

Web 应用基础 Project 2

需求文档

2015-05-04

目录

1. 引言	4
1.1 项目目的	4
1.2 项目范围	4
1.3 项目答疑	4
1.4 时间安排	4
2. 项目概述	5
2.1 项目背景	5
2.2 功能及业务简介	5
2.3 项目目标	5
3. PROJECT 2 需求	6
3.1 总体要求	6
3.1.1 技术要求	6
3.1.2 质量要求	6
3.1.3 多语言支持	7
3.1.4 浏览器兼容性	7
3.1.5 代码风格	7
3.1.6 其它	7
3.2 注册及登陆页面	7
3.3 首页	8
3.4 发现页	8
3.5 问题页	9
3.6 个人信息页面	9
3.7 其它功能	9
3.7.1 提问 / 回答 / 评论	9
3.7.2 关注功能	10
3.7.3 私信功能	10
3.7.4 评论展开 / 收起	10
3.7.5 分页显示功能	11
3.7.6 更改个人信息	11

3.7.7 搜索功能	11
4. PROJECT 2 详细评分点	12
5. PROJECT 2 提交物	13
6. PROJECT 2 可利用资源	13
6.1 工具	13
6.2 技术建议	14

1. 引言

1.1 项目目的

项目目的在于巩固和检验 Web 应用基础课程中学到的知识，通过进行一个实际的项目开发，从而锻炼同学们的 Web 综合开发能力。

1.2 项目范围

本学期 Project 所涉及的主要技术大部分在课程中都有介绍和讲解，部分技术还会在平时的 Lab 中有所体现。Project 2 主要涉及：

- 网站后台
- 数据库

1.3 项目答疑

- 需求方面的疑问请联系 Project 2 负责 TA（11302010035@fudan.edu.cn）。
- 技术方面的疑问请按照课程分组向对应的负责 TA 或任课教师询问。

1.4 时间安排

Project 2 截止时间：2015 年 6 月 25 日（星期四）20:59:59（面试截止时间）

TA 在面试截止时间前接受面试预约，请同学们至少提前三天向各负责 TA 确认面试时间。面试之后请将源代码上传至 FTP。

延期一天在项目总得分中扣除 5%，延期两天扣除 10%，依此类推。请大家尽快开始 Project 2 的设计和开发工作，切忌在 Deadline 临近时突击赶工。

2. 项目概述

2.1 项目背景

在完成了 Project 1 的工作之后，前端的界面设计已经完成了大半。Project 2 主要是设计系统的后台，从而使 Project 1 的工作更加有意义，成为一个真正可以使用的网站系统。

2.2 功能及业务简介

- 用户可以注册、登陆。
- 用户登陆后，可以提问（仅文字）、回答（文字和图片，不要求音频和视频）、对回答作出评论（仅文字）。
- 用户之间可互相关注、互发私信（互发私信不要求关注关系）。
- 用户可以搜索包含特定关键字的问题和用户。

为了简化系统的实现，该系统没有话题的概念，没有对问题的关注，也没有点赞或取消点赞功能。

具体功能及要求在后面会详细说明。

2.3 项目目标

Project 2 的目标是在 Project 1 的样板页面的基础上，实现初步的后台程序，并通过简单的数据库操作对用户所有相关信息进行记录和查询。

3. Project 2 需求

3.1 总体要求

3.1.1 技术要求

该课程 Project 主要是为了让大家熟悉并体验 Web 开发的一般流程，因此对技术的选用不会有严格的要求。

- 后台语言：不限
- 框架：允许使用，不限
- 数据库：不限
- 前端库和框架：允许使用，不限

技术的选用不作为评分点，即最终评分主要看功能的实现效果。

该课程中主要涉及到的技术有：JSP/Servlet、MySQL、jQuery

3.1.2 质量要求

- JS 脚本能够正确执行，在要求的浏览器中不得出现脚本错误。
- 系统性能良好，从用户在浏览器提交操作到完全输出响应页面时间间隔不超过 5 秒。
- 无论用户输入正确的还是错误的数据，系统均能正确响应。如果用户输入错误的数据，应当返回数据输入页面，并提示输入的数据不符合要求；不得显示程序的堆栈信息报错页面或 HTTP 5XX 页面。此项要求仅需要保证正常用户的使用便捷性，对于蓄意的攻击（如 SQL 注入等）不在考察范围内。
- 系统需要全面支持中文。
- 对未登录用户应有一定的访问限制（例如只能访问发现页面而不能查看问题的具体回答。具体限制哪些功能大家自行设计）。

3.1.3 多语言支持

所有页面应当一致使用 UTF-8 编码，做到正确支持 CJK 字符的显示和输入。任何页面的任何位置都不能出现乱码。

另外 URL 中尽量避免传中文，如果一定要传，请 Encode 之后再传。

3.1.4 浏览器兼容性

页面应在如下浏览器中有较为一致的显示(“较为一致”指的是各功能区位置相对关系、色彩、背景图片及其位置、文本显示样式、元素存在性没有明显差异, 不包括浏览器内建的 HTML5 控件的显示差异), 脚本执行应当一致:

