**毕业设计（论文）题目：Web游戏化设计模式的研究与实现**

**学 院：** 软件学院

**专 业：** 软件工程

**班 级：** 08211101

**姓 名：** 张梦菲

**指导教师：** 赵丰年

**校外指导教师：**无

# 题目内容

游戏化（Gamification）简单来说就是将游戏的思维和游戏的机制运用到其他的领域，来引导用户互动和使用的方法。它能在互联网、医疗、教育、金融等领域中影响到用户使用时的心理倾向，进而促进用户的参与和分享。有关游戏化最完整和可靠的定义来源于Deterding、Dixon、Khaled和Nacke ，他们将游戏化定义为“在非游戏背景下使用游戏设计元素（The use of game design elements in non-game contexts）”。在社交网络甚至普通商业网站上，游戏化的应用比比皆是（例如，百度百科中的升级、任务、进度条等机制）。设计模式（Design pattern）是一套被反复使用、多数人知晓的、经过分类编目的、代码设计经验的总结。使用设计模式是为了可重用代码、让代码更容易被他人理解、保证代码可靠性。 本课题要求对游戏化的设计模式（例如任务系统、升级、关卡、社交联系、经验值、虚拟物品、排行榜等）进行研究，总结Web gamification的元素和软件实现方法，并设计开发一个Elearning Web案例（在线学习网站）对研究内容进行验证。

# 任务书

本课题要求对游戏化的机制和对应的设计模式（例如例如任务系统、升级、关卡、社交联系、经验值、虚拟物品、排行榜等）进行研究，结合目前主流网站上的游戏化设置，总结Web gamification的元素和软件实现方法，并使用自己熟悉的Web开发平台设计开发一个ELearning Web案例（在线学习网站）对研究内容进行验证。本系统的设计和开发要求完全按照规范化的软件项目来做，最后实现的系统既要有完整的注释完备的源代码，也要有所有合格软件产品要求的各种文档，并且实现的系统具有一定的可扩展性。

**候选人要求**

1. 精通以下3种主流Web开发工具之一：PHP、.NET或Java；
2. 精通Web设计，熟练掌握HTML、CSS和Web设计模式；
3. 对游戏有热情，熟悉计算机游戏的运行机制，熟悉Web上的游戏机制。

**具体任务**

1. 对游戏化(gamification)和设计模式(design patterns）进行研究；
2. 对Web游戏化（Web gamification）和游戏化设计模式（gameful patterns）进行文献综述；
3. 总结Web gamification的开发方法和开发模式；
4. 使用任意一款主流Web开发平台，设计一个游戏化的ELearning Web系统（可以是C语言学习网上课程、英语学习网站等）；
5. 系统编码，实现实例网站；
6. 优化网站视觉效果；
7. 对整个网站进行测试；
8. 完成毕业设计论文；
9. 完成毕业论文英文资料翻译。

**进度安排**

2014年12月20日~2015年2月15日：研究游戏化和Web游戏化；

2015年2月15日~3月15日：总结Web游戏化开发方法和设计模式；

2015年3月15日~5月15日：设计案例游戏化Web系统，软件开发与实现；

2015年5月15日~5月30日：系统测试及功能测试

2015年4月15日~6月10日：撰写毕业论文，完成相关翻译

# 摘要

在这个计算机科学技术和互联网飞速发展的时代，伴随着网络游戏的出现和普及，游戏化的概念正在逐渐被大范围推广和大规模应用。现代意义上的游戏化首次出现于2003年，而就在2010年，游戏化已经开始趋于热门。短短不到10年的时间，游戏化是以怎样惊人的速度在发展？它为何能拥有如此强大的力量去引人入胜？而我们将如何把游戏化的技术巧妙地应用于各种非游戏的情景之中，从而发挥其效益去创造更多的价值？这都是本文讨论的重点所在。

所谓游戏化，是指在非游戏环境中对于游戏元素和游戏设计技巧的应用。我们在研究游戏化的机制和对应的设计模式之前，首先应该了解游戏化的发展背景，明白游戏化的研究价值和发展前景，然后结合目前一些主流的游戏化应用案例，从中体会和总结出游戏化网站的元素特点和设计框架，最终有效地利用游戏化原理，采用Java语言和J2EE架构去实现高效的在线学习网站系统。

**关键字：**游戏化；设计模式；游戏元素；J2EE架构

# Abstract

In this era of rapid development of computer technology and the Internet, along with the emergence and popularity of online games, the concept of the gamification is being promoted gradually and having a large-scale applications. Gamification of the modern sense first appeared in 2003, however, it has become popular in 2010. In less than 10 years, what an alarming speed of development of gamification it is! Why it has such power to be attractive? And how we will apply the skill of the gamification to adapt all kinds of non-gaming situation, and thus bring into play its function to create more and more value? This is the focus of the article.

The so-called gamification, refers to the gameplay elements and game design skills applied in a non-gaming environment. Before researching the mechanism of the gamification and the corresponding design patterns, we should first know the development background of gamification and understand the value and prospects of it. Then, we can combine with the current mainstream application cases of gamification to conclude the characteristic of the gameplay elements and the frame of design. Finally, we will apply the effective principle of gamification, at the same time, make use of the Java and J2EE architecture to achieve efficient online learning website system.

**Keywords:** gamification; design pattern; gameplay elements; J2EE architecture

**目录**

[第1章 绪论 1](#_Toc421224958)

[1.1 概要叙述 1](#_Toc421224959)

[1.2 背景介绍 1](#_Toc421224960)

[1.2.1 早期的游戏化 1](#_Toc421224961)

[1.2.2 现代游戏化的出现 2](#_Toc421224962)

[1.2.3 游戏化趋于热门 3](#_Toc421224963)

[1.3 研究目的 4](#_Toc421224964)

[第2章 游戏化原理 6](#_Toc421224965)

[2.1 什么是游戏化 6](#_Toc421224966)

[2.2 游戏化三大要点 7](#_Toc421224967)

[2.2.1 游戏元素 7](#_Toc421224968)

[2.2.2 游戏设计技巧 7](#_Toc421224969)

[2.2.3 非游戏环境 7](#_Toc421224970)

[2.3 游戏化的价值 8](#_Toc421224971)

[第3章 游戏化案例 10](#_Toc421224972)

[3.1 游戏化分类 10](#_Toc421224973)

[3.2 外部游戏化 10](#_Toc421224974)

[3.2.1 Psych 10](#_Toc421224975)

[3.2.2 Stack Overflow 13](#_Toc421224976)

[3.3 内部游戏化 15](#_Toc421224977)

[3.4 行为改变游戏化 16](#_Toc421224978)

[第4章 游戏化框架 18](#_Toc421224979)

[4.1 运用游戏元素进行设计 18](#_Toc421224980)

[4.2 金字塔型框架 19](#_Toc421224981)

[4.2.1 动力层 20](#_Toc421224982)

[4.2.2 机制层 21](#_Toc421224983)

[4.2.3 元素层 21](#_Toc421224984)

[第5章 游戏化设计模式 23](#_Toc421224985)

[5.1 常见的游戏元素 23](#_Toc421224986)

[5.1.1 积分 23](#_Toc421224987)

[5.1.2 徽章 24](#_Toc421224988)

[5.1.3 排行榜 26](#_Toc421224989)

[5.2 游戏元素的局限性 27](#_Toc421224990)

[第6章 Web游戏化需求分析 31](#_Toc421224991)

[6.1 引言 31](#_Toc421224992)

[6.1.1 编写目的 31](#_Toc421224993)

[6.1.2 名词定义 31](#_Toc421224994)

[6.2 任务概述 31](#_Toc421224995)

[6.2.1 一般描述 31](#_Toc421224996)

[6.2.2 需求分析 31](#_Toc421224997)

[6.2.3 运行环境 32](#_Toc421224998)

[第7章 Web游戏化详细设计 33](#_Toc421224999)

[7.1 总体设计 33](#_Toc421225000)

[7.1.1 处理流程 33](#_Toc421225001)

[7.1.2 关系结构 34](#_Toc421225002)

[7.1.3 数据库设计 35](#_Toc421225003)

[7.2 接口设计 37](#_Toc421225004)

[7.2.1功能与用户界面 37](#_Toc421225005)

[7.2.2 内部接口 52](#_Toc421225006)

[7.3 类和函数的设计 52](#_Toc421225007)

[7.3.1 类的属性 52](#_Toc421225008)

[7.3.2 类的方法 53](#_Toc421225009)

[第8章 Web游戏化系统实现 59](#_Toc421225010)

[8.1 增加模块 59](#_Toc421225011)

[8.1.1 用户注册 59](#_Toc421225012)

[8.1.2 添加日志 60](#_Toc421225013)

[8.1.3 留言与回复 62](#_Toc421225014)

[8.1.4 颁发荣誉徽章 63](#_Toc421225015)

[8.2 删减模块 64](#_Toc421225016)

[8.2.1 用户注销 64](#_Toc421225017)

[8.2.2 删除日志 65](#_Toc421225018)

[8.3 更新模块 66](#_Toc421225019)

[8.3.1 修改个人信息 66](#_Toc421225020)

[8.3.2 上传头像 67](#_Toc421225021)

[8.3.3 访问与点赞 70](#_Toc421225022)

[8.3.4 累计积分 71](#_Toc421225023)

[8.4 查询模块 71](#_Toc421225024)

[8.4.1 验证登录 71](#_Toc421225025)

[8.4.2 信息加载 73](#_Toc421225026)

[8.4.3 社交搜索 74](#_Toc421225027)

[8.4.4 排行榜 75](#_Toc421225028)

[致谢 76](#_Toc421225029)

[参考文献 77](#_Toc421225030)

# 

# 第1章 绪论

## 1.1 概要叙述

游戏化，就是从游戏中学习。举个例子，从《愤怒的小鸟》中，我们可以学到什么呢，如此流行的游戏，以至于各种版本被下载超过10亿次，难道这10亿用户的选择是错的吗？比这更重要的是，游戏化所谓的向游戏学习，不仅指学习游戏本身，更要理解为游戏如何成功地引人入胜，理解游戏能做什么以及它为什么有这么大的力量，与此同时，我们能够将这些技术细致周到并且巧妙地应用于其他各种非游戏的情境之中。

我们从《三星王国》说起，这是三星企业网站的一部分，也是一个应用所谓游戏元素和游戏机制来解决三星商业问题的系统。由于三星集团希望更多的客户访问他们的网站，并在网站上通过撰写评论、观看视频等方式与产品进行互动，从而了解更多信息并注册已购买的产品，因此建立了这样一个网站，将游戏中的简单元素应用其中，如排行榜、成就徽章、积分系统等等。以上就是游戏化的一个应用案例，尽管它绝不是游戏化的唯一应用方式，但作为起点，对于介绍游戏化来说，这是一个很好的开始。

## 1.2 背景介绍

游戏化到底从何而来？事实上，它的起源可以追溯到上个世纪初，在那个时期，Cracker Jack公司已经开始在每个产品的包装盒里放进一个小玩具作为惊喜，从此，很多公司都用游戏、玩具或者其他有趣的方式销售产品，相关的例子有不少，例如飞行常客奖励计划等，或多或少都有一些游戏化元素，但从把事物变得更像游戏以提升商业效果的意义上来说，这些还不算是我们现在所说的游戏化。

### 1.2.1 早期的游戏化

游戏化这个专业术语首次出现于1980年，来自英国埃塞克斯大学的著名游戏设计师兼研究学者Richard Bartle，同时作为MUD1项目的成员之一。这里简单介绍下MUD1，它在本质上来说是第一个大型多人在线游戏，这也使人们第一次体验到多人共享的虚拟世界，我们可以将它理解为《魔兽世界》等当代游戏的前身。Bartle在这项计划里的工作就是对合作平台进行游戏化，使之与游戏更加相似。那时候游戏化的意思是，将某些非游戏的事物变成游戏，而现在的游戏化概念则将游戏分为不同的构成元素，即把真正的游戏向非游戏的方向改造。

Bartle当时所做的工作也叫游戏化，虽然它与我们现在所说的游戏化不大相同。当时还有一些研究，为我们现在所说的游戏化奠定了基础，其中一项就是教育学者对电子游戏和学习进行的研究。Tom Malone，现就职于麻省理工学院斯隆商学院，从很早之前就开始对早期电子游戏进行研究，上世纪80年代期间，电脑上只有很简单的游戏，但他还是证明了儿童可以通过电子游戏进行学习。此后，不计其数的研究者们又展开了类似或更加复杂的研究，其中的一名来自亚利桑那州立大学的研究者，James Paul G，写过很多关于电子游戏的书籍。他曾在书中谈到：“包括像《古墓丽影》系列这种商业娱乐导向的热门游戏在内，电子游戏涵盖了强大的知识创造与学习机制，几乎可以涉及到所有关于人类学习的深度研究”。这也推动了当代游戏化的第二次浪潮，严肃游戏运动。

严肃游戏运动由Ben Sawyer和David Rejecsk创立于2002年，吸引了私营企业、学术界与军队等机构。该运动采用成熟游戏方式进行非游戏目的模拟训练，如军队对战场模拟等，利用游戏作为训练机制是一件有趣并且高效的事情，这种机制每年都能为军队训练成千上万的军人和其他人员。这也使越来越多的公司有着类似的兴趣。严肃游戏的定义与我们现在所说的游戏化依旧不同，它类似于为特定目标建立模拟情景的成熟完善的游戏，这是一件有强大力量的事情，并且已经获得了很多了不起的成就。尽管严肃游戏的做法与现在的游戏化有着相似的形式，但它们依然不尽相同。

### 1.2.2 现代游戏化的出现

现代意义上的游戏化首次出现于2003年，由一名来自英国的开发者兼设计师Nick Pelling提出。当时他建立了一家咨询公司Conundra，主要业务是促进消费品的游戏化推广，他想把硬件产品改造得更像游戏，但是并没有取得太大的成功，而他的公司也没有运营太久。尽管如此，这也体现了一种理念，表明游戏机制和游戏概念能够应用于消费品或者其他情景。后来，在2005年的时候，一家名为Bunchball的公司成立了，这是第一个真正意义上的现代游戏化平台，虽然人们不知道如何称呼它，因为“游戏化”这个词在当时并不常用。之所以这么说，是因为这是第一个结合积分和排行榜等机制协助公司提高参与度的游戏化平台。在不久之后，市场中又出现了一批新的竞争对手，包括Gigya、Bigdoor和Badgeville等公司，他们为自己的客户提供了游戏化平台，并可以提供特殊定制服务。还有一些公司，如KEAS、Pratically Green和Rypple等，则专为特定领域提供游戏化服务。与此同时，也出现了不少自己建立游戏化服务和系统的公司。

### 1.2.3 游戏化趋于热门

2010年，游戏化开始趋于热门。一方面，游戏化行业已经取得了一定的规模，业内一致同意把“游戏化”作为通用术语，另一方面，由于出现了一系列相关演讲，终于明确定义了游戏化的概念。在这里，不得不提的是Jesse Schell的演讲，他是著名的游戏开发者并且创办了一家名为Schell Games的公司，目前在卡内基梅隆大学教书。这是一次在游戏业界知名峰会——DICE峰会中发表的演讲，演讲中有一个经典的片段：“这个世界会是怎样的？我认为是这样。你早上起床要刷牙，而牙刷检测到你正在刷牙，很好！你因为刷牙获得了10个积分，牙刷记录了你刷牙所用的时间，如果你应该在3分钟内刷完，而你完成了任务，你就会因此获得额外的奖励。如果你本周每天都刷牙，那么你同样会获得额外的奖励。说这么多，到底谁会在乎这些呢？当然是生产牙膏和牙刷的公司。你刷牙越频繁，就会用掉越多牙膏，而他们也将获得更多的经济效益。之后你就去吃早餐，早餐有玉米片，玉米片包装盒的背后有一个小型的网页游戏，而你，再也不用看包装盒上的字来打发时间了。你可以边吃早餐边玩游戏，并且因为吃了玉米片，你又获得了10个积分。由于连接了wifi，绑定了Facebook和各种账号，所以你还可以看到自己的好友列表上，究竟哪些人也吃了玉米片并获得了多少积分。接下来，你会因为在玉米片游戏中击败了一个好友而获得5个额外积分。然后你出门去坐公交车，为什么要坐公交车呢？因为政府会给你发放积分，政府选择给那些使用公共交通工具出行的人发放积分，而这些积分可以用来抵税。在乘坐公交车上班的路上，你玩了俄罗斯方块，又获得了一些零散的积分。你突然想起昨晚的梦，你梦见你的妈妈在和一个巨型的百事可乐罐跳舞，你突然意识到，这原来是那个放在你耳朵里的装置——快速眼动娱乐系统。它检测到你进入睡眠状态后就会开始投放小广告，从而尝试去影响你梦境中的内容。之后你可以填一些简单的调查问卷，看看那些广告是否成功地进入了你的梦境，一旦成功，你就会获得大量的积分。”

Jesse的演讲真的非常棒，尽管有些讽刺意味，但却在很大程度上推动了人们去思考游戏化的发展趋势，并有利于设想游戏化的潜在价值。同年，有类似影响力还有Jane McGonigal的著作《游戏改变世界》以及她在TED大会上发表的演讲。Jane同样是一名游戏设计师，她因为对平行实境游戏，也就是嵌入真实世界的游戏的研究而出名。在她看来，游戏能够切实解决人类的重大问题，不仅能够帮助人们应对气候变化或第三世界中的不平等，而且可以帮助人们变得更加健康、幸福、积极和成功。

