



# 广东工业大学

## 本科毕业设计（论文）

### 基于微信小程序的游戏微店

学    院 计算机学院

专    业 网络工程

年级班别 2015 级（4）班

学    号 3115005534

学生姓名 李楚豪

指导教师 张海笑

2019 年 5 月



## 摘 要

微信小程序是最近几年才提出的新概念。伴随着微信在人群中的广泛使用，不仅意味着腾讯拥有了新的摇钱树，而且这也是其它商家利用微信的巨量用户群体拓展自身的潜在客户群体的机会。微信小程序其独特的技术优势在于，微信小程序不需要进行客户端下载，而是基于 HTML5 内核进行页面渲染。因此微信小程序可以对大部分移动设备提供完美的技术支持。不仅具有便携性，随时随地都可以使用，而且能够非常好的体现出移动设备的优势。

随着人们的生活水平不断提高，精神生活对人们的重要性越来越大，而游戏则是年轻人最喜欢的娱乐方式。本设计最终实现一个基于微信小程序的游戏微店，并且为用户提供游戏购买服务。不仅具有简洁明了的微信小程序界面，而且也拥有传统的网上商务的后台管理系统以及服务端支持。商家通过后台管理系统管理商店的用户、商品以及用户购买的订单等各种各样的纷繁信息。

本设计将会采用 Java 语言来为游戏微店提供技术支持，服务器使用 SpringBoot 框架技术来进行搭建，能够快速地搭建一个简洁有效率的 Java web 项目；前端部分将使用 Vuejs 构造项目，Vuejs 能够将普通的网页组件化，方便代码的复用降低耦合度。

**关键词：** 微信小程序，游戏微店，Java，Vuejs

## **Abstract**

The WeChat applet is a new idea that has only been created in recent years. With the extensive use of WeChat in the crowd, it not only means that Tencent has a new cash cow, but it is also an opportunity for other merchants to expand their potential customer base by using WeChat's huge user base. The unique technical advantage of the WeChat applet is that the WeChat applet does not require client-side downloading, but instead renders the page based on the HTML5 kernel. So WeChat applet can provide perfect technical support for most mobile devices. Not only is it portable, it can be used anytime, anywhere, and it can very well reflect the advantages of mobile devices.

With the continuous improvement of people's living standards and the increasing demand for entertainment, games have become an indispensable part of people's lives. This design ultimately implements a WeChat applet-based game micro store and provides users with game purchase services. Not only has a simple and clear WeChat applet interface, but also has a traditional online business background management system and server support. The merchant can manage a wide variety of information such as users of the store, merchandise, and orders purchased by the user through the background management system.

This design will use the Java language to provide technical support for the game micro-store. The server uses the Spring Boot framework technology to build, and can quickly build a simple and efficient Java web project; the front-end part will use the Vuejs construction project, Vuejs will be able to Ordinary web page componentization facilitates code reuse and reduces coupling.

**Key words:** WeChat applet , game store , Java , Vuejs

# 目录

1 绪论.....	1
1.1 研究背景.....	1
1.2 研究现状.....	1
1.3 研究目标.....	2
2 相关技术介绍.....	3
2.1 Java.....	3
2.2 微信小程序.....	3
2.3 Spring Boot.....	3
2.4 Mybatis.....	4
2.5 MySQL.....	4
2.6 Vuejs.....	4
2.7 JavaScript.....	5
2.8 C3P0.....	5
3 需求分析.....	6
3.1 功能需求.....	6
3.1.1 微信小程序.....	6
3.1.2 后台管理系统.....	6
3.2 开发环境需求.....	6
3.3 可行性分析.....	7
3.3.1 技术可行性.....	7
3.3.2 操作可行性.....	7
4 总体设计.....	8
4.1 系统功能概述.....	8
4.2 前台模块设计.....	8
4.3 后台模块设计.....	9
4.4 数据库设计.....	10
4.4.1 概念模型设计.....	10
4.4.2 数据库表设计.....	15

