



安徽工程大学机电学院
Anhui Polytechnic University, Mechanical & Electrical College

2018 届本科生毕业设计（论文）

题 目	Dota2 资料库智能管理平台的设计与实现
学生姓名	陈劲骁
学 号	314202050102
专业班级	计算机科学与技术 1401
系 部	计算机与软件工程系
指导老师	张云玲
完成日期	2018 年 05 月 19 日

教务处 制

Dota2 资料库智能管理平台的设计与实现

摘 要

当今社会，游戏产业蓬勃发展，如 PC 端的绝地求生、坦克世界、英雄联盟，再到移动端的王者荣耀、荒野行动的火爆。都离不开科学的游戏管理系统，游戏管理系统当今已经受到开发者和运营者的高度重视，通过游戏管理可以对游戏自身及玩法进行系统性、战略性的调整和协调，以提高可玩性。通过对权限的控制将游戏管理系统转变为面向玩家的资料库，玩家可以通过资料库对 Dota2 信息进行由浅入深的了解，从而进一步提高游戏可玩性。

本项目是基于 SpringMVC+Spring+Mybatis+SpringBoot 开发的一款 Dota2 资料库智能管理平台，数据库采用了 MySQL，前端采用了 ExtJs+Query 技术。

关键词：管理平台；Spring MVC；SpringBoot；ExtJs

Design and Implementation of Dota2 Database Intelligent Management Platform

Abstract

In today's society, the game industry is booming, such as player unknown's battlegrounds, world of tanks, the league of legends of the PC, the honor of the kings, and wild and action of the mobile, are inseparable from the scientific game management system, management system today has been attached great importance to developers and operators, the game itself and gameplay will be systematic and strategic adjusted to and coordinated through the game management, in order to improve playability. The game management can be transformed into a player oriented database management system through the permissions control, the player can obtain information from the game database on the game, thus he can get further enhance gameplay.

This project is a Dota2 database intelligent management platform based on SpringMVC+Spring+MyBatis+SpringBoot. MySQL is used in the database, and ExtJs and jQuery are used in the front end.

Keywords: Management platform; Spring MVC; SpringBoot; ExtJs

目 录

引言	- 1 -
第 1 章 绪论	- 2 -
1.1 课题研究的背景及意义	- 2 -
1.2 课题的研究现状和发展趋势	- 2 -
1.2.1 国内外发展现状	- 2 -
1.2.2 发展趋势	- 3 -
第 2 章 开发工具及相关技术介绍	- 4 -
2.1 开发技术	- 4 -
2.1.1 SpringBoot	- 4 -
2.1.2 SpringMVC 框架	- 4 -
2.1.3 ExtJs	- 4 -
2.1.4 MyBatis	- 4 -
2.2 开发工具	- 5 -
2.2.1 IntelliJ IDEA	- 5 -
2.2.2 Visual Studio Code	- 5 -
第 3 章 需求分析	- 6 -
3.1 可行性分析	- 6 -
3.1.1 经济可行性	- 6 -
3.1.2 技术可行性	- 6 -
3.1.3 操作可行性	- 6 -
3.2 系统分析	- 6 -
3.2.1 业务流分析	- 6 -
3.2.2 数据流分析	- 8 -
3.2.3 数据字典	- 9 -
3.2.4 数据库需求分析	- 10 -
第 4 章 概要设计	- 12 -
4.1 系统模块结构设计	- 12 -
4.1.1 系统模块	- 12 -
4.2 系统总体结构设计	- 14 -
4.2.1 总体结构设计功能	- 14 -
4.2.2 系统 E-R 图	- 15 -
4.3 数据库表设计	- 16 -
第 5 章 系统详细设计	- 18 -
5.1 系统登录模块	- 18 -
5.2 用户操作模块	- 21 -
5.3 游戏运营人操作模块	- 26 -
第 6 章 系统测试	- 28 -
6.1 系统测试的概述	- 28 -
6.2 功能性测试	- 28 -
6.3 非功能性测试	- 29 -
结论与展望	- 31 -
致谢	- 32 -

参考文献	- 33 -
附录	- 35 -

插图清单

图 3-1 Dota2 资料库管理平台玩家流程图	- 7 -
图 3-2 Dota2 资料库管理平台运营人流程图	- 8 -
图 3-3 Dota2 资料库管理平台数据流图	- 9 -
图 4-1 系统功能模块结构图	- 12 -
图 4-2 Dota2 资料库管理平台玩家模块流程图	- 13 -
图 4-3 Dota2 资料库管理平台游戏运营人模块流程图	- 14 -
图 4-4 Dota2 资料库管理平台总体结构设计功能图	- 15 -
图 4-5 Dota2 资料库管理平台的 E-R 图	- 15 -
图 5-1 Dota2 资料库管理平台平台登陆首页图	- 18 -
图 5-2 Doat2 资料库登录界面图	- 19 -
图 5-3 Dota2 资料库登录界面图	- 20 -
图 5-4 查看英雄信息	- 21 -
图 5-5 查看物品信息	- 22 -
图 5-6 查看天赋信息	- 23 -
图 5-7 物品模拟界面	- 24 -
图 5-8 天赋模拟界面	- 25 -
图 5-9 战绩信息模块	- 26 -
图 5-10 用户修改表单	- 26 -
图 5-11 角色分配	- 27 -
图 5-12 权限分配	- 27 -

表格清单

表 3-1 用户表 - 9 -

表 3-2 角色表 - 9 -

表 3-3 权限表 - 9 -

表 3-4 英雄表 - 10 -

表 3-5 物品表 - 10 -

表 3-6 天赋表 - 10 -

表 4-1 用户表 - 16 -

表 4-2 角色表 - 16 -

表 4-3 权限表 - 16 -

表 4-4 英雄表 - 16 -

表 4-5 物品表 - 17 -

表 4-6 天赋表 - 17 -

表 6-1 重置用户密码测试用例 - 28 -

表 6-2 角色分配测试用例 - 28 -

表 6-3 新增权限功能测试用例 - 29 -

表 6-4 物品模拟功能测试用例 - 29 -

表 6-5 性能测试用例 - 29 -

引言

随着社会的高速发展，互联网行业不断发展，尤其是游戏产业的快速发展，越来越多的人开始接触游戏，游戏产业进入了高并发、大数据和数据共享时代。当今时代，游戏运营是游戏厂商所要应对的问题，并且玩家对游戏信息获取的需求日渐提高。因此一个科学的游戏管理平台可以改善开发者和运营者对数据的管理，Dota2 资料库管理平台操作简单，玩家根据自己的需求一些获取关于 Dota2 的信息。

本系统是基于 SpringBoot 的 SSM 开发，该框架是目前风靡的一种 Web 应用程序开源框架，其中 Spring 是一个轻量级的 Java 开发框架。前端使用 jQuery、ExtJs 等框架，数据访问层使用 MyBatis，采用 MVC 设计模式。

第1章 绪论

1.1 课题研究的背景及意义

近年来,随着游戏产业的大爆发,产生了大量优秀的网络游戏作品,游戏已经融入很多人的生活中,游戏品质是决定一款游戏是否受欢迎的直接因素,但渠道商的发行与宣传、后期的运营、版本的更新、强大的游戏管理系统等因素交织在一起才能决定它是否能延续成功,只有真正有想法的团队、有品质的游戏才能经得起时间的考验,在竞争激烈的游戏市场上存活下来。

考虑到游戏策划和运营人员不太了解编程开发,信息的可视性和灵活性较差,难以对游戏进行有效的数据管理、便捷地对开发人员提出需求等,因此游戏信息化、系统化管理就显得迫在眉睫。游戏厂商需要专业化的管理系统,提高游戏策划和运营人员管理效率及游戏开发人员开发效率;玩家需要获取游戏数据,例如历史战绩、英雄使用情况和装备属性等,通过更加专业的角度来了解这款游戏,如通过网页、手机浏览器。以往玩家在游戏过程中了解游戏数据会浪费玩家的游戏时间,带来的效率问题已经日益突出。

游戏管理系统首先应能提供与游戏相关的数据,无论是游戏内容数据还是记录玩家行为的数据,都应该完整地呈现给工作人员,必要时还需提供精确查找,并能够生成图表。对于玩家,管理系统应提供部分游戏内容数据。

1.2 课题的研究现状和发展趋势

1.2.1 国内外发展现状

(1) 框架技术国内外的研究现状

REST 全称为表述性状态转移(Representational State Transfer),一种分布式软件架构风格,可有效地降低系统开发的难度,增加系统可扩展性,王正华认为,具有面向资源性、无状态性、连通性强、方法统一等特征的 REST 式架构优于面向活动的 RPC 和面向服务的 SOA^[3]。盛伟翔等人^[4]从架构风格的角度,研究了如何统一操作接口,降低系统耦合度,以轻量级的架构方法提高系统性能,增强系统的可见性和容错性,优化交互方式,减少无意义的服务器任务^[16]。

秦姣华等人^[1]提出了基于 ExtJS3.2 的调优方案,根据自己的需要延迟加载 JS 库,江雪等人^[1]认为,可以通过 ExtJS 的 ViewModel 的继承特性来完成不同需求的数据绑定,李亚丽等人^[2]提出了 ExtJS 可以用来开发 RIA,即富客户端的 Ajax 应用,是一个用 JavaScript 编写的,主要用于创建前端用户界面,是一个与后台技术无关的前端 Ajax 框架^[17],Sencha 中文站在 2017 年 5 月 24 日发布 ExtJS6.5 版本,推出了 ExtJS6.5 Modern toolkit,可以为移动设备,也可以为桌面应用带来丰富的体验^[18],推出了更多的新组件以及优化大量数据管理、无缝滚动等。

(2) 游戏管理系统国内外的研究现状

蒋伟^[7]使用基于 JSF-EJB3.0 开发模式设计了网络游戏管理系统,但是该系统对远程调用的安全机制等没有相应的讨论。王贞^[10]设计实现的网络游戏管理系统中,代理服务器使用了基于集群的负载均衡,失败恢复技术,前端使用了基于静态页面缓存的 Web 加速器,前端代理服务器将大部分的静态资源缓存起来,直接返回给访问者,减少应用服务器的负载,该系统将数据按照不同策略方案划分到不同的服务器上,分散数据库的压力。郭杰^[12]对游戏资源管理系统的各方面做了充分的讨论和阐述,详细的分析 OGRE 中资源管理框架的设计模式,并且分析了 OGRE 资源管理框架的优点和缺点。施海华等人^[8]设计了基于 NGB 的互动游戏管理系统,有以下三个模块:互动电视游戏管理审核模块、互动电视游

戏管理发布模块及互动电视下载应用服务模块，基于 NGB 的互动游戏管理系统的建设解决了各游戏厂商提供的游戏无法直接被网络运营商运营发布的问题，实现了对来自不同厂商的游戏的审核、发布等统一管理功能。董欢^[9]在游戏管理系统中加强了后台数据安全的管理，他在系统中采用了“所见即所得”设计方式，使用户在不具备操作数据库能力的条件下也可以实现对数据库数据的备份与恢复操作，但是该系统是在 B/S 架构上进行设计开发的，管理人员必须通过 PC 端网页进行数据的获取，不适用于手机、平板电脑等移动设备。费国俊^[6]提出，随着 Symbian 系统的没落，iOS 和安卓、winPhone 系统的兴起，针对新兴的系统开发的手机游戏的管理、推广运营将是新的业务需求。张士铎等人^[11]采用基于 LAMP 的 Web 开发平台，敏感数据的传输采用报文加密，他们认为，通过对基本的游戏数据统计进行数据挖掘，分析用户的行为模型，为游戏的发展战略提供数据指导，具有更深的意义。张灿^[5]将游戏管理系统分为五大模块，其中数据日志模块包括输入 SQL 语句的查询，以及其他相关日志的查询。游戏服务器每时每刻产生的记录大量用户行为的日志，游戏管理系统会定时将这些日志从游戏服务器迁移到本地，并将这些日志读取后存储至数据库，工作人员通过游戏管理系统查询具体日志信息，该管理系统还包括服务器配置模块，工作人员可通过该模块设置服务器配置，因为游戏有不同的物理服务器，针对不同的玩家会有不同的查询结果和操作结果，因此需要配置不同的服务器信息。

1.2.2 发展趋势

2017 年上半年国内游戏市场收入达 997.8 亿元（同比+26.7%，环比+14.9%）；移动端市场收入达 561.4 亿元（同比+49.8%，环比+26.3%）。其中，上半年 PC 端游戏市场实际销售收入达到 319.5 亿元^[20]。

2018 年，游戏行业发明了新的方式，如 loot boxes（战利品宝箱）和现实增强（AR）类游戏来吸引和娱乐玩家。游戏行业的未来将发生改变，既有娱乐性，也充满了创新和争议。

预计 2019 年，电子游戏市场整合将急剧加速，其中 80% 的收入来自大型上市公司。不过，2018 年也有一些小型游戏制作公司发布了畅销的游戏，如《PUBG（绝地求生）》和《Cuphead（茶杯头）》，而且这种趋势将持续下去。独立开发者没有顶级公司的资源，因为较小的预算通常意味着更大的误差幅度，但他们确实有创新的自主权^[26]。

第 2 章 开发工具及相关技术介绍

2.1 开发技术

本系统采用的是 Java 语言开发, 系统主要采用了 Spring MVC 框架, 基于 SpringBoot, 数据库采用了 MySQL, 前端使用了 ExtJs、jQuery 等技术^[4]。

2.1.1 SpringBoot

多年以来, SpringIO 平台饱受争议的一点就是大量的 XML 配置以及复杂的依赖管理。SpringBoot 的目标之一就是实现免 XML 的开发体验。Boot 所实现的功能超出了这个任务的描述, 开发人员不仅不再需要编写 XML, 而且在一些场景中甚至不需要再编写繁琐的 import 语句^[13]。

SpringBoot 的目的不在于为已解决的问题提供新的处理方式, 而是为平台带来另一种开发体验, 从而简化这些已有技术的应用。对于已经熟悉 Spring 生态系统的开发人员来说, Boot 是一种很完美的选择, 不过对于使用 Spring 的新人来说, Boot 提供一种更简约的方式来运用这些技术。

对 Java 项目而言, 利用 maven 对第三方 jar 包进行统一管理, 丰富的第三方库减小了开发量, 提高了工作质量。

2.1.2 SpringMVC 框架

MVC 是一个以设计界面应用程序为基础的设计模式。它主要利用分离模型、视图及控制器在应用程序中所扮演的角色将业务逻辑从界面中分离。通常, 模型负责封装应用程序, 数据在视图层展示。视图仅仅展示这些数据, 不包含任何业务逻辑。控制器担任接收用户的各种请求的责任, 并调用后台服务 (service 和 dao) 来处理业务逻辑。经过处理, 后台业务逻辑层返回数据在视图层展现, 控制器搜集这些数据及模型在视图层展示。MVC 模式的核心思想是将业务逻辑从界面中分离出来, 允许它们单独改变而不会相互影响^[14]。