## 1.3 研究目的

尽管游戏化产业仍然是一个新兴产业，有新的观点不断被提出，但是不可否认的是，它已经开始走向成熟。研讨会和越来越多的公司都开始涉足产业中的不同领域，市场调研报告也反映了游戏化产业十分诱人的前景，游戏化已经越来越流行了，而这个产业也将在未来取得更加重要的地位。

游戏化是否能使世界变得更加美好？是否能提高人们的生活品质？很显然，在一定程度上，答案是肯定的。游戏化是一种激励工具。它可能会激励你更多地购买某品牌的早餐谷物食品，也可能会激励你去填写一份市场调查问卷。但是，它也可以用于鼓励你养成更加健康的饮食习惯，鼓励你节约能源、减少碳排放、学习个人理财技巧、摆脱贫困等等。所以，显然游戏化能够为社会公益做出贡献，而我们也需要对游戏化在这一领域的应用加以关注。在社会影响或社会公益等方面，游戏化面临着特殊的挑战，但与此同时，它也面临着一些有趣的机遇，或者说存在某些技巧，能够让这些领域内的应用更加有效。可贵的是，当今世界中，游戏化的应用已经涉及到了很多领域，不少机构都已经把游戏化作为激励工具，去鼓励人们做对自己或是社会有利的事情。这里的社会公益包括多种类型，其中之一是某件事情给社会带来的利益超过了它给实施这一项目的机构，或者说给当事人所工作的机构带来的好处，这就属于一种社会公益领域内的游戏化。而另一种则是能够帮助人们变得更好、变得更快乐、变得更健康的一系列事物，这种方式在属于广义上的社会公益的同时，也可能与某些公司的商业利益有关。但从根本上说，它们的目的都是让人们变得更好、更出色，所以这两种类别都可以归于社会公益的范畴之内。我们应该知道，当游戏化被应用于社会公益时，在游戏以及类似游戏的活动中，会产生很多正面的能量、情感和吸引力。Jane McGonigal在她的著作《游戏改变世界》中详细阐述了这一点，游戏会产生积极的能量，能够促进社交，还可以使人们更加关注于手头的任务等等。所以把事物变得更像游戏会有很多潜在的好处。值得注意的是，在任何任务中，纯粹地追求乐趣和参与感，寻找所谓的解决问题的游戏精神，这些都是积极的社会价值。

而我们之所以耗费如此巨大的精力去研究游戏化，是因为现在的游戏化还是个比较年轻的领域，发展速度很快，甚至每个月都会有新的变化。所以，游戏化的前景确实值得我们去思考。不妨设想一下，如果游戏化能够持续发展壮大，等到它真的无处不在的那一天，这也将意味着关于游戏化的研究直接或间接地涉及到了世界范围内的各个领域，我想，蕴含在这其中的价值自然是不言而喻的。

# 第2章 游戏化原理

## 2.1 什么是游戏化

游戏化是一个新的概念，人们在很多方面都对其存在误解。那么游戏化到底是什么呢？游戏化是在非游戏环境中，对游戏元素和游戏设计技巧的应用。该定义有三个部分：游戏元素、游戏设计技巧、非游戏环境。

举一个例子解释一下吧，大家都知道耐克（Nike）生产运动鞋，作为造鞋厂商，它当然希望人们多去跑步，于是为耐克设计了一种叫做加速度感应计的设备放在鞋底。这个东西会跟踪记录你跑过的每一步并且知道你跑步的距离和速度，它将与你的智能手机或者电脑进行无线通讯以收集所有相关数据。此外，耐克就上述功能开发了一系列应用，这些应用可以告诉你跑了多远和跑得多快，可以让你明确地知道自己跑步的最远记录和最快记录以及各种跟踪数据，跟有趣的是，你可以设置目标来挑战自己，如果你成功实现了这些目标则可以获得奖杯和奖牌。这一切都使跑步的体验更像是在参与游戏，但对于用户而言，这仍然是外出跑步，而不是真的坐在电脑前玩赛跑游戏。这也在很大程度上激励了用户，让跑步变得更加丰富多彩。

还有一个例子，虽然同样是为了鼓励人们去参与跑步，但却看起来很不一样。有一个名为《Zombies Run》的游戏，游戏情境中，用户会被吸血僵尸群疯狂追赶，其实这是一个较为常见的游戏主题，但是其目的不是让用户坐着玩弄智能手机，而是带上手机外出跑步。在用户奔跑的时候会听到大波僵尸正在靠近的提示音，并能够看到僵尸们在地图上的位置。用户可以获得超能力或者加速奖励包，而用户的朋友也可以加入其中展开一场混战。其实这只是一个轻量级游戏，但是它却十分巧妙地将乐趣添加在跑步中，让跑步这一体验升级到了一种新的层面。

相比于耐克应用中的挑战、奖励等元素，《Zombies Run》则是提供了一种更加令人沉迷的类游戏体验。其实这两者的最终目的与游戏并无关系，但他们都使用了游戏元素和游戏设计技巧。

## 2.2 游戏化三大要点

### 2.2.1 游戏元素

游戏化的第一要点是游戏元素，那么，游戏元素是什么呢？事实上，可以将其想象为工具箱，如果你想要使用游戏中的东西，游戏元素则是你必须要使用的部分。这里的游戏不是指游戏本身，而是组成游戏的常规设计模式。举个简单的例子，《水果忍者》（Empires and Allies），这是一款热门的休闲益智游戏，它拥有与其他游戏类似的图形界面，并与各种细节一起构成了游戏式的体验，即便不知道其玩法，也知道这是一款游戏。但是由表及里地看，我们不难分解出其中的游戏元素。如分数这一元素，这是玩家在游戏中积累成绩的方式，也是各种游戏中常见的内容。我们理所当然地认为很多游戏，尤其是网游或单机游戏，都有分数的设置。可以说，这是很多游戏采用的常规模式，与此相类似，我们可以在这里收集和升级相关资源，在游戏的世界中，这些资源能让你制造出更加复杂的物品，以供不断提取和利用。我们需要做的，是通过游戏化，去考虑不用层面的游戏元素，并将这些元素巧妙地应用到游戏以外的服务之中。

### 2.2.2 游戏设计技巧

游戏化的第二要点是游戏设计技巧，之所以要讨论所谓的设计技巧，是因为游戏不仅仅包含前面介绍的那些元素，毕竟游戏不是一堆元素的随意堆砌，为了增加趣味性，设计游戏时必须要有系统的步骤和详细的规划，并赋予其艺术美感。一个游戏设计师所要做的，除了软件工程之类的工作外，还涉及很多，包括大量的算法和相关技术。游戏设计同样对美感有要求，这需要以一定的方式去考虑问题，包括采取一些通用的设计概念和一些新颖的、特别适合游戏设计的概念。这种做法就像在游戏之外应用游戏元素，一些游戏化的案例会着重于强调游戏元素，而另一些则偏向于强调游戏模式，还有一些两者兼顾，这也是定义包含两者的原因所在。游戏设计技巧更多地涉及一种思维方式，这不只是一套约定俗成的做法，更是处理问题的方式。

### 2.2.3 非游戏环境

第三个要点是非游戏环境，这里指的是游戏之外的事物。玩游戏时，你是为了享受乐趣，但是如果是为了自己的业务或者公司的目标，为了学习与工作相关的知识技能，这就是非游戏环境了。这看上去像在玩游戏，但最终目的却在游戏之外。你的目标使游戏体验具备了独立的用途和目的，尽管之前举了许多商业背景的例子，但这不一定要和商业相关。比如跑步，虽然对于耐克的销售业务来说十分重要，但对于一些想通过跑步保持健康或实现其他利益的人而言也很重要，所以，非游戏环境可以是任何事物，只要你的最终目标是在游戏之外。

把游戏元素和游戏设计技巧结合在一起，为超出游戏本身的目标而服务，这就是游戏化。

## 2.3 游戏化的价值

为什么越来越多的人开始学习游戏化呢？我认为游戏化的价值所在，可以从以下四个方面来体现。

首先，作为一种重要的商业实践，游戏化正在逐渐兴起，我们可以看到越来越多游戏化应用的案例，正如《财富》杂志曾在2011年10月刊中提到：“游戏化是一个热门的新型商业概念”。尽管盲目地随波逐流并不被提倡，但是，很多世界知名企业已经开始采纳了这一概念，无论是小型初创企业或者大型企业。还有一篇来自《华尔街日报》的文章中提到：“很多企业正在将电子游戏的元素融入到工作环境中”。事实上，电子游戏已成为一个颇具影响力的产业，它不仅可以通过销售获利，还成为了解决工作和工作相关问题的基础，因此，游戏化正在普及并且成为了一种非常普遍的现象。调查发现，不少企业都已经较好地应用了游戏化，小到新生的初创科技公司，大到以生产消费品为主的大型实体企业，其中一些公司已经下意识去尝试将游戏化应用到商业实践之中，更有一些公司研发出了一些具有游戏性质的东西，但最初他们纯粹是觉得这只是一个处理特定问题的好办法。就上述情况来看，人们已经开始认同游戏化是一种具有价值的技术，并且适用范围十分广泛。

第二个游戏化的价值体现在游戏具有强大的力量，这或许也是最有说服力的理由了。我想，很多人都曾经从内心深处发出过“这个游戏好上瘾”的感叹，或是玩游戏的时候觉得时光飞逝。这些游戏不仅仅是电子游戏，还可能是桌游、纸牌或体育运动等等。游戏对我们的的确确有着巨大的吸引力，那么到底是什么让游戏如此令人着迷呢？又是什么让游戏能够支持复杂的思维和学习？这都是很重要的问题，也正是游戏研究者们真正关注的问题。当游戏元素和游戏设计技巧被应用于非游戏领域中时，这些问题会变得相当重要，也非常值得我们去进行深入研究。

而学习游戏化的第三个原因，即游戏化价值所在的第三点是，我们能够从游戏化中学习到其他领域的只是，我们会逐渐发现在研究游戏化的过程中，我们也在接触着心理学。游戏之所以能够贯穿整个人类历史，是因为它与我们人类思维的基础运作方式息息相关。因此，要掌握如何有效地进行游戏化设计，我们就需要理解一些东西，比如“动机”，简单来说，是什么让人想去做某件事情。除了心理学之外，我们还会接触到设计。游戏化是一项设计实践活动，为了懂得如何去实践，我们必须去探究复杂的设计世界。接下来是战略，组织内部的游戏化与许多商学院中人们谈及的话题有关，比如“如何做生意”、“榜样在工作中意味着什么”、“设计一个有效的营销活动有何意义”等。而只有把这些不同领域的知识汇总在一起，才能更好地理解游戏化的基本原理。最后是技术，游戏将始终伴随着我们，许多游戏并不包含复杂的技术，但是当今的网络信息技术能够创造出丰富且个性化的体验，不仅实现了实时跟踪互动，还能收集整理数据以分析利用，这一切对于游戏化的应用而言都是难以置信的强大力量。由于游戏化目前主要被应用于数字化的网络环境之中，所以理解什么样的环境能使游戏化获得成功则为了解技术问题提供了一个有效的途径。

最后一点就是，游戏化并不是大家想象中的那样容易和浅显。一旦理解了什么是游戏化，你会觉得这是一件很棒的事情。你以为你效仿耐克，在商业活动中加入挑战和奖杯的概念，就一定会大受欢迎吗？事实上并不是那么简单的事情，要把游戏化做好且做得合乎道德规范，符合长期的商业目标，真的很不容易，因为这一切都需要你去思考和运用各种不同领域的知识，这不是一份操作手册或一张游戏元素清单就能解决的事情，而是需要你在更加广阔的范围内去理解它们。

# 第3章 游戏化案例

## 3.1 游戏化分类

一般来说，游戏化可以分为三种主要类型，或者说能从三个方面为人们带来价值，即外部、内部和行为改变三种类型。

外部游戏化是指对你所在公司或组织而言的外部，这种游戏化通常针对客户或潜在客户，例如应用于营销或销售的环境中。内部游戏化是指针对公司内部人员的游戏化应用，通常针对雇员。游戏化的第三种类型是行为改变，通常，人们会想要做些什么，或者及时能意识到做某件事的价值，却无法克服困难真正做到它，比如你知道应该多去体育馆锻炼但执行起来却很难。而游戏化提供的动力或许能改变你的行为，去把它变成一种习惯，也就是行为改变类型的游戏化。我们会看到，它的应用范围并不局限于社会影响方面，有时也会涉及到一些个人或团体利益，关键在于，虽然人们知道这是他们想做的事情，但如何才能让他们真正地参与进来。

这就是游戏化的三大分类，下面让我们来看看各个类型的案例。

## 3.2 外部游戏化

### 3.2.1 Psych

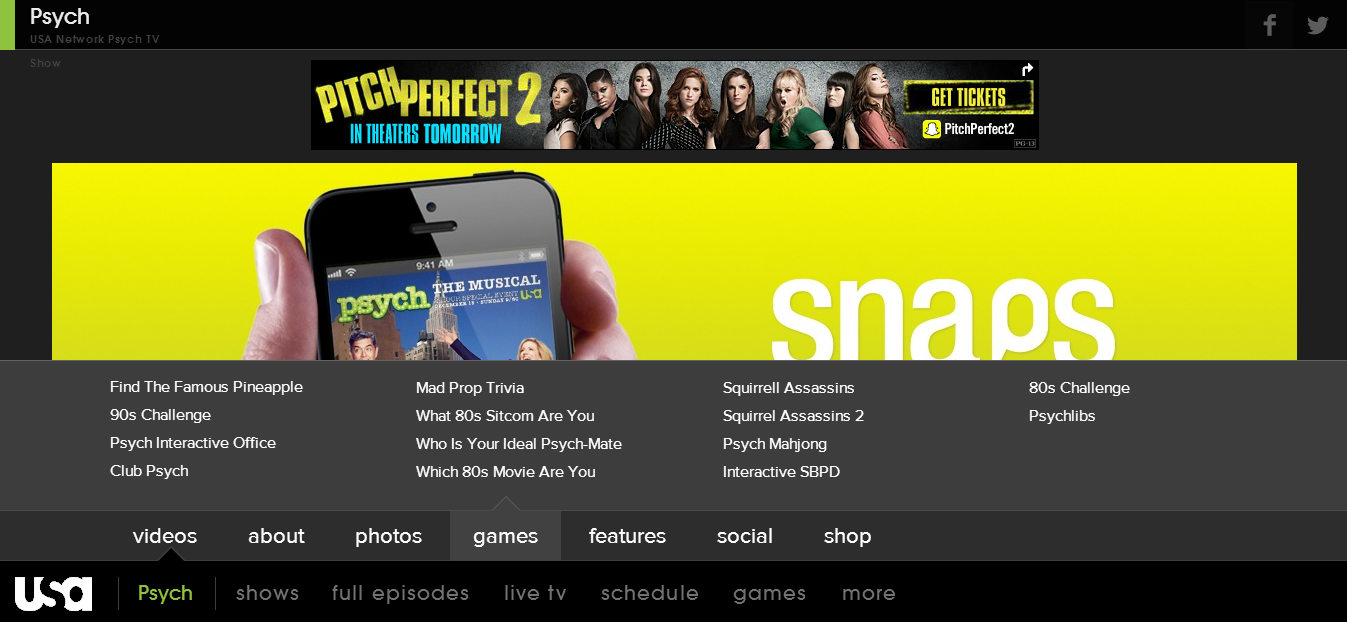


图3-1 灵异妙探俱乐部游戏化模块

首先，介绍一个网站，叫灵异妙探俱乐部（Psych），这是有一家美国的有线电视台建立的，网站的主题是一部名为灵异妙探的流行美剧。事实上，这个电视台之前有一个常规的网站，就像我们国产的芒果TV那样，但是几年前，他们新建了这个游戏化网站，希望人们可以与节目去进行更多的互动。

如图7-1所示，灵异妙探俱乐部这个网站使用了很多游戏机制，我们可以发现不少之前介绍过的游戏元素。你可以通过完成很多事情去获得积分，积分最终可以帮助你赢得相应的奖励，这是十分常见的流程，尤其是针对外部营销的游戏化案例。你还可以赢得徽章，这里的徽章看起来像个菠萝，如果你有看过灵异妙探的话，就会知道这是一个在剧中反复出现的物品。对于灵异妙探俱乐部而言，这个网站将游戏机制与节目中的诸多方面进行了巧妙的整合。这不是一个普通的游戏化应用，它与节目主题紧密结合，使观众们觉得这是节目体验的一种延续。



图3-2 灵异妙探俱乐部网页游戏

从图中，我们可以看到排行榜、玩家头像以及一些挑战和任务。这都是为了鼓励用户去做一些事情，也都是我们所常见的游戏机制。那么，美国电视台对这个网站进行游戏化的结果如何呢？应该说非常成功。这使美国电视台的节目收视率足足提高了30%。从收入的角度来看，在线商品销售的营业额提高了50%，而该网站的总浏览量上升了130%，节目相关的网页浏览量更是翻了不止一倍，通过让用户在Facebook上分享内容（这也是作为游戏化服务中任务的一部分），他们获得了30万次的转载，这意味着有4000万用户将能够看到那些内容。事实上，灵异妙探的观众大概只有450万，所以这对于他们而言，是一个巨大的市场提升，而这一切都得益于这些相对简单常见却被巧妙实施的游戏机制。