5 详细设计与实现.....	17
5.1 微信小程序的设计与实现.....	17
5.1.1 导航栏模块.....	17
5.1.2 游戏显示模块.....	18
5.1.3 订单显示模块.....	20
5.1.4 用户登录.....	22
5.1.5 用户修改邮箱.....	23
5.1.6 发送邮件.....	24
5.2 后台管理系统的设计与实现.....	26
5.2.1 整体结构.....	26
5.2.2 导航栏组件.....	27
5.2.3 单行查询表单.....	28
5.2.4 用户信息修改表单.....	29
5.2.5 游戏列表显示.....	29
5.2.6 游戏信息修改表单.....	29
5.2.7 订单信息列表显示.....	30
5.2.8 订单信息修改表单.....	30
6 运行与测试.....	32
6.1 测试目的.....	32
6.2 测试方法.....	32
6.2.1 单元测试.....	32
6.2.2 集成测试.....	33
6.2.3 系统测试.....	33
6.2.4 验收测试.....	33
6.3 测试流程.....	34
6.3.1 用户测试.....	34
6.3.2 游戏测试.....	35
6.3.3 订单测试.....	36
6.4 测试反馈.....	38

结论.....	39
参考文献.....	40
致 谢.....	41





# 1 绪论

## 1.1 研究背景

在 2017 年 1 月，腾讯推出了微信的新功能——微信小程序。其最大的特点就是即用即开，不用可删。减少了用户的手机空间需求。而微信的内核采用的是具有响应式设计的 H5 框架，能够适应几乎所有的平台与设备。根据相关报道，截止至 2018 年 3 月，在短短的 1 年的时间内，微信小程序的用户规模就已经突破了 4 亿，其中小游戏类小程序占比达 28%。这足以看出微信小程序的未来发展前景。

在多年的编程语言排行榜中，Java 语言和 C 语言基本占据前两名。其中 Java 语言是一款自从问世后就极其受欢迎的计算机编程语言。目前 Java 语言最主要的应用场景是企业级 web 应用程序开发。本系统采用 Java 语言作为服务器开发的语言。在现有的服务器开发框架中，最流行的框架莫过于 Spring Boot，Spring Boot“约定优于配置”的设计思想，令服务端开发变得更具有适用性，并且更好地支持 Web 应用程序开发。

根据以上的背景分析，可以非常清楚地看到微信小程序开发将会成为极具商业利用价值以及实用价值的方向。而采用 Java 进行服务端开发不仅可以优秀地提供服务支持，而且也能为以后功能的添加修改提供了框架。因此本设计一是为了在微信小程序开发这一具有重大价值的方向进行尝试，也是因为 Java 开发能够减少开发的工作量和降低开发难度，以及能够更好地为网络应用开发提供服务支持。

## 1.2 研究现状

从 2017 年微信创始人张晓龙正式发布微信小程序后，有关微信小程序相关的知识，网络上的资源还是相对比较少，而且没有完整的体系，直到目前也没有一个被大多数开发者接受的开发框架。微信小程序的框架被称为 MINA，核心是响应式的数据绑定与视图渲染系统，这其中很大一部分参照了流行的 Vuejs，因此在进行开发的时候可以去学习 Vuejs 的相关知识，能够更好的掌握微信小程序的开发工作。大多数微信小程序开发者都是一边摸索着开发，一边总结自己的经验为其它开发者提供帮助。本设计在进行开发时，最主要参照的是微信官方的 API 文档，采用微信原生的组件并且使用微信开发者工具进行开发与调试。

### 1.3 研究目标

基于以上的背景和现状，本设计旨在实现一个完整的、基于微信小程序的在线游戏商城（游戏微店）。本设计不仅为了在开发过程中进一步掌握已学的知识和技术，而且也是在新兴的技术方向进行一次具有创新性的尝试。此设计基于传统的网上商城，能够实现商品信息查看，商品购买，订单管理等功能。

