实现简单的微博平台

摘 要

本次课程综合设计我们组完成一个简单的微博平台。我们通过XAMPP（Apache+MySQL+PHP+PERL）集成软件包，主要使用PHP、JSP（Java Server Pages）和Html（HyperText Mark-up Language）等语言完成了用户注册、发表微博、关注或取关其他用户、评论、查看关注用户和自己所发微博等所有基本要求，此外我们还加入了微博大厅等新的内容。最后我们使用Html+CSS（Cascading Style Sheets）来美化界面。

首先我们实现了用户的注册和登陆功能，首先在登陆界面，如果是新用户则需要点击注册进入注册界面，在进行表单的填写与提交后，后台通过链接MySQL实现用户数据的存放，并在这个过程中测试了用户名是否合法等问题。最后回到登陆界面，用户输入合法的账号与正确的密码即可登陆微博系统。

接着在用户登陆成功后界面将跳转到微博大厅，在微博大厅用户可以查看所有用户发表的微博，可以点击自己或者其他用户查看该用户的个人界面。然后在用户个人界面会显示当前用户所发的所有微博，如果是处于查看其他用户状态则能够关注或者取关该用户，如果是处于查看自己个人界面则能够点击关注查看自己的所有关注用户。

然后在微博大厅或者用户个人界面均可以发表微博，同时点击单条微博可以查看该微博的所有历史评论，并且在下面还能添加自己的评论内容。在这里为了避免用户的重复输入，我们选择当用户输入文字超过指定长度后就立即提示超出字数限制并截取字数限制的内容的方法来保证用户拥有一个较好的体验。

最后使用前端技术美化了整个微博系统的界面。

**关键词**：微博；PHP；XAMPP

目 录

[第1章 所完成的工作任务 1](#_Toc483664958)

[1.1所要求的微博详细功能描述 1](#_Toc483664959)

[1.2所完成的微博详细功能描述 1](#_Toc483664960)

[第2章 需求分析 2](#_Toc483664961)

[2.1背景 2](#_Toc483664962)

[2.2术语与缩写解释 2](#_Toc483664963)

[2.3产品面向的用户群体 2](#_Toc483664964)

[2.4微博详细功能描述 2](#_Toc483664965)

[2.5功能性需求分类 3](#_Toc483664966)

[第3章 概要设计 3](#_Toc483664967)

[3.1数据库设计 3](#_Toc483664968)

[3.1.1 user表 3](#_Toc483664969)

[3.1.2 entries表 4](#_Toc483664970)

[3.1.3 fomments表 4](#_Toc483664971)

[3.1.4 follownfans表 4](#_Toc483664972)

[3.2 系统功能结构设计 5](#_Toc483664973)

[3.2.1注册 5](#_Toc483664974)

[3.2.2登陆 5](#_Toc483664975)

[3.2.3个人主页 5](#_Toc483664976)

[3.2.4微博大厅 5](#_Toc483664977)

[3.2.5发布微博 6](#_Toc483664978)

[3.2.6评论 6](#_Toc483664979)

[3.2.7关注或取关 6](#_Toc483664980)

[3.3 前端设计 7](#_Toc483664981)

[3.4 微博系统功能流程图 7](#_Toc483664982)

[第4章 系统功能设计 8](#_Toc483664983)

[4.1 用户信息管理 8](#_Toc483664984)

[4.1.1用户注册 8](#_Toc483664985)

[4.1.2用户登陆 9](#_Toc483664986)

[4.2 微博信息管理 10](#_Toc483664987)

[4.2.1发表微博 10](#_Toc483664988)

[4.2.2评论微博 11](#_Toc483664989)

[4.2.3删除微博 11](#_Toc483664990)

[4.3 好友信息管理 12](#_Toc483664991)

[4.3.1查看粉丝 12](#_Toc483664992)

[4.3.2查看关注 12](#_Toc483664993)

[4.3.3关注用户 13](#_Toc483664994)

[4.2.4取关用户 14](#_Toc483664995)

[4.4 主界面 15](#_Toc483664996)

[4.4.1微博大厅 15](#_Toc483664997)

[4.4.2个人界面 16](#_Toc483664998)