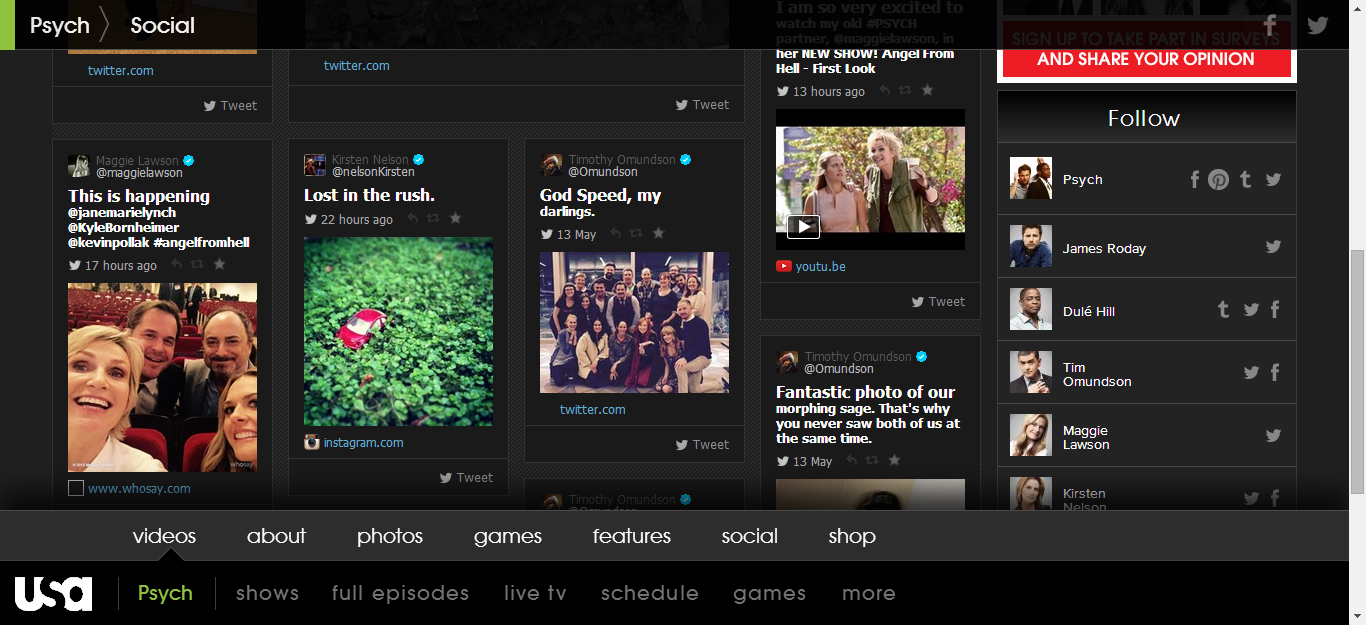


图3-3 灵异妙探俱乐部社交化模块

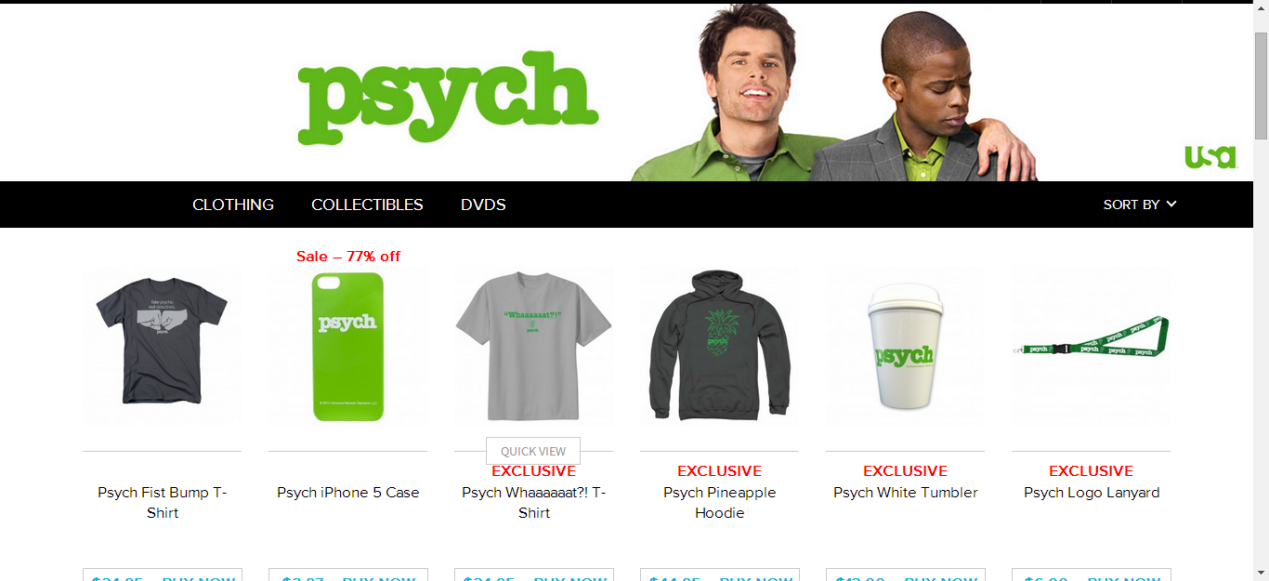


图3-4 灵异妙探俱乐部在线商品销售

### 3.2.2 Stack Overflow

当面临商业难题时，游戏化是一种很好的方法。那么，如果确保游戏化能够被全面深入且有意义地实施呢？最好的说明就是下面这个使用了PBL元素，并且能够将三者紧密有序地结合在一起的典型案例。

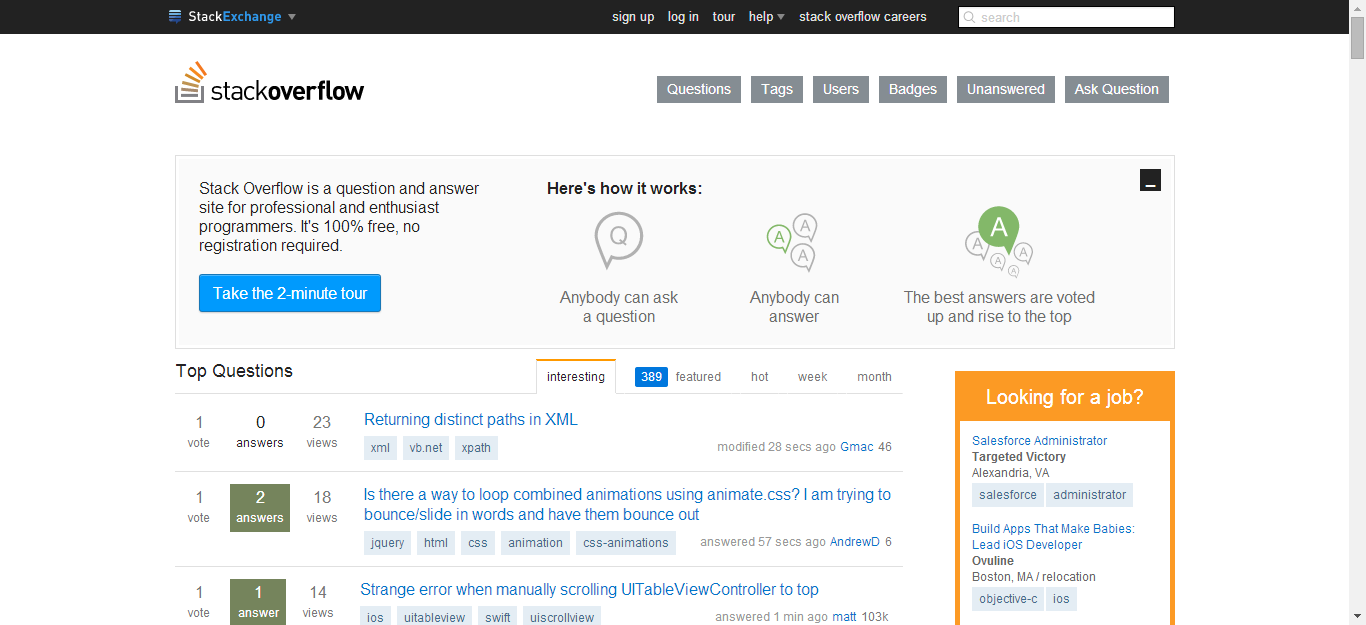


图3-5 Stack Overflow首页

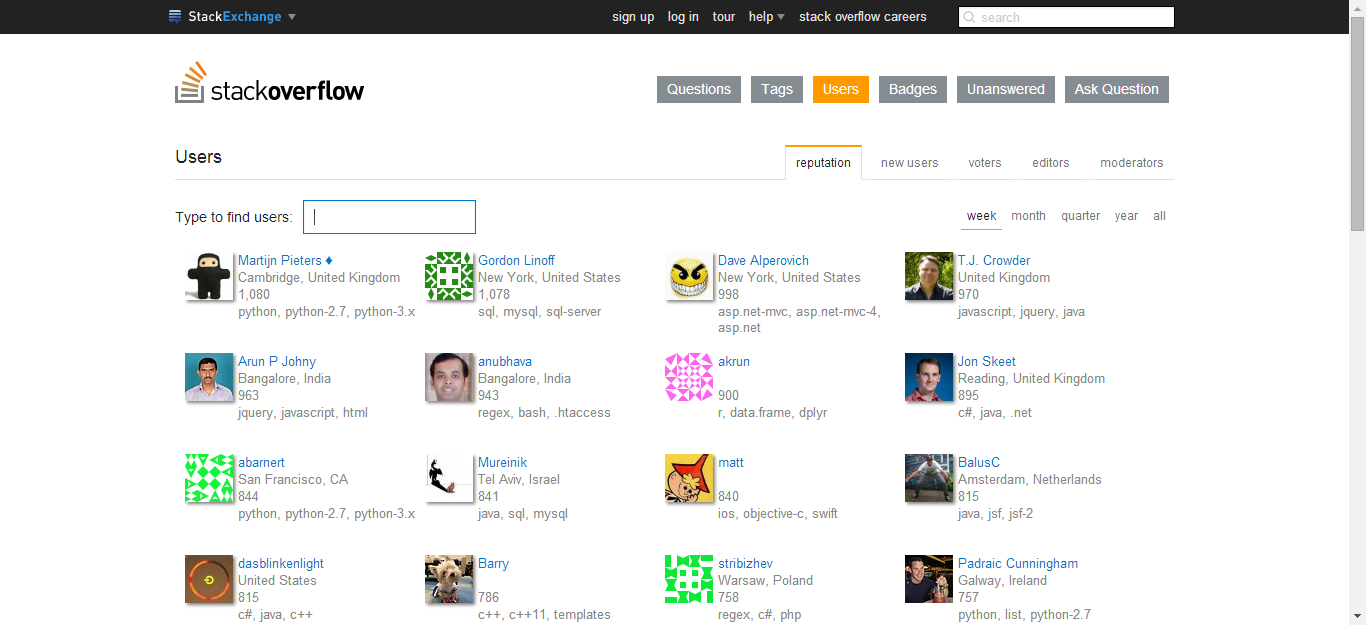


图3-6 Stack Overflow社交化模块

这个网站叫Stack Overflow，它现在是一个更大的网站群组Stack Exchange的一部分，以上展示了这个网站的首页截图。在这里我将引用网站创始人之一Jeff Atwood的一片博文，在这篇文章中，他说明了他与其他联合创始人使用游戏化的原因和具体的做法。其中很重要的一点是，在最初搭建网站的时候，他根本不知道游戏化是什么。他们并不是因为游戏化是一种新潮有趣的设计实践才做出了相关的设计决定，他们只是想尽可能地解决他们所面临的问题，而当时他们的做法恰好和我们现在定义的游戏化差不多。

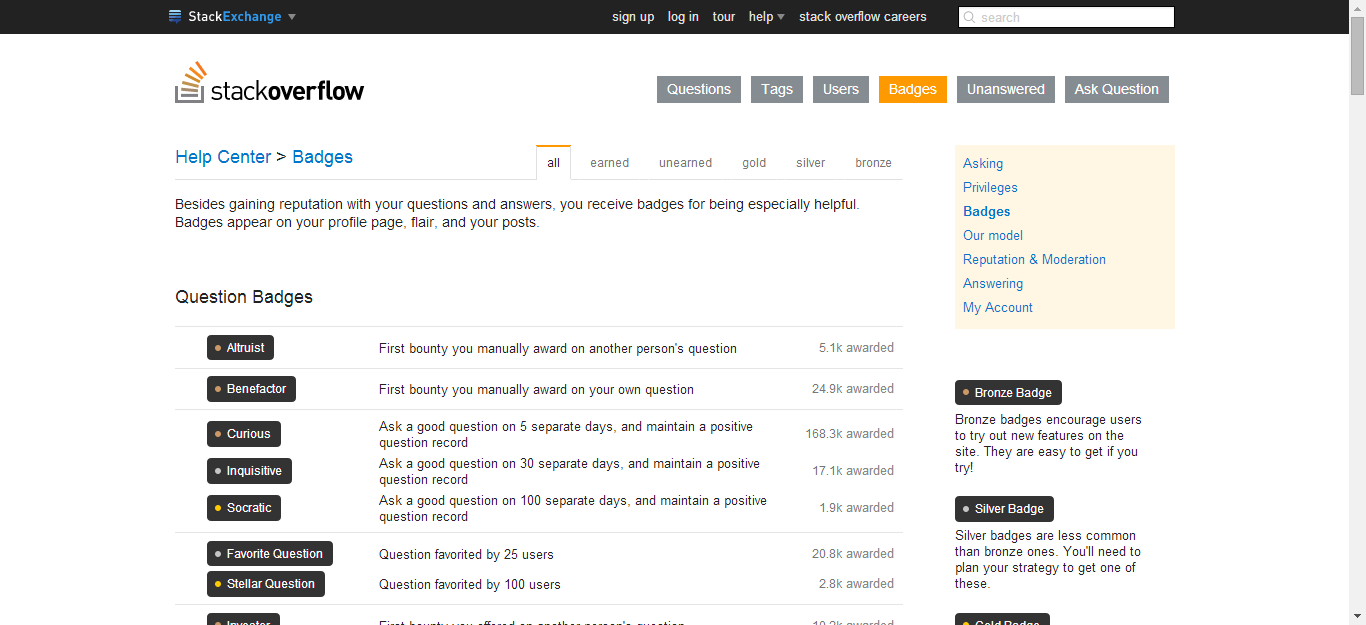


图3-6 Stack Overflow徽章系统

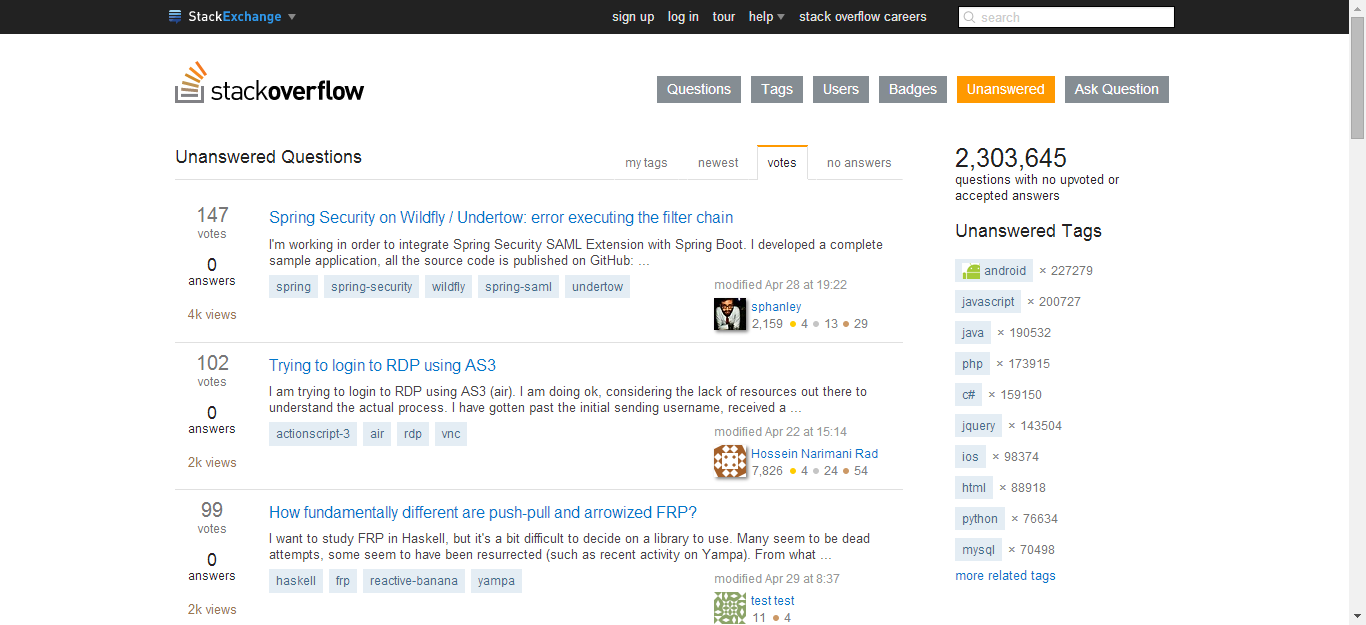


图3-6 Stack Overflow积分系统

Stack Overflow是一个面向程序员的问答网站，你可以在网站上提问，比如如何编写软件或是有关具体代码的问题，其他用户则会回答你的问题。这似乎不是一件很有趣的事情，仅仅是提出或回答有关编程的问题。尤其是怎样才能让人们自愿花费时间来回答一些陌生人的编程问题呢？但事实上，Stack Overflow在这方面做得非常成功，这个网站和它合并起来的一系列网站拥有超过两百万注册用户，平均每天会出现5000多个新问题和超过10000个新答案。他们成功地激励了人们加入这项活动，那么成功背后的经验是什么呢？