## 2 相关技术介绍

本设计将会主要采用 Java 语言、微信小程序、Vuejs 作为项目的主要使用技术。下面将详细地介绍一下使用到的技术。

### 2.1 Java

Java 语言是一种分布式的面向对象语言，具有面向对象、平台无关性、简单性、解释执行、多线程、安全性等很多特点<sup>[1,2]</sup>。其中面向对象、平台无关性和多线程这三个特性最被开发者看重。面向对象思想是指将所有可编程内容都可以抽象的视为一个对象，程序则是对象的集合，对象之间的消息通知构成了程序的工作流程。平台无关性则是指 Java 是一种源码编译成字节码在依靠 JVM 虚拟机进行运行，任何平台只要有安装 JVM 都可以毫无障碍的运行 Java 程序。多线程是指程序在同一时刻完成多个任务，在如今的大数据时代，多线程编程的重要性不言而喻<sup>[3,4]</sup>。

### 2.2 微信小程序

微信小程序是微信上提供给用户服务的应用。由于移动端上的应用程序功能逐渐向着服务单一化，简单化发展，应用程序都想着把服务做好做精，但是服务功能又没有持续性，因此每次用户使用某项功能需要去安装专门的软件，使用完成后又没有使用服务的需求，就会把应用卸载。这个过程令用户耗费了大量的时间和流量来下载应用。应用开发者也需要对各种各样的移动平台进行适用性配置。而微信小程序的出现大大地满足了双方的需求。

微信小程序不要进行下载安装，用户可以直接在微信打开对应的小程序，即可使用相关功能。微信则提供一个适用平台给开发者，开发者只需要考虑功能的实现，而不需要浪费时间在设备适用性上面<sup>[5]</sup>。

### 2.3 Spring Boot

Spring Boot 是近些年来比较受欢迎的服务器开发框架。Spring Boot 框架为搭建服务器提供了一套适用性提高的配置，开发者可以直接使用默认配置启动服务器，因此开发

人员就可以不需要定义大部分配置，只去更改个性化配置。在其默认配置中提前设置了非常多的流行框架，这些流行框架可以很轻易的被开发者拿来开发，这些框架原本都需要进行非常繁杂的配置，而 Spring Boot 为这些框架添加了默认配置，如果开发者需要进行框架的个性化使用，可以在配置文件中对需要特别处理的某些配置项进行配置，而不需要对整体配置进行修改。Spring Boot 的理念就是“约定优于配置”<sup>[6]</sup>。

## 2.4 Mybatis

MyBatis 是服务端开发中的 ORM 持久层的框架，开发者通过 ORM 框架将服务器开发与数据库数据修改连接起来。Mybatis 对比起其他的 ORM 框架，它支持定制 SQL、通过 XML 文件解析成为对应的 SQL 语句<sup>[7]</sup>。Mybatis 可以使用 XML 和 Java 注解来将 Java 中的原生类型映射成为数据库中的记录信息。尽管 Mybatis 是一款游戏 ORM 框架，但是使用 XML 进行映射其实会加重开发者的负担，因为 XML 文件是专门的解释型文件，是给 Java 虚拟机检索的而不是给开发者维护的。另外，XML 文件中的 SQL 语句是定制的可以随意调用，但是却没有动态生成 SQL 语句框架的灵活性。

## 2.5 MySQL

MySQL 是一种小型的关系型数据库管理系统（DBMS）。因为 MySQL 是开发源码的，所以很多初学者和小型的公司都会使用 MySQL。这也同样的令 MySQL 成为了最被广泛使用的数据库管理系统。但是自从甲骨文（Oracle）公司收购了 MySQL 项目后不久，就发行了 MySQL 的企业版。MySQL 企业版会比较社区版有更强大的数据库内核以及更好的数据库安全性。