SpringMVC 属于 SpringFramework 的后续产品, 已经融合在 Spring Web Flow 里面。Spring 框架提供了构建 web 应用程序的全功能 MVC 模块。使用 Spring 可插入 MVC 架构, 从而在使用 Spring 进行 Web 开发时, 可以选择使用 Spring 的 SpringMVC 框架或集成其他 MVC 开发框架, 如 Struts, Struts2 等^[21]。

在 SpringMVC 应用程序中, 模型通常由 POJO 对象组成, 它在业务层中被处理, 在持久层中被持久化^[22]。

2.1.3 ExtJs

ExtJs 源于 YUI, 最先由 Jack Slocun 开发, 采用 OOP 设计思想建立类库与后台语言无关的 JavaScript 框架^[15], 它将 OOP 思想发挥得淋漓尽致, 然而 ExtJs 不仅仅是个 JavaScript 框架, 它可以独立于其他 JavaScript 框架通过 Adapter (适配器)^[16]。对于后台程序员来说, 不用深知前端网页开发技术, 就可以快速开发出类似桌面应用的用户体验很高的界面。一方面减少了与美工的交流, 自己来开发客户端界面, 另一方面, 页面交互能力强, 能够很好地实现后台功能。

Ajax 作为 Web2.0 的核心技术之一, 带来了 Web 用户体验革命性的转变。ExtJs 作为 Ajax 的客户端 UI 框架, 很大程度上减少了系统开发成本, 同时又保证了系统的美化以及页面响应速度。利用 Ajax+ExtJS 组合, 有效地避开了传统 B/S 结构用户体验差的特点, 又具有了 C/S 结构的强交互能力, 同时还具有很好的可移植性和可维护性^[1]。

2.1.4 MyBatis

MyBatis 是支持简单 SQL 查询、存储过程和高级映射的优良持久层框架。

MyBatis 避免了一切 JDBC 代码和参数的手工设置以及结果集的检索。MyBatis 使用简单的 XML 或注解用于配置和原始映射，将接口和 Java 的 POJOs（Plain Ordinary Java Objects）映射成数据库中的记录。

每个 Mybatis 应用程序主要都是使用 `SqlSessionFactoryBean` 实例的，一个 `SqlSessionFactoryBean` 实例，对该实例设置 `dataSource` 和映射文件 `xxx(mapper).xml`，它将构成一个基本的 MyBatis 应用程序。

MyBatis 的功能架构分为三层^[14]：

（1）API 接口层：提供给外部使用的接口 API，开发人员通过这些本地 API 来操纵数据库，接口层接收到调用请求就会调用数据处理层来完成具体的数据处理。

（2）数据处理层：负责具体的 SQL 查找、SQL 解析、SQL 执行和执行结果的映射处理等。它主要的目的是根据调用的请求完成一次数据库操作。

（3）基础支撑层：负责最基础功能的支撑，包括连接管理、事务管理、配置加载和缓存处理等，将他们抽取出来作为最基础的组件。为上层的数据处理层提供最基础的支撑。

2.2 开发工具

2.2.1 IntelliJ IDEA

本系统使用 IntelliJ IDEA 开发工具，数据库使用可视化编辑工具 Navicat for MySQL，组合完成了一套完整的资料库管理系统，IDEA 是 Java 语言开发的集成环境，IntelliJ 在业界被公认为最好的 Java 开发工具之一，尤其在智能代码助手、代码自动提示、重构、J2EE 支持、各类版本工具(git、svn、github 等)、JUnit、CVS 整合、代码分析、创新的 GUI 设计等方面的功能可以说是超常的。IDEA 是 JetBrains 公司的产品，这家公司总部位于捷克共和国的首都布拉格，开发人员以严谨著称的东欧程序员为主。

2.2.2 Visual Studio Code

本系统前端使用 Visual Studio Code 开发工具，用来编写 Html, CSS, jQuery。Visual Studio Code 是一个运行于 Mac OS X、Windows 和 Linux 之上的，针对于编写现代 Web 和云应用的跨平台源代码编辑器。

该编辑器也集成了所有一款现代编辑器所应该具备的特性，包括语法高亮（syntax high lighting），可定制的热键绑定（customizable keyboard bindings），括号匹配（bracket matching）以及代码片段收集（snippets）。Somasegar 也告诉笔者这款编辑器也拥有对 Git 的开箱即用的支持。

第3章 需求分析

3.1 可行性分析

可行性分析主要就是研究这个项目是否可以施行,并且具有必要性、迫切性、科学性等,不但能产生可观经济效益,而且能产生巨大社会效益、生态效益等等。用往年的数据和资料证明研究该项目是有意义的^[12]。

可行性分析是通过对项目的主要内容,如配套条件、设施选型、资源供给、建设规模、工艺路线、市场需求、环境影响、资金筹措、盈利能力等,从技术、经济、工程等方面进行考查研究和分析比较,并对项目建成以后可能获得的财务、经济效益及社会环境影响进行预测,从而提出该项目是否值得投资和如何进行建设的征询意见,为项目决策提供根据的一种综合性的系统分析措施。可行性分析应具备公正性、预见性、科学性和可靠性的特点。

可行性研究是要求以全面、系统的分析为主要方法,经济效益为核心,围绕影响项目的各种因素,运用大量的数据资料论证拟建项目是否可行。对整个可行性研究提出综合分析评价,指出优缺点和建议。为了结论的需要,往往还需要加上一些附件,如试验数据、论证材料、计算图表、附图等,以增强可行性报告的说服力^[23]。

3.1.1 经济可行性

Dota2 资料库管理平台,提供高效便捷的服务,该平台提供给已注册的用户使用,平台的运行环境是 SpringBoot+Windows+MySQL,该环境成本低,所需要的成本主要集中在开发成本和推广成本。Dota2 资料库管理平台一旦投入使用,游戏运营人员可以对游戏数据进行有效管理,也可以让玩家对游戏信息进行深入了解。该平台可以通过广告来营收,后期的经济效益将大于系统开发和推广成本。

3.1.2 技术可行性

本系统基于 SSM 框架开发,开发过程中分为两大模块,一个模块是后端模块,一个模块是前端模块。后端框架 Spring+Spring MVC+Mybatis,数据库使用 MySQL。前端使用 ExtJs 框架,物品天赋模块前端使用 jQuery,从所用的框架来看,都是已经成熟的框架,从开发语言来看,Java 和 Html 语言也是非常熟悉的,所以开发本系统在技术上是可行的。

3.1.3 操作可行性

Dota2 资料库管理平台是方便游戏运营人员对游戏数据进行有效管理和使玩家了解游戏动向。平台提供一套完整的控制体系,通过角色权限控制,游戏运营人员可以多游戏数据进行增删改操作,而玩家只能浏览信息,如查询物品信息、战绩信息,并进行物品和天赋的模拟。经过简单的注册就可以成为该平台的用户。玩家登陆平台就可以浏览游戏信息,游戏开发与运营人员拥有更高权限的账号。平台的设计首先考虑到用户使用起来简单,用户体验问题。所以 Dota2 资料库管理平台是可行的。

3.2 系统分析

3.2.1 业务流分析

Dota2 资料库管理平台所有用户都可以浏览英雄、物品、天赋信息,并且允许进行物品合成模拟和天赋加点模拟,玩家不可以对游戏信息进行增删改操作,运营人员可以对用户、角色、权限、英雄、物品、天赋进行增删改操作。Dota2 资料库管理平台玩家流程图如图 3-1 所示,Dota2 资料库管理平台运营人流程图如图 3-2 所示。

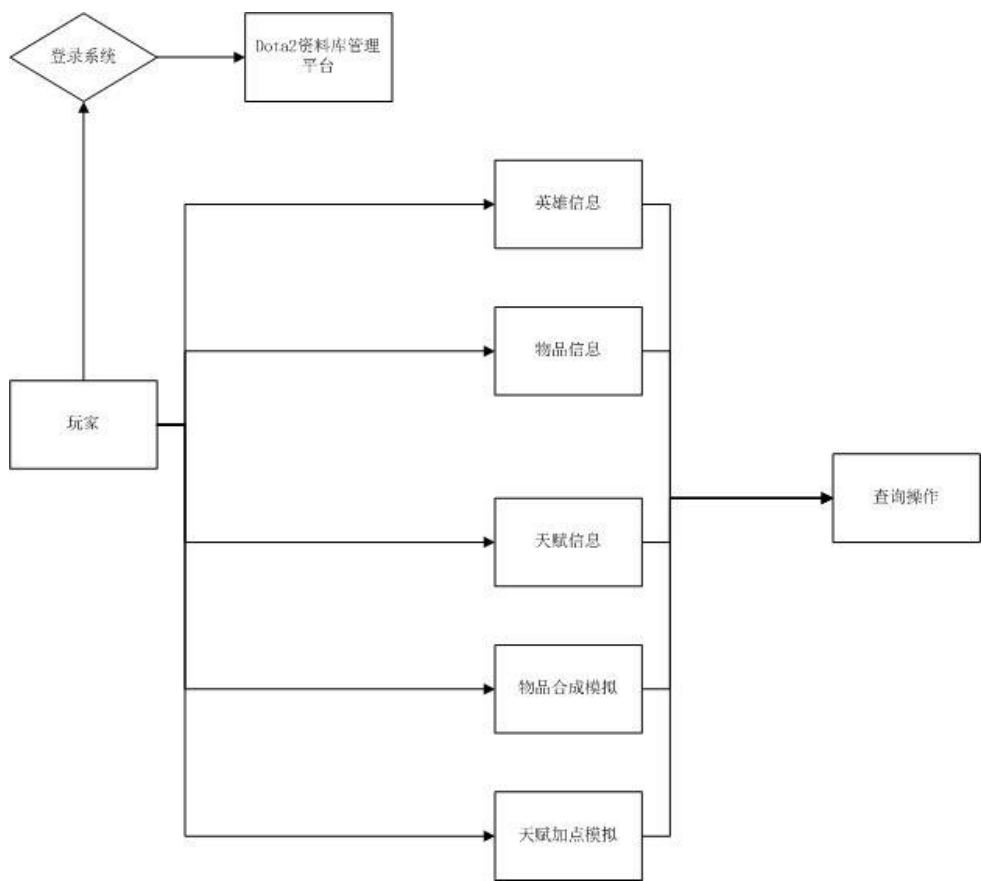


图 3- 1 Dota2 资料库管理平台玩家流程图

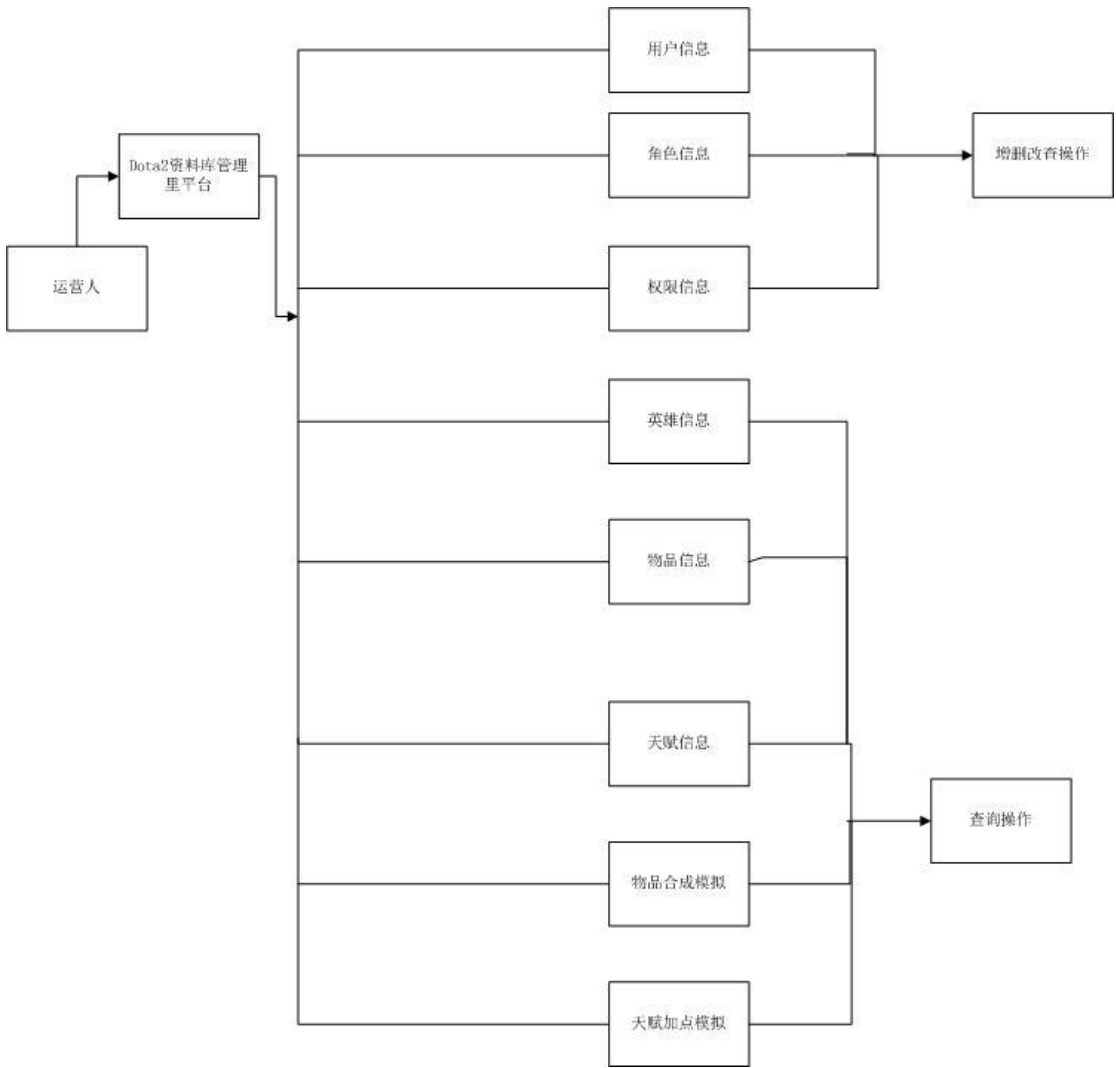


图 3- 2 Dota2 资料库管理平台运营人流程图

3.2.2 数据流分析

Dota2 资料库管理平台的数据流是依赖平台进行流向的，录入操作是输入数据，查询操作是输出数据。输入和录入数据保持同步，Dota2 资料库管理平台数据流图如图 3-3 所示。