第一就是从正确的角度思考问题。编程是什么？很多人都认为编程是一件平淡无聊的事情，回答关于编程的问题也同样无聊，而且很难。但也有人可能觉得它很酷，非常神奇有趣，就像游戏一样，而这就是网站设计师们的切入点。Atwood说，编程是一个需要不断学习的领域，而学习和游戏有着密切的联系。很明显，对于程序员来说，编程就应该是有趣的，这也自然就成了一种真正的游戏化方式。这不是强迫他们去做原本不会做的事情，也不是用一堆奖品来吸引用户，他们的做法是从用户的角度考虑问题。而他们解决的第二个问题是实现一个面向程序员群体的系统，了解你的用户、了解什么样的群体会参与这项活动非常重要。因为特定的群体会赋予游戏化特定的意义。什么才是有意义的？对于每个群体而言，答案是不同的，这取决于这些群体的行为模式以及他们参与的活动类别。Stack Overflow很好地利用了徽章这一游戏元素，但他们的徽章却别具一格，由于这是一个程序员和Geek的网站，所以更加注重数字和信息，很少有花哨的视觉效果，以尽可能帮助人们快速获得信息要点。

Atwood谈到，游戏元素应该服务于更高的目标，他认为在为人们改善网络面貌的同事，自己也因此变得更有学识和更加善于交流。不是每个游戏化网站都敢说自己能够为人们改善互联网络，那种仅仅是为了吸引更多用户参与的营销网络，很难从根本上帮助互联网做出改善，这些其实并不矛盾，只要我们往好的方向努力，尝试寻找对人们有意义、有趣的事情，寻找合适的方式，加入游戏元素来促进合作与沟通，服务于集体利益，那么我们的游戏化网站就更有可能变得意义深远、经久不衰！

## 3.3 内部游戏化

内部游戏化的案例来自于微软，微软面临着一个巨大的挑战，他们要把Windows和Office之类的软件提供给上百个不同国家的数百万人，让软件正常运行实在是个挑战。他们有一大群人负责测试和质量保证，而测试小组面临的挑战是有太多的人在太多的地方使用微软的软件，即使是像微软这样巨大并且资源丰富的企业，也未必能雇用到足够的人手去仔细检查所有内容。试想一下，Windows系统有几十种不同的语言，即使微软愿意，也未必能在每个国家都雇用到当地的测试公司，以确保软件中所有的文本、对话框和界面元素都没有错误。因此，微软测试组想到了一个方案，巧妙地解决了软件语言的本地化问题。

他们采用了游戏化，微软测试组的组长Ross Smith开发了一个游戏叫《语言质量竞赛》。测试组把这个游戏提供给在那个国家工作的微软员工，他们是微软员工中的志愿者，不是从其它公司雇来的。测试组宣称这不仅是一个为公司出力的好机会，还可以通过这个游戏和其他微软员工比赛，并设置一个排行榜，用于记录每个人检测出的错误数量，如果对话框看上去没有问题就点击“确定”，如果有问题就点击“报错”，然后错误就会被记录下来。

结果微软的员工们开始相互比赛了。他们想，我们一定要把我们的Windows语言本地化做出全公司第一，尽管这个任务非常乏味，仅仅是坐在电脑前阅读对话框。但这个任务类似游戏的一面是，它被放进了一个类似游戏的竞争性架构，并取得了显著的成效。结果显示，超过4000名微软员工自愿无偿工作去检查这些对话框，而他们一共检查出了50多万个对话框，找到了近7000个错误，其中数百处是微软应该修正且实际存在的语言错误。所以是游戏化机制激励员工们做到了这一切，他们不出于盈利目的，也不是为了得到个人认可，这是微软员工在正常工作职责之外的企业公民意识的一部分，而需要强调的是，正是游戏结构让工作变得有趣好玩，同时也巧妙地调动了员工的积极性。

## 3.4 行为改变游戏化

这个例子和超速有关，警察是怎样阻止人们超速驾驶的呢？传统方法是拿着一把雷达枪，坐在监视器前。如果你被抓到超速驾驶，就会收到罚单并且必须交罚款，效果还过得去。但这并不完美，人们只有在觉得前面有警察时才会减速行驶，而警察不可能随时随地守着，因此一些警察局的做法是放置一些显示屏或者电子设备，显示你经过时的速度，即使它们并没有与警察局相连，但确实产生了效果。它们仅仅是显示了你的车速，提供了一种反馈，这种实时反馈带来了对应的行为回馈，使人们对此作出反应。所以尽管你知道自己开得有多快，或者可以从车速表中看到速度，但是在正前方看到一个显示实时车速的指示牌时，还是会受到影响，因此降低车速。

接下来，我们可以更进一步。大众汽车举办了一个叫“趣味理论”的竞赛，这是一个营销项目，鼓励大家提交有趣的想法，使用游戏和娱乐元素来解决现实中的问题。获胜作品是“超速摄像头彩票”，它的作者工作于美国MTV电视台。之所以获胜，是因为他的想法别具一格。与其只找到超速的车主，不妨设置屏幕监控设备，追踪人们的驾驶情况，显示车辆的实时速度，同时设有摄像头，照下车主和车辆的牌照。它会记录下有哪些人开车驶过和他们的车速，但与其仅仅为了找到超速的车主，不如做点不同的事情。让警察站在路边，用雷达枪找到超速的车主，同时那些未超速的车主会被纳入彩票系统，得到一张彩票，而买彩票的钱来自于那些超速车主的罚单，其中一些本来会上交政府的款项则会进入一个资金池，并在某一时刻开奖，获奖者从未超速的车主中抽出。

事实上，大众汽车正在瑞典的斯德哥尔摩试用这个系统，效果相当显著。人们真的慢下来了，在试验该系统的三天内，人们的车速普遍降低了20%以上，而改变的部分仅仅是，如果不超速，他们就可能会赢取奖金。正是附加的彩票、游戏或竞赛元素改变了人们的行为，这恰恰就是行为改变的例子。我之前说过，很多行为改变的例子能够带来一定的社会效益，我想，对于超速的车主而言，这个例子就产生了效益，这种政府部门负责实施的新的方式，正如我们所见，的确是有效的！

# 第4章 游戏化框架

## 4.1 运用游戏元素进行设计

可以说，游戏元素是我们实现Web游戏化的工具箱，比如你要盖房子的时候不可能跑到五金店去买一把“盖房枪”，然后射出来一栋完完整整的楼房。你要做的是购买螺丝刀、钉子、锤子、木材和水泥板等各种不同用途的工具和材料，然后用它们建成你想要的房子。换句话说，你要将这些元素进行不同的搭配，形成各种组合，这也正是提到游戏元素时想要表达的意思。

游戏元素是一种固定的模式，是可以从任意游戏中提取的工具，我们可以利用这些工具通过不同的方法使商业实践更接近游戏。有个大家比较熟悉的游戏《帝国与盟军》（Empires and Allies），它和大多数游戏一样，向玩家展示了各种不同的游戏元素，比如积分、任务、等级等，这也算得上是一个应用游戏元素的模板了。但是如果深入来考虑什么是游戏元素，我想举下面这个例子。

井字游戏，一种非常流行的连珠游戏，在3\*3的格子里由两个玩家分别代表O和X以留下标记，如果一方在任意一条直线上成功连接3个标记，则该方获胜。尽管很多人对这个游戏再熟悉不过，但是如果我们不从游戏的角度去思考，也就是说不去考虑玩游戏本身这件事情，而是从游戏元素的角度考虑，那么，井字游戏包含了那些元素呢？而这其中又有着怎样的规律？是什么东西连接着美学和体验从而产生了游戏的可玩性？

我想，大家应该不难发现一些相对容易找到的游戏元素。游戏“棋盘”算一个，因为它划分出了游戏所需的9个格子，还有用于游戏的“记号”，也就是“O”和“X”。接下来是一些看不到的游戏元素，这些元素隐藏于游戏的规则和架构之中。对于井字游戏而言，其中之一则是两个玩家之间的“竞争”，一方获胜则另一方阵亡，这不是一群人共同合作去寻找解决方案，每次游戏都会分出输赢或是平局。还有“回合”，一方先走，然后轮到另一方。此外，游戏还定义了其中一方能够获胜的特定情形，比如当这3个“O”连成一条直线的时候则某一方胜利，这被称为是“获胜状态”。同样，如果双方填满了棋盘却都还没有连成直线，这样则定义为平局。说了这么多，想要讨论的就是游戏架构，即决定游戏输赢的元素，它们对游戏性有何影响？人们在游戏中感受如何？这样能够获得多少乐趣？继续玩下去的愿望是否强烈？然后我们再将这些融入到游戏化之中，思考如何系统地运用这些所谓的工具。当然，这里没有一套固定的元素搭配是绝对正确或绝对错误的，它们需要针对不同的游戏环境进行相应的搭配。此外需要说明的是，井字游戏不包含游戏进程，这是它的一个局限。在游戏过程中，它并不会进步或改变，也不存在关卡的概念，玩完了这一局，玩家不会因此从第一关进入第二关。它也没有任何分数的概念，仅仅就是去玩，然后分出胜负。

从以上分析来看，井字游戏并不是一个好游戏，那么一个简单又容易平局的游戏，实在是无法维持住成年人的兴趣，能够流行这么多年也是不可思议。尽管推动游戏进程或者通过分数提供反馈不是必须的，但它们确实能够让游戏更加引人入胜。如果从另一个角度来看待这个问题，即把“游戏”这个概念放在最中间，它的上层是“体验”，也就是人们在玩游戏时的感受。首先，游戏和体验并不相同，游戏是一系列的规则、美感以及各种不同游戏元素的集合，而体验则是玩家在玩游戏时的感觉。两者虽然紧密联系但并不能沦为一谈。当你设计游戏时，你所能控制的是游戏，而你想要创造的是某种体验。在“游戏”的下层是“元素”，如果体验是一种整体的影响或感受，那么元素就是那些相对细节的部分，是构成游戏的零件。在思考游戏化时，“体验”和“元素”其实是两种不同的角度，虽然在这一章节里，我们将主要从元素的角度去探讨问题，但是游戏化仍然需要我们像游戏设计师那样思考，去理解如何能运用游戏元素和技巧，从而创造出类似游戏的整体体验。

通过分析井字游戏，我们会发现它包含很多不同的游戏元素，如果去思考所有你可能想到的游戏，你会发现大量不同的游戏元素和设计模式，它们都可以用于其它游戏或各种游戏化之中，包括web游戏化。而当你深入分析游戏化中的那些优秀案例时，你会发现很多元素都在重复出现，其中绝大部分源于社交网络游戏和其他网络游戏。

## 4.2 金字塔型框架

我们不难为游戏化设计这样一个框架，其中包括游戏化的常见元素以及这些元素所构成的相应结构。虽然这个框架并不一定出现在每一个优秀的游戏化案例中，但针对怎样以不同的方式使用各种类型的游戏元素这一问题上，或多或少总能为大家带来一些启发。它是一个分为三层的金字塔结构，其中顶层是“动力”，中间层是“机制”，底层是“元素”。需要强调的是，它们只是组件，并不是游戏的全部，围绕在它们之外的是游戏的全局体验，是游戏元素不能完全表达的那一方面，是一个比所有部分加起来更重要的整体。

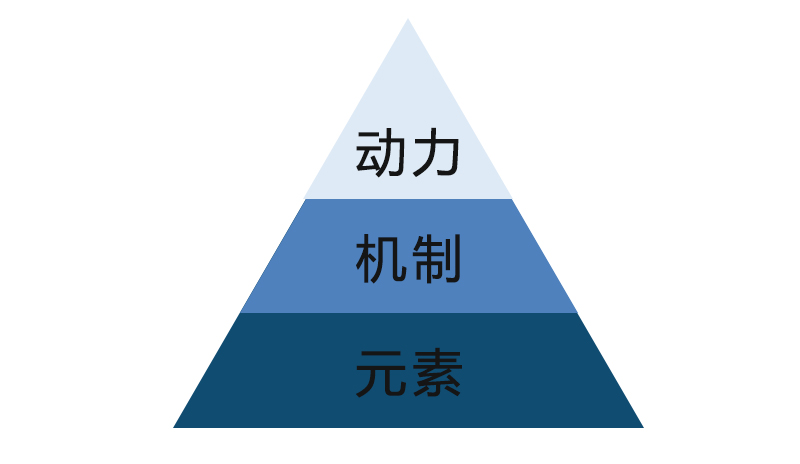


图4-1 游戏化设计——金字塔框架

### 4.2.1 动力层

现在我们来看看游戏元素的金字塔。在它的顶部是游戏“动力”，它们是一个游戏或游戏化系统中等级最高的概念元素，可以将它们比喻成“语法”，即一种形成固定模式与整体体验的内隐结构。它们和规则不一样，任何游戏都有明确的规则，规则是一种外露的表现形式，而动力则包括许多构成游戏框架的概念性元素，如“约束”，每个游戏都有一些约束，通过限制玩家的自由，游戏创造了有意义的选择和各种有趣的障碍，所以“怎样为玩家设计约束”和“设计怎样的约束”这样的概念成为重要的动力元素之一，这也是所有游戏设计师都需要思考的问题。接下来是“情绪”，无论是喜悦、悲伤或任何一种情绪，几乎都可以被游戏所引发，但对于情绪的选择，游戏化会受到较多的限制，因为我们需要应对的是游戏之外的真实世界，伤心、痛苦、沮丧、失落等这些负面情绪并不适合工作、营销或锻炼这样类型的背景环境。然而，还是有很多不同层次的情绪可以使用户体验更加丰富，如成就感、愉悦感等。在大多数游戏化案例中，强化那些能够促使用户更多参与的情绪是一件非常重要的任务。然后是“叙事”，它将游戏化系统中的各部分整合为一个关联整体。叙事可以明确，也可以含蓄，如果没有叙事感，游戏化系统就会存在问题，它会仅仅只是一堆抽象的东西。例如“你获得了积分”或者“你获得了一枚徽章”，如果这些完全脱离了某种整体关联并与玩家的生活无关，这必将限制游戏化的最终效果。还有“过程”，这里的过程是指你从一个位置开始，沿阶梯上行，直到抵达另一个更高的位置。在大多数游戏化案例中，过程都非常重要，它向玩家提供了一种有机会进步的感觉，或者至少可以从起点开始不断前进，而不是反反复复去做相同的事情。此外，过程未必需要使用到积分或等级，但是为了增强游戏的这种动力，这些游戏组件的确经常被使用。最后是“关系”，人们总是在相互影响，因此，朋友、队友、敌人、对手等这些普遍存在的社交动力对于游戏体验而言非常重要。

### 4.2.2 机制层

中间层是游戏“机制”，可以将其比喻为游戏或游戏化中的“动词”，是一些推进用户行为的东西。尽管在这里无法详细说明所有元素，因为它们实在是太多了，但是有一部分却不得不加以强调。首先是“挑战”，游戏设定了一些需要你达成的目标。然后是“机率”，这里含有一些运气的成分，比如通过投掷骰子等方式产生随机的结果，而并不完全取决于玩家的行为本身。还有“合作”与“竞争”，它们是相互对立的，但两者都是有价值的游戏机制，它们使用户在一起互助合作或者激烈竞争，并产生了输赢的概念。接下来是“反馈”，只有让用户随时看到自己正在做的事情，才能有效地促使他们继续参与下去。此外，“获取资源”也很重要，即游戏中给玩家机会去获得继续玩游戏所需要的东西，如《帝国与盟军》中需要玩家去采集石油来建造飞机之类的战斗物品。还有“奖励”和“交易”，奖励是指一些因为游戏成绩而获得的好处，交易则是指买卖物品或者与其他玩家或非玩家角色交换物品。最后是“回合”和“获胜状态”，回合自然很容易理解，而所谓获胜状态是指一种被定义的可以获胜的情形。再次强调，以上这些都是推进用户行为，促使玩家从一种状态前进到另一种状态的工具。

### 4.2.3 元素层

底层是游戏“元素”，这也是最具服务性的一层，就是用具体的例子和方法来实现“动力”与“机制”层面描述的事物。例如“成就”，它与挑战有所不同，对于完成了某些特定任务的玩家，成就会给与一些特定奖励。还有“虚拟形象”，这是玩家角色的一种视觉体现。“徽章”会在之后详细讨论，对于成就或者其他更高层次的动力或机制而言，徽章是一种特定的视觉体现形式。接下来是“Boss战”，相信玩游戏的人对此再熟悉不过，在游戏的每一关或某些部分的最后，一般都会有一个高难度挑战，要进入下一关，玩家必须击败一个高级怪物，而这也是游戏的难点所在。然后是“收集”，通过集齐一堆不同的东西，如收集某些物品的碎片样本、收集一堆符咒或勋章来填满游戏中的储藏柜。对于“战斗”，相信大家都很熟悉。“内容解锁”意味着玩家需要完成一些事物，可能是一项成就，也可能是升到一定的级别，从而可以接触到游戏中一些新的内容。“赠送礼物”，不言而喻，送给别人东西时，人们通常会觉得很有趣，这也在一定程度上增强了人们玩游戏的意愿。“排行榜”，即按照成绩对玩家进行排序，下面会做详细解说。“等级”和“积分”也会在之后重点描述。还有“任务”，它和成就类似，是一种游戏中的概念，意味着你需要去完成一些游戏中明确要求的事情。“社交图谱”，玩过《帝国与盟军》的用户都见过它，就是找到参与游戏的朋友并与他们互动从而进行合作或对抗，这在很大程度上使游戏成为人们社交网络的体验延伸。至于“团队”，也是大家所熟悉的，在之前也提到过。最后是“虚拟商品”，即就是游戏中的东西，在后面会详细分析。