## 2.6 Vuejs

Vuejs 是一套前端开发者使用来进行构建友好的、具有渐进式风格的前端界面框架。Vuejs 与其它的一些前端开发大型框架拥有一些特别不同的地方，它的设计模式是“化零为整”的逐层开发模式，页面可以由各种不同的组件进行组合，甚至同一个组件传入的参数不同也会有不同的表现形式。Vuejs 的官方核心库的关注点只在视图层（View 层）的具体表现方式，而没有对页面的之间的逻辑关系以及数据的动态显示。这样的开发方

式不仅对前端的新手开发者来说非常易于上手，而且还能够非常方便地与第三方库积极整合在一起<sup>[9]</sup>。然而在另一方面，Vuejs 与其它流行的工具库或者其它一些支持类库一起进行开发工作时。也能够完成页面逻辑相对复杂的单页应用项目。

## 2.7 JavaScript

说到 JavaScript 就有很多人与 Java 混淆在一起，但是 JavaScript 其实是一种对网页节点比如 div 节点进行操作以及根据事件进行驱动的语言<sup>[10]</sup>。事件驱动的令客户端浏览器的页面展示从静态页面转变成动态页面，增加用户的使用感受。由于 JavaScript 是在客户端浏览器运行的，用户可以查看并修改 JavaScript 文件，因此，JavaScript 并没有高度的安全性。所以一些关键的、重要的数据操作比如用户密码校验等，必须加密之后，向服务器发送加密数据。

## 2.8 C3P0

C3P0 是一个开源的 JDBC 连接池，支持 Java 语言动态的获取数据库连接，连接释放之后会重新进入连接池。同时，可以通过设置最大连接数可以将有效的避免数据库连接太多，引起系统内存占用过多。它实现数据源和 JNDI 绑定，并支持 JDBC3 规范和 JDBC2 标准扩展。这样的功能支持可以让开发者不需要手动地使用传统的 JDBC 在每次与数据库进行交互时都需要静态的创建连接。目前使用它的开源项目有 Hibernate, Spring 等。

## 3 需求分析

### 3.1 功能需求

本设计需要实现的功能主要包括两个方面：微信小程序以及后台管理系统。下面将详细介绍系统的功能需求。

#### 3.1.1 微信小程序

- 1、用户管理：用户使用微信登录；用户绑定邮箱完成用户信息
- 2、游戏显示：显示全部游戏；按照游戏类别显示游戏；查看游戏详情
- 3、订单管理：查看游戏详情后购买（需要绑定邮箱）增加订单；查看订单详细信息；将订单发送至邮箱

#### 3.1.2 后台管理系统

- 1、用户管理：根据用户 ID 查询；根据用户名查询
- 2、游戏管理：查看所有游戏；根据游戏 ID 查询；根据游戏类型查询；添加新游戏；修改游戏信息
- 3、订单管理：根据游戏查询库存；查询用户的订单；添加游戏库存

### 3.2 开发环境需求

项目构建在 Windows 8.1（Intel core i3-4005U） 四核处理器，8G 内存系统环境下，而且使用了多种编辑器以及不同语言进行项目的开发，因此系统必须还要安装微信开发者工具（微信小程序开发）、Visual Studio Code（后台管理系统页面开发）、IntelliJ IDEA（服务端开发）以及 Google Chrome（页面调试）等开发工具。

### 3.3 可行性分析

#### 3.3.1 技术可行性

该系统采用 WeChat applet、Vuejs、Java 以及 MySQL 四个部分共同开发，在 Windows8.1 操作系统下分别使用微信开发者工具开发微信小程序，使用 Vuejs 技术开发后台管理系统页面，使用 MySQL 作为数据库管理系统开发来完成增删查改等功能。而且各种技术都是非常灵活且易于维护的技术，因此，使用上述技术是开发轻型平台的最佳组合，以说明该系统在技术上是具备一定的可行性的。

#### 3.3.2 操作可行性

目前，绝大部分的计算机系统以及云主机都可以毫无障碍地运行本系统，而且本系统的安装，操作都不会对用户的系统造成任何修改。游戏购买者只需要在微信小程序中实现游戏的购买和查询，而商家只需要在后台管理系统完成对商品的添加和修改，以及订单的管理。