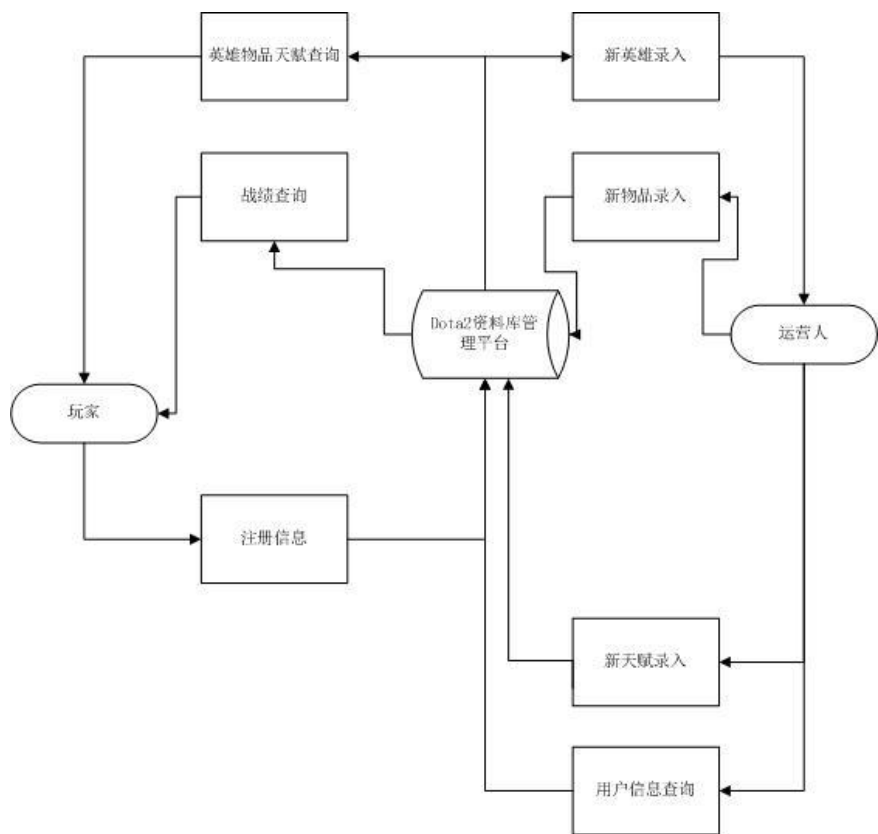


图 3-3 Dota2 资料库管理平台数据流图

3.2.3 数据字典

数据字典是指对数据的数据存储、处理逻辑、数据流、数据项、数据结构、外部实体等进行定义和描述，其目的是对数据流程图中的各个元素做出详细的说明，使用数据字典为简单的建模项目。数据字典是一种用户可以访问应用程序元数据和记录数据库的目录。主动数据字典是指在对数据库或应用程序结构进行修改时，其内容可以由 DBMS 自动更新的数据字典。被动数据字典是指修改时必须手动更新里面的内容的数据字典^[24]。

Dota2 资料库管理平台的用户数据模型表的数据字典，是由用户详细信息组成的，而这些信息是保存在数据库中，用户可以通过该信息登录平台系统。如表 3-1 所示。

表 3-1 用户数据模型表

数据项存储名	Dota2 资料库管理平台的用户信息
数据项来源	Dota2 资料库管理平台的用户表
数据项去向	Dota2 资料库管理平台的用户登陆模块
数据项组成	管理平台的用户信息=管理平台的用户账号+管理平台的用户密码+管理平台的用户操作权限
数据项描述	保存了管理平台用户的基本信息

角色数据模型表是保存用户角色信息的数据字典，游戏运营人用户进行角色分配、修改和读取的同时，这些信息会记录到数据库中。如表 3-2 所示。

表 3-2 角色数据模型表

数据项存储名	Dota2 资料库管理平台的角色信息
数据项来源	角色信息表

数据项去向	用户模块、角色模块、权限模块
数据项组成	角色信息=角色名称+角色状态等
数据项描述	保存了用户的角色信息

权限数据模型表是保存角色权限信息的数据字典，游戏运营人用户进行权限读取、分配和修改的同时，这些信息会记录到数据库中。如表 3-3 所示。

表 3-3 权限数据模型表

数据项存储名	Dota2 资料库管理平台的权限信息
数据项来源	权限信息表
数据项去向	用户模块、角色模块、权限模块
数据项组成	权限信息=权限名称+权限状态等
数据项描述	保存了角色的权限信息

英雄数据模型表是保存英雄信息的数据字典，游戏运营人用户和玩家用户进行英雄查询和修改的同时，这些信息会记录到数据库中。如表 3-4 所示。

表 3-4 英雄数据模型表

数据项存储名	Dota2 资料库管理平台的英雄信息
数据项来源	英雄信息表
数据项去向	英雄管理模块、天赋模拟模块
数据项组成	英雄信息=英雄名称+英雄图片路径等
数据项描述	保存了英雄的基本信息

物品数据模型表是保存物品信息的数据字典，游戏运营人用户和玩家用户进行物品查询、修改和物品模拟合成的同时，这些信息会记录到数据库中。如表 3-5 所示。

表 3-5 物品数据模拟表

数据项存储名	Dota2 资料库管理平台的物品信息
数据项来源	物品信息表、物品合成表
数据项去向	物品管理模块、物品模拟模块
数据项组成	物品信息=物品名称+物品图片路径等
数据项描述	保存了物品的基本信息

天赋数据模型表是保存天赋信息的数据字典，游戏运营人用户和玩家用户进行天赋查询、修改和天赋模拟的同时，这些信息会记录到数据库中。如表 3-6 所示。

表 3-6 天赋数据模拟表

数据项存储名	Dota2 资料库管理平台的天赋信息
数据项来源	天赋信息表
数据项去向	天赋管理模块、天赋模拟模块
数据项组成	天赋信息=天赋名称+天赋图片路径等
数据项描述	保存了天赋的基本信息

3.2.4 数据库需求分析

Dota2 资料库管理平台使用 MySQL 数据库，MySQL 是一种关系数据库管理系统，关系数据库将数据保存在不同的表中，MySQL 所使用的 SQL 语言是用于访问数据库的最常用标准化语言^[25]，系统需要利用数据库的增加，修改，删除，查询语句来实现信息的及时更新。

因此要求数据库的底层架构能承受各种信息的输入和输出，以及输入和输出

数据的安全性能、时间性能和完整性。在进行数据库需求分析的时候，我们需要对系统需求进行详细的分析，这样才可以准确地把数据库设计出来，避免在设计数据库中存在一些逻辑上的问题。通过对系统的功能需求分析以及系统的数据库需求分析，Dota2 资料库管理平台根据用户对象不同各个模块有如下具体功能需求信息：

（1）管理员功能需求分析

- 1) 用户、角色和权限信息的增删改查功能；
- 2) 英雄、物品和天赋信息的增删改查功能；
- 3) 项目需求和玩家意见的增删改查功能。

（2）玩家功能需求分析

- 1) 英雄、物品和天赋信息的查询功能：玩家可以对游戏信息进行检索；
- 2) 天赋模拟功能：玩家可以自行对天赋加点进行模拟；
- 3) 物品模拟功能：玩家可以自行对物品进行模拟合成；
- 4) 战绩信息查询功能：玩家可以自行玩家战绩进行检索。

第 4 章 概要设计

经过 Dota2 资料库管理平台的详细需求分析之后，Dota2 资料库管理平台开发进入到概要设计阶段，也称系统设计阶段，本阶段主要对 Dota2 资料库管理平台进行基本的流程设计、架构设计、模块设计、功能设计、接口设计以及数据设计等，为资料库管理平台软件的详细设计提供基础。

4.1 系统模块结构设计

本设计主要实现 Windows 平台下的 Dota2 资料库管理平台。此系统根据角色主要包括游戏运营人（管理员）模块和玩家模块。Dota2 资料库管理平台的系统功能模块如图 4-1 所示。

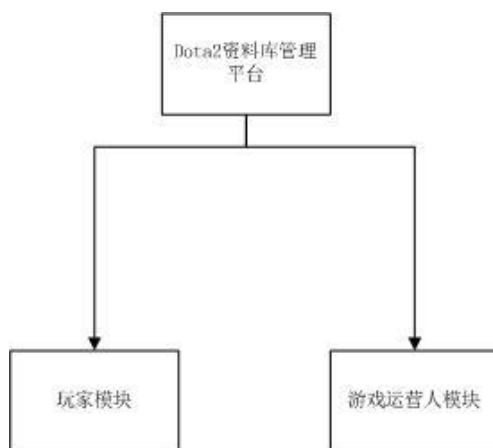


图 4-1 系统功能模块结构图

玩家模块包括英雄、物品、天赋和战绩信息的检索，以及物品模拟和天赋模拟。

游戏运营人模块包括英雄、物品、天赋和战绩信息的增加、删除、修改、查询，以及用户、角色和权限信息的增加、删除、修改、查询。

4.1.1 系统模块

（1）玩家模块流程图

首先判断玩家有没有登录管理平台，若没有，则不能访问平台系统的信息。如果登录成功，可以访问英雄、物品、天赋和战绩信息，以及模拟物品合成和天赋加点，玩家模块流程图如图 4-2 所示。

（2）游戏运营人模块流程图

首先判断游戏运营人有没有登录管理平台，若没有，则不能访问平台系统的信息。如果登录成功，可以对英雄、物品、天赋和战绩信息进行增加、删除、修改、查询，以及对用户、角色和权限信息增加、删除、修改、查询，游戏运营人模块流程图如图 4-3 所示。