以上列出了几十种不同的游戏元素，需要明确的是，这不是一个万能的完整列表，但它却很好地展示了游戏元素的丰富性，包括了在实施游戏化系统时必须考虑到的很多选项。这个金字塔结构意味着，我们有一个狭窄的顶部和一个宽阔的底部，也就是说，属于动力的组件比属于元素的组件要少得多。这些也进一步表明了关键部分在于较高层次的组件，底层的元素则是实现其上层结构的途径，这正是整个系统的构成方式。

# 第5章 游戏化设计模式

## 5.1 常见的游戏元素

尽管所有的游戏元素都是平等的，但不可否认，有些游戏元素确实比其它一些游戏元素更为常见。在那些游戏化的经典案例中，它们的影响也更大。在这里，我将强调并详细说明其中三点，它们的缩写为PBL，分别代表积分（Points）、徽章（Badges）和排行榜（Leader boards），因为在游戏化中，人们总会不由自主地使用它们，部分原因在于它们拥有多种功能，比一般人们所能想到的更加丰富。很多常见的web游戏化案例中，至少是那些比较标准的游戏化应用，都把PBL作为其核心元素。

### 5.1.1 积分



图5-1 中国移动官网的积分模块

让我们从“积分”说起，那么积分有什么意义呢？相信图5-1这个页面大多数人都很熟悉，这是中国移动官网的积分商城，它采用了典型的游戏化设计模式，其中，积分这个元素较为醒目。

首先，积分是一种计分方式，如果一个用户有100分，而另一个用户有5000分，我们自然可以知道后者取胜。也就是说，积分可以衡量玩家在游戏中的表现，显示玩家的相对排名或者用来界定胜利条件，比如第一个得到5000分的用户就是赢家。但是，积分也可以有其它功能，它可以和奖励联系起来，比如说，你需要用5000分区兑换一份奖励，可以是一些免费的小礼品或是游戏化系统中的某些徽章。积分还能提供反馈，比如拿到了5000分的用户就能很好地证明自己比那些拿到了100分的用户做得更好。所以，作为一种反馈机制，积分能实时反应你在游戏中的表现。积分也是一种显示过程的方式，我们在此之前解释过“过程”这一顶层动力组件，这种前往阶梯更高位置的动力十分重要，所以一方面积分会给我们反馈，另一方面也告诉我们，你正在沿着阶梯一步步向上移动。

积分还能为游戏设计师提供数据，他们会知道你在游戏中获得了多少分、在哪里得分、需要花费多长的时间来得到这些积分等等。而这些数据可以用于进一步改善游戏或游戏化系统。

最后，积分的优点在于，一分就是一分，每一分都是相等的，也都是平等的。因此，只要你设计出一个积分系统，你就能设计出拥有多个独立积分系统的游戏或游戏化系统。总体而言，每一分的价值都是相同的，这也意味着积分可以用来代表所有事物，它可以被作为通用货币，举个例子，我们可以建立一个系统，在这个系统中的一系列行为，比如“和小伙伴们一起做任务”，在某种程度上和其他行为是相等的并且可比较的。以上这些都是分数能够起到的作用，它们对于web游戏化而言都很有价值，值得我们去学习和借鉴。

### 5.1.2 徽章

徽章是成就的一种体现，它从视觉上表明了玩家已经达到了一定的等级或者完成了游戏化系统中设定的一些目标。它们通常都是纽扣状的图标，被显示在玩家档案或者其他玩家可以看见的地方，以展示玩家所获得的成就。

徽章的好处很多，现在很多优秀的游戏化服务都在以不同的方式使用徽章。首先，徽章极具灵活性，可以用来代表一切事物，它就是一个像盾牌或者纽扣之类的小东西，这就是一块空白的画布，可以作为任何一件事的奖励或者纪念。举个例子，或许你会因为第一次做某事而得到一枚徽章，又或许重复做某事100次而得到一枚徽章，甚至是在某天早上，你无意中就获得了一枚徽章，只因为设计师想送给你一个小小的惊喜。徽章可以代表设计师想到的任何事物，这是非常强大的，因为游戏化系统的目的就在于想要激发用户的某些特定行为，从而得到商业或者其他背景下的特定结果，于是徽章成为了一种很好的传递方式。

此外，徽章还能表达风格，因为徽章是一种图案，可以有自己的美术风格，它的设计元素和模式可以体现一种氛围或游戏化系统的整体美学。徽章还可以体现重要程度，它可以指出游戏中重要的东西，比如游戏中可能会有成就、等级或者积分等各种元素，而徽章能够体现出在这个游戏或游戏化系统中，什么是重要的，什么是值得鼓励的。徽章还有认证的功能，它会告诉所有看到它的人，这是当前用户做过的事情，就像我们每个人的大学学位证书一样。



图5-2 国外网站中的徽章模块

与此同时，徽章还具有收藏价值。如图5-2所示，如果有一个柜子可以存放各种徽章，那么对于玩家而言，就会产生一种诱惑去把它填满。因为他们渴望完整，想要从不同的地方集齐东西来填满它，而填满柜子可就变成了他们的目标，这是一种结构或者机制，能够有效地促使人们乐于参与并且玩得更多。

最后，徽章还是身份的象征，它向人们展示了当前用户完成了某些任务或者某件独一无二的事情。如今的游戏化中，徽章未必一定要体现为身份的象征，但如果一个游戏化系统中需要彰显玩家的身份与地位，那么徽章自然就成为一种典型的选择。

### 5.1.3 排行榜

最后是“排行榜”，通俗来讲，就是排名。如图5-3，这是百度音乐排行榜页面的一部分，相信大家再熟悉不过。



图5-3 百度音乐中的排行榜模块

那么排行榜的作用是什么呢？它可以告诉玩家相对于其他玩家在游戏中所处的位置。有一种看法是，与我们现在讨论的所有这些元素一样，排行榜也是一种反馈形式，确切的说，这种反馈存在于竞争之中。也就是说，和游戏或游戏化系统中的其他用户相比，你做的如何？排行榜在电子游戏和街机游戏中十分常见，但是它存在一定的问题，如果你看到自己的分数，假如是500分，然后你去看排行榜，发现排在第一的玩家有5000000000分，你会觉得这个差距简直大到难以逾越，而这种感觉并不好，会让你对自己的得分丧失信心，你会下意识地认为自己玩不好这个游戏。以上这些可能会导致部分玩家得不到应有的乐趣并因此放弃游戏，所有大多数社交游戏对排行榜进行了一些改造和创新，我们称其为“个人定制排行榜”。举个例子，你有500分，而你看到的并不是所有玩家的排名，你将出现在排行榜的中间部分，排在你前面的那个玩家有800分，排在你后面的那个玩家有400分，还有个玩家有300分，等等。现在你的分数看起来排在中间，你只能看到你所处的那个分数段的排名，而看不到那种比你高出很多分的情况。还有一种方式叫做“好友排行榜”，在这个定制的排行榜中，你只能看到自己在好友中的排名，他们都在你的社交图谱中，这会使你不用和陌生人去比较，而是和那些你所熟悉的人去竞争。以上都是社交游戏中对排行榜的一些革新，它们同样可以运用到web游戏化中。

但是，排行榜仍然存在一些风险，这也是所有游戏元素都面临的问题，我们将在之后做详细讨论。研究显示，很多情境下。排行榜反而会挫败用户的积极性，它经常被用于销售竞赛，比如本周业绩谁获得了第一就能得到相应的奖励，名字也会显示在排行榜的顶部。公司的目的本应该是通过这种方式推动大家的积极性，看到别人排在你前面或者看到自己落后了一点点，你都可能会更加努力。但是在很多使用排行榜的案例中却并非如此，有时候竞争反而使人们失去了参与的兴趣，由于排行榜使用户的关注点有所转移，他们反而会变得并不愿意去努力尝试，甚至干脆选择放弃。因此，我们不得不讨论使用游戏元素，尤其是PBL进行设计时，需要注意的局限和问题。

## 5.2 游戏元素的局限性

现在，我们已经了解了各种游戏元素，尤其是积分、徽章和排行榜，然而游戏元素只是游戏化的起点，是可以供我们使用、部署的原料与工具。坦白的说，游戏化还存在一个严重的问题，即很多公司认为，只要将这些游戏元素都扔进业务流程，就会像变魔术一样，几乎不需要任何努力，就能把事物变得更像游戏，变得有趣并且吸引人。

但是，游戏元素只是起点，并不是全部。想象一下，如果一个游戏化系统中只有这些元素，比如在飞行常客奖励计划中，你会得到积分和免费里程奖励，你的级别会升高，假如你是白金会员，还能因为积累飞行里程而获得徽章，就像这些代表不同级别的信用卡，还有人们一定会喜欢的“进度条”和“地位”。人们通常认为，地位在游戏化中作用巨大，因为人们喜欢去炫耀自己的积分和徽章等。但是，这样就会有很多人把所有时间都花在飞行常客奖励计划上而不停地去玩这个飞行常客“游戏”吗？事实上并非如此，人们并不是这样看待这个体验过程的。他们喜欢获得飞行常客里程积分并不意味着他们认为这个游戏吸引人，只有很小一部分核心客户才会非常投入地去挖掘常客奖励计划能够带来的好处。

更重要的一点是，游戏元素不是游戏，在此之前我们就讨论过，体验和游戏不同，元素和游戏也不同，游戏处于元素和体验之间，如果我们只注重游戏元素，自然会觉得这个网站有徽章就很好了，有成就系统就很不错了，还可以做任务简直妙极了，但这些都不能决定网站的成功与否，真正让这些元素成功的方法是把它们结合在一起，这通常会涉及到“动力”这些较高层次的概念，而我们已经知道积分、徽章和排行榜都处于金字塔的最底层，它们只是属于服务级别的组件，所以，元素本身并不能告诉你，体验是否有趣或是否吸引人。

还有一方面，如果我们过分关注这些游戏元素，尤其是积分、徽章和排行榜，就容易高估奖励的作用。心理学研究表明，奖励在某种程度上反而会降低人们的积极性，也就是说得到徽章甚至现金的奖励实际上会降低人们的参与度，也会降低人们全力以赴的可能性，这也不难说明奖励和乐趣并不相同。那么，乐趣是什么呢？大多数理论都指出乐趣可以是奖励之外的东西，可以是朋友之间的互动，可以是精力的释放，可以是解决一项难题，可以是探索发现等。很多事情都可以是有趣的，事实上，得到奖励并不等于就是有趣的体验，而有趣的体验也并不意味着必须和奖励有关。在游戏化中，奖励并不是PBL的必要条件，但在具体实施 时，它却又经常成为人们关注的焦点。这是存在问题的地方，奖励本身未必有错，但是如果设计师只关注到这一点就容易产生巨大的危险，系统就不能从用户的实际参与中得到真实的结果。

最后，如果积分、徽章和排行榜是我们游戏化网站的核心，那么它看起来可能会和其他拥有PBL的游戏化网站一样，而这也会导致以下两个问题。首先，用户可能会觉得这些网站都大同小异。其次，用户可能会产生疲劳，他们会觉得自己已经在其他网站上收集过徽章，即使在刚开始的时候挺有趣并且吸引自己继续参与下去，但为什么又要在这里重新开始收集？这并不好玩。

所以，过分关注积分、徽章和排行榜等游戏元素是目前游戏化系统中存在的普遍问题。还有很多理由能够说明我们不应该仅仅局限于游戏元素。举一个谷歌的例子，从各个层面上来说，谷歌都是一个相当优秀和成功的公司，但它也有过并不成功的项目案例，比如谷歌新闻徽章。在2011年夏天，谷歌宣布将在谷歌新闻中增添游戏化元素，他们宣布在浏览和阅读谷歌新闻时，根据阅读主题的不同，用户将得到相应的徽章。假如一个用户读了很多关于篮球的文章，就会获得一枚篮球徽章。可怕的是，这些徽章都是突然冒出来的，谷歌并没有提供一个对应的专栏，用户只是因为读了很多有关篮球的文章就能突然看见这些徽章了。谷歌当时还列举了一大堆理由来说明徽章的好处，比如，这可以作为用户的阅读记录而向其他用户展示当前用户的阅读内容，还可以用来和其他用户比较在某个领域读过的文章数量，但这些理由真的具有说服力吗？我认为没有。假如我喜欢读关于篮球的文章，可是我并不需要徽章来告诉我，我读了很多这方面的内容，即使真的需要又能怎么样？这些徽章并没有真正给与我。换句话说，徽章体现成就，而这里所谓的“成就”，只是表明我正好阅读了很多相同主题的文章。因此，我并不清楚谷歌新闻徽章是如何激励和吸引读者去做他们从未做过的事情。实际上，谷歌最近宣布取消了新闻徽章这个功能。那么到底是哪里出现了问题呢？看起来他们已经看到了游戏化的吸引力并且觉得应该去尝试。但这么想其实很危险，因为这可能会让你在网页上放一些并不能真正创造商业价值的东西。

需要强调的是，元素可能是有用的，是游戏化的良好开端，但是它并不是游戏本身，如果只专注于这些元素，那么那些让事物变得像游戏一样的有意义的选择又在哪里？在点击鼠标一次和一百次之间做决定，或是在看视频和上网之间做决定，这些都不是真正重大的选择。而重大的选择需要人们进行思考，权衡利弊之后再做出决定。所以，并不是只要有选择就能吸引用户，同样，挑战也未必就是难题，设想一下，如果你要做的某件事，需要耗费你更多的精力，就像点击鼠标一千次比点击一次需要耗费更多精力一样，但是点击一千次也并没有什么挑战性，显然我们不会将它看作是难题。而难题是需要解决问题、审慎思考和创造革新等能力的。不是每一个游戏化网站都具有上述的元素，这些元素也未必都适合所有的个例。需要说明的是，一大堆徽章或者积分确实可以证明你的能力，或者证明你是某个方面的专家。如果只是为了熟练而设置阶梯型的熟练度上升过程，这也许会有一点点价值，但是如果能够给予人们真正掌握一些技能的机会，那么这样的体验才会变得更加丰富和吸引人。还有社区，之前我们提过，社交互动在游戏中是极其强大的，但通常基于PBL的网站，不会有与他人互动的设定，只是不断地为自己收集东西。最后，每个人是不同的，如果系统框架只提供了某一类的任务，这就很难吸引到其他不同动机的用户，毕竟每个人对于“乐趣”这个概念有着不同的定义。

以上这些就是只关注游戏元素的局限性，尽管游戏元素非常重要，我们只有掌握它们才能继续web游戏化的道路，但我们也绝不能止步于此。

# 第6章 Web游戏化需求分析

## 6.1 引言

### 6.1.1 编写目的

陈述游戏化网站系统BIT微课堂的基本功能、总体设计和模块实现。

### 6.1.2 名词定义

1. 游戏化：将游戏的思维和游戏的机制运用到其他领域，从而引导用户互动和使用的方法。
2. J2EE：Java2平台企业版，核心是一组技术规范与指南，其中所包含的各类组件、服务架构及技术层次，均有共同的标准及规格。
3. JSP：全称为Java Server Pages，是由Sun Microsystems公司倡导、许多公司参与建立的一种[动态网页](http://baike.baidu.com/view/348756.htm)技术标准。
4. Servlet：全称Java Servlet，是用Java编写的[服务器端程序](http://baike.soso.com/ShowTitle.e?sp=S服务器端程序)，其主要功能在于[交互式](http://baike.soso.com/ShowTitle.e?sp=S交互式)地浏览和修改数据，生成动态Web内容。
5. JavaBean：一种用Java语言编写的可重用组件。

## 6.2 任务概述

### 6.2.1 一般描述

BIT微课堂是一个将游戏元素和游戏化设计模式融入其中的在线学习网站系统，用户群体为北京理工大学的全体学生。该系统的构建旨在为同学们提供一个课堂之外的更为广阔的学习空间。而网站系统的实现采用了J2EE架构下的JSP + Servlet + JavaBean技术模式，便于后期开发和维护。