## 4 总体设计

### 4.1 系统功能概述

本设计是典型的网上购物电子商务平台模型，所需要实现的功能包括用户登录注册、游戏查询购买，订单查询等各种基本功能。在本系统的设计开发过程中，开发者必须严格按照标准的代码格式进行开发，比如前端开发中的 ESLint 规范，努力实现系统的稳定性和高可用性。

### 4.2 前台模块设计

如图 4.1 所示，微信小程序的具体流程图从用户进入小程序开始，进行微信登录后，进入主界面查看游戏，挑选到合适的游戏后可以查看游戏的详细信息，然后可以进行购买，购买成功后会在订单管理中显示新的订单信息。最后用户可以选择将订单信息发送到绑定的邮箱完成购买。

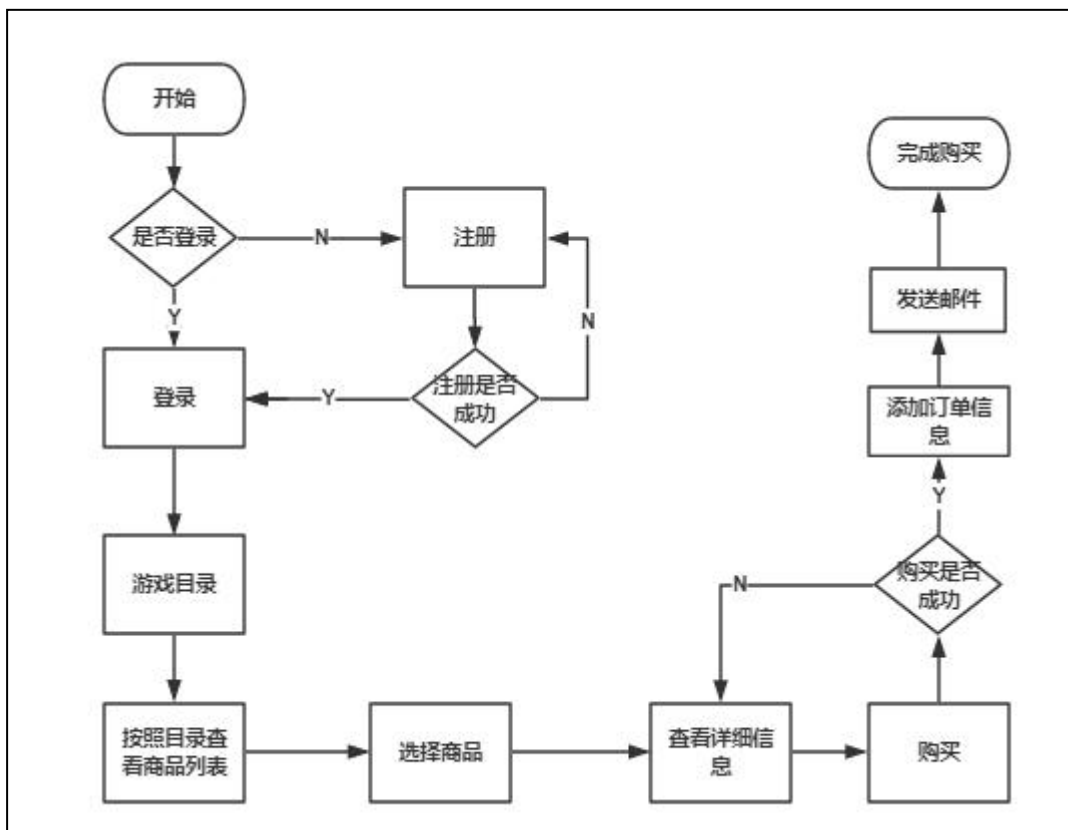


图 4.1 微信小程序购买流程图



根据系统功能分析，可以顺利绘制出系统的功能模块图。

如图 4.2 所示，微信小程序用户进入小程序后可以使用用户管理，游戏显示和订单管理三大功能模块。其中用户管理功能模块包括微信登录和修改绑定邮箱功能；游戏显示功能模块包括：显示全部游戏、按类别显示游戏和点击游戏查看游戏详情；订单管理功能模块则包括：购买游戏增加订单信息、查看订单详情和发送订单到邮箱功能。