[第5章 总结与展望 17](#_Toc483664999)

# 第1章 所完成的工作任务

## 1.1所要求的微博详细功能描述

微博平台的功能包括：

1. 用户注册；
2. 用户发表不超过139字的文本（即微博）；
3. 用户关注和取消关注其它用户；
4. 用户对微博加入评论（不超过100字）；
5. 用户主页可查看关注用户和自己发表的微博。

## 1.2所完成的微博详细功能描述

1. 用户注册与登陆：实现了新用户的注册包括主要的用户ID与密码，用户的其他信息如年龄等。登陆则实现了已注册用户登录到微博系统。
2. 用户发表不超过139字的文本（即微博）：在微博大厅或者当前用户个人界面均可以点击发表以发表微博，每条微博包括标题和不超过139字的文本内容。当输入文本超过字数限制后会自动提示并只截取前139个字。
3. 用户关注和取消关注其它用户：在其他用户个人界面下可以点击关注或者取关来关注或取关该用户。
4. 用户对微博加入评论（不超过100字）：点击进入单条微博后，可以看到该微博的历史评论，并且可以在下面添加自己的微博评论（不超过100字）。当输入文本超过字数限制后会自动提示并只截取前100个字。
5. 用户主页可查看关注用户和自己发表的微博：在当前用户个人界面可以点击关注查看自己所关注的用户列表。
6. 微博大厅：在这里会显示所有用户所发的微博，可以查看陌生人的微博，进入陌生人的个人界面进行关注、评论等等操作，增强与其他用户互动。

# 第2章 需求分析

## 2.1背景

人离不开社交，精神生活方式丰富多彩。微博只言片语的“语录体”让人们能够即时表露自己的状态，这更加贴切现在人的生活节奏和习惯。微博下的留言互动让人参与到社交媒体中来，用户可以在微博上分享一些即时的新闻。为了人们更好的社交，更及时的分享自己身边的事以及用户之间的社交互动。为了适应用户需求，我们推出微博以满足人们这方面的社交需求。

## 2.2术语与缩写解释

XAMPP（Apache+MySQL+PHP+PERL）是一个功能强大的建站集成软件包。

PHP（外文名:PHP: Hypertext Preprocessor，中文名：“超文本预处理器”）是一种通用开源脚本语言。

JSP全名为Java Server Pages，中文名叫java服务器页面，其根本是一个简化的Servlet设计，它是由Sun Microsystems公司倡导、许多公司参与一起建立的一种动态网页技术标准。

Apache（Web服务器）是世界使用排名第一的Web[服务器](http://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8)软件。

mySQL是一个关系型数据库管理系统。

## 2.3产品面向的用户群体

此次的微博主要面向的用户群体：社会人群。

## 2.4微博详细功能描述

微博平台的功能包括：

1. 用户注册；
2. 用户发表不超过139字的文本（即微博）；
3. 用户关注和取消关注其它用户；
4. 用户对微博加入评论（不超过100字）；
5. 用户主页可查看关注用户和自己发表的微博。

## 2.5功能性需求分类

表1 功能性需求分类表

|  |  |  |
| --- | --- | --- |
| 功能类别 | 功能名称 | 描述 |
| 微博用户 | 注册 | 注册一个新的账号 |
| 登陆 | 使用已经注册成功的账号登陆微博系统 |
| 发表微博 | 发表一条不超过139字的文本（微博） |
| 评论微博 | 在微博下面评论（不超过100字） |
| 关注用户 | 关注其他用户 |
| 取消关注 | 取消已经关注的用户 |
| 查看关注用户列表 | 查看已经关注的所有用户列表 |
| 查看用户个人主页 | 查看某个用户的个人主页（微博、粉丝等） |
| 进入微博大厅 | 能够看见系统上陌生用户所发的微博 |

# 第3章 概要设计

## 3.1数据库设计

针对该微博的主要功能，我们设计了如下数据库，数据库名为blogs，包含4个数据表，用户信息资料表user，发表内容entries，评论内容comments和粉丝表follownfans，下面给出这4个表的字段设计和基本属性。