### 6.2.2 需求分析

通过走访调研和问卷调查，我们将游戏化在线学习网站系统——BIT微课堂的用户需求和预期功能总结如下：

1. 用户使用邮箱注册个人账号，通过填写相关的注册信息，成为BIT微课堂的认证用户；
2. 注册成功后，用户可以使用邮箱账号和密码进行验证登录；
3. 登录成功后，用户进入个人主页，个人主页主要分为学习空间、学友之家、成就展厅、超级榜单、消息提示和修改资料六大版块；
4. 在个人主页中设置风采展示版块，用户在该版块中可以进行头像的上传和更新，并可以查看自己的积分值、徽章数、获赞数、访客量和魅力指数等游戏化相关的数值；
5. 在学习空间版块中，用户可以查看自己的日志列表，并可以通过添加日志的功能上传学习日志，供其他用户阅览学习；
6. 在学友之家版块中，用户可以选择搜索条件或填写相关的搜索信息来查找该网站的注册用户或学习资料；
7. 在成就展厅版块中，用户可以查看自己获得的系统荣誉徽章，与此同时，也可以了解徽章对应的获取方式和获取人数；
8. 在超级榜单版块中，用户可以查看各种排行榜的前十名用户，包括积分排行榜、获赞排行榜和点赞排行榜等；
9. 在消息提示版块中，用户可以查看收到的留言并选择是否回复；
10. 在修改资料版块中，用户可以更改自己的个人信息和学习宣言，也可以选择重置登录密码；
11. 如果用户决定停用当前账号，可以进入注销页面进行账号注销；
12. 用户通过多种方式查找到其他用户，并可以进入他们的主页进行访问；
13. 访问页面展示其他用户的基本资料、荣誉徽章和日志列表等，用户可以在访问页面中对被访问的用户进行点赞。

### 6.2.3 运行环境

该网站系统开发的操作系统为Windows7，使用My Eclipse进行编译，其中数据库使用My SQL，服务器为Tomcat7。

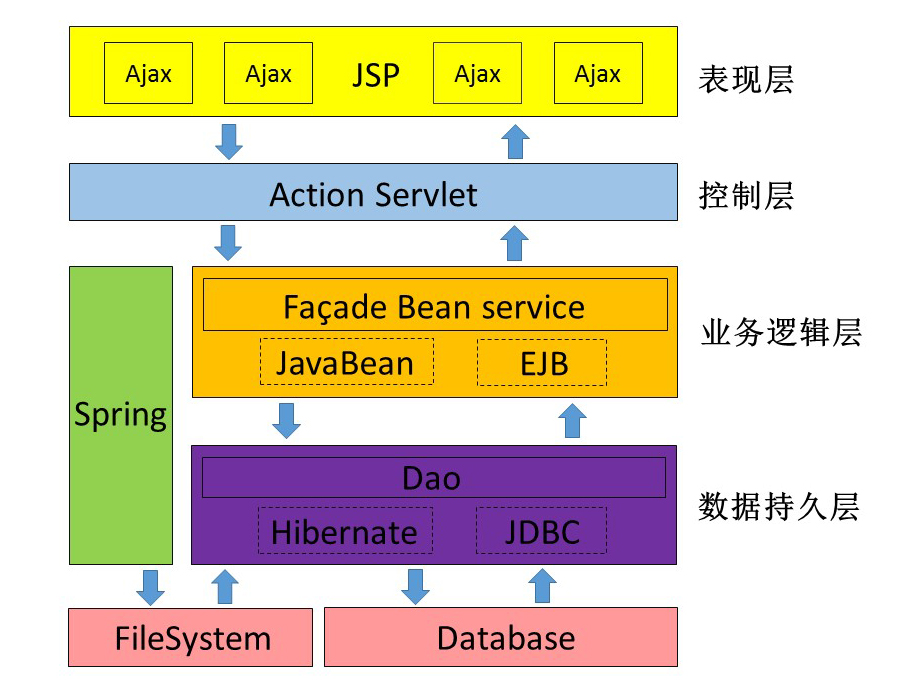
硬件要求为大于等于1.0GHz32位及以上的CPU、大于等于1G的内存（32位）或2G的内存（64位）、大于等于16G的硬盘（32位）或20G的硬盘（64位）、带有WDDM1.0或更高驱动、支持DirectX9.0以上的设备。

# 第7章 Web游戏化详细设计

## 7.1 总体设计

### 7.1.1 处理流程

BIT微课堂网站系统采用图7-1所示的J2EE架构开发实现。

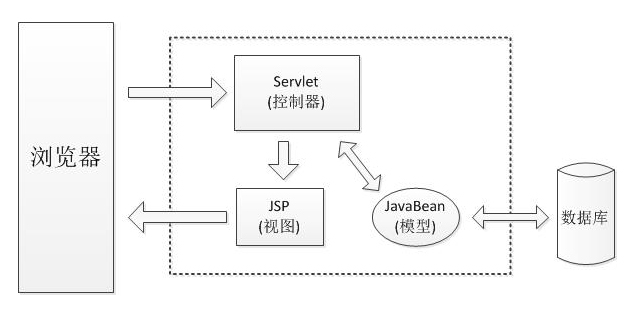


7-1 J2EE模型架构图

结合J2EE架构图，我们可将BIT微课堂网站系统的处理流程总结如下：

1. 用户通过前端页面发出了一个请求，该请求被转发给控制器；
2. 控制器收到该请求后，检查该请求并决定用怎样的业务逻辑来处理，然后控制器会将该请求转发给一个相应的业务组件去处理；
3. 模型中包含处理该用户请求的所有业务组件，并能够执行用户所需要的全部数据存取，用户查询检索出的任何数据都被打包返回给控制器；
4. 控制器接收从模型返回的数据，并选择显示这些数据的相关视图，然后将视图返回给用户。

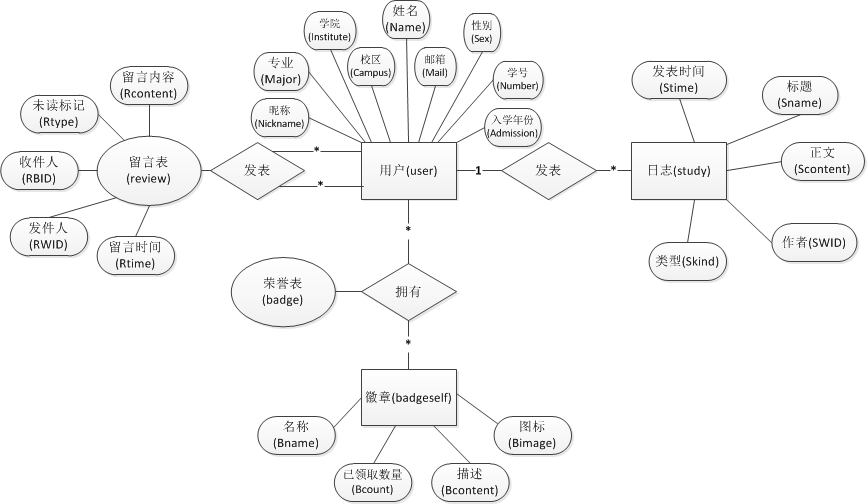
将以上过程进行提炼归纳后，可呈现为图7-2所示的系统处理流程模型。



7-2 BIT微课堂系统的处理流程

### 7.1.2 关系结构

根据需求分析和概要设计，最终可将该网站系统的总体结构用图7-3所示的E-R模型来描述。



7-3 BIT微课堂系统E-R模型

其中，本系统共有三个实体，分别是用户、日志和徽章。由于用户这一实体的属性过多，故无法在模型中进行穷举。

### 7.1.3 数据库设计

考虑到网站的整体功能、用户的实际需求和开发人员的可实现程度，本网站的数据库将由user、study、review、badge和badgeself五张表构成。

user表的结构如表7-1所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 表7-1 user表 | | | |
| 字段 | 类型 | 是否为空 | 默认 |
| ID | int(20) | not null | 无 |
| Nickname | varchar(200) | not null | 无 |
| Password | varchar(200) | not null | 无 |
| Name | varchar(200) | not null | 无 |
| Mail | varchar(200) | not null | 无 |
| Sex | varchar(20) | not null | 无 |
| Campus | varchar(200) | null | null |
| Admission | int(10) | null | null |
| Institute | varchar(200) | null | null |
| Major | varchar(200) | null | null |
| Number | varchar(200) | null | null |
| Score | int(10) | null | 0 |
| Count | int(10) | null | 0 |
| VisitorNumber | int(10) | null | 0 |
| Good | int(10) | null | 0 |
| Badge | int(10) | null | 0 |
| Intro | Longtext | null | null |
| Photo | varchar(2000) | null | null |
| Mei | int(10) | null | 0 |

其中，ID为主键，且设置为自增。

study表的结构如表7-2所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 表7-2 study表 | | | |
| 字段 | 类型 | 是否为空 | 默认 |
| SID | int(20) | not null | 无 |
| Sname | varchar(100) | not null | 无 |
| SContent | longtext | not null | 无 |
| SWID | int(20) | not null | 无 |
| Skind | varchar(100) | not null | 无 |
| Stime | varchar(100) | not null | 无 |

其中，SID为主键，且设置为自增。SWID为外键，参考user表的ID属性。

review表的结构如表7-3所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 表7-3 review表 | | | |
| 字段 | 类型 | 是否为空 | 默认 |
| RID | int(20) | not null | 无 |
| RContent | Longtext | not null | 无 |
| Rtype | int(3) | not null | 无 |
| RBID | int(20) | not null | 无 |
| RWID | int(20) | not null | 无 |
| Rtime | varchar(100) | not null | 无 |

其中，RID为主键，且设置为自增。RBID和RWID均为外键，都参考user表的ID属性。

badge表的结构如表7-4所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 表7-4 badge表 | | | |
| 字段 | 类型 | 是否为空 | 默认 |
| BID | int(20) | not null | 无 |
| BBID | int(20) | not null | 无 |
| BUID | int(20) | not null | 无 |

其中，BID为主键，且设置为自增。BBID和BUID均为外键，BUID参考user表的ID属性，BBID参考badgeself表的BBID属性。

Badgeself表的结构如7-5所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 表7-5 Badgeself表 | | | |
| 字段 | 类型 | 是否为空 | 默认 |
| BBID | int(20) | not null | 无 |
| Bname | varchar(100) | not null | 无 |
| Bcount | int(20) | not null | 0 |
| Bcontent | varchar(500) | not null | 无 |
| Bimage | varchar(100) | not null | 无 |

其中，BBID为主键，且设置为自增。

## 7.2 接口设计

### 7.2.1功能与用户界面

本次项目开发将网站风格定义为活泼萌动，因为网站的用户主要面向广大北理工学生，即用户群体偏向年轻化和个性化，这样的风格更容易吸引用户，拉近与用户之间的距离，使其产生好感。

如图7-4所示，这是BIT微课堂网站系统的索引页面，该页面作为新老用户的引导页面，在引导新用户快速注册的同时，也能够方便地引导老用户跳转到登录页面。



图7-4 BIT微课堂索引页面

如图7-5所示，这是BIT微课堂网站系统的登录页面，已经成功获取账号的用户可以通过注册邮箱和密码成功进入个人主页，没有获取账号的用户可以跳转到登录页面。



图7-5 BIT微课堂登录页面

如图7-6所示，这是BIT微课堂网站系统的注册页面，没有账号的用户可以在该页面填写必要的注册信息获取登录账号，已有账号的用户可以直接跳转到登录页面进行登录。



图7-6 BIT微课堂注册页面

如图7-7所示，这是BIT微课堂网站系统的用户个人主页页面。在该页面中，左半部分是用户的风采展示栏，用户可以方便地上传头像来替换系统默认头像。头像下方分别是用户的当前积分值、荣誉徽章数、被赞次数、访客数量和魅力指数，其中，魅力指数是指被赞次数占访客数量的比例。这一栏很好地体现了游戏化设计的理念，能够有效地激励用户去做一些有助于提高积分的任务，而这些任务包括发表学习日志，留言和回复他人留言等。

右半部分是日志列表，用户可以查看已发表的学习日志，也可以点击“添加日志”按钮，跳转到日志编辑页面。其中，上方的导航栏也可以方便地跳转到任何相应的功能页面。这些页面包括：搜索用户和日志页面（学友之家）、查看已获荣誉徽章页面（成就展厅）、系统排行榜页面（超级榜单）、查看与回复留言页面（消息提示）和更改个人资料页面（修改资料）。



图7-7 BIT微课堂个人日志列表页面

如图7-8所示，这是BIT微课堂网站系统的日志查看页面，页面中显示学习日志的详细内容，包括日志标题、日志作者、日志内容和日志发表时间等。在个人主页的学友空间中点击任意一篇文章的链接即可跳转到该页面。

在日志的详细内容下面，点击“返回日志列表”即可返回到学友空间页面，点击“删除日志”即可删除当前查看的学习日志。

而左半部分依旧是风采展示栏，与学友空间中一样，这样做的好处是，能够从始至终贯彻游戏化的设计理念，提醒用户时时刻刻不忘专注自己的积分值、徽章数和魅力指数等，从而激励用户坚持学习。



图7-8 BIT微课堂日志查看页面

如图7-9所示，这是BIT微课堂网站系统中的日志编辑页面。用户可以在该页面中轻松便捷地发布自己的学习日志。填写日志标题后，选择日志类型，最后在文本编辑器中自由发挥即可。考虑到不同学科的内容差异性，本系统选择了UM-Editor这样一款功能强大的开源编辑器，用户可以自由地上传图片、公式和表格等相关内容，以便更好更深入地讲解各科知识。最后，当用户点击“发布日志”按钮后，即可跳转回到学习空间的日志列表，并且最新发布的日志将排在列表的首位，方便查看阅览。

如图7-10所示，这是BIT微课堂网站系统中的搜索页面。之所以名为“学友之家”，是因为用户可以在该页面搜索到理想的学习用户和学习资料。通过选择查找条件和填写查找内容，点击“查找用户”或“查找日志”按钮，即可跳转到对应的搜索结果页面。



图7-9 BIT微课堂添加日志页面



图7-10 BIT微课堂搜索页面



图7-11 BIT微课堂用户搜索结果页面



图7-12 BIT微课堂日志搜索结果页面

如图7-11所示，这是BIT微课堂网站系统中的用户搜索结果页面，通过在学友之家页面选择用户性别、校区，或通过填写用户昵称、入学年份、学院和专业查找指定用户，查找结果呈现在该页面中。

如图7-12所示，这是BIT微课堂网站系统中的日志搜索结果页面，通过在学友之家页面选择日志类型，或通过填写日志标题中的关键字查找目标日志，查找结果呈现在该页面中。

如图7-13所示，这是BIT微课堂网站系统中的游戏化模块之一，徽章展示页面。该页面显示当前登录用户所获取的个性徽章列表，包括徽章图标、徽章简介和徽章获取人数，这样有利于激励用户不断探索BIT微课堂网站系统的新功能，也有助于激发用户连续登录网站和坚持发表日志。



图7-13 BIT微课堂徽章展示页面

如图7-14所示，这是BIT微课堂网站系统的排行榜页面，与积分版块和徽章版块一样，这也是典型的游戏化设计模块之一。用户可在该页面中点击查看积分排行（超级学霸榜）、获赞排行（分为魅力男神榜和魅力女神榜）、点赞排行（点赞狂魔榜）等。通过了解自己在榜单中的地位，有助于激励用户去做一些促进排名的事情，如发布日志获取积分、浏览他人主页、给他人留言、为他人点赞等。



图7-14 BIT微课堂排行榜页面

如图7-15所示，这是BIT微课堂网站系统的超级学霸排行榜页面。所谓超级学霸榜，顾名思义，就是按学习狂热程度来排名。而该网站中能够体现出学习狂热程度的虚拟物品为用户积分值。积分值越高，代表用户发表学习日志的频率越高，也意味着用户为该系统贡献的学习资料越多。与此同时，针对其他用户的日志给其留言，或者回复其他用户的留言问题，都可以获得相应的积分值。故此排行榜实际上反映了用户的积分排名。

此外，魅力男神榜页面、魅力女神榜页面和点赞狂魔榜页面分别显示了获赞最多的男性用户排名、获赞最多的女性用户排名和点赞最多的用户排名，这些排名作为游戏化设计模式的表现方式，都在很大程度上促进用户参与网站系统的功能使用，同时促使BIT微课堂网站系统更加趋于社交化，而这也是当前网站设计的主流发展方向。由于这些榜单的界面与学霸排行非常相似，故没有在此全部展示。



图7-15 BIT微课堂超级学霸榜页面

如图7-16所示，这是BIT微课堂网站系统的消息提示页面，而消息列表主要由用户留言组成。如果用户已查看了某个留言的详细内容，则该留言进入历史消息版块，否则标红并显示未读，以提醒用户去查看并及时回复留言。在此页面中，用户还可以点击访问留言用户的个人主页。

如图7-17所示，这是BIT微课堂网站系统的回复消息页面。在消息提示页面中点击任意消息后的“查看详情”即可跳转到该页面。进入页面后不仅可以看到留言内容，而且可以在文本框中编辑回复内容，发送成功后，该回复将显示在对方个人主页的新消息列表中。



图7-16 BIT微课堂消息提示页面



图7-17 BIT微课堂回复消息页面

如图7-18所示，这是BIT微课堂网站系统的修改个人资料页面，用户可以在该页面中选择和填写想要更改的个人信息，包括昵称、校区、学院、专业和学习宣言等。此外，用户还可以在此页面重置登录密码。



图7-18 BIT微课堂修改个人资料页面

如图7-19所示，这是BIT微课堂网站系统的用户注销页面。当用户决定停用账号时，可以点击任意页面右上方的“注销账号”进入到该页面。如果坚持停用，点击“残忍注销”按钮即可。注销成功后将跳转到网站系统的索引页面，如果想要再次进入系统，必须重新注册账号。



