

# 题库设计文档

## 1. 功能设计

本系统由 Java EE 技术开发，主要用于题库、专题库、素材库和动画库的展示，辅以一些方便操作的简化接口。每个库都是单页面应用，页面切换和过度效果由 Javascript 控制，所以在性能和用户体验方面有所优势。

### 1.1 题库

该部分的功能（前台和后台）较为完善，页面如图 1 所示。

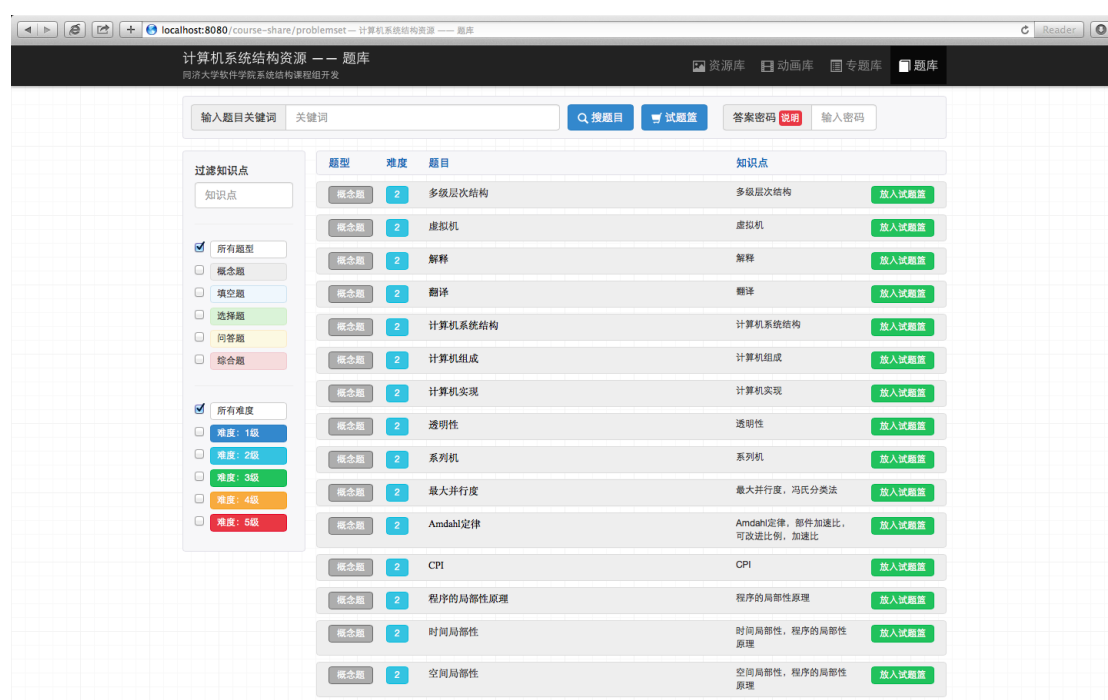


图 1 题库主页示例

题库的功能包括：

- 题目浏览  
每道题目分为题型、难度、题目内容、知识点四部分，采用列表和滚动加载的方式展示，不同难度和题型的题目分别用明显的颜色加以区别。
- 题目搜索与过滤  
搜索功能位于页面上方，主要针对的题目内容和知识点，通过关键词匹配得到搜索结果。过滤功能位于页面的左边，以复选框的形式筛选题目。
- 题目答案  
答案在输入密码后才可见。
- 试题组卷  
每道题目均可以先放入试题篮，然后再在试题篮中删除题目，或者自动组卷，导出成

doc 格式的文档。

- 题目导入和清空

该功能用于将题目从 doc 文本导入数据库或清空数据库，由于只是一次性的任务，所以仅利用 URL 来完成这些操作，必要时可以完善。

## 1.2 其它库

对于专题库、素材库和动画库，后台功能包括资源的导入和展示，但页面风格需要重新实现，目前主页上的链接暂时重定向至 <http://xuexi.tongji.edu.cn>。