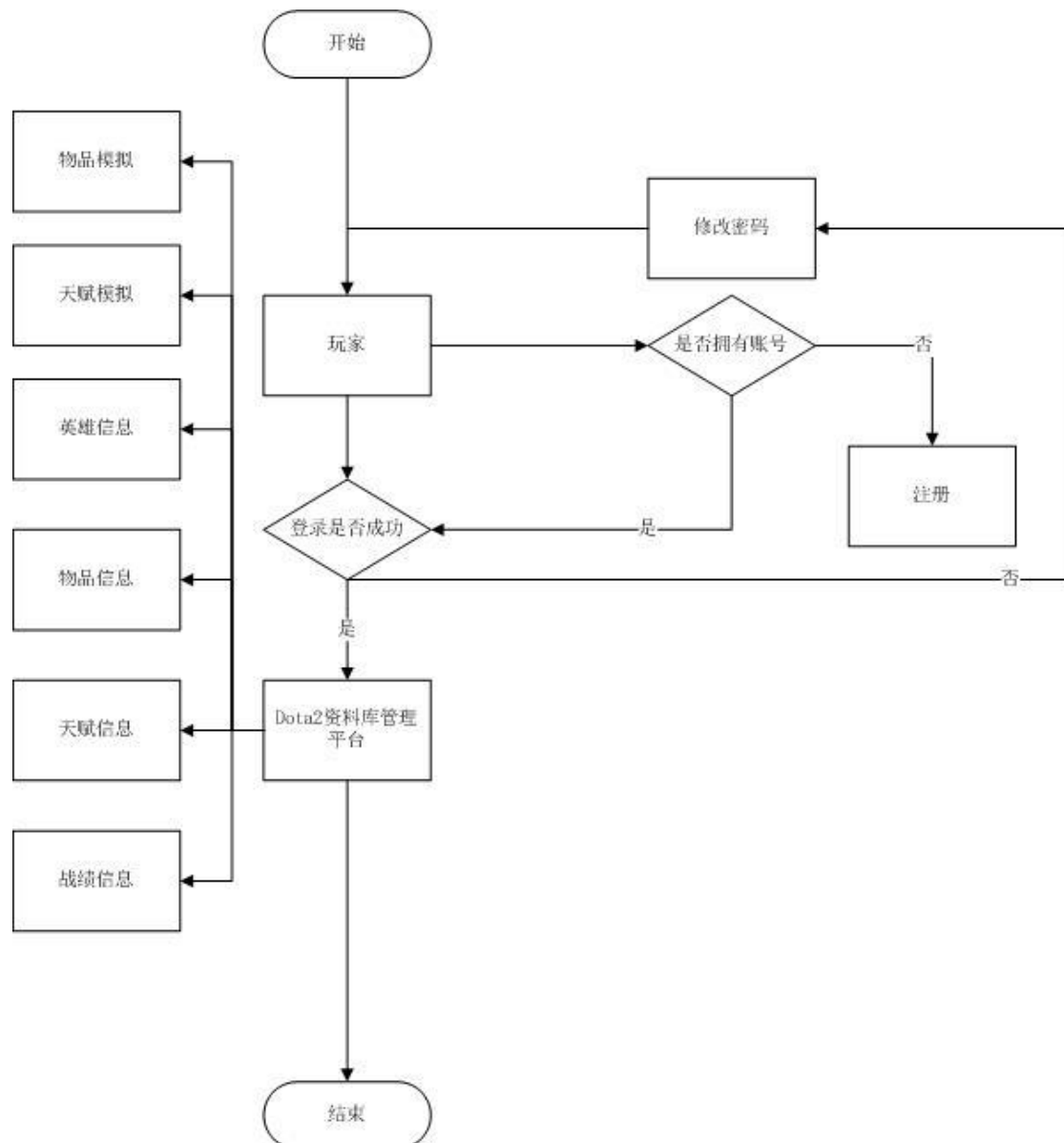


图 4-2 Dota2 资料库管理平台玩家模块流程图

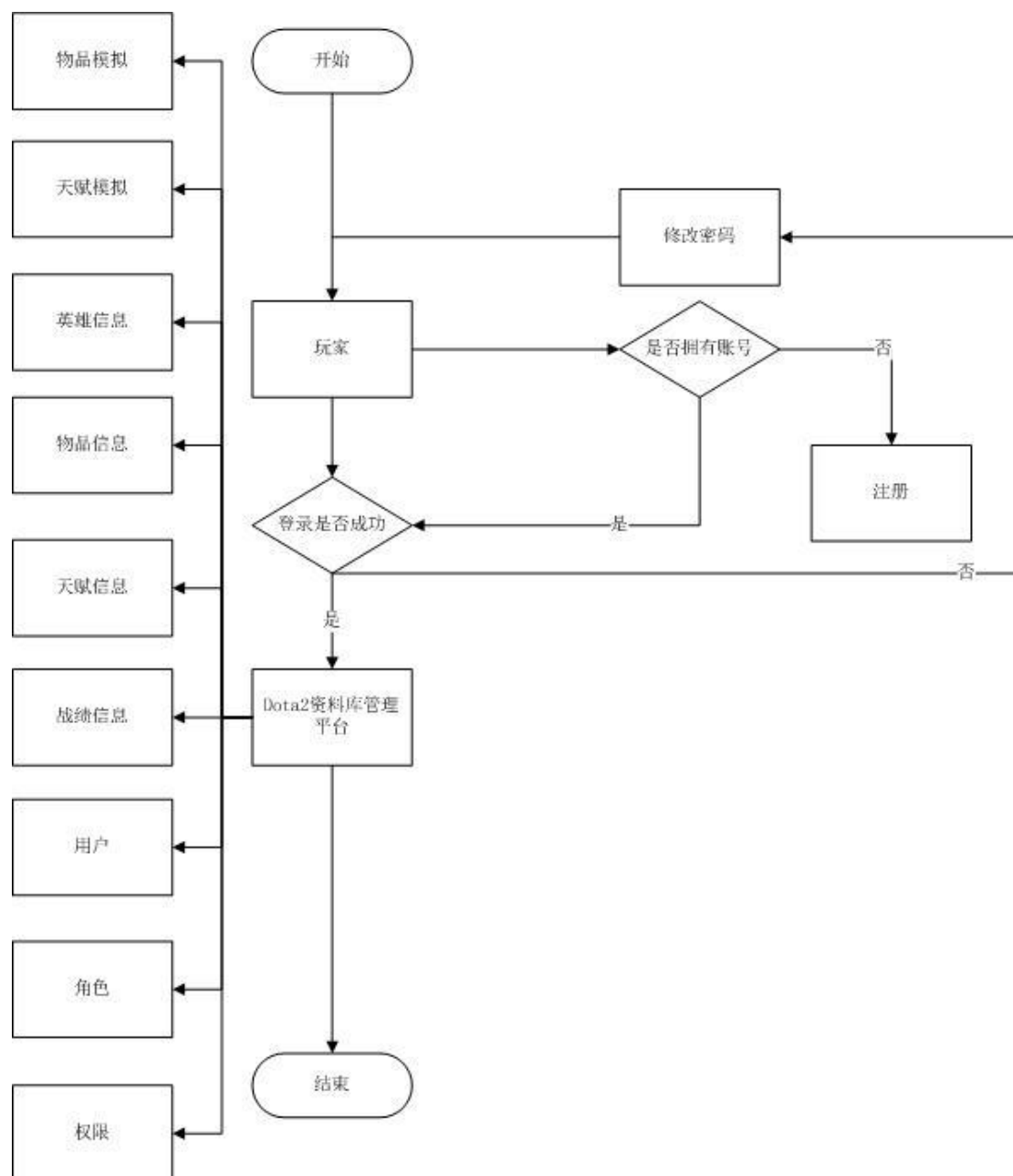


图 4-3 Dota2 资料库管理平台游戏运营人模块流程图

4.2 系统总体结构设计

4.2.1 总体结构设计功能

Dota2 资料库管理平台设计包含了玩家模块和运营人模块，可以对游戏信息进行检索处理，极大提高了游戏的版本迭代效率和游戏可玩性，Dota2 资料库管理平台总体结构设计功能图如图 4-4 所示。

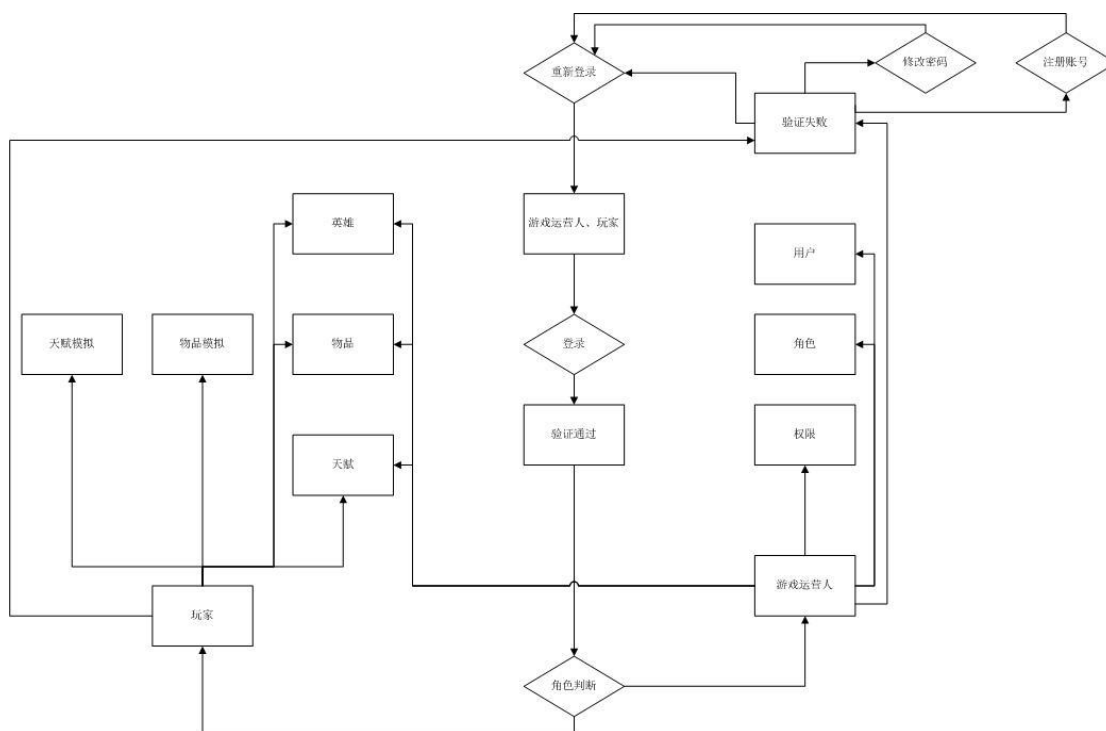


图 4-4 Dota2 资料库管理平台总体结构设计功能图

4.2.2 系统 E-R 图

E-R 图也称实体-联系图，提供了表示属性、实体类型和联系的方法，用来描述真实世界的概念模型。Dota2 资料库管理平台的 E-R 图如图 4-5 所示。

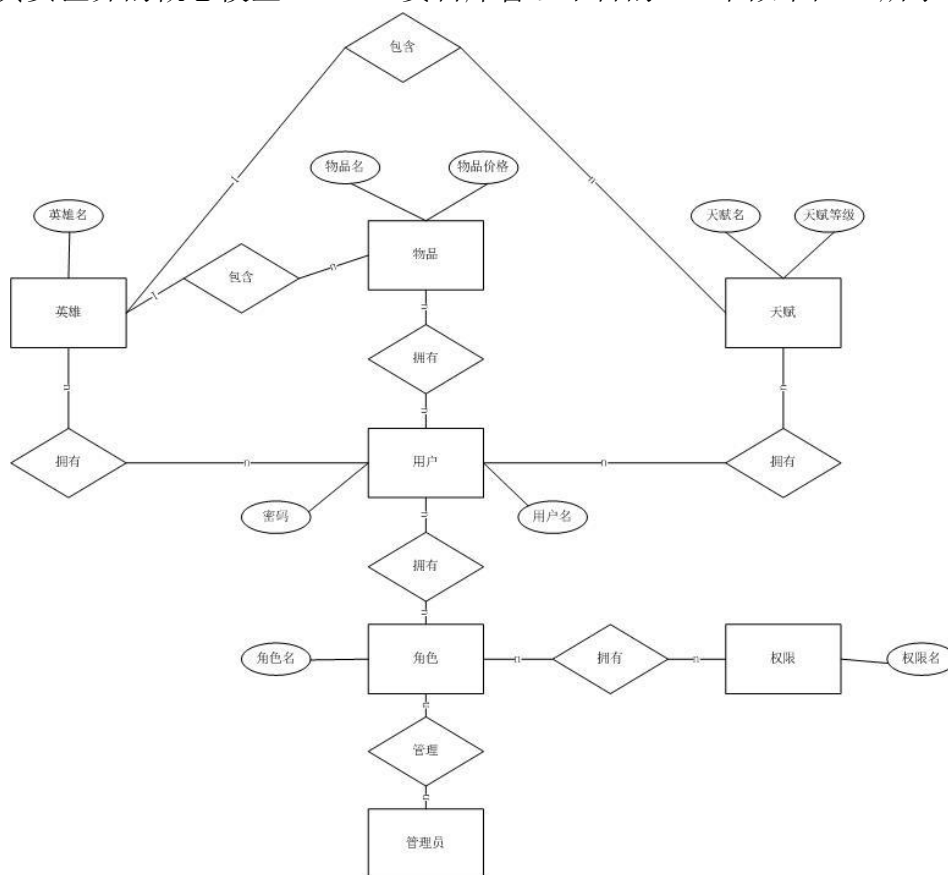


图 4-5 Dota2 资料库管理平台的 E-R 图

4.3 数据库表设计

用户表 CT_USER 如表 4-1 所示，CT_USER 表是存储了用户信息的数据字典，用户进行注册时，用户信息会加入到此表中。

表 4-1 用户表

字段名	数据类型	允许非空	默认值	备注
ID	INT	NO		用户 id
USERNAME	VARCHAR	NO		用户名
REALNAME	VARCHAR	YES	NULL	真实姓名
PASSWORD	VARCHAR	NO		密码
SEX	TINYINT	YES		性别
PHONE	VARCHAR	YES		手机号
STATUS	TINYINT	NO		状态，0 禁用 1 正常
UPDATE_TIME	TIMESTAMP	NO	CURRENT_TIMESTAMP	修改时间
CREATE_TIME	DATETIME	NO	'0000-00-00 00:00:00'	创建时间

角色表 CT_ROLE 如表 4-2 所示，CT_ROLE 表是用来存储角色信息的数据字典，与用户表构成多对多关系。

表 4-2 角色表

字段名	数据类型	允许非空	默认值	备注
ID	INT	NO		角色 id
NAME	VARCHAR	NO		角色名
DESCRIPTION	VARCHAR	YES	NULL	描述
STATUS	TINYINT	NO		状态
UPDATE_TIME	TIMESTAMP	NO	CURRENT_TIMESTAMP	修改时间
CREATE_TIME	DATETIME	NO	'0000-00-00 00:00:00'	创建时间

权限表 CT_RESOURCE 如表 4-3 所示，CT_RESOURCE 表存储了权限信息，与角色表构成了多对多关系。

表 4-3 权限表

字段名	数据类型	允许非空	默认值	备注
ID	INT	NO		角色 id
NAME	VARCHAR	NO		权限名
TYPE	VARCHAR	NO	NULL	描述
ICON	VARCHAR	YES		权限图标
PARENT_ID	INT	NO		父类 ID
PERMISSION	VARCHAR	NO		权限地址
URL	VARCHAR	NO		
STATUS	TINYINT	NO		状态
UPDATE_TIME	TIMESTAMP	NO	CURRENT_TIMESTAMP	修改时间
CREATE_TIME	DATETIME	NO	'0000-00-00 00:00:00'	创建时间

英雄表 HERO 如表 4-4 所示，HERO 表是用来存储英雄信息的字典。

表 4-4 英雄表

字段名	数据类型	允许非空	默认值	备注
ID	INT	NO		英雄 ID

NAME	VARCHAR	NO		英雄名
LOCALIZED_NAME	VARCHAR	YES	NULL	英文名
HEADPORTRAIT_PATH	VARCHAR	NO		头像路径
HERO_PATH	VARCHAR	NO	CURRENT_TIMESTAMP	英雄图片路径

物品表 ITEM 如表 4-5 所示，ITEM 表是用来存储物品信息的字典。

表 4-5 物品表

字段名	数据类型	允许非空	默认值	备注
ID	INT	NO		物品 ID
NAME	VARCHAR	NO		物品名
COST	INT	NO	NULL	花费
SECRET_SHOP	TINYINT	NO		秘密商店
SIDE_SHOP	TINYINT	NO		路边商店
RECEIPE	TINYINT	NO		是否卷轴，0 否，1 是
UPGRATED_ITEM	VARCHAR	NO	0	是否更新物品
ITEM_PATH	VARCHAR	YES		物品图片路径
LOCALIZED_NAME	VARCHAR	YES	CURRENT_TIMESTAMP	英文名
CHINESE_NAME	VARCHAR	YES	'0000-00-00 00:00:00'	中文名

天赋表 TALENT 表如表 4-6 所示，TALENT 表用来存储英雄天赋信息。

表 4-6 天赋表

字段名	数据类型	允许非空	默认值	备注
ID	INT	NO		天赋 ID
TEXT	VARCHAR	NO		天赋名
HERO_ID	INT	NO		英雄 ID
GRADE	TINYINT	NO		天赋等级
TYPE	TINYINT	NO		天赋类型，类型 1，类型 2
IS_DELETE	TINYINT	NO	0	是否已删除，0 否，1 是

第 5 章 系统详细设计

5.1 系统登录模块

系统登录模块实现了登录注册功能，注册需要用户名和密码，真实姓名、性别等其他信息选填，一个用户名只能注册一次。用户登录需要用户名和密码，连续输入用户名密码错误五次十分钟内不允许登录系统。登录界面如图 5-1 所示。

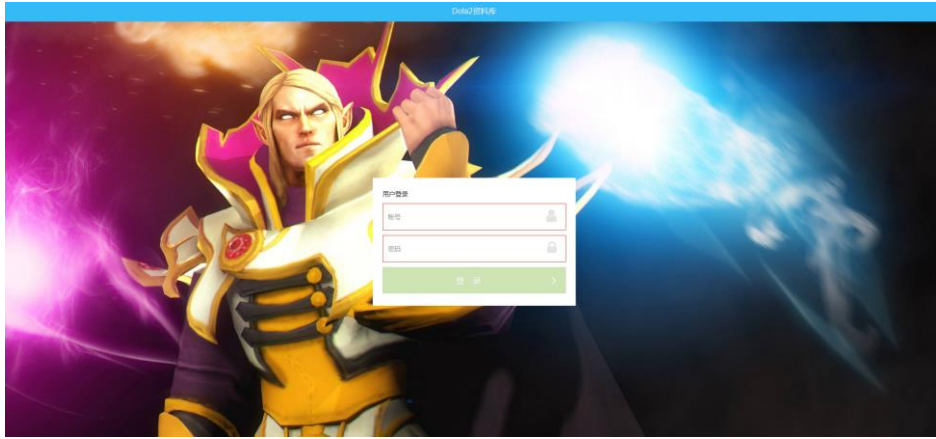


图 5-1 Dota2 资料库管理平台平台登陆首页图

核心代码：

```
@POST
@Path("/login")
@Produces(MediaType.APPLICATION_JSON)
// 此方法不处理登录成功,由 shiro 进行处理.
public PcsResult login(@Context HttpServletRequest request,@FormParam("username")
String username,
                        @FormParam("password") String
password,@FormParam("roomcode")String roomcode) throws Exception {
    PcsResult result = new PcsResult();
    SecurityUtils.setSecurityManager(securityManager);
    Subject subject = SecurityUtils.getSubject();
    OfficeUsernamePasswordToken token=new
OfficeUsernamePasswordToken(username,password,roomcode);
    try {
        subject.login(token);
    } catch (AuthenticationException exception) {
        // 登录失败从 request 中获取 shiro 处理的异常信息。
        // shiroLoginFailure: 就是 shiro 异常类的全类名.
        EnumPcsServiceError msg = EnumPcsServiceError.BUSINESS_DATA_NONE;
        if (exception != null) {
            if (exception instanceof UnknownAccountException) { // 账号不存在
                msg = EnumPcsServiceError.BUSINESS_USER_NONE;
                logger.warn("login failed, account not existed. account={}", username);
            } else if (exception instanceof IncorrectCredentialsException) { // 密码不正确
```

```

        msg =
EnumPcsServiceError.BUSINESS_USER_PASSWORD_ERROR;
        logger.warn("login failed, password not correct. account={}",
username);
    } else if (exception instanceof ExcessiveAttemptsException){//登录失败多次
        msg = EnumPcsServiceError.BUSINESS_USER_LOGIN_ERROR;
        logger.warn("login failed, login failed times too much. account={}",
username);
    } else if ("kaptchaValidateFailed".equals(exception)) { // 验证码错误
        msg =
EnumPcsServiceError.BUSINESS_USER_VERIFYCODE_ERROR;
    } else if (exception instanceof IncorrectOfficeException) {
        msg = EnumPcsServiceError.BUSINESS_ROOM_EXISTED;
        logger.warn("login failed, login failed not in office. account={}",
username);
    }
}
result.setCode(msg.getCode()).setMessage(msg.getDesc());
}
return result;
}