图7-19 BIT微课堂注销页面

如图7-20所示，这是BIT微课堂网站系统的好友主页页面。用户登录后，可以在学友之家版块的用户搜索结果页面、超级榜单版块的各种排行榜页面和消息提示版块的消息列表中，通过点击任意用户的链接，即可前往该页面查看相关用户的个人信息和学习资料。

在好友主页页面中，用户可以查看他人的简要介绍，包括入学年份、校区、学院和专业等，也可以在互动留言版块给他人留言。与此同时，可以浏览他人的日志列表，也可以看到他人的学习宣言和获取的荣誉徽章。



图7-20 BIT微课堂访问用户页面

如图7-21所示，这是BIT微课堂系统的点赞成功页面。用户在他人主页的学友名片版块点击“为他点赞”或“为她点赞”，即可跳转到该页面中。点赞成功后，用户不仅可以看到自己的点赞数量，而且可以根据提示跳转到不同页面继续浏览信息。



图7-21 BIT微课堂点赞成功页面



图7-22 BIT微课堂关键字搜索结果页面

如图7-22所示，这是BIT微课堂网站系统的好友主页关键字搜索结果页面。在好友主页的日志列表版块中，通过输入关键字可快速精确查找到相关的学习日志，查找结果呈现在该页面中。

如果7-23所示，这是BIT微课堂网站系统的查看用户日志页面。在好友主页的日志列表或通过关键字查找到的日志列表中，通过点击任意日志的链接，即可跳转到该页面中查看日志详细内容并进行学习。点击日志内容下方的“返回日志列表”即可返回到好友主页中查看其他日志。



图7-23 BIT微课堂查看用户日志页面

### 7.2.2 内部接口

在本次项目开发中，内部接口采用了JSP + Servlet + JavaBean的技术模式，这是一种采用Java编程语言来实现CGI功能的方式。官方给出的解释是Servlet技术是运行在Web服务器上用以生成Web页面，该技术适用于服务器端的处理和编程，并且Servlet将长期驻留在固定位置。在大多数实际的项目开发过程中，前端的页面设计者可以非常方便地使用普通的HTML工具来开发JSP页面，而Servlet交给后端的开发者即可。

总体而言，这种JSP + Servlet + JavaBean的技术模式结合并充分体现了JSP和Servlet技术的优点。在这种开发模式下，通过JSP页面来表现页面，通过Servlet来处理大量的事务。其中，Servlet就像一个幕后的控制者，它不仅创建JSP页面所需要的Bean和Object，还将根据用户的请求行为决定将哪个JSP页面发送回去。

## 7.3 类和函数的设计

### 7.3.1 类的属性

在该软件系统中共建立了三个包，分别是javabean，servlet和dao。为了方便查找和使用，我们将各种功能相似的类放在同一个包中。而本系统中类的总体设计分类如表7-6所示。

|  |  |  |
| --- | --- | --- |
| 表7-6 类的属性表 | | |
| 类的名称 | 类的属性 | 类的访问权限 |
| Database | none | public |
| UserBean | id, mail, password, name, nickname, sex, campus, admission, major, institute, number, score, count, visit, good, badge, intro, photo, mei | public |
| StudyBean | sid, swid, swname, sname, scontent, skind, stime | public |
| ReviewBean | rid, rbid, rwid, rtype, rcontent, rtime, rwname, rwsex | public |
| BadgeBean | bid, bbid, buid | public |
| BadgeselfBean | bbid, bcount, bname, bcontent, bimge | public |
| UserServlet | none | public |
| StudyServlet | none | public |
| ReviewServlet | none | public |
| BadgeServlet | none | public |
| UploadUserIcon | none | public |
| UserDao | none | public |
| StudyDao | none | public |
| ReviewDao | none | public |
| BadgeDao | none | public |

### 7.3.2 类的方法

由于本系统存在较多的类，每个类涉及过多的方法，而很多类和方法在功能和逻辑上类似，故不在此穷举所有的类和全部的方法。以UserServlet、Database、UserBean和UserDao这四个类为例，其包含的方法可描述为表7-7、表7-8、表7-9和表7-10。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表7-7 UserServlet类的方法设计 | | | | |
| 方法名称 | 基本功能 | 输入 | 输出 | 程序逻辑 |
| doPost | 获取JSP页面提交的表单信息，并调用给相关函数 | HttpServletRequest, HttpServletResponse | none | 顺序执行  条件执行 |
| saveregister | 对index页面的快速注册栏进行Session设置 | HttpServletRequest, HttpServletResponse | none | 顺序执行 |
| addUser | 获取register页面的用户注册信息并传递到后台 | HttpServletRequest, HttpServletResponse | none | 顺序执行  条件执行 |
| validateUser | 查看register页面要求必填的注册信息是否为空 | HttpServletRequest, HttpServletResponse | boolean | 顺序执行  条件执行 |
| validateLogin | 对login页面获取的用户登录相关信息进行验证 | HttpServletRequest, HttpServletResponse | none | 顺序执行  条件执行 |
| validateLogin2 | 对index页面的用户登录信息进行Session设置 | HttpServletRequest, HttpServletResponse | none | 顺序执行 |
| validateLogin3 | 跳转页面时重新获取homepage页面的日志列表 | HttpServletRequest, HttpServletResponse | none | 顺序执行 |
| userupdate | 获取home\_upda页面的用户更改信息并传递到后台 | HttpServletRequest, HttpServletResponse | none | 顺序执行  条件执行 |
| searchuser | 获取home\_friend页面的用户搜索条件并传递后台 | HttpServletRequest, HttpServletResponse | none | 顺序执行 |
| boybank | 获取home\_bank页面的魅力男神榜榜单列表 | HttpServletRequest, HttpServletResponse | none | 顺序执行 |
| girlbank | 获取home\_bank页面的魅力女神榜榜单列表 | HttpServletRequest, HttpServletResponse | none | 顺序执行 |
| goodbank | 获取home\_bank页面的点赞狂魔榜榜单列表 | HttpServletRequest, HttpServletResponse | none | 顺序执行 |
| scorebank | 获取home\_bank页面的超级学霸榜榜单列表 | HttpServletRequest, HttpServletResponse | none | 顺序执行 |
| deleteUser | 获取logoff页面的用户ID并传递到后台进行注销 | HttpServletRequest, HttpServletResponse | none | 顺序执行  条件执行 |
| friendpage | 获取用户ID加载用户信息并跳转friendpage页面 | HttpServletRequest, HttpServletResponse | none | 顺序执行 |
| addGood | 获取点赞用户ID和被赞用户ID并传递到后台处理 | HttpServletRequest, HttpServletResponse | none | 顺序执行  条件执行 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表7-8 Database类的方法设计 | | | | |
| 方法名称 | 基本功能 | 输入 | 输出 | 程序逻辑 |
| createCon | 创建数据库连接 | none | none | 顺序执行 |
| getStm | 获取Statement对象 | none | none | 顺序执行 |
| executeUpdate | 对数据库的增加、修改和删除操作 | String | boolean | 顺序执行  条件执行 |
| executeQuery | 查询数据库操作 | String | boolean | 顺序执行 |
| closed | 关闭数据库的操作 | none | none | 顺序执行  条件执行 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表7-9 UserBean类的方法设计 | | | | |
| 方法名称 | 基本功能 | 输入 | 输出 | 程序逻辑 |
| getId | 获取用户ID值 | none | int | 顺序执行 |
| setId | 设置用户ID值 | int | none | 顺序执行 |
| getMail | 获取用户的注册邮箱 | none | String | 顺序执行 |
| setMail | 设置用户的注册邮箱 | String | none | 顺序执行 |
| getPassword | 获取用户的登录密码 | none | String | 顺序执行 |
| setPassword | 设置用户的登录密码 | String | none | 顺序执行 |
| getName | 获取用户的真实姓名 | none | String | 顺序执行 |
| setName | 设置用户的真实姓名 | String | none | 顺序执行 |
| getNickname | 获取用户的昵称 | none | String | 顺序执行 |
| setNickname | 设置用户的昵称 | String | none | 顺序执行 |
| getSex | 获取用户的性别 | none | String | 顺序执行 |
| setSex | 设置用户的性别 | String | none | 顺序执行 |
| getNumber | 获取用户的学号 | none | String | 顺序执行 |
| setNumber | 设置用户的学号 | String | none | 顺序执行 |
| getCampus | 获取用户的校区 | none | String | 顺序执行 |
| setCampus | 设置用户的校区 | String | none | 顺序执行 |
| getMajor | 获取用户的专业 | none | String | 顺序执行 |
| setMajor | 设置用户的专业 | String | none | 顺序执行 |
| getAdmission | 获取用户的入学年份 | none | String | 顺序执行 |
| setAdmission | 设置用户的入学年份 | String | none | 顺序执行 |
| getInstitute | 获取用户的学院 | none | String | 顺序执行 |
| setInstitute | 设置用户的学院 | String | none | 顺序执行 |
| getIntro | 获取用户的学习宣言 | none | String | 顺序执行 |
| setIntro | 设置用户的学习宣言 | String | none | 顺序执行 |
| getGood | 获取用户的被赞次数 | none | int | 顺序执行 |
| setGood | 设置用户的被赞次数 | int | none | 顺序执行 |
| getCount | 获取用户的点赞次数 | none | int | 顺序执行 |
| setCount | 设置用户的点赞次数 | int | none | 顺序执行 |
| getBadge | 获取用户的徽章个数 | none | int | 顺序执行 |
| setBadge | 设置用户的徽章个数 | int | none | 顺序执行 |
| getVisit | 获取用户的访客数量 | none | int | 顺序执行 |
| setVisit | 设置用户的访客数量 | int | none | 顺序执行 |
| getScore | 获取用户的积分值 | none | int | 顺序执行 |
| setScore | 设置用户的积分值 | int | none | 顺序执行 |
| getMei | 获取用户的魅力指数 | none | int | 顺序执行 |
| setMei | 设置用户的魅力指数 | int | none | 顺序执行 |
| getPhoto | 获取用户的头像路径 | none | String | 顺序执行 |
| setPhoto | 设置用户的头像路径 | String | none | 顺序执行 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表7-10 UserDao类的方法设计 | | | | |
| 方法名称 | 基本功能 | 输入 | 输出 | 程序逻辑 |
| setSession | 创建数据库连接 | none | none | 顺序执行 |
| frienduser | 获取Statement对象 | none | none | 顺序执行 |
| searchscore | 根据积分值对数据库记录进行排序并输出列表 | UserBean | ArrayList<UserBean> | 顺序执行条件执行  循环执行 |
| searchgirl | 根据被赞次数对数据库中性别为女的记录进行排序并输出列表 | UserBean | ArrayList<UserBean> | 顺序执行条件执行  循环执行 |
| searchboy | 根据被赞次数对数据库中性别为男的记录进行排序并输出列表 | UserBean | ArrayList<UserBean> | 顺序执行条件执行  循环执行 |
| searchgood | 根据点赞数量对数据库记录进行排序并输出列表 | UserBean | ArrayList<UserBean> | 顺序执行条件执行  循环执行 |
| userLogin | 根据用户登录信息从数据库中获取用户其他信息 | String  String | UserBean | 顺序执行  条件执行 |
| operationUser | 根据运算字符对user表相关记录进行增、删、改 | String  UserBean | boolean | 顺序执行  条件执行 |
| addGood | 根据ID给点赞用户的点赞数量加1，并给被赞用户的被赞次数加1 | int  int | boolean | 顺序执行 |
| addVisit | 根据用户ID给其访客数量加1 | int | none | 顺序执行 |
| addScore | 根据用户ID给用户积分值加10分 | int | none | 顺序执行 |
| addScore2 | 根据用户ID给用户积分值加3分 | int | none | 顺序执行 |
| checkUserExsit | 根据注册邮箱查询用户是否存在 | String | boolean | 顺序执行  条件执行 |
| searchuser | 构造条件语句查询目标用户列表 | UserBean | ArrayList<UserBean> | 顺序执行  条件执行  循环执行 |
| closeConnection | 关闭类与数据库之间的连接 | none | none | 顺序执行 |

# 第8章 Web游戏化系统实现

该游戏化网站的模块总体上分为：增加模块、删减模块、更新模块和查询模块，分别对应数据库中的增、删、改、查操作。接下来，我将细化和分类各模块，并给出不同模块对应的实现方式。

## 8.1 增加模块

该模块参照网站系统的功能可细化为用户注册、添加日志、留言与回复和颁发荣誉徽章四个较小的模块，其核心思想都是Insert。

### 8.1.1 用户注册

此模块本质上是为user表添加一条新的记录，在注册相关的JSP页面使用Post方法传递用户填写的信息，在UserServlet中获取相关的信息后，结合UserBean调用UserDao中封装好的方法即可。

实现该模块的核心代码如下：

|  |
| --- |
| public void addUser(HttpServletRequest request,  HttpServletResponse response) throws ServletException,  IOException, SQLException {  String forward = null;  UserDao userDao = new UserDao();  boolean flag = validateUser(request, response);  boolean exist = userDao.checkUserExsit (request.getParameter ("email"));  if (exist == false) {  if (flag) {  UserBean userBean = new UserBean();  userBean.setSex(request.getParameter("sex"));  userBean.setCampus(request.getParameter("campus")); userBean.setAdmission(request.getParameter("admission"));  userBean.setInstitute(request.getParameter("institute"));  userBean.setMajor(request.getParameter("major"));  userBean.setName(request.getParameter("name"));  userBean.setNumber(request.getParameter("number"));  userBean.setMail(request.getParameter("email"));  userBean.setNickname(request.getParameter("nickname"));  userBean.setPassword(request.getParameter("password"));  userBean.setIntro(request.getParameter("intro"));  boolean mark = userDao.operationUser("add", userBean);  if (mark) {  forward = "../login.jsp";  } else {  forward = "../register.jsp";  }  } else {  PrintWriter out = response.getWriter();  out.println("<script>window.history.go(-1)</script>");  }  response.sendRedirect(forward);  } else {  PrintWriter out = response.getWriter();  out.println("<script>window.history.go(-1)</script>");  }  userDao.closeConnection();  } |

### 8.1.2 添加日志

此模块本质上是为study表添加一条新的记录，在添加日志的JSP页面使用Post方法传递用户已编辑的文章，在StudyServlet中获取相关的信息后，结合StudyBean调用StudyDao中封装好的方法即可。

实现该模块的核心代码如下：

|  |
| --- |
| public void usereditor(HttpServletRequest request,  HttpServletResponse response) throws ServletException,  IOException, SQLException {  String forward = null;  UserBean user = new UserBean();  HttpSession session=request.getSession();  user = (UserBean)session.getAttribute("user");  StudyDao studyDao = new StudyDao();  boolean flag = validateStudy(request, response);  if(flag){  StudyBean studyBean = new StudyBean();  studyBean.setSname(request.getParameter("sname"));  studyBean.setSkind(request.getParameter("skind"));  studyBean.setScontent(request.getParameter("content"));  studyBean.setStime(getTime());  studyBean.setSWId(user.getId());  boolean mark = studyDao.operationStudy("add", studyBean);  UserDao userdao = new UserDao();  userdao.addScore(user.getId());  UserBean user2 = new UserBean();  UserDao userDao2 = new UserDao();  user2 = userDao2.setSession(user.getId());  session.setAttribute("user", user2);  ArrayList<StudyBean> studylist = new  ArrayList<StudyBean>();  studylist = studyDao.search\_user\_study(user.getId());  studyDao.closeConnection();  session.setAttribute("studylist", studylist);  if(mark){  forward = "../homepage.jsp";  }else{  forward = "../editor.jsp";  }  response.sendRedirect(forward);  }else{  PrintWriter out = response.getWriter();  out.println("<script>window.history.go(-1)</script>");  } |

### 8.1.3 留言与回复

此模块本质上是为review表添加一条新的记录，在回复和留言相关的JSP页面使用Post方法传递用户输入的信息，在ReviewServlet中获取相关的信息后，结合ReviewBean调用ReviewDao中封装好的方法即可。

实现该模块的核心代码如下：

|  |
| --- |
| public void addnews(HttpServletRequest request,  HttpServletResponse response) throws ServletException,  IOException, SQLException {  String forward = null;  ReviewBean news = new ReviewBean();  HttpSession session = request.getSession();  UserBean user = new UserBean();  user = (UserBean)session.getAttribute("user");  ReviewDao reviewDao = new ReviewDao();  String rcontent = request.getParameter("Rcontent");  if(rcontent == null || rcontent.equals("")){  PrintWriter out = response.getWriter();  out.println("<script>window.history.go(-1)</script>");  }else{  news.setRcontent(request.getParameter("Rcontent"));  news.setRWId(user.getId()); news.setRBId(Integer.parseInt  (request.getParameter("rbid")));  news.setRtime(getTime());  news.setRtype(0);  boolean mark = reviewDao.operationReview("add", news);  if(mark){  UserDao userdao = new UserDao();  userdao.addScore2(user.getId());  UserBean user2 = new UserBean();  UserDao userDao2 = new UserDao();  user2 = userDao2.setSession(user.getId());  session.setAttribute("user", user2);  forward = "../homepage.jsp";  }else{  PrintWriter out = response.getWriter(); out.println("<script>window.history.go(-1)  </script>");  }  response.sendRedirect(forward);  }  reviewDao.closeConnection();  } |

### 8.1.4 颁发荣誉徽章

此模块本质上是为Badge表添加一条新的记录，与以上三个模块不同的是，该模块不需要获取用户输入的信息，而是系统根据用户的活跃程度自动授予其相应的奖励，然后结合BadgeBean和BadgeselfBean调用BadgeDao中封装好的方法即可。