## 2. 系统架构

### 2.1 开发环境

参数	配置	链接
Supported OS	Windows, Linux, Mac OS X	N/A
Java Version	JDK 1.7	<a href="http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html">http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html</a>
Web Server	Tomcat 7.0	<a href="http://tomcat.apache.org/download-70.cgi">http://tomcat.apache.org/download-70.cgi</a>
Java Library	Spring 3.1	<a href="http://projects.spring.io/spring-framework/">http://projects.spring.io/spring-framework/</a>
Javascript	jQuery 1.9.1	<a href="http://jquery.com/">http://jquery.com/</a>
Library	Bootstrap 3.0	<a href="http://getbootstrap.com/">http://getbootstrap.com/</a>
Database	MySQL 5.6.13	<a href="http://dev.mysql.com/downloads/">http://dev.mysql.com/downloads/</a>
IDE	Eclipse Java EE IDE (Kepler)	<a href="http://www.eclipse.org/downloads/">http://www.eclipse.org/downloads/</a>

2.2 软件架构

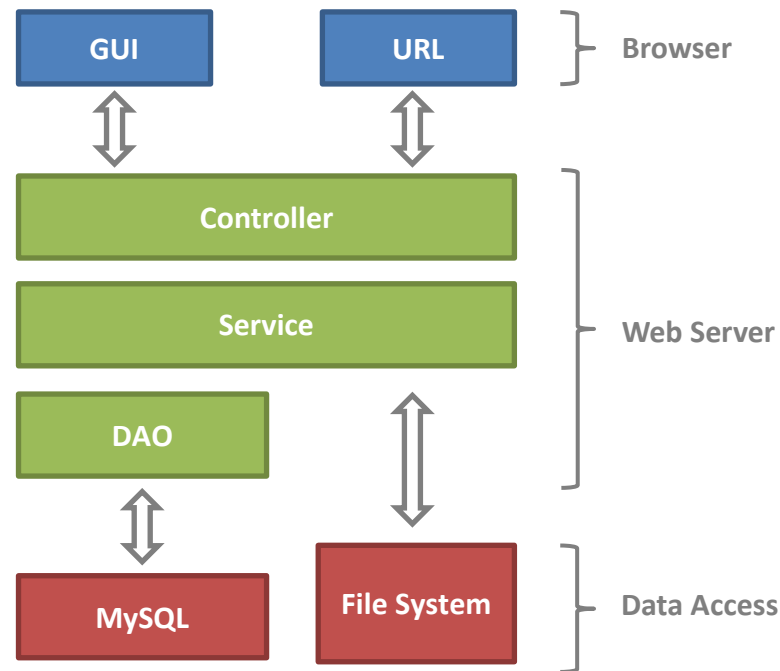


图 2 系统架构图

如图 2 所示，用户接口的本质由 URL 驱动，遵循 REST 原则，GUI 的用户界面由 AJAX 与后台 web server 交换数据。Web server 由 Tomcat 支持，其上运行 Java EE 的程序，包括 Controller 层、Service 层和 Dao 层，分别处理与浏览器的数据交换、具体的业务逻辑和数据库的读写。数据存储由 MySQL 数据库和操作系统的文件系统组成，一般数据均存在 MySQL 中，初始化待导入的文本均在文件系统中。

3. 详细设计

3.2 Controller 接口设计

Controller 层由 Spring MVC 支持，用于和浏览器通信，向外界提供 RESTful API 形式的接口，交换数据的格式一般为 JSON，也支持图片和动画等格式。

3.2.1 ProblemsetController 类

该 controller 处理题库相关的所有请求，根 URI 为 “/problemset”，以下是各个 action 对应的方法。

● index

URI	一般为空或者 “/”；（GET）
功能	仅返回题库主页。

● importProblems

URI	“/import/{problems}”；（GET）
功能	导入磁盘上的试题文档至数据库。URI 中的 problems 为文档名称，多个文档用

	英文逗号 “,” 隔开。注意，此接口仅为导入试题方便，不建议在日常运行时继续使用。
举例	当导入名为 “problem1.doc” 和 “problem2.doc” 两个文档时，对应的 URI 为 “xxx/problemset/import/problem1,problem2”。

● clearProblems

URI	“/clear”; (GET)
功能	清空数据库中题库的全部数据，使用时要小心。

● listProblems

URI	“/list”; (GET)
功能	返回符合过滤条件的题目列表，返回类型为 JSON。URL 中可以附带的过滤参数包括，problem_type 指定题目类型；difficulty 指定题目难度；problem_content 指定题目内容的关键词；offset 用于分页，一般不应人为指定。

● picture

URI	“/resource/{id}”; (GET)
功能	获得题目中的图片文件，URI 中的 id 为图片标识，返回类型为具体的图片类型。该方法不应人为使用。