### 3.1.1 user表

此表包含了用户的注册信息

表2 user表

|  |  |  |  |
| --- | --- | --- | --- |
| 逻辑字段 | 物理字段 | 类型（长度） | 规则 |
| 用户名 | userid | varchar(10) | 主键 |
| 用户年龄 | age | int(4) | 非空 |
| 密码 | password | varchar(20) | 非空 |

### 3.1.2 entries表

此表包含发布的微博信息

表3 entries表

|  |  |  |  |
| --- | --- | --- | --- |
| 逻辑字段 | 物理字段 | 类型（长度） | 规则 |
| 微博编号 | id | int(5) | 主键 |
| 标题 | title | varchar(50) | 非空 |
| 日期 | dates | date | 非空 |
| 内容 | Contents | text | 非空 |
| 发布人 | userid | varchar(10) | 非空 |

### 3.1.3 fomments表

此表包含了评论信息

表4 fomments表

|  |  |  |  |
| --- | --- | --- | --- |
| 逻辑字段 | 物理字段 | 类型（长度） | 规则 |
| 评论用户 | Userid | varchar(10) | 非空 |
| 评论日期 | Dates | date | 非空 |
| 评论内容 | Contents | text | 非空 |
| 评论编号 | Commentsid | Int(5) | 主键 |
| 微博编号 | Entryid | int(5) | 非空 |

### 3.1.4 follownfans表

此表包含了关注情况

表5 follownfans表

|  |  |  |  |
| --- | --- | --- | --- |
| 逻辑字段 | 物理字段 | 类型（长度） | 规则 |
| 用户名 | Userid | varchar(10) | 非空 |
| 粉丝用户名 | Followorfans | varchar(10) | 非空 |
| 关注/粉丝 | Tag | int(1) | 非空 |

## 3.2 系统功能结构设计

### 3.2.1注册

注册界面，通过注册建立账户。注册界面主要是实现用户的账号、密码、确认密码以及其他信息的输入，当点击确定后会通过从已有数据库核对注册信息是否合法，两次密码输入是否一致，用户名是否已被占用等。如果全部无误就显示注册成功并跳转到登陆界面。否则显示注册失败信息并停留在注册页面。

### 3.2.2登陆

登陆界面，通过登陆进入微博大厅。登录界面通过输入用户账号以及密码，提交后将通过连接数据库来检查输入是否合法，用户账号是否合法在已有数据库中，密码是否正确。若都正确无误，那么提示登陆成功并跳转到微博大厅界面。否则返回登陆界面。同时若没有账号可以通过注册按钮跳转到注册界面。

### 3.2.3个人主页

显示当前登录用户的个人主页，或是所查看的其他用户的个人主页，在个人主页界面中提供跳转回微博大厅的入口，用户能够在该页面查看自己或是所查看其他用户所发布的所有微博内容，并且能够对微博进行评论。若所查看为其他用户，则可以添加关注或取消关注，若所查看为自己的个人主页，则可以对自己发布过的微博进行删除管理。

### 3.2.4微博大厅

有关微博大厅的数据库内容表项有微博正文、发表微博的用户、微博ID、发表时间等，在微博大厅页面实现群平台微博的展示，按照时间最近为优先进行排列，登陆用户可以在该页面查看全平台用户发表的微博，并且可以对微博进行评论，也可以对感兴趣的陌生用户添加关注，或是对已经添加关注的用户取消关注，同时提供发布微博的入口，使用户可以发布自己的微博。同时在页面中显示当前用户的登录名，提供前往用户个人主页的入口。也提供当前登录用户的关注人数和粉丝人数显示，并且可以通过点击关注数查看自己关注的用户列表，并能通过列表中的用户名进入用户个人页面。

### 3.2.5发布微博

点击发布按钮后，当前登陆的用户可以发布新的微博，发表后提示发布成功，并返回道微博大厅。同时会实时检测所写内容是否超过字数限制，若超过则会给出提示。

### 3.2.6评论

点击某条微博的评论按钮后，能够对点击的微博内容发表评论，发表后提示评论成功，并返回到微博大厅。同时会实时检测所写内容是否超过字数限制，若超过则会给出提示。

### 3.2.7关注或取关

为了完成功能“用户关注和取消关注其它用户”，我们将要完成以下内容：

（1）用户的微博大厅界面显示关注数/粉丝数

（2）点击关注数/粉丝数可以查看详细名单

（3）点击详细名单里的用户名可以进入该用户的个人界面

（4）关注/取关按钮

## 3.3 前端设计

为了使前端的设计更加简单和高质量，项目中使用bootstrap前端框架进行开发。bootstrap是一个由Facebook开发并使用的前端框架，其中提供了很多已有的样式表以及各种组件，可以很方便地写出比较美观的响应式布局。