```

Dota2 资料库管理平台界面如图 5-2、5-3 所示。

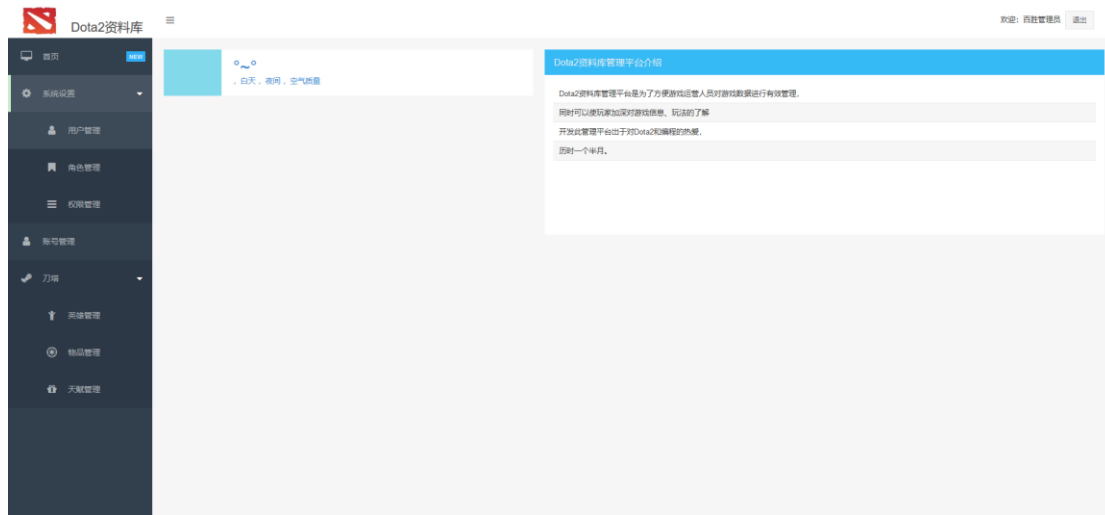


图 5-2 Doat2 资料库登录界面图

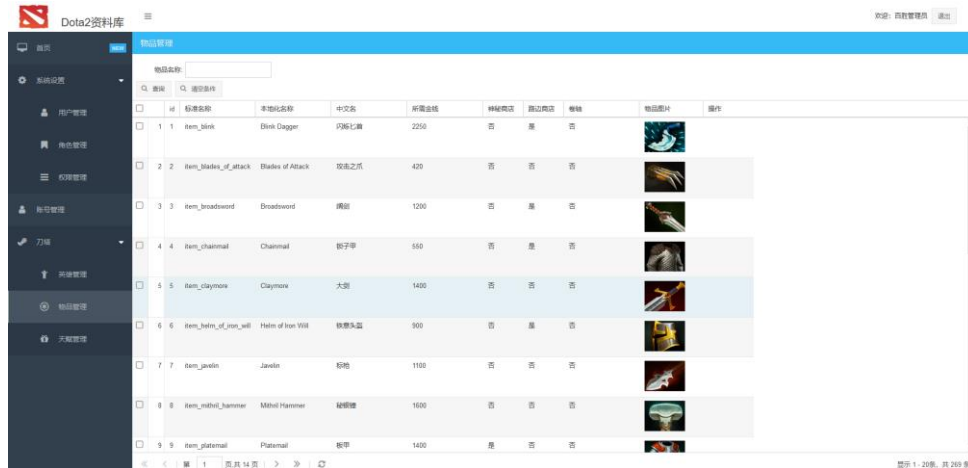


图 5-3 Dota2 资料库登录界面图

核心代码：

```
Ext.define('Admin.view.main.Main', {
    extend: 'Ext.Container',
    xtype: 'main',
    requires: [
        'Ext.button.Button',
        'Ext.container.Container',
        'Ext.list.Tree',
        'Ext.toolbar.Fill'
    ],
    controller: 'main',
    viewModel: 'main',
    cls: 'sencha-dash-viewport',
    itemId: 'mainView',
    layout: {
        type: 'vbox',
        align: 'stretch'
    },
    initComponents: function () {
        var me = this, boxData = {}, record = {}, modifyPassword = 1;
        Common.util.Util.doAjax({
            url: Common.Config.requestPath('System', 'Users', 'info'),
            method: 'get',
            async: false
        }, function (data) {
            record = data.data;
            boxData.realname = record.realname;
            boxData.userId=record.userId;
            modifyPassword = record.modifyPassword;
            Common.permission.Permission.server_permcollections = record.buttonList;
            Common.Config.user.userid = record.userId;
        });
    }
});
```

```

Common.Config.IMAGE_ADDRESS=record.dicMap.IMG_ADDRESS;
});
//construct navigationTree store data
})

```

5.2 用户操作模块

玩家登录平台后，玩家可以查询英雄、物品、天赋信息，还可以对物品合成模拟和对天赋加点模拟。查看英雄信息如图 5-4 所示。

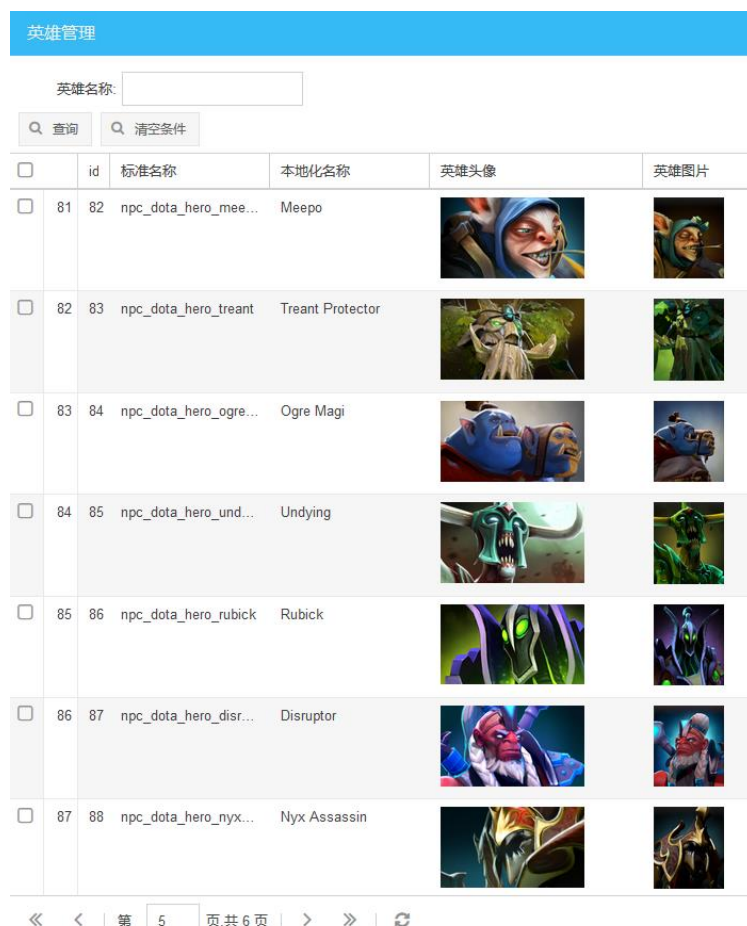


图 5-4 查看英雄信息

核心代码：

```

@GET
@Path("queryHero")
@Produces(MediaType.APPLICATION_JSON)
//@RequiresPermissions(value="hero:queryHero")
public PcsResult queryHero(@Query HeroBean heroBean){
    Map map = heroService.queryHero(heroBean);
    if(map.size()==0){
        return
newResult(false).setCode(EnumPcsServiceError.ERROR_OPERATE.getCode()).setMessage(Enu
mPcsServiceError.ERROR_OPERATE.getDesc());
    }else {

```

```

        return newResult(true).setData(map);
    }
}

```

查询物品信息如图 5-5 所示。










物品管理									
物品名称: <input type="text"/>									
Q 查询 Q 清空条件									
<input type="checkbox"/>	id	标准名称	本地化名称	中文名	所需金钱	神秘商店	路边商店	卷轴	物品图片
<input type="checkbox"/>	1	item_blink	Blink Dagger	闪烁匕首	2250	否	是	否	
<input type="checkbox"/>	2	item_blades_of_attack	Blades of Attack	攻击之爪	420	否	否	否	
<input type="checkbox"/>	3	item_broadsword	Broadsword	阔剑	1200	否	是	否	
<input type="checkbox"/>	4	item_chainmail	Chainmail	锁子甲	550	否	是	否	
<input type="checkbox"/>	5	item_claymore	Claymore	大剑	1400	否	否	否	
<input type="checkbox"/>	6	item_helm_of_iron_will	Helm of Iron Will	铁意头盔	900	否	是	否	
<input type="checkbox"/>	7	item_javelin	Javelin	标枪	1100	否	否	否	
<input type="checkbox"/>	8	item_mithril_hammer	Mithril Hammer	秘银锤	1600	否	否	否	
<input type="checkbox"/>	9	item_platemail	Platemail	板甲	1400	是	否	否	
<< < 第 1 页共 14 页 > >>									

图 5-5 查看物品信息

核心代码：

```

@GET
@Path("queryItem")
@Produces(MediaType.APPLICATION_JSON)
//@RequiresPermissions(value="item:queryItem")
public PcsResult queryHero(@Query ItemBean itemBean){
    Map map = itemService.queryItem(itemBean);
    if(map.size()==0){
        return
newResult(false).setCode(EnumPcsServiceError.ERROR_OPERATE.getCode()).setMessage(EnumPcsServiceError.ERROR_OPERATE.getDesc());
    }else
    {return newResult(true).setData(map); }
}

```

查看天赋信息如图 5-6 所示。

天赋管理

天赋名称:

🔍 查询

🔍 清空条件

<input type="checkbox"/>	id	天赋名称	英雄id	所需等级	类型
<input type="checkbox"/>	1	1 +25 Movement Speed	102	10	1
<input type="checkbox"/>	2	2 +20% XP Gain	102	10	2
<input type="checkbox"/>	3	3 +6 Armor	102	15	1
<input type="checkbox"/>	4	4 +50 Mist Coil Heal/Damage	102	15	2
<input type="checkbox"/>	5	5 +15% Cooldown Reduction	102	20	1
<input type="checkbox"/>	6	6 +90 Damage	102	20	2
<input type="checkbox"/>	7	7 +250 Aphotic Shield Health	102	25	1
<input type="checkbox"/>	8	8 +25% Curse of Avernus Slow	102	25	2
<input type="checkbox"/>	9	9 +25 Attack Speed	73	10	1
<input type="checkbox"/>	10	10 -5s Unstable Concoction	73	10	2
<input type="checkbox"/>	11	11 +50 Damage	73	15	1
<input type="checkbox"/>	12	12 +350 Health	73	15	2
<input type="checkbox"/>	13	13 +25% Cleave	73	20	1
<input type="checkbox"/>	14	14 +360 Unstable Concoction Damage	73	20	2
<input type="checkbox"/>	15	15 +30 Chemical Rage Regeneration	73	25	1
<input type="checkbox"/>	16	16 -0.2 Chemical Rage Base Attack Time	73	25	2
<input type="checkbox"/>	17	17 +60 Gold/Min	68	10	1
<input type="checkbox"/>	18	18 +10% Spell Amplification	68	10	2
<input type="checkbox"/>	19	19 +20 Health Regen	68	15	1
<input type="checkbox"/>	20	20 -1.5s Ice Vortex Cooldown	68	15	2

<<

<

第

1

页,共 46 页

>

>>

🔄

图 5-6 查看天赋信息

核心代码：

```

@GET
@Path("/queryTalent")
@Produces(MediaType.APPLICATION_JSON)
//@RequiresPermissions(value="hero:queryHero")
public PcsResult queryTalent(@Query TalentBean talentBean){
    Map map = talentService.queryTalent(talentBean);
    if(map.size()==0){
        return
newResult(false).setCode(EnumPcsServiceError.ERROR_OPERATE.getCode()).setMessage(EnumPcsServiceError.ERROR_OPERATE.getDesc());
    }else
    { return newResult(true).setData(map); } }

```

物品模拟如图 5-7 所示，点击右侧的物品即可在右下方看到模拟结果。

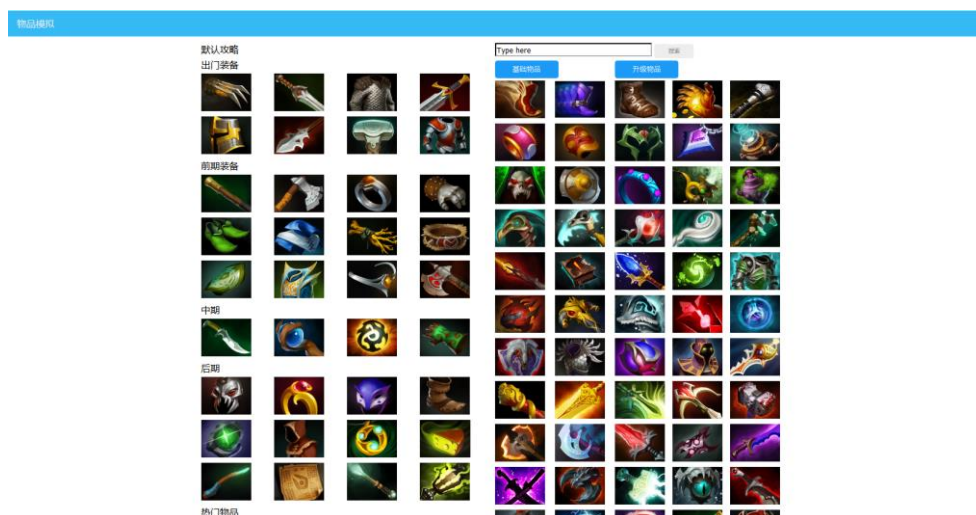


图 5-7 物品模拟界面

核心代码：

```
@GET
@Path("/getTalentByHeroId")
@Produces(MediaType.APPLICATION_JSON)
//@RequiresPermissions(value="hero:getTalentByHeroId")
public PcsResult getTalentByHeroId(@QueryParam("heroId") Integer heroId){
    Map map = talentService.getTalentByHeroId(heroId);
    if(map.size()==0){
        return
newResult(false).setCode(EnumPcsServiceError.ERROR_OPERATE.getCode()).setMessage(EnumPcsServiceError.ERROR_OPERATE.getDesc());
    }else {
        return newResult(true).setData(map);
    }
}
```