● paper

URI	“/paper”; (GET)
功能	给定题目 ID 列表生成试卷，返回类型为 doc 格式。URL 中可以带的参数为 pids，多个 ID 用英文逗号隔开。
举例	当要把 ID 为 1,2,3 的题目组成试卷时，对应的 URI 写成 “xxx/problemset/paper?pids=1,2,3”

### 3.2.2 SubjectController 类

该 controller 处理专题库相关的所有请求，根 URI 为 “/subject”，以下是各个 action 对应的方法。

● index

URI	一般为空或者 “/”; (GET)
功能	仅返回专题库主页。

● importSubjects

URI	“/import/”; (GET)
功能	导入磁盘上的专题文档至数据库。

● listSubjects

URI	“/list”; (GET)
功能	返回所有专题内容，格式为 JSON。

### 3.2.3 ImageController 类

该 controller 处理素材库（一般为图片）相关的所有请求，根 URI 为 “/image”，以下是各个 action 对应的方法。

- index

URI	一般为空或者 “/”；（GET）
功能	仅返回素材库主页。

- importImages

URI	“/import/”；（GET）
功能	导入磁盘上的素材信息至数据库，以及图片实体文件至文件系统。

- listImages

URI	“/list”；（GET）
功能	返回所有素材信息的数据，格式为 JSON。

- readImage

URI	“/read{imageNo}”；（GET）
功能	返回指定素材的实体文件，格式为图片本身的格式。URI 中的 imageNo 为素材的 ID。

### 3.2.4 FlashController 类

该 controller 处理动画库（一般为 Flash）相关的所有请求，根 URI 为 “/flash”，以下是各个 action 对应的方法。

- index

URI	一般为空或者 “/”；（GET）
功能	仅返回动画库主页。

- importFlashes

URI	“/import/”；（GET）
功能	导入磁盘上的动画信息至数据库，以及动画实体文件至文件系统。

- listFlashes

URI	“/list”；（GET）
功能	返回所有动画信息的数据，格式为 JSON。

- readFlash

URI	“/read{flashNo}”；（GET）
功能	返回指定动画的实体文件，格式为图片本身的格式。URI 中的 flashNo 为动画的 ID。

### 3.2.5 ConfigController 类

该类用于在浏览器端直接设置系统的参数，待改进。

- setConfig

URI	“/set”; (GET)
功能	用于手动配置系统参数，但不建议使用。目前可以接受的参数为 url，即设置项目的系统路径

### 3.3 Service 接口设计

Service 层用于处理四个库的具体业务逻辑，为 Controller 层提供接口，数据交换格式为 Java Bean（见 bean package，这里不详述）。

#### 3.3.1 ImplProblemsetService 类

该 Service 处理题库相关业务逻辑，实现 ProblemsetService 接口，具体 public 方法如下。

- convertDoc2Html

参数	String docPath, String htmlPath
返回	是否成功
功能	将 doc 文档转换成 html 文本，便于以后解析。参数 docPath 为 doc 的绝对路径，htmlPath 为 html 保存的绝对路径。转换过程由 Apache 的 POI 库提供支持。

- splitProblem

参数	String htmlPath
返回	导入题目的数量
功能	解析 html 文本，对每道题目进行切割，并提取出题目的相关信息，包括题型、难度、知识点、题干和答案，并保存至数据库。 对于本来的 doc 文档，有一些事先规定的记号用于标识题目的内容。其中，“//”开头的行为注释，不用解析。 “\$”之后为题型，且独占一行，在遇到下一个“\$”前，题目都为该题型。 “&”之后为难度，且独占一行，在遇到下一个“&”前，题目都为该难度。 “@1”独占一行，表示在遇到下一个标记前，接下来几行的内容为题干。 “@2”独占一行，表示在遇到下一个标记前，接下来几行的内容为答案。 “@3”独占一行，表示在遇到下一个标记前，接下来几行的内容为知识点。 Doc 的文档一般保存为 UTF-8 编码，且允许插入 jpg，png 等常用格式的图片。

- findProblems

参数	String[] types, Integer[] diffs, String[] contents, String[] knows, int offset
返回	Problem 类型的 List
功能	搜索符合给定条件的题目。五个参数分别对应题目的五个过滤条件，即类型，难度，题干，知识点和题目列表中的位置偏移。