界面的风格为bootstrap的默认风格。所有的功能页面（即除了登录的其他页面）顶端均有一个导航条，其最左边为返回主页（index.php）的链接，最右端为当前登录的用户名。

在主页（index.php）中，页面的右侧设置了一个显示当前用户信息，即用户名、关注数和粉丝数的侧栏，处在中间的正文部分则显示各个其他用户发表的文章，包括文章标题、发布时间以及发布人，同时加上一个评论的入口连接。个人主页部分的正文也与以上类似。

对于显示关注人和粉丝名单的两个页面，使用流式布局来展示出各个用户的用户名。

## 3.4 微博系统功能流程图

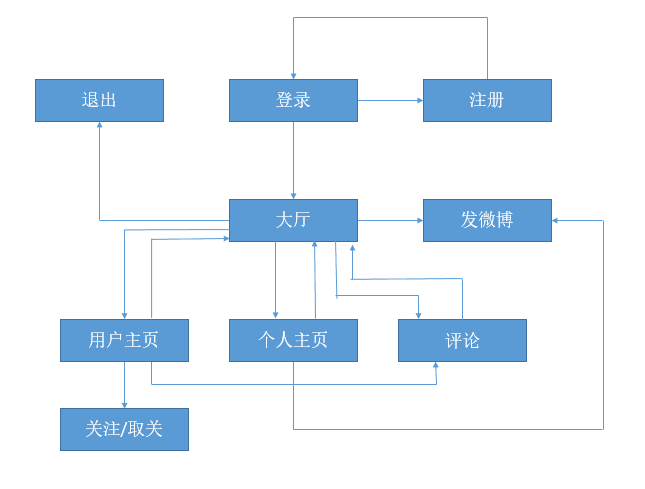


图3.1 微博系统功能流程图

# 第4章 系统功能设计

## 4.1 用户信息管理

### 4.1.1用户注册

注册登录系统，通过注册建立账户。注册界面Register.php主要是实现用户的账号、密码、确认密码以及其他信息的输入，当点击确定后会通过RegisterCheck.php从已有数据库核对注册信息是否合法，两次密码输入是否一致，用户名是否已被占用等。如果全部无误就显示注册成功并跳转到登陆界面。否则显示注册失败信息并跳转回去。

以下是注册界面：



图4.1 用户注册



图4.2 注册成功

### 4.1.2用户登陆

登陆模块，通过登陆进入微博大厅。登录模块Login.php通过输入用户账号以及密码，提交后将通过LoginCheck.php来检查输入是否合法，用户账号是否合法在已有数据库中，密码是否正确。若都正确无误，那么提示登陆成功并跳转到微博大厅界面。否则返回登陆界面。同时若没有账号可以通过注册按钮跳转到注册界面。