天赋模拟如图 5-8 所示，点击左侧的英雄图像，在右侧会显示出天赋模拟的信息。

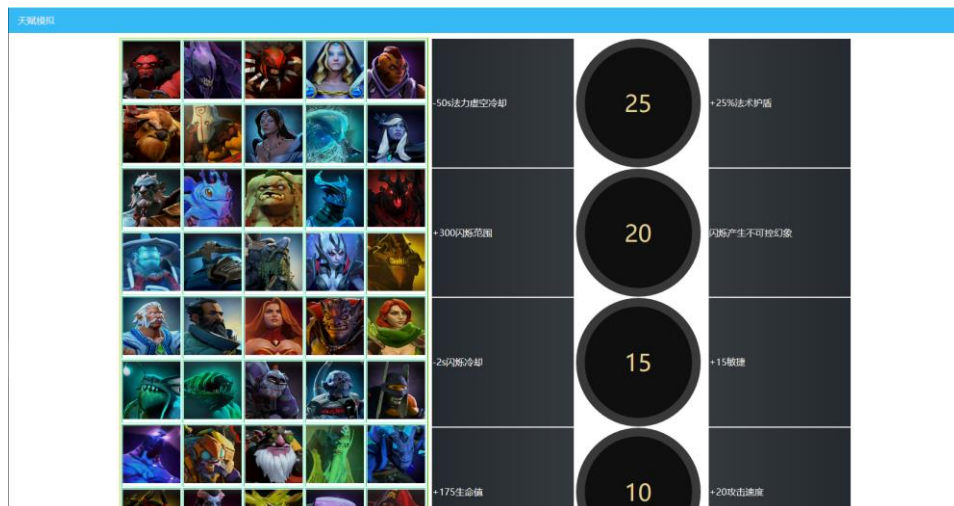


图 5-8 天赋模拟界面

核心代码：

```

@GET
@Path("getMatchDetail")
@Produces(MediaType.APPLICATION_JSON)
@RequiresPermissions(value="match:getMatchDetails")
public Map<String,Object> getMatchDetails(@QueryParam("match_id") Long match_id)
throws IOException {
    Map<String,Object> modelMap = new HashMap<String,Object>();
    String jsonArray = "";
    if (match_id == null || match_id.equals("")) {
        jsonArray =
URLUtil.getUrlForMatchDetail("https://api.steampowered.com/IDOTA2Match_570/GetMatchDetails/v1/?key=" + steamKey + "&match_id=" + "3839053706");
    }else{jsonArray =
URLUtil.getUrlForMatchDetail("https://api.steampowered.com/IDOTA2Match_570/GetMatchDetails/v1/?key=" + steamKey + "&match_id=" + match_id);
    }
    List<MatchDetail> matchDetails =
GsonUtil.getObjectList(jsonArray,MatchDetail.class);
    for (MatchDetail matchDetail:matchDetails){
        String startTime = String.valueOf(matchDetail.getStart_time());
        String normalStartTime = DateUtil.Timestamp2Date(startTime);
        matchDetail.setString_start_time(normalStartTime);
    }
    Map<String,Object> data = new HashMap();
    modelMap.put("data",data);
    data.put("list",matchDetails);
    data.put("total",matchDetails.size());
    modelMap.put("success",true);
    return modelMap;
}

```

}

战绩信息如图 5-9 所示，输入 SteamId，就可以查询该 SteamId 的比赛信息，双击比赛信息，会弹出一个记载着比赛详细信息的窗口。

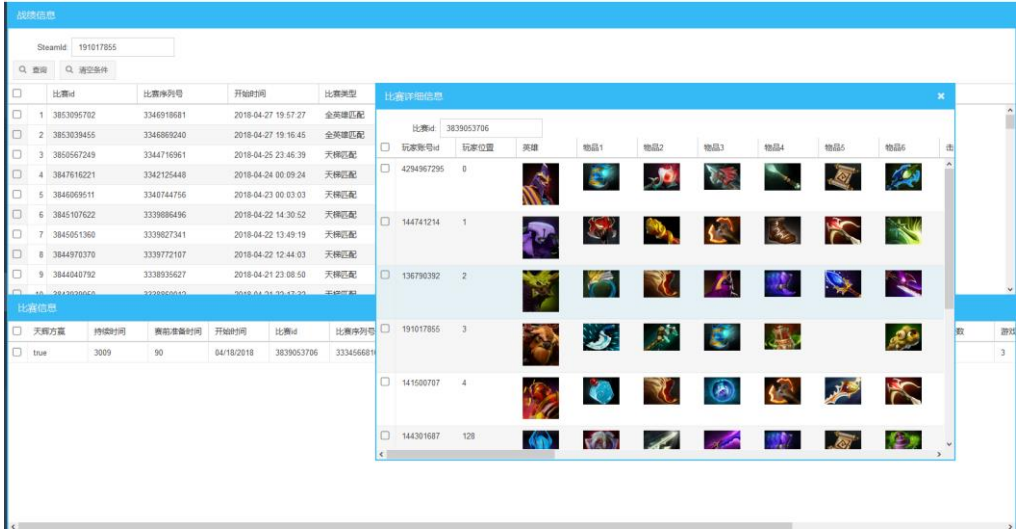


图 5-9 战绩信息模块

5.3 游戏运营人操作模块

游戏运营人登录平台之后，可以对用户、角色、权限、英雄、天赋、物品进行增加、删除、修改、查询操作。用户的修改如图 5-10 所示，进入用户管理，点击该行右侧的修改图标，会弹出来用户修改表单，输入信息即可修改。

用户修改

用户名:ceshi6

昵称:是兄弟就来砍我

QQ:634208959

电话号码:18895358020

用户类型:一级管理

头像:

选择图片...

查看

描述:是

确定

取消

图 5-10 用户修改表单

角色的分配如图 5-11 所示，进入用户管理，点击该行右侧的编辑图标，弹出来权限分配表单，下拉选择要分配的权限即可分配。

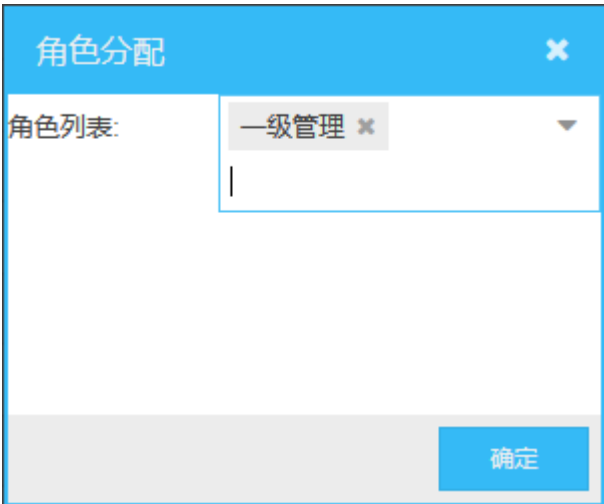


图 5-11 角色分配

权限的分配如图 5-12 所示，进入角色管理，单击选择该行右侧的权限分配图标，在弹出表单的右侧树状的权限打钩即可完成权限的分配。

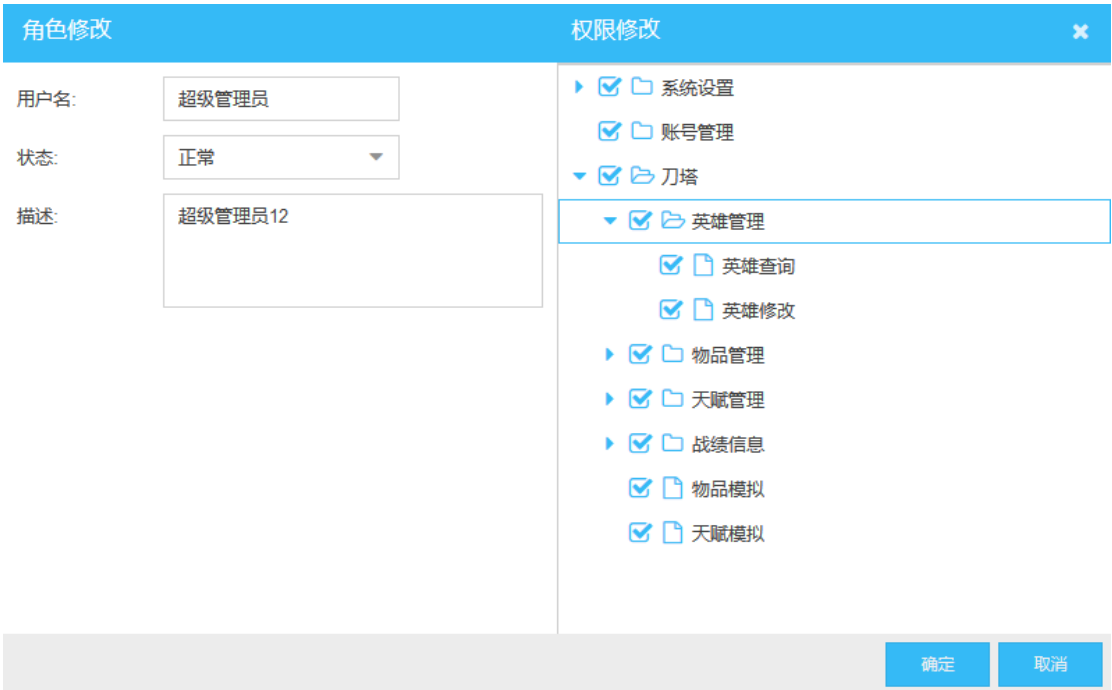


图 5-12 权限分配

第 6 章 系统测试

6.1 系统测试的概述

系统测试是软件开发流程中至关重要的环节，也是项目上线之前的必经之路。只有通过系统全面的测试，才能保证管理平台在功能和性能方面均已达到上线的要求。本章首先介绍系统测试的功能性测试部分，其次介绍非功能性测试，最后介绍系统针对性能做出的改进和优化。

6.2 功能性测试

功能性测试又称为黑盒测试，它是指对系统各项功能进行测试与验证，从而使其能满足用户的需求^[5]，本节主要介绍 Dota2 资料库管理系统各模块部分功能的测试用例。

用户管理模块用户密码重置功能的测试用例如表 6-1 所示，重置密码是指将用户密码重置为初始密码（123456），该用例旨在验证重置用户密码功能是否能成功地重置用户的密码。

表 6-1 重置用户密码测试用例

测试用例编号	TC003
测试用例标题	重置用户密码功能测试
测试目的	验证能否成功重置用户密码
优先级	1
测试流程	(1) 点击进入系统设置用户管理模块 (2) 在操作中点击重置图标 (3) 在弹出表单中确认重置 (4) 使用该用户名和重置后的密码，能否登录系统
预期结果	用户可以登录系统
实际结果	(1) 成功登录系统 (2) 密码已重置
结论	重置用户密码功能测试符合预期结果，测试通过

角色管理模块角色分配功能的测试用例如表 6-2 所示，角色分配是指给用户分配指定角色，并拥有相应的权限。该用例旨在验证角色分配功能能否成功地给用户分配指定的角色。

表 6-2 角色分配测试用例

测试用例编号	TC005
测试用例标题	角色分配功能测试
测试目的	验证能否成功进行角色分配
优先级	1
测试流程	(1) 点击进入系统设置用户管理模块 (2) 在操作中点击用户的角色分配图标 (3) 在弹出表单中下拉框选择角色并确认 (4) 在用户管理查看是否已分配角色，并重新登录检查是否拥有相应角色的权限
预期结果	用户获得相应的角色和相应的权限
实际结果	(1) 角色分配成功 (2) 拥有权限
结论	角色分配功能测试符合预期结果，测试通过

权限管理模块新增权限的测试用例如表 6-3 所示，权限分配是指往数据库 CT_RESOURCE 表里新增权限内容，与后台权限内容相对应。该用例旨在验证新增权限功能能否成功地新增权限。

表 6-3 新增权限功能测试用例

测试用例编号	TC008
测试用例标题	新增权限功能测试
测试目的	验证能否成功新增权限
优先级	1
测试流程	(1) 点击进入系统设置权限管理模块 (2) 在操作中点击添加的图标 (3) 在弹出表单中填入权限名称、图标、地址 (4) 在权限管理查看是否已添加权限，并将该权限分配到角色，再把角色分配给用户，登录验证
预期结果	用户获得相应的角色和相应的权限
实际结果	(1) 新增权限成功 (2) 拥有权限
结论	新增权限功能测试符合预期结果，测试通过

物品模拟模块物品模拟的测试用例如表 6-4 所示，物品模拟是指玩家进入物品模拟界面，点击物品，可以看到物品的父物品（该物品可以合成什么）和子物品（该物品需要哪些配方）。该用例旨在验证物品模拟功能能否成功地进行物品模拟。

表 6-4 物品模拟功能测试用例

测试用例编号	TC0011
测试用例标题	物品模拟功能测试
测试目的	验证能否成功物品模拟
优先级	1
测试流程	(1) 点击进入刀塔物品模拟模块 (2) 在操作中点击升级物品按钮 (3) 随机点击一个物品 (4) 在右下角区域观察是否有该物品的父物品和子物品出现
预期结果	该物品、其父物品和其子物品均出现
实际结果	(1) 物品模拟成功
结论	物品模拟功能测试符合预期结果，测试通过

6.3 非功能性测试

非功能性测试同样重要，主要是测试平台在性能、兼容性以及可用性方面的表现，本节以性能测试为例对系统进行非功能性测试。

性能测试的用例如表 6-5 所示，平台面向运营人员和玩家，并发量较高，因此平台涉及了大量数据查询，所以在响应时间方面有较高的要求。

表 6-5 性能测试用例

测试用例编号	TC050
测试用例标题	性能测试
测试目的	验证英雄信息查询的响应时间，系统能否在合理的时间返回英雄信息
优先级	2

测试流程	从系统服务器日志中抽取 1000 条与英雄信息查询相关的日志，统计每次相关操作的相应时间，并对其分析
预期结果	根据用户的体验，理想的响应时间应该在 1s 以内返回所有数据，因此预期结果在 1s 以内
实际结果	95%的处理时间在 1s 以内；3.2%的处理时间在 1s 以上，1.8%的结果是操作失败
结论	实际结果在目标范围内，测试通过

结论与展望

近些年来，游戏市场规模越来越庞大，国内仅手游市场收入达 819.2 亿元。在 Dota 界，国内代理商完美世界未曾做过专门的游戏数据分析软件，只有 DotaMAX 游戏数据分析做得较为专业，然而 DotaMAX 并不支持物品和天赋模拟。Dota2 资料库管理平台在国内第一次实现物品和天赋模拟。