- readPicture

参数	int prId
返回	DSPicture 类型的 bean

功能	读取题目中包含的图片信息和实体文件。参数为图片的 id。
----	------------------------------

● makePaper

参数	Integer[] pids
返回	Doc 文件的 byte 数组
功能	根据题目 ID 的列表，组成试卷。试卷的标题、题型顺序都预先固定。

● removeAll

参数	无
返回	无
功能	完全删除数据库中题目的数据，以及文件系统上题目图片文件。

### 3.3.2 ImplSubjectService 类

该 Service 处理专题相关业务逻辑，实现 SubjectService 接口，具体 public 方法如下。

● importSubject

参数	String path
返回	导入专题的数量
功能	从 csv 文件导入专题至数据库。 对于 csv 文件，每列分别为专题名称、专题描述、主题分类和外部 URL。

● findSubjects

参数	无
返回	Subject 类型 List
功能	获取所有专题。

● findSubjects

参数	String[] contents
返回	Subject 类型 List
功能	根据条件搜索专题内容。参数 contents 为多个关键词。

### 3.3.3 ImplImageService 类

该 Service 处理素材相关业务逻辑，实现 ImageService 接口，具体 public 方法如下。

● importImage

参数	String path
返回	导入素材的数量
功能	根据磁盘上素材文件，将其描述信息导入数据库。

● findImages

参数	无
返回	Image 类型 List
功能	获取所有素材信息。

- findImages

参数	String[] contents
返回	Image 类型 List
功能	根据条件搜索素材信息。参数 contents 为多个关键词。

- readImage

参数	byte[] imageNo
返回	素材文件的字节流 byte[]
功能	根据参数 imageNo 给定的素材 ID 读取素材文件。

### 3.3.4 FlashController 类

该 Service 处理动画相关业务逻辑，实现 FlashService 接口，具体 public 方法如下。

- importFlash

参数	String path
返回	导入动画的数量
功能	根据磁盘上动画文件，将其描述信息导入数据库。

- findFlashes

参数	无
返回	Flash 类型 List
功能	获取所有动画信息。

- findFlashes

参数	String[] contents
返回	Flash 类型 List
功能	根据条件搜索动画信息。参数 contents 为多个关键词。

- readFlash

参数	byte[] flashNo
返回	动画文件的字节流 byte[]
功能	根据参数 flashNo 给定的动画 ID 读取动画文件，一般为 swf 格式。

## 3.4 DAO 接口设计

### 3.4.1 JdbcBaseDao

该类为所有 DAO 的基类，使用 Spring 的 JdbcTemplate 来管理数据库连接。所有的 CRUD 操作均由基本的 SQL 实现，特别之处是，每个 CRUD 的方法均封装为以 Java Bean 为交换数据的接口，即对数据库和 Bean 之间做了映射，使上层模块调用更加方便。实现原理利用 Java 的反射机制和 JdbcTemplate 类中提供的方法进行封装，这里不再详述。

### 3.4.2 其它 DAO

剩余的 Dao 均继承自 JdbcBaseDao 并实现相关接口。



### 3.5 其他接口设计

Config 类为系统整体的配置信息，程序启动时会自动初始化。程序会根据不同的操作系统，判断文件系统上默认的存储路径（详见 Config 类），**系统使用前需要将必要的文件资源先放在这些目录下。**

ROOT_PATH	文件系统上的存储资源的根路径
PROBLEM_PATH	题库资源的根路径
TMP_PATH	临时目录的根路径

LibType 枚举类表示各个库，ProblemType 枚举类表示题目类型。这些类一般为常量。

### 3.6 Web 设计

目前比较完善的页面为题库，用 jQuery 和 Bootstrap 重新实现，其余三个库的页面仍然保持着 Extjs 的风格，需要重新实现。

对于题库页面（具体实现参考 jq-problemset.js 文件），普通的事件响应由 jQuery 实现，页面滚动加载参考修改后的 scrollpagination.js 插件。提示框由 jQuery 的 qtip 插件支持。除了上述提到的 JS 库以外，还用到了 underscore.js（模板复用），ie-css.js（IE 兼容性）等。

## 4. 小结

总的来说，本系统基本实现了各个库的展示功能，其中题库的实现相对最完善。其中，系统的优势包括几点：

- 以 REST 形式实现的接口达到了很好的松耦合状态；
- 界面部分实现由之前的 Extjs 转换为现在的 jQuery，从而对比较古老的浏览器也有比较好的支持；

可改进的地方有如下几点：

- 将危险的 controller 接口改为 POST 方式，集成至统一的管理界面，让管理员管理各个资源的导入和删除；
- 对资源的管理略显简单，CRUD 的操作可以具体化；
- 增加用户权限模块；
- 增加其它功能需求；