图4.3 登陆界面



图4.4 用户登录

## 4.2 微博信息管理

### 4.2.1发表微博

在发表微博界面可以选择返回微博大厅，也可以选择写下自己的微博并发表。

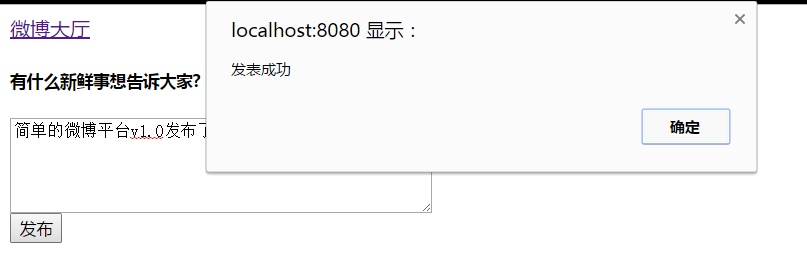


图4.5 发布微博



图4.6 发布微博后的个人主页

当微博内容超过字数限制后，会自动给出提示。

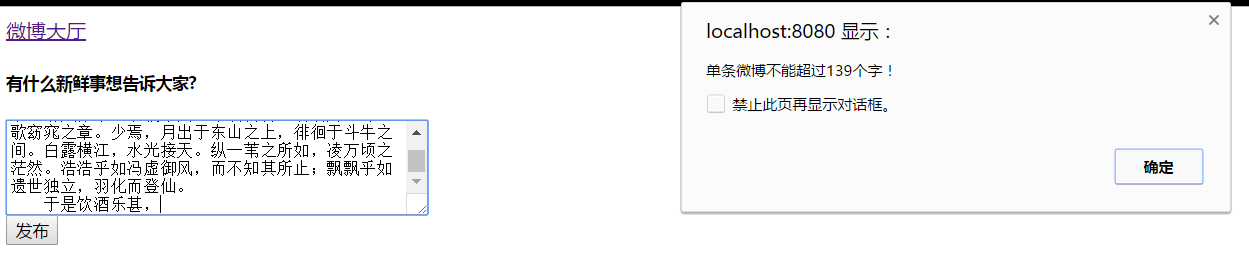


图4.7 发微博超过字数限制的提示

成功发布一条微博后可以在微博大厅和个人主页里面看到该微博信息。



图4.6 发布微博后的微博大厅

### 4.2.2评论微博

用户在微博下点击评论后，可以进入该微博的评论界面，查看该微博的历史评论，并可以在末尾给出自己的评论。

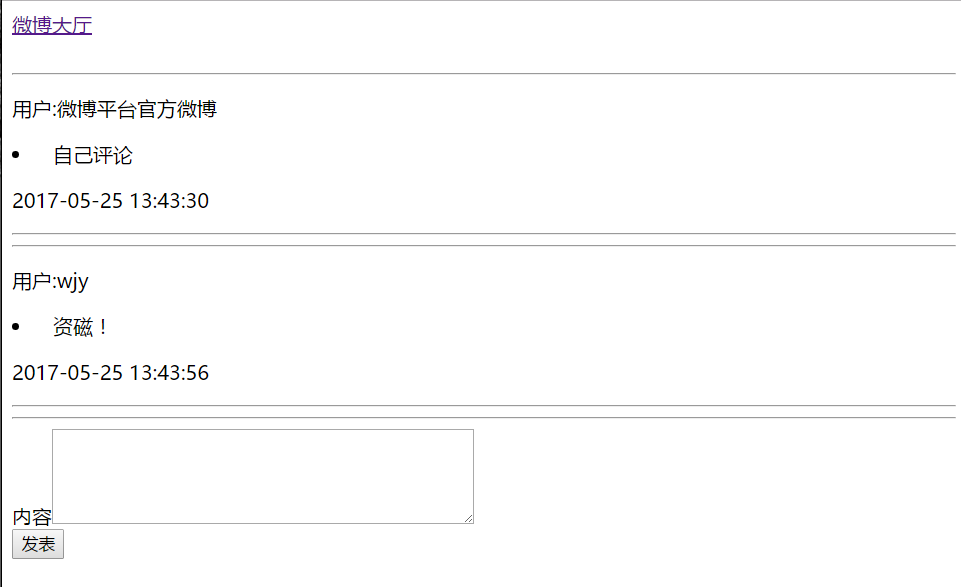


图4.8 微博评论

当评论超过字数限制后，会自动给出提示。

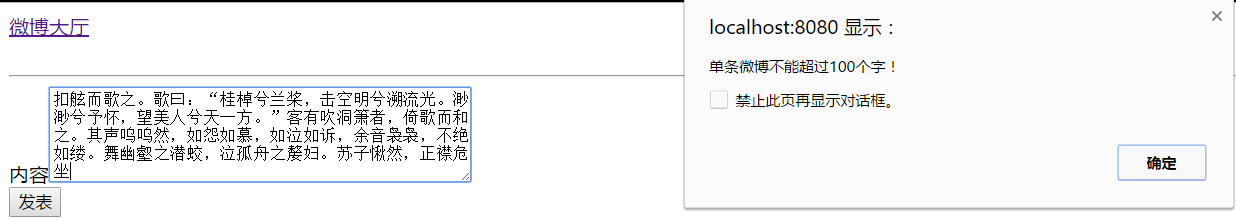


图4.7 发评论超过字数限制的提示

### 4.2.3删除微博

用户可以选择删除自己所发的微博。



图4.9 微博删除



图4.10 微博删除后的个人主页

## 4.3 好友信息管理

### 4.3.1查看粉丝

用户可以在个人主页里查看关注自己的所有粉丝，并且可以点击用户名进入该用户的个人主页。



图4.11 微博平台官方微博的粉丝

### 4.3.2查看关注

用户可以在个人主页里查看自己所关注的所有用户，并且可以点击用户名进入该用户的个人主页。



图4.12 微博平台官方微博的关注

### 4.3.3关注用户

在其他用户的个人主页界面，如果当前用户和该用户是陌生人关系，则可以选择“添加关注”关注该用户。



图4.13 关注陌生用户

关注该用户后，“添加关注”消失，取而代之的是“取消关注”。



图4.14 关注陌生用户成功后的界面

### 4.2.4取关用户

在其他用户的个人主页界面，如果当前用户关注了该用户，则可以选择“取消关注”取关该用户。



图4.15 取关关注用户

取关该用户后，当前用户和该用户是陌生人关系，则显示“添加关注”按钮。



图4.16 取关后的界面