经过历时一个半月的开发，主要研究成果如下：

（1）采用开源软件和开源框架，减少了开发成本

IntelliJ IDEA 社区版完全免费，前端框架 ExtJs 有 trial 版本，阿里云的诸多 SDK 在一定的用户并发量下可以免费调用，项目部署在阿里云学生服务器上，价格也是相当便宜。

（2）采用市面较新的框架和生产环境

采用基于 SSM 的 SpringBoot 框架和 Maven 仓库管理，摒弃了 Spring 的复杂配置、tomcat 的安装部署和繁杂的 jar 包管理

（3）加强了不同用户之间的交流

玩家可以对游戏运营人员留言，提出改进意见，运营人员进行回复，促进游戏的版本更迭，以玩家需求为导向。

（4）便捷的用户角色权限机制

管理员对用户分配角色、对角色分配权限，合理控制每个用户能拥有哪些权限，方便管理。

（5）提供了智能数据统计功能

玩家不仅可以对战绩信息进行检索，也可以看到经过后台计算处理过的战绩详情。

然而，Dota2 资料库管理平台存在一些问题。

（1）缺少移动端的支持

本项目主要是在 B/S 架构上进行设计和开发的，在移动端上无法顺利运行，给移动端用户带来了不便。

（2）物品模拟和天赋模拟未与属性相结合

物品和天赋模拟不能计算出攻击力、智力、力量和其他基础属性的值，也没有对这些属性值结果进行运算处理，这些未完善的功能需要更长的时间来开发完成。

致谢

我要感谢我的指导老师，她为人友善认真，在论文的用词她总会以专业标准要求我，从选题开始，她就给我细心的指导，开题报告也经过导师的指导反复修改，导师认真负责，从平台开发到结束遇到了许多困难，导师给我鼓励，最终使我克服重重困难，完成了 Dota2 资料库管理平台的发展，在指导教师的引领下，我完成了毕业论文。

我要感谢计算机与软件工程系提供一个良好的学习氛围，感谢学院对我的辛勤栽培，我在学校四年来学到了很多有用的知识，感谢计算机工程学院为我提供了教育资源，感谢导师给我的悉心指导，在此我还要感谢曾经实习的同事和班里的同学，在我遇到难题时给了我方向，使我不再迷茫，感谢你们。

作者：陈劲骁
2018 年 4 月 30 日

参考文献

- [1] 秦姣华,袁智威,王振,刘纯和. 利用 Ajax 与 ExtJS 改善用户体验[J]. 电子设计工程,2011,19(10):1-4.
- [2] 江雪,曹子钰,李洋,钟逸.ExtJS 中 ViewModel 的数据继承[J].中国新通信,2017,19(17):91-91.
- [3] 李亚丽,梁华国,张国平.基于 ExtJS 和 SSH2 的软件工程资源共享平台[J].信息技术,2017(07):99-102.
- [4] 王正华. 基于 REST 与 Shiro 的遗留系统再工程技术研究[D].太原理工大学,2014.
- [5] 盛伟翔,王昊,董晓睿,谢桂华.基于 REST 架构风格的精品课程平台建设[J].南昌大学学报(理科版),2015,39(03):251-254.
- [6] 张灿. 游戏管理系统的设计与实现[D].北京交通大学,2016.
- [7] 费国俊. 基于 Struts 框架的手机游戏产品管理系统的设计与实现[D].北京工业大学,2012.
- [8] 蒋伟.Java EE 简洁模式设计及其在网络游戏管理系统中应用[D].哈尔滨工业大学,2007.
- [9] 施海华,陈宝霞,吴坚等.基于 NGB 的互动游戏管理系统的设计[J].有线电视技术,2012,(4):27-29.
- [10] 董欢.网络游戏管理系统的设计与实现[D].南开大学,2012.
- [11] 王贞.网络游戏管理系统设计与实现[D].华中科技大学,2009.
- [12] 张士铎,刘克.网络游戏后台管理系统的关键技术研究[J].中国传媒大学学报(自然科学版),2016,23(5):62-66.
- [13] 郭杰.基于 OGRE 框架的 3D 网络游戏资源管理系统设计与实现[D].北京交通大学,2010.
- [14] 杨家炜.基于 Spring Boot 的 web 设计与实现[J].轻工科技,2016,32(07):86-89.
- [15] 武志强. 基于 SpringMVC 和 MyBatis 框架的理财规划系统的设计与实现[D].北京交通大学,2016.
- [16] 雷镞,陈俊. 基于 ExtJS 及 Pushlet 实现 Web 及时交流[J].软件导刊,2010,9(2):118-120.LEI Lei, CHEN Jun. Implementation of Web communication by ExtJS and pushlet [J]. Software Guide,2010, 9(2):118-120.
- [17] 程冬梅,王瑞聪,刘燕,秦伟俊.基于 REST 架构风格的物联网服务平台研发[J].计算机工程与应用,2012,48(14):74-78+157.
- [18] 死神的丧钟. 几款 Web 富客户端(RIA)框架[EB/OL].
<https://blog.csdn.net/accountwcx/article/details/47835261>.
- [19] Sencha 中文站. ExtJS 6.5 和 Sencha Cmd 6.5 GA 版本发布[EB/OL].
<http://extjs.org.cn/node/818>.
- [20] 刘雅琴. 浅析 Oracle 大型数据库的性能优化[J].中国科技博览,2013(2):307-307.
- [21] 中国产业信息. 2017 年我国文化产业占 GDP 比重及相关产业企业营收增速分析[EB/OL]. <http://www.chyxx.com/industry/201711/587273.html>.
- [22] 仙剑. Java Web 学习笔记（四） 基于 SpringMVC+Bootstrap 创建 WebApp[EB/OL]. <https://blog.csdn.net/jian200801/article/details/54379381>.
- [23] Franson. Spring MVC 完整示例[EB/OL].

- <http://www.cnblogs.com/franson-2016/p/5603651.html>.
- [24] 百度百科. 可行性分析[EB/OL].
<https://baike.baidu.com/item/%E5%8F%AF%E8%A1%8C%E6%80%A7%E5%88%86%E6%9E%90>.
- [25] 百度百科. 数据字典[EB/OL].
<https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%AD%97%E5%85%B8>.
- [26] 吴沧舟,兰逸正,张辉.基于 MySQL 数据库的优化[J].电子科技,2013,26(09):182-184.
- [27] 半宇白肌.
<http://baijiahao.baidu.com/s?id=1595717762932558144&wfr=spider&for=pc>[M]. 百家号:百家号, 2018.

附录

附录 A 引用的外文文献及其译文

Critical success factors to improve the game development process from a developer's perspective

Saiqa, Aleem, Luiz, Fernando, Capretz, Faheem, Ahmed. Critical success factors to improve the game development process from a developer's perspective[M].

Canada:Thompson River University, 2018. 1-5

Abstract

The growth of the software game development industry is enormous and is gaining importance day by day. This growth imposes severe pressure and a number of issues and challenges on the game development community. Game development is a complex process, and one important game development choice is to consider the developer perspective to produce good-quality software games by improving the game development process. The objective of this study is to provide a better understanding of the developer's dimension as a factor in software game success. It focuses mainly on an empirical investigation of the effect of key developer factors on the software game development process and eventually on the quality of the resulting game. A quantitative survey was developed and conducted to identify key developer factors for an enhanced game development process. For this study, the developed survey was used to test the research model and hypotheses. The results provide evidence that game development organizations must deal with multiple key factors to remain competitive and to handle high pressure in the software game industry. The main contribution of this paper is to investigate empirically the influence of key developer factors on the game development process.

Keywords : Developer's perspective; Software games; Empirical investigation; Good-quality games; Game development process; Game developer's factors.

1. Introduction

The first software game was created half a century ago. In the world of software gaming, many things have changed during this time period. Now the software game industry has reached the point that it rivals other well-established industries such as music and cinema. As a result, the soft-ware gaming business has grown enormously, has made billions of dollars in profit, and has started to mature over time . The game development process has also had an impact on the industry, which now counts on special methodologies and mature processes for its development, ultimately leading to an enhanced game development process. Game developers try to produce games that are different from any other game in the market. This difference can be achieved through by introducing new perspectives, new gameplays, new genre combinations, enhanced graphics, or new characters. Therefore, almost all games must be novel, and

their success depends on their overall quality . Only good-quality games are able to retain their players, and this has become an important factor for any software game to succeed commercially. In other words, if a game is not of good quality, players can easily switch to another game. Hence, it has become mandatory for the software game industry to try to morph and adapt to the preferences and demand of its players.

One of the main concerns in game development process is that developers need to follow best practices and procedures from software engineering discipline to develop good quality games. The game development process involves four main phases: concept, pre-production, production, and post-production. It consists of various activities such as synopsis, background research, script writing, visualization and concept art, level and inter-action design, animation, programming, media editing, integration, testing, and publishing. Soft-ware games are also characterized based on the category into which they fall, which is called the genre of the game. Genres include action, shooters, fighting, racing, adventure, sports, role playing, strategy, simulations, puzzles, dance, music, and others. Each genre has its own requirements which must be taken into consideration during the pre-production phase. For this reason, software game development is considered as a complex process that involves multidisciplinary collaborative team efforts and processes (including sound, gameplay, art, artificial intelligence, control systems and human factors) to develop a creative product. Fundamentally, game development is a form of software development process with several additional requirements such as creative design, artistic aspects, and visual presentation. In this context, game development organizations can apply the same software engineering principles to improve their development processes. However, many studies have discussed the challenges of applying software engineering principles to the game development process.

Kultima highlighted these challenges from the game design perspective. Blow discussed their implications from the perspective of technical frameworks and development techniques. Blow and McGill discussed issues even for the required technical skills for game development. Software game development also requires a range of skills that include design, project management, development, and asset creation. It also involves team members from heterogeneous disciplines, e.g., game designers, artists, programmers, and software developers. Knowledge of best practices for game development is very important and has become crucial to sustain the growth of the software game industry. Finally, this knowledge will help game developers make correct game development decisions at the right time. An investigation of key success factors from a developer's perspective will contribute to the understanding of current development process implications and will help developers improve the game development process.

Exploring diverse developers' preferences for software game development will provide a significant benefit to improve the development process by generating valuable insights. No research has been done to date on including developer centered factors in the software game development process. This study will help identify key factors empirically from the developer's perspective, an effort that will ultimately help improve the software game development process to produce good-quality software

games. To identify key factors, a quantitative survey was conducted, and the results are reported here. The survey was used to test the re-search model and several hypotheses. Finally, the results show that consideration of key factors from a developer's perspective helps identify important game development choices and their implications for the current process.

1.1 Research Background

The software game domain covers a great variety of player modes and genres. The complexity of digital games has posed many challenges and issues in software development because it involves diverse activities in creative arts disciplines (storyboarding, design, refinement of animations, artificial intelligence, video production, scenarios, sound effects, marketing, and finally sales) besides technological and functional requirements. This inherent diversity leads to a greatly fragmented domain from the perspectives of both underlying theory and design methodology. The software game literature published in recent years has focused mainly on technical issues.

Issues of game production, development, and testing reflect only the general state of the art in software engineering. Pressman stated that a game is a kind of software which entertains its users, but game development faces many challenges and issues if only a traditional software development process is followed.

Many researchers have discussed game development challenges. Pertillo et al. surveyed the problems faced by game development organizations. The overall game development process combines both an engineering process and the creation of artistic assets. Ramadan and Widjani compared various game development strategies from a management perspective, and some researchers have proposed frameworks for game development. To effectively manage and improve the game development, key developer's factors are required. Tschang and Petrillo et al. highlighted the issues in the game development process and its differences from traditional software development practices.

In traditional software engineering, the development phase usually involves activities like application design and implementation, and the production phase is when the software actually runs and is ready for use. However, in the game development, the production phase includes the development process, which is the pre-production phase of the software engineering process, and the production phase of software engineering is actually the post-production phase of the game development life cycle. Therefore, the game development is different from the traditional software engineering process, and many researchers have studied the challenges faced by this domain. Kanode and Haddad stated that an important incorrect assumption has been made that game development follows the waterfall method. More recently, researchers have agreed that it must follow the incremental model because it combines the waterfall method with an iterative process. Petrillo et al. reported a major concern, that developers for software creation in the game industry commonly use very poor development methodologies. The game development life cycle (GDLC) is the object of questions on many forms, which attempt to determine what types of practices are used. However, this question has no single answer. The most prominent observation made in these studies is that to

address the challenges faced by the GDLC, more rigorous software engineering strategies must be used. However, the proposed GDLCs do not ensure the quality of the development process. Hagan et al. published a systematic literature review of software process models used for game development. They concluded that agile and hybrid approaches are used by most organizations for game development. They also reported that Scrum, Kanban, Rapid Development Application (RAD), XP, and incremental methodologies are used by game development organizations. The major difference in software development and game development is in the design phase because design of game may undergo major change in late development. The other differences are content development and quality criteria. Managing game development has become a much harder process that anyone could have initially imagined, and because of the fragmented nature of the domain, no clear picture of its advancement can be found in the literature.

From the above discussion, it can be easily concluded that game development process is different from traditional software development process. Kasurinen et al. argued that current software engineering knowledge is unable to bridge the gap between software engineering and certain aspect of game development. The overall development process to produce a game includes art, audio and gameplay other than software development discussed above. In the game development process, the content and production activities are performed in tandem with the development and engineering activities. Further, it is well agreed that the game development process is a multidisciplinary activity that involves the merging of creative and technical talent to bring a concept to life, where the main activities can be categorized into content and production, and engineering at each phase of the development process.

Moreover, sometime game development organizations reduce their development process due to of high competition and extreme market demand so they can be first to market. This reduction of the development process definitely affects game quality. Therefore, they do not strictly follow the software engineering standards and practices. Because of these types of complex project-management tasks, the game development process diverges from traditional software development. Nevertheless, the differences between software engineering and games development are not exclusive; it seems that traditional software development does not fully support game development activities and provide process assessment procedures. So, we need key success factors to improve game development process that may overlap with traditional software development factors or just exclusive to game development. Therefore, it has become important now to investigate the critical success factors for game development organizations in developing good quality games from developer's perspective.

1.2 Research Motivation

Game development has become incredibly challenging due to rapid changes in game technology such as game platforms, game engines, and reuse of code modules for different genres. During the 1990s, game development was usually carried out by small team members and involved simple architectures consisting of 2D graphics, sound, simulation, and input/output streaming. The first software games were developed by a few talented individuals from diverse backgrounds like mathematics, computer science,

and physics with no educational background in engineering or computer science. At that time, developers were mainly focused on how to develop interesting games rather than on architecture or software engineering principles. The current success of the game industry, continuous enhancements in game technology, and the need to meet the ever-higher expectations of the players resulted in a complex game development process.

