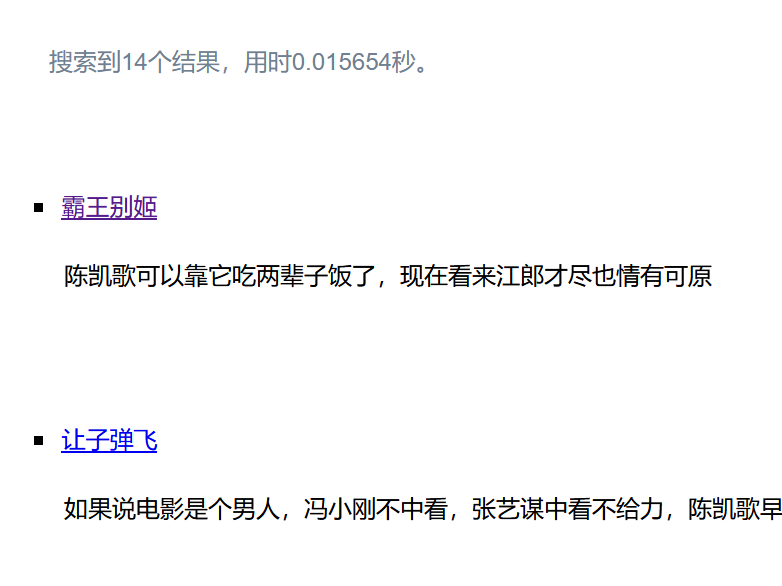
演员信息包括姓名、简介、个人照片和性别。

**2、查询时间统计**

影视搜索：以搜索“霸王”为例，搜索到2个结果（包括霸王别姬等电影），用时0.568432秒。



演员搜索：以搜索“张”为例，搜索到114个结果（包括张国荣等演员），用时0.593827秒。

影评搜索：以搜索“陈凯歌”为例，搜索到14个结果（包括霸王别姬等电影），用时0.015654秒。

1. **各个功能涉及的技术和实现方式**

**1、爬虫**

利用requests库获取网页内容，利用浏览器信息伪装头部，同时每爬一次间隔2-5秒。尽管如此，还是被ban了n个ip。最后在开飞行换ip/开代理/开热点/借用舍友电脑等操作下，爬完了1002部电影信息。

利用lxml库中的etree解析html内容。通过xpath定位，获得想要的信息。将每部电影信息存入一个dict，通过pickle库存入pkl文件。

**2、django**

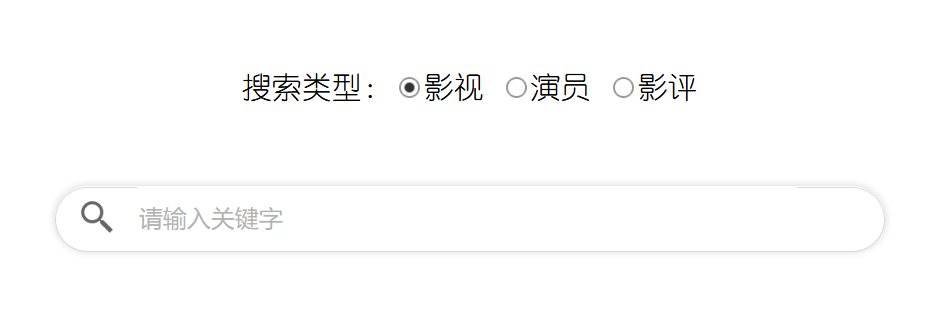
建立app。在model中建立“Actor”“Movie”和代表电影、演员关系的类“Membership”。将pkl中存储的数据导入sqlite3数据库，存在这3个model中建立表格，方便索引查询。通过查找每个演员参演的电影，再查找该电影的参演演员，计算合作次数。最后将这些信息以变量形式传入html进行展示。

**3、html**

如何在html中使用变量和传回单选框及关键字是两个难题。html不支持[]调用变量，所以要使用数字索引需要movie.0的格式。通过request.GET.get('keyword')等函数获得用户输入的关键字和单选选项，传回后端进行操作。

**4、css**

为了让页面看起来美观，通过css的盒模型等进行美化。

其中最满意的外观效果是搜索栏。