## 4.4 主界面

### 4.4.1微博大厅

登陆后，进入微博大厅界面，在微博大厅界面可以看见所有微博用户所发的微博，在右边个人主页可以进入当前用户的个人主页，点击关注和粉丝数可以查看所有关注的用户和所有关注自己的粉丝。

图4.17 微博大厅

### 4.4.2个人界面

在当前用户个人界面，可以查看自己所发的微博，发表新微博，还可以对单条微博进行评论或者删除操作。

图4.18 当前用户的个人主页

在其他个人用户界面，可以查看该用户所发的微博，还可以对单条微博进行评论操作。

图4.19 其他用户的个人主页

# 第5章 总结与展望

此次综合课程设计，我们小组选择了难度最高的题目，因为我们小组没有人接触过PHP和Python，在一两个月内自学其中一门语言并用其完成课程设计的要求对我们来说无疑是一个不小的挑战，但是我们决定迎难而上去攻克这个最难的课设题目，通过初期的讨论，我们确定了选择PHP来实现课程设计的要求，于是小组成员们便开始马不停蹄的自学PHP，经过一段时间的学习，我们决定开始着手尝试着做做看，同时确定了使用XAMPP集成软件包配合Atom编辑器作为开发软件，开发过程中，我们一边参照着教程代码，一边完成自己的功能，在不断的报错、Bug中不断地熟悉PHP语言的使用，终于在截止日期临近之时完成了此次课程设计。

在此过程中我们也犯了很多的错误，首先是小组成员之间的沟通有很多不到位的地方，出现功能实现有偏差的情况，拖慢了工程进度，其次，因为没有使用版本控制软件，导致每个人手上的版本非常混乱，在整合版本的过程中也浪费了不少时间。通过这些错误，我们学到了不少东西，一方面意识到了合作过程中沟通的重要性，其次认识到版本控制的重要性，受益良多。

总的来说，此次课程设计是大学期间一次不可多得的软件开发经验，虽然我们做的东西距离真正的产品还有十万八千里，但是从中我们依旧能够一窥软件开发过程的一些问题，对今后走上软件开发岗位有些微的用处，我们也会牢记此次课程设计中的失误和教训，尽量避免再次犯同样的错误，尤其是在团队协作问题上，能够尽可能的提升自己的能力，做好自己的本职工作。

**参考文献**

[1] 胡崧.HTML从入门到精通[M].北京：中国青年出版社，2007.

[2] 陈会安.JavaScript基础与实例教程[M].北京：中国电力出版社，2007.

[3] 陈湘扬，陈国益. PHP5+MySQL网页系统开发设计[M].北京：电子工业出版社，2007.