由于判断活跃程度涉及过多繁杂的算法，在这里不做详细赘述。该模块因为无需通过post方法传输数据，故在BadgeServlet中不作处理。

## 8.2 删减模块

该模块参照网站系统的功能可细化为用户注销和删除日志两个较小的模块，其核心思想都是Delete。

### 8.2.1 用户注销

此模块本质上是删除user表的一条记录。用户登录后其ID通过登录模块进行了存储，当用户不愿再登录该网站并且不希望被他人查看资料时进行注销将是最好的选择，通过在UserServlet中获取用户的ID即可调用UserDao中封装的方法对记录进行删除。

实现该模块的核心代码如下：

|  |
| --- |
| public void deleteUser(HttpServletRequest request,  HttpServletResponse response) throws ServletException,  IOException, SQLException {  String forward = null;  UserDao userdao = new UserDao();  UserBean userbean = new UserBean();  userbean.setId(Integer.parseInt  (request.getParameter("userid")));  System.out.println(userbean.getId());  boolean flag = userdao.operationUser("delete", userbean);  userdao.closeConnection();  if(flag){  forward = "../index.jsp";  }else{  forward = "../homepage.jsp";  }  response.sendRedirect(forward);  } |

### 8.2.2 删除日志

此模块本质上是删除study表的一条记录。用户在查看个人日志时，日志对应的SID将通过Post方法传递到StudyServlet中，当用户选择删除日志，后台通过获取到的SID调用StudyDao中封装的方法对记录进行删除。

实现该模块的核心代码如下：

|  |
| --- |
| public void deleteUser(HttpServletRequest request,  HttpServletResponse response) throws ServletException,  IOException, SQLException {  HttpSession session = request.getSession();  UserBean user = new UserBean();  user = (UserBean)session.getAttribute("user");  int sid = Integer.parseInt(request.getParameter("sid"));  StudyDao studyDao = new StudyDao();  boolean mark = studyDao.deletestudy(sid);  if(mark){  ArrayList<StudyBean> studylist =  new ArrayList<StudyBean>();  StudyDao studyDao2 = new StudyDao();  studylist = studyDao2.search\_user\_study(user.getId());  studyDao2.closeConnection();  session.setAttribute("studylist", studylist);  }else{  PrintWriter out = response.getWriter();  out.println("<script>window.history.go(-1)</script>");  }  studyDao.closeConnection();  response.sendRedirect("../homepage.jsp");  } |

## 8.3 更新模块

该模块参照网站系统的功能可细化为修改个人信息、上传头像、访问与点赞和累计积分四个较小的模块，其核心思想都是Update。

### 8.3.1 修改个人信息

此模块本质上是更新user表的一条记录，在更改个人信息相关的JSP页面使用Post方法传递要更新的字段和更新后的值，在UserServlet中对这些信息进行获取，结合UserBean和UserDao进行类和方法的调用即可。

实现该模块的核心代码如下：

|  |
| --- |
| public void userupdate(HttpServletRequest request,  HttpServletResponse response)throws ServletException,  IOException, SQLException {  String password1 = request.getParameter("password1");  String password2 = request.getParameter("password2");  int userid = Integer.parseInt(  request.getParameter("userid"));  System.out.println(password1);  System.out.println(password2);  if(password1.equals(password2)){  String forward = null;  UserDao userDao = new UserDao();  UserBean userBean = new UserBean();  userBean.setId(userid);  userBean.setNickname(request.getParameter("nickname")); userBean.setCampus(request.getParameter("campus"));  userBean.setInstitute(request.getParameter("institute"));  userBean.setMajor(request.getParameter("major"));  userBean.setIntro(request.getParameter("intro"));  userBean.setPassword(password1);  boolean mark = userDao.operationUser("update", userBean);  if(mark){  HttpSession session = request.getSession();  UserBean user = new UserBean();  user = userDao.setSession(userid);  session.setAttribute("user", user);  forward = "../homepage.jsp";  }else{  forward = "../home\_upda.jsp";  }  response.sendRedirect(forward);  userDao.closeConnection();  }else{  PrintWriter out = response.getWriter();  out.println("<script>window.history.go(-1)</script>");  }  } |

### 8.3.2 上传头像

此模块本质上是更新user表的Photo字段值，当登录用户在任意页面点击上传头像时，被访问用户的ID使用Post的方法传递到uploadUserIcon中，结合Session机制调用UserDao中的方法更新相应的字段值。其中，用户头像以相对路径的方式存储在数据库中，文件命名格式为用户ID.jpg。

实现该模块的核心代码如下：

|  |
| --- |
| public void userupdate(HttpServletRequest request,  HttpServletResponse response)throws ServletException,  IOException, SQLException {  request.setCharacterEncoding("utf-8");  response.setContentType("text/html");  response.setCharacterEncoding("utf-8");  response.setHeader("Cache-Control","no-store");  response.setHeader("Pragrma","no-cache");  response.setDateHeader("Expires",0);  HttpSession session = request.getSession();  String basePath = this.getServletConfig().  getServletContext().getRealPath("/").replace('\\', '/');  String filePath = "upload/";  response.setContentType("text/html");  java.io.PrintWriter out = response.getWriter( );  isMultipart = ServletFileUpload.isMultipartContent(request);  if( !isMultipart ){  out.println("<html>");  out.println("<head>");  out.println("<title>Servlet upload</title>");  out.println("</head>");  out.println("<body>");  out.println("<p>No file uploaded</p>");  out.println("</body>");  out.println("</html>");  return;  }  DiskFileItemFactory factory = new DiskFileItemFactory();  factory.setSizeThreshold(maxMemSize);  factory.setRepository(new File(basePath + "/uploadtemp/"));  ServletFileUpload upload = new ServletFileUpload(factory);  upload.setSizeMax( maxFileSize );  try{  List fileItems = upload.parseRequest(request);  Iterator i = fileItems.iterator();  int myid = ((UserBean)session.  getAttribute("user")).getId();  while ( i.hasNext () ){  FileItem fi = (FileItem)i.next();  if ( !fi.isFormField () ){  String fileName = myid + ".jpg";  file = new File( basePath + filePath + fileName) ;  fi.write( file ) ;  String sql = "update user set Photo=\""+filePath  + fileName+"\" where ID="+ myid;  connection.executeUpdate(sql);  connection.closed();  }  }  UserDao userDao = new UserDao();  UserBean user = new UserBean();  user = userDao.setSession(myid);  session.setAttribute("user", user);  response.sendRedirect("../homepage.jsp");  }catch(Exception ex) {  System.out.println(ex);  out.println("<script> window.history.go(-1)</script>");  }  } |

### 8.3.3 访问与点赞

此模块本质上是更新user表的VisitorNumber和Good字段值，当登录用户在任意页面查找并访问其他用户的主页时，被访问用户的ID使用Post的方法传递到UserServlet中，结合UserBean调用UserDao中的方法更新相应的字段值。当用户在他人主页点赞时，其Count值和被点赞用户的Good值也会通过相同的方法进行更新。

以点赞功能为例，其核心代码如下：

|  |
| --- |
| public void addGood(HttpServletRequest request,  HttpServletResponse response) throws ServletException,  IOException, SQLException {  HttpSession session = request.getSession();  int myid = ((UserBean)session.getAttribute("user")).getId();  UserDao userDao = new UserDao();  UserBean friend = new UserBean(); friend.setId(Integer.parseInt(  request.getParameter("friendid")));  int friendid = friend.getId();  boolean mark = userDao.addGood(myid, friendid);  String forward = null;  UserBean user = new UserBean();  UserDao userDao2 = new UserDao();  user = userDao2.setSession(myid);  session.setAttribute("user", user);  if (mark) {  forward = "../good.jsp";  } else {  forward = "../friendpage.jsp";  }  response.sendRedirect(forward);  userDao.closeConnection();  } |

### 8.3.4 累计积分

此模块本质上是更新user表的Score字段值，当用户添加学习日志或回复留言时，其ID值通过JSP页面用Post的方法传递到StudyServlet中，结合UserBean调用UserDao中的方法更新当前的积分值。其中，每次添加一篇日志获得10分，每次留言或回复留言获得3分。

实现该模块的核心代码如下：

|  |
| --- |
| public void addScore(int myid) throws SQLException {  String sql = "update user set Score=Score+10 where ID="+ myid;  connection.executeUpdate(sql);  connection.closed();  } |

## 8.4 查询模块

该模块参照网站系统的功能可细化为验证登录、信息加载、社交搜索和排行榜等四个较小的模块，其核心思想都是Select。

### 8.4.1 验证登录

此模块主要方法是通过查找与用户账号一致的记录，验证密码与用户输入的密码是否一致。同样，在JSP页面通过Post方法传递账号和密码信息，在UserServlet中结合UserBean调用UserDao中封装的类和方法进行信息的加工处理，并将结果返回给页面，从而实现页面之间的跳转。

实现该模块的核心代码如下：

|  |
| --- |
| public void validateLogin(HttpServletRequest request,  HttpServletResponse response) throws ServletException,  IOException, SQLException {  String forward = "";  String mail = request.getParameter("email");  String password = request.getParameter("password");  UserDao userDao = new UserDao();  UserBean user = new UserBean();  user = userDao.userLogin(mail, password); userDao.closeConnection();  if (user != null) {  HttpSession session = request.getSession();  session.setAttribute("user", user);  ArrayList<StudyBean> studylist =  new ArrayList<StudyBean>();  StudyDao studyDao = new StudyDao();  studylist = studyDao.search\_user\_study(user.getId()); studyDao.closeConnection();  session.setAttribute("studylist", studylist);  forward = "../homepage.jsp";  response.sendRedirect(forward);  } else {  PrintWriter out = response.getWriter();  out.println("<script> window.history.go(-1)</script>");  }  } |

### 8.4.2 信息加载

此模块本质上是通过用户ID进行各种表连接操作，以查找用户相关的全部记录，并将这些记录显示在各种JSP页面的相应位置。在登录模块中保存了用户账号信息后，结合Bean机制调用UserDao、StudyDao、ReviewDao和BadgeDao中的查找方法，并将结果进行Set，通过Session机制即可在相应的JSP页面中Get并打印。

由于信息加载模块涉及的页面较多，在此不多赘述。以好友主页的各项信息展示为例，其核心代码如下：

|  |
| --- |
| public void friendpage(HttpServletRequest request,  HttpServletResponse response) throws ServletException,  IOException, SQLException {  HttpSession session = request.getSession();  int friendid = Integer.parseInt(  request.getParameter("friendid"));  UserDao userDao = new UserDao();  UserDao userDao2 = new UserDao();  BadgeDao badgeDao = new BadgeDao();  StudyDao studyDao = new StudyDao();  userDao2.addVisit(friendid);  UserBean user = userDao.frienduser(friendid);  ArrayList<BadgeselfBean> badgelist = new  ArrayList<BadgeselfBean>();  badgelist = badgeDao.friendbadge(friendid);  ArrayList<StudyBean> studylist = new ArrayList<StudyBean>();  studylist = studyDao.search\_user\_study(friendid);  userDao.closeConnection();  session.setAttribute("friend", user);  badgeDao.closeConnection();  session.setAttribute("badgelist", badgelist);  studyDao.closeConnection();  session.setAttribute("studylist", studylist);  response.sendRedirect("../friendpage.jsp");  } |

### 8.4.3 社交搜索

该模块的本质是构造一条特定的Sql语句进行单条件或者多条件下的联合搜索，并将搜索结果通过Session机制存储在各种Bean机制或ArrayList中，最后在搜索结果相关的JSP页面中选择性地打印即可。同样，在搜索页面中使用Post方法传递用户选择或填写的搜索条件后，由相应的Servlet机制获取信息，调用对应的Bean和Dao中封装好的类和方法加工处理即可。

以搜索页面的用户搜索为例，其核心代码如下：

|  |
| --- |
| public void searchuser(HttpServletRequest request,  HttpServletResponse response) throws ServletException,  IOException, SQLException {  ArrayList<UserBean> userlist = new ArrayList<UserBean>();  HttpSession session = request.getSession();  UserDao userDao = new UserDao();  UserBean userBean = new UserBean();  userBean.setSex(request.getParameter("sex"));  userBean.setCampus(request.getParameter("campus"));  userBean.setNickname(request.getParameter("nickname"));  userBean.setAdmission(request.getParameter("admission"));  userBean.setInstitute(request.getParameter("institute"));  userBean.setMajor(request.getParameter("major"));  userlist = userDao.searchuser(userBean);  userDao.closeConnection();  session.setAttribute("userlist", userlist);  response.sendRedirect("../result\_user.jsp");  } |

### 8.4.4 排行榜

该模块的本质是构造一条按照某个字段值排序的Sql语句，通过调用封装好的UserDao方法结合UserBean机制，将排序后的结果返回到UserServlet中，通过Session机制存储在ArrayList中，最后在对应榜单的JSP页面中进行选择性打印即可。

以超级学霸榜为例，其核心代码如下：

|  |
| --- |
| public void scorebank(HttpServletRequest request,  HttpServletResponse response) throws ServletException,  IOException, SQLException {  ArrayList<UserBean> userlist = new ArrayList<UserBean>();  HttpSession session = request.getSession();  UserDao userDao = new UserDao();  UserBean userBean = new UserBean();  userlist = userDao.searchscore(userBean);  userDao.closeConnection();  session.setAttribute("userlist", userlist);  response.sendRedirect("../bank\_score.jsp");  } |

# 致谢

此时此刻，我的毕业设计已经进入最后的补充完善阶段，这也预示着我的大学生活接近尾声。一路走来，坚持到现在，要感谢的人太多。

首先，感谢母校的培养，现在的我能够自豪地告诉全世界，我的青春给了北京理工大学无怨无悔。感谢赵丰年老师，作为我大学四年的班主任和毕业设计指导老师，他给与过我很多指导性的意见和建议。感谢老师开设了这样一个吸引并且适合我的课题，让我能够顺利地完成大学生活的最后一个挑战，感谢老师在百忙之中抽出时间来解答我毕设过程中遇到的疑惑和问题，也感谢老师在未来的研究生学习期间能够继续指导我不断进步和成长。

其次，要感谢那些对游戏化进行过深入研究和探索的前辈，尽管种族和肤色不同，我依然深深地膜拜着前辈们，在他们的间接指引下，我才能拨开层层迷雾去寻找到一条正确的道路继续前行。也正是因为站在巨人的肩上，我才能跳脱自己浅显的眼界，在游戏化的道路上看得更高更远。

最后，感谢小伙伴们的支持和陪伴，这也是支撑我大学四年不断学习和进步的动力源。没有小伙伴们的帮助和鼓励，就没有现在这个豁达开朗、积极乐观的我，也不会有这么一个异彩纷呈的校园时光让我终生难忘。大学四年因为你们所以快乐，也希望今后不管走到哪里，我们都能像现在这样亲密无间，祝我们友谊常在，青春永驻。

# 参考文献

[1] McGonigal J. Reality is broken: Why games make us better and how they can change the world[M]. Penguin, 2011.

[2] Werbach K, Hunter D. For the win: How game thinking can revolutionize your business[M]. Wharton Digital Press, 2012.

[3] 汪通. 游戏化用户体验背景下的交互设计方法探讨[J]. 美与时代: 创意 (上), 2013 (12): 94-97.

[4] 约翰. 赫伊津哈[J]. 游戏的人, 1996, 9.

[5] Zichermann G, Linder J. The gamification revolution[J]. 2013.

[6] Prensky M. Digital game-based learning[J]. Computers in Entertainment (CIE), 2003, 1(1): 21-21.

[7] Zichermann G, Cunningham C. Gamification by design: Implementing game mechanics in web and mobile apps[M]. " O'Reilly Media, Inc.", 2011.

[8] Hate S. Enterprise Gamification Architecture Strategy[J]. Gamification: Rediscover the Power of Engagement, 55.

[9] Van Eck R. Digital game-based learning: It's not just the digital natives who are restless[J]. EDUCAUSE review, 2006, 41(2): 16.

[10] Alur D, Malks D, Crupi J, et al. Core J2EE Patterns (Core Design Series): Best Practices and Design Strategies[M]. Sun Microsystems, Inc., 2003.

[11] Johnson M. Designing enterprise applications with the J2EE platform[M]. Addison-Wesley Professional, 2002.

[12] Lindholm T, Yellin F. Java virtual machine specification[M]. Addison-Wesley Longman Publishing Co., Inc., 1999.

[13] The Java language specification[M]. Addison-Wesley Professional, 2000.

[14] Deterding S, Dixon D, Khaled R, et al. From game design elements to gamefulness: defining gamification[C]//Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments. ACM, 2011: 9-15.

[15] Kapp K M. The gamification of learning and instruction: game-based methods and strategies for training and education[M]. John Wiley & Sons, 2012.

[16] Muntean C I. Raising engagement in e-learning through gamification[C]//Proc. 6th International Conference on Virtual Learning ICVL. 2011: 323-329.

[17] Hamari J, Koivisto J, Sarsa H. Does gamification work?--a literature review of empirical studies on gamification[C]//System Sciences (HICSS), 2014 47th Hawaii International Conference on. IEEE, 2014: 3025-3034.

[18] Deterding S. Gamification: designing for motivation[J]. interactions, 2012, 19(4): 14-17.