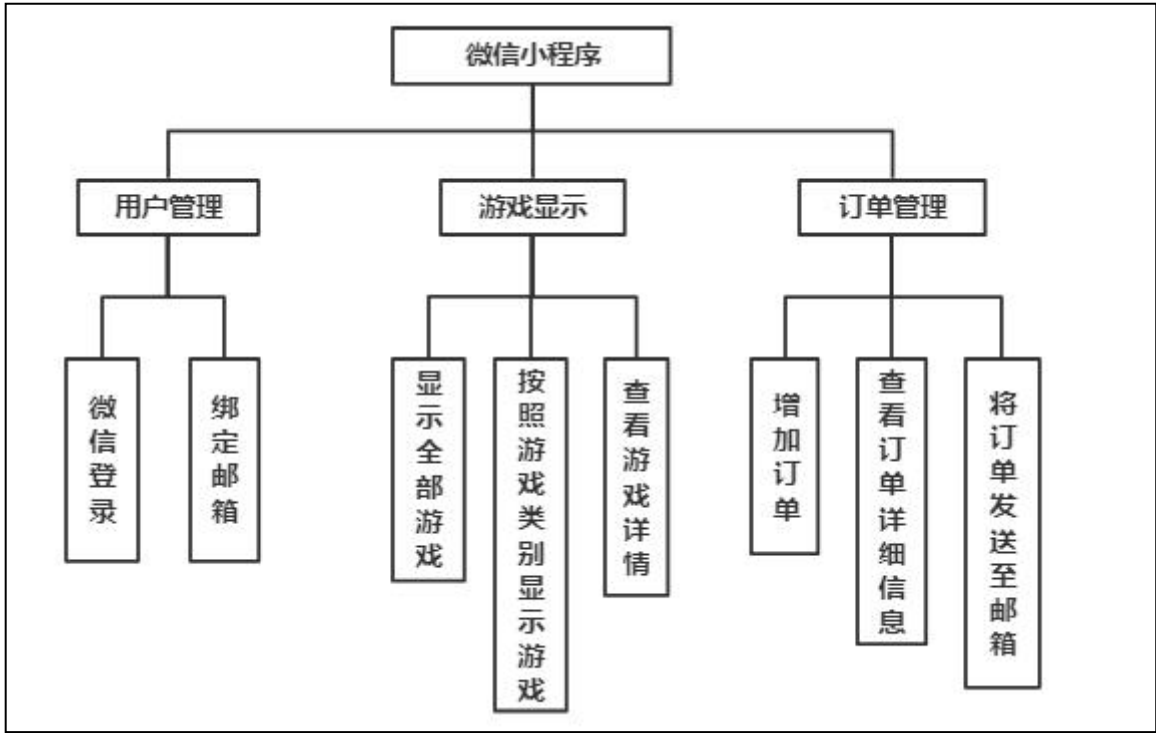


图 4.2 前台功能模块设计

### 4.3 后台模块设计

如图 4.3 所示，管理员可以使用后台管理系统进行用户管理、游戏管理和订单管理三大功能模块的操作。用户管理功能模块包括根据用户 ID 查询用户信息和根据用户名查询用户信息。游戏管理功能模块包括查看所有游戏功能、根据游戏 ID 查询游戏功能、根据游戏类别查询游戏功能、修改游戏信息功能和添加新游戏功能。订单管理包括根据游戏查询游戏库存功能、查询用户订单功能和添加游戏库存功能。