- IE9+
- Chrome 最新版本
- Firefox 最新版本

3.1.5 代码风格

- 代码清晰可读，应有适当的注释。
- 前后台尽量分离，例如 JSP 中不得出现连接数据库之类的操作。

3.1.6 其它

- 所有提交页面均不得抄袭他人的或网络上的页面代码。
- 鼓励同学们实现文档中未要求的功能，但请自己编写代码。

3.2 注册及登陆页面

必须完成的业务逻辑点如下：

- 前端和后台都要进行表单验证：前端表单验证不通过的情况下表单不能提交，后台表单验证不通过的情况下应当在前端显示错误提示信息。
- 注册：如果用户输入的用户名或邮箱在数据库中已经存在，则不予注册。注

册成功后跳转到发现页面。

- 登陆：登陆成功后跳转到首页。登陆失败时，页面应当提示“用户名或密码错误”，但不得分开提示“用户名错误”或“密码错误”。具体提示方式不限。
- 热门用户：通过查询数据库，按用户的粉丝数降序排列，取前若干个进行展示。
- 热门问题：通过查询数据库，按问题的回答数降序排列，取前若干个进行展示。

3.3 首页

首页是用户登陆成功后自动跳转到的页面，主要展示该用户所关注的人的动态。该页面的整体排版与问题页面类似，是左右两栏结构。

在此页面，动态有如下定义：

- 该用户所关注的人提出了一个问题，动态时间为提问时间
- 该用户所关注的人回答了一个问题，动态时间为回答时间

将上述动态按照时间降序排列显示在页面上，显示内容为：

- 如果是提问，则显示问题标题（不用显示问题描述）、提问者、提问时间
- 如果是回答，则显示原始问题标题，回答者，回答时间，以及回答内容

上述内容的显示可自行排版，但是应当符合一般的审美，不能够过于简陋（例如使用表格）。同时需要注意：

- 问题标题点击后可跳转到相关的问题页面
- 用户名点击后可跳转到相应的用户个人信息页面

3.4 发现页

发现页面主要显示系统中的所有问题，并按照最新活跃时间降序排列。

最新活跃时间定义如下：

- 如果该问题没有回答，则使用该问题的提问时间
- 如果该问题有回答，则使用该问题的最新回答时间

在页面上显示如下：

- 如果该问题没有回答，则显示问题标题、提问时间
- 如果该问题有回答，则显示问题标题、最新回答相关信息（回答者、回答时间、回答内容，不需要评论数及评论内容）

3.5 问题页

问题页面主要显示一条问题及其相关的回答、评论等内容。需要完成的功能点：

- 回答的排列按照时间升序
- 每条回答包括用户头像、用户名、用户简介、回答内容、回答时间、评论数。
- 默认情况下评论是关闭的，当点击了评论之后才展开评论框。评论框内的评论排列按照时间升序。

3.6 个人信息页面

该页面主要展示用户的基本信息，基本布局和内容显示按照 Project 1 中的要求。

其中最近我关注的人和最近关注我的人取前若干个进行展示。

3.7 其它功能

3.7.1 提问 / 回答 / 评论

提问：

- 用户点击提问按钮之后，会弹出提问框，让用户输入问题标题（必填）和问题描述（可选）。具体可参考知乎的效果。
- 提问成功后，自动跳转到相关的问题页面。

回答：

- 回答可以是纯文字，可以是纯图片，也可以是文字加图片，但是不能够什么也没有。
- 需要有图片上传功能。可以是点击提交表单的时候上传，也可以 Ajax 异步上传。实现方式对评分无影响。