The main research motivations behind this study are the rapid and continual changes in technology and the severity of competition in game development organizations. Ultimately, these factors will not only affect the business, but also will have a major impact on the game development process. Nowadays, games are developed by large teams because game projects have grown in size and complexity. Various stakeholders are involved in the development process and have different expectations and world views. For example, the game designer does not know the level of complexity involved in implementing artificial intelligence to represent the behavior of a non-player character. A software engineer may think that some features in the game design document are infeasible to implement due to time deadlines or technical constraints. Another important requirement that must be part of the game is the fun, flow, and enjoyment factors.

基于开发者角度提高游戏开发进程的决定性因素

加拿大:汤姆逊河大学, 2018.

摘要

随着游戏软件开发产业巨大的增长, 游戏软件产业获得了越来越多的重视。这种增长增加了游戏开发社区压力和严峻的挑战。游戏开发是一个复杂的过程, 一个重要的游戏开发方案是通过提高游戏开发过程考虑到开发者角度来制作优良的软件。这种研究的目的是提供一种更好的理解, 将开发者维度作为重要因素考虑在游戏的成功中。它主要聚焦于在游戏开发过程中关键开发者因素和最终将要完工的游戏质量的经验性调查上。一定数量的调查被开展用来确定关键开发者因素在已被提高游戏开发进程所发挥的作用。对于这个研究, 被开展的调查被用作测试研究模型和假设。这些结果提供了游戏开发组织活动必须处理多因素来保证有竞争力和处理在游戏软件产业的高压力的证据。该研究的主要贡献在于经验性地调查了关键开发者因素在游戏开发进程中的影响。

关键词: 开发者视角, 游戏软件。经验性调查, 高质量游戏, 游戏开发过程, 游戏开发者因素。

1. 引言

半个世纪前第一款游戏软件问世, 在游戏软件世界里, 许多事在此时期发生改变。现在游戏软件产业已经到达了可以与其他成熟的产业如音乐和电影相媲美的程度。结果是游戏软件商业化极速扩张, 有数十亿美元的收入, 已经开始随着时间成熟, 游戏开发过程同样对游戏产业有影响。游戏产业依赖于特殊的方法学和开发的成熟过程, 最终引向加强的游戏开发过程。游戏开发者尝试生产和市场中其他游戏不同的游戏。这种差异也可以通过新视角、新的游戏设置、先进的图像处理或者引入新角色来达到。因此, 几乎所有游戏必须来源小说, 他们的成功依靠总体的质量。只有高质量的游戏才能赢得玩家的垂青, 这已经成为任何一个游戏软件商业成功的重要因素。换句话说, 如果一个游戏不具备高质量, 玩家可以轻易地选择其他游戏。因此, 高质量成为对游戏软件产业尝试达到高图像标准和满足玩家的偏好强制的要求。

一个主要的担忧在于开发过程是开发者需要根据最好的实践和软件工程开发高质量产品的原则。游戏开发过程包括四个阶段: 概念、预生产、生产和后期制作。它包含不同的活动, 如大纲、背景研究、脚本编写、视觉化、概念种类、水平和交互动作设计、动画、变成、媒体处理、集成、测试和出版。游戏软件也具有基于他们所属的类别, 这叫做游戏的流派。流派包括动作、射击者、打斗、比赛、冒险、运动、角色扮演、策略、模拟、解谜、跳舞、音乐和其他。每个流派都有自己的、必须在预生产阶段就被考虑进去要求。因为这个原因。游戏软件开发被认为是一个包括多学问的和多合作的团队努力和过程(包括声音、游戏设置、美术、人工智能、控制系统和人为因素)来开发一个创造性的产品的复杂的过程。根本地, 游戏开发是一种有着服务附加要求如创造性设计、艺术形式和视觉展现的软件开发过程的形式。在本文中, 游戏开发组织可以应用同样的软件工程原则来提高开发流程。然而, 许多研究已经讨论过针对于游戏开发过程的应用软件工程原则的挑战。

Kultima 强调了这些来自于游戏设计角度的挑战。Blow 讨论了从科技框架和开发技术的角度的影响。Blow 和 McGill 甚至讨论了游戏开发必备技能的问题。游戏软件开发也需要一定的技能, 包括设计、项目管理、开发和资产创建。它包

含了具备多种技术的团队成员，例如游戏设计者、美术家、编程者和软件开发人员。游戏开发知识的最佳实践是至关重要的，已经成为支撑游戏软件产业成长的重要的基石。最终，知识会在正确的时间帮助游戏开发人员做出正确的游戏开发决定。在对基于开发人员角度的关键成功因素的调查中会有助于开发过程影响的理解，并且帮助开发人员提高游戏开发过程。

探究不同的开发人员对游戏软件开发的偏好会提供一个有意义的优势通过产生有价值的洞察力来提高开发过程。没有研究记录过在游戏软件开发中以开发人员为中心的作用。此研究会帮助人们经验性地找出基于开发人员角度的关键因素，我们所做的努力会最终帮助提高软件开发过程来制造高质量的游戏软件。为了找出关键因素，大量的调查会被试验，结果会在这里被报告。这份调查被用作测试研究模型和许多假设。最终，结果展现了基于开发人员角度帮助提高游戏开发选择和他们对现阶段进程的影响关键因素的考量。

1.1 研究背景

游戏软件领域包括一大堆玩家模型和流派。电子游戏的复杂性已经在游戏软件开发领域形成了许多挑战和问题，因为它包含了在创造性的美术原则中不同的活动（情节制作、设计、动画的精制、人工智能、影像设计、剧本、声音效果、市场和最后销售）除了科技和功能的要求。内在的多样性将我们引入了成碎片性的领域基于潜在的理论 and 设计方法学。近些年游戏软件文学出版物大多数关于科技问题的探讨。

游戏生产、开发的问题，并且在软件工程中测试仅仅影响普通美术的基本状态。Pressman 陈述了游戏是一种让用户快乐的软件，如果仅仅采用传统的软件开发过程，游戏开发会面对许多挑战和问题。

许多研究者已经讨论过游戏开发挑战。Petrillo 调查了许多游戏开发组织活动的所面临的问题。总体的游戏开发过程混合了工程过程和艺术的创造。Ramadan 和 Widyanib 根据管理的角度比较了不同的游戏开发策略。一些研究者计划将框架应用于游戏开发中。为了有效地管理和提高游戏开发，需要关键开发人员因素。Tschang 和 Petrillo 强调了在游戏开发过程中的问题和与传统软件开发实践的差异。

在传统的软件工程中，开发阶段通常包含活动，如应用设计和实现，生产阶段是当软件事实上运行和准备投入使用的时候。然而，在游戏开发阶段，生产阶段包含了开发过程，是软件工程过程的预生产阶段，并且软件工程的生产阶段事实上是游戏开发生命周期的后期处理阶段。因此，游戏开发和传统的软件工程过程不同，许多研究者研究了该领域所面临的挑战。Kanode 和 Haddad 陈述了一个重要的错误的开发使用瀑布模型的假设已经被证实。最近，研究者同意遵守增加的模型，因为它包含了用迭代流程的瀑布方法。Petrillo 表达了担忧，游戏工业的软件开发人员通常在开发方法上很薄弱。游戏开发生命周期是许多形式问题的目标，它能够决定使用哪种实践方案。然而，这种问题已经没有唯一的答案。最显著的观察使这些研究处理游戏开发生命周期面对的挑战，更多严谨的软件工程策略会被使用。然而，提出的游戏生命开发周期不保证开发过程的质量。Hagan 出版了一个回顾软件过程模型使用在游戏开发的系统文献。他们总结了敏捷和混开发。他们也报告了快速开发应用和增加的被使用在游戏开发组织活动的方法学。软件开发和游戏开发的主要不同在于设计阶段，因为游戏设计在后期会经历很多变化。其他的不同在于满意的开发和质量标准。管理游戏开发已经成为很艰难的

过程,任何人都可以想象并且因为碎片化的自然领域,在文献中没有进步的迹象能找到。

从上面的讨论,可以很容易的总结出游戏开发过程是不同于传统的软件开发过程。**Kasurinen** 持有异议,他认为目前的软件工程知识是不能够在软件工程和特定的游戏开发方面架起桥梁弥补这之间的间隙。总体的开发过程生产一个游戏,包括美术、声音和游戏设置。在游戏开发过程中,内容和生产活动与开发和工程活动相串联。更进一步来说,大家都知道游戏开发过程是一个多学科的活动,它包括创造和科技才能的混合来给生活带来一个全新的概念,主要活动(每个开发过程阶段的)可以被归类为内容和生产、制造和工程。

此外,因为竞争激烈和市场需求有时游戏开发组织活动减少了开发过程,所以这些产品能率先进入市场。这种开发过程的缩减肯定会影响游戏的质量。因此,他们不能严格地遵循软件工程标准和实践。由于这些复杂的项目管理任务的种类,游戏开发过程偏离了传统软件开发。然而,这种在软件工程和游戏开发之间的差异不是可排除的;看起来传统的软件开发不能完全地支持游戏开发活动并且提供流程评定步骤。所以,我们需要关键成功信息来提高能和传统软件开发过程重叠的或者仅仅为游戏开发服务的游戏开发过程。因此,调查基于游戏开发者角度的游戏开发组织活动的决定性的成功因素现在已经变得非常重要。

1.2 研究动机

因为游戏科技快速的改变如游戏平台、游戏引擎、和不同流派的代码模块的重用,游戏开发已经难以置信地变得愈来愈具有挑战性。在九十年代,游戏开发通常由一些小型队伍成员完成,这些游戏项目包括简单的结构设计,包含了 2D 图像、声音、模拟和输入输出流。第一个游戏软件是由有以为有天赋的人,不像数学家、计算机科学家和物理学家,他没有软件工程和数学的教育背景。在当时,开发者专注于如何开发有趣的游戏而不是研究架构和软件工程原则。最近游戏产业的成功,持续的游戏科技进步和满足玩家的天马行空的需求,导致出现了复杂的游戏开发流程。

该研究的主要的研究动机是快速和持续不断的科技变化和游戏开发组织中的激烈竞争。最终,这些因素不仅仅只影响商业利益,也会对游戏开发进程有重要的影响。现如今,游戏由大型团队开发,因为游戏项目已经在量和复杂度上增加。不同的利益相关者与开发进程有关,他们有不同的期望和世界观。例如,游戏设计者不知道实现人工智能来代表非人来玩家的行为的复杂水平。一个软件工程师可能基于项目开发时间和技术限制认为游戏设计文档中的一些方案是不可行的。另外一个很重要的要求是游戏必须有趣、流畅和乐趣。

附件 B 主要参考文献的题录及摘要

1、利用 Ajax 与 ExtJS 改善用户体验

【篇名】利用 Ajax 与 ExtJS 改善用户体验

【作者】秦姣华, 袁智威, 王振, 刘纯和

【关键词】Ajax; ExtJS; 富网络应用框架; B/S 结构; 用户体验;

【摘要】随着 Web 技术的不断发展,传统的 B/S"瘦客户端,胖服务器"结构已不能满足用户正常体验和实时交互要求。Ajax 技术可以实现用户异步请求,减轻网络负载,ExtJS 作为 Ajax 的 RIA 框架,使得界面构建更加容易。从 Ajax 原理,ExtJS 在 OA 型系统中的应用及性能调优出发,设计实现了三农电子商城后台管理系统,并探讨了 ExtJS 在基于 B/S 结构的 OA 型电子商务系统中对用户体验的改善。系统界面美观,操作方便,开发效率高,维护规范,可以满足用户正常体验和实时交互需求。

4、基于 REST 与 Shiro 的遗留系统再工程技术研究

【篇名】基于 REST 与 Shiro 的遗留系统再工程技术研究

【作者】王正华

【关键词】基于 REST 与 Shiro 的遗留系统再工程技术研究;

【摘要】遗留系统再工程为遗留系统的复用提供了现实可行的途径,可将现存的遗留系统重新构造为新的形式。随着互联网及软件工程技术的迅猛发展,Web 服务给遗留系统再工程提供了一种很好的解决方案。传统的 RPC(Remote Procedure Call,简称 RPC)式 Web 服务,接口设计复杂且很难标准化,而基于表述性状态转移(Representational State Transfer,简称 REST)的设计原则更契合 Web 服务复用的理念。面向资源架构的 REST 风格渐已成为 Web 服务设计趋势的主流,然而在复杂的互联网环境下 Web 服务的安全性显得尤其重要,Shiro 作为一个强大而灵活的安全框架,具有直观和易于使用的特性,集成 Shiro 至遗留系统,可以简化应用程序的安全操作。本课题针对传统的 RPC 式 Web 服务存在的缺陷,引入 REST 风格的 Web 服务至遗留系统再工程领域,并且结合 Shiro 安全框架能够加强遗留系统的身份认证的安全性以及权限控制的灵活性,从而保证 REST 服务的安全性,为此,本文提出一种基于 REST 与 Shiro 的遗留系统再工程技术方案。本文首先分析遗留系统再工程的国内外研究现状,对遗留系统、软件再工程、Web 服务以及 Shiro 框架等背景知识作出详细的阐述,对比了几种不同的 Web 服务方式,引出 REST 风格的 Web 服务的特征,在分析其优势的基础上,从设计的角度详细分析了 REST 式 Web 服务理论及相关技术。文中详细阐述了基于 REST 与 Shiro 的遗留系统再工程技术方案,结合实际项目背景,设计并实现了基于 Shiro 重构身份认证以及统一用户管理系统权限的实例以及基于 REST 包装“账户”服务的实例。通过实例验证说明 REST 与 Shiro 在遗留系统再工程中应用的优越性。

11、网络游戏管理系统设计与实现

【篇名】网络游戏管理系统设计与实现

【作者】王贞

【关键词】网络游戏; 后台管理系统; PHP 语言; B/S 模型; 安全;

【摘要】随着 Internet 和计算机网络的普及,网络已经走进千家万户,人们的生活越来越离不开网络。学习,工作闲暇之余人们可以通过网络来放松自己,也正因为如此网络游戏开始走进我们的生活。比如说日本麻将就是一个必须依托于互联网进行,可以允许多人同时参与的网络游戏,通过用户与用户之间的互动达到交流,休闲和娱乐的目的。正由于网络游戏允许多用户参与,所以在后台对与用户的管

理是必不可缺的。针对麻将游戏系统后台用户管理的需要,构建了一个基于 PHP 与 MySQL 的麻将用户管理系统。该系统运用健全的网络体系,建立一个集与 B/S 结构与先进的 PHP 加 MySQL 技术为一体的系统。系统中涉及到用户登陆,用户信息管理,用户游戏得分排名,大会的具体情况等各个环节,具有对多层次用户的完善的管理功能。设计时使用了 Apache+PHP+MySQL 三者结合的先进开发模式。通过使用系统,用户可以更好地了解游戏,进入游戏,享受游戏。同时也方便管理员更好的管理用户的信息,满足了系统管理员和游戏玩家等的信息管理需求,达到了用户,游戏,管理员三者一体的目的。用户只需要在浏览器上操作就可以实现系统的所有功能。系统的界面人性化,操作方便。系统使用了多级安全保护措施,充分保证了系统的安全运行。