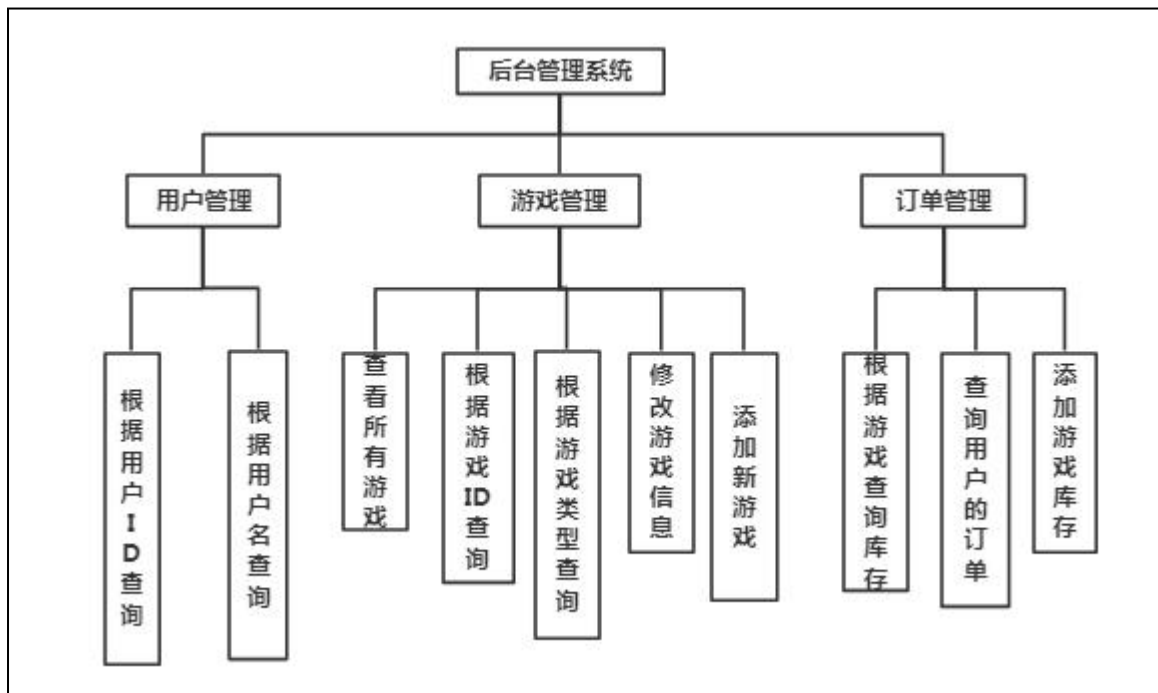


图 4.3 后台管理系统功能图

## 4.4 数据库设计

数据库设计的主要目的是管理数据在数据库中的存储方式以及如何高效地读取并修改大量数据信息。这些功能的实现离不开数据库管理系统（DBMS），本次数据库设计的实现采用了 MySQL 作为数据库管理系统。下面将会从概念模型设计和数据库表的设计两个方面分析。

### 4.4.1 概念模型设计

概念模型为了将服务器端开发中的所需要进行存储的对象存放到数据库中。在数据中建立对象的模型，必须与实际开发中的对象属性保持一致，这样做是为了方便开发者维护数据库和服务器端，形成一个规范的开发流程。

在本设计中，需要存放到数据库中的对象有 3 个，包括用户、游戏以及订单。用户对象需要完成查询校验、增加用户、修改用户信息的操作。游戏对象需要完成增加游戏，修改游戏信息、查询游戏信息等操作。订单对象需要完成查询游戏库存、查询用户订单、

购买游戏等操作。以上的各种操作需要在数据库概念模型设计的过程中就必须考虑到，操作是否符合数据库开发的规范，执行了上述操作之后数据库中的信息是否还能正常使用等各种情况。