- 为了简化实现，不要求多图上传。另外在显示回答的时候图片显示在文字最后即可，不要求图文混排。
- 不要求富文本功能。

评论：

- 评论只需要纯文字即可。

3.7.2 关注功能

关注功能需要实现即可，具体交互方式不限，以下交互供参考（可以实现其中一种，也可以全部实现，也可以使用你自己设计的交互方式）：

- 进入其他用户个人信息页面，该页面上有“关注 / 取消关注”按钮
- 当鼠标移到用户名上时，会弹出小卡片，显示用户基本信息，并有“关注 / 取消关注”按钮

3.7.3 私信功能

该功能实现即可，具体交互方式不限，以下交互供参考（可以实现其中一种，也可以全部实现，也可以使用你自己设计的交互方式）：

- 点击进入其他用户的个人信息页面，该页面上有“发私信”的按钮，点击按钮进行发私信
- 有一个专门的私信页面，在表单里输入对方用户名和私信内容，从而实现发私信

其他要求：

- 私信页面显示无特殊要求，同学们可自行设计布局
- 要求对已读和未读的私信进行标注区分
- 用户登录后，如果有未读的私信，应当在菜单栏里有所提示

3.7.4 评论展开 / 收起

该功能的实现主要有两种方式：

- 在加载页面的时候一次性将所有数据取出，然后默认情况下评论不显示，点击展开后将评论显示出来
- 当点击显示评论的时候通过 Ajax 从后台取数据，然后展示

具体实现方式不做硬性要求，但是鼓励大家使用第二种方式。使用第一种方式的在评分时将会扣除 1 分。

3.7.5 分页显示功能

该功能只针对回答。评论不作要求，全部显示就行。可选方案有以下两种：

- 使用传统的分页方式，即点击页码进入相应页面。
- 不显式分页，使用类似知乎那样的瀑布流方式，滚动到页面底部的时候点击“显示更多”来显示更多的答案。

具体实现方式二选一即可。对于任一方式，都有真分页和假分页的区别：

- 假分页：页面加载时一次性取出所有数据，然后隐藏部分数据，点击进入下一页或者显示更多的时候才显示另外的一部分数据。
- 真分页：加载时只加载当前显示的数据，当点击进入下一页或者显示更多的时候，实时地向服务器发请求获取数据。

使用假分页方式的将会扣除 1 分。

只需要在问题页面对用户的答案实现该功能即可。首页和发现页不做分页要求。

3.7.6 更改个人信息

用户可以编辑更改个人信息，包括：个人简介、密码、头像。用户名和邮箱一旦注册成功不可更改。

- 个人信息更改后，其他用户应能够看到这个改变。
- 头像的更改过程中，无需对图片做任何裁剪、压缩等操作。

3.7.7 搜索功能

- 用户在搜索框中输入字符串之后，程序将在数据库中搜索包含该字符串的问题（只需要匹配问题标题即可，不需要匹配问题内容）和用户名，最终显示在前端页面上。
- 搜索结果可以分类（按用户和问题分类）显示，也可以不分类。页面布局可自行设计。
- 不要求搜索结果中匹配字符串的高亮显示。

4. Project 2 详细评分点

满分为 100 分。同学们可参考表格中的得分点进行页面设计，评分时严格按照表格中的各项进行评分，理论上不会扣除表中不存在项的分数。

功能	检查点	分值
总体要求（10）	中文支持，无乱码	2
	代码风格良好，简洁明了，有适当注释，无过多冗余代码	2
	性能良好	2
	未登录用户权限限制	2
	浏览器兼容性	2
注册及登陆页面（14）	注册整体功能逻辑	3
	用户名或邮箱已存在时错误提示	2
	登陆功能整体逻辑	3
	登录失败提示	2
	热门用户、热门问题展示正确	4
首页（8）	动态内容展示正确	6
	问题标题和用户名点击后跳转到相应的页面	2
发现（8）	页面主体内容展示正确	6
	问题标题和用户名点击后跳转到相应的页面	2
问题页面（16）	用户答案展示正确	4
	评论展示正确	4
	评论可展开 / 折叠	4
	答案分页展示	4
个人信息页面（4）	提问和回答能够正确展示	2
	最近我关注的人和最近关注我的人显示正确	2
提问 / 回答 / 评论（12）	点击提问按钮弹出浮层，显示提问框	2
	提问表单逻辑正确，非法表单内容不可提交	1
	提问成功后自动跳转	1
	回答能够正确提交，提交后可显示在该页面上	2
	回答中的图片上传	4

	评论能够正确提交，评论提交后显示在该页面上	2
关注功能（4）	用户之间可互相关注 / 取消关注	4
私信功能（5）	用户之间可以互发私信	2
	私信有已读和未读标识，用户登陆后如果有未读私信应当在菜单栏里有所提示	3
个人信息更改（5）	个人简介、密码、头像可更改，用户名和邮箱不可更改	2
	头像能够正确更改	3
搜索功能（4）	能够正确进行搜索	4
整体效果分（10）	整体界面设计良好，各信息正确显示，布局合理。用户体验良好，交互流畅。	10

5. Project 2 提交物

- 所有相关的源代码文件、数据库表结构的 SQL 备份、通过演示之后系统数据的 SQL 备份、素材等。

以上内容打成一个压缩包，命名为“学号_姓名.zip”，上传至 FTP。

6. Project 2 可利用资源

6.1 工具

- 如果使用 Java 进行开发的话，Eclipse 是非常有用的代码编辑器和调试器；同时可选用 Tomcat 作为服务器。
- 如果使用的是 MySQL 数据库，MySQL WorkBench 或 Navicat for MySQL（收费软件，有破解版）对于数据库的修改和查询非常有帮助。

6.2 技术建议

以下为 TA 的建议，感兴趣的同学可以作为一个学习方向进行参考学习。（不作为加分项）

- 前端允许使用 CSS 框架（例如 **Bootstrap** 等），允许使用各种 JavaScript 库（例如 **jQuery**、**Underscore** 等）和框架（例如 **AngularJS**、**Backbone**、**React** 等）。有兴趣的同学可以自行学习使用。
- 前后端分离：可以通过“前端模板+Ajax+Restful 接口”的方式实现前后端的分离，从而使整体结构更加清晰，也可以避免后台大量的标签使用或者字符串拼接。