确定了对象的建立之后，必须设计对象之间的联系（relationship）。联系指的是对象之间是否存在着某种联系，对一个对象的修改、增加、删除等操作是否会对其他对象产生影响。或者对象本身的属性之间是否还存在着联系。在一个对象中，会设置有一个主键索引，可以通过这个索引找到一条准确的数据记录。此外，开发者还可以设置一些特别的索引，提高数据库系统的查询、修改效率。在本设计中，用户、游戏以及订单对象都有设置主键索引，分别是 UID、GID 以及 RID。此外，因为用户购买了游戏才会产生联系，所以还在订单对象中设置了对用户和游戏的外键（UID，GID）。上述的主键索引都是 unique 索引，可以有效地避免数据信息的冲突。

进行了以上内容的设计之后，开发者就可以清晰地了解到数据库中的对象以及联系。E-R 图是最为常用的表现对象与联系的方法，在本设计中，除了将会对每个对象绘制单独的 E-R 图，还会对整个数据库绘制总 E-R 图。

### 1、E-R 图

整个项目的所有对象的联系如图 4.4 所示，主要对象包括用户对象、游戏对象和订单对象。每个单独的用户可以购买多个游戏因此用户和游戏之间是 1：n 的关系；每个游戏都可以有很多库存，商家可以为游戏添加库存信息，因此游戏与库存之间的关系是 1：n 的关系；用户购买游戏产生订单信息，消耗了一个库存信息，因此将库存信息和订单信息整合成为一个对象。

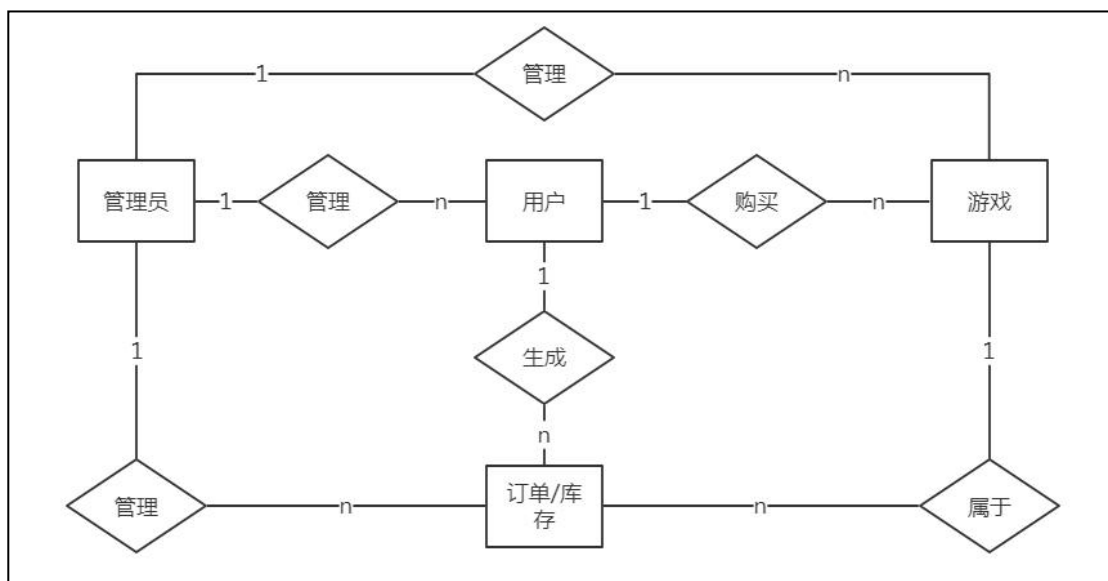


图 4.4 全局 E-R 图

## 2、实体图

如图 4.5 所示，用户实体包括了用户 ID、用户名和用户邮箱三个属性，其中用户 ID 是对象的主键，每个用户对象都有一个唯一的用户 ID。用户名记录了用户的昵称，该属性在一般情况下是唯一的。用户邮箱属性记录的是用户绑定的邮箱，用户可以修改自己绑定的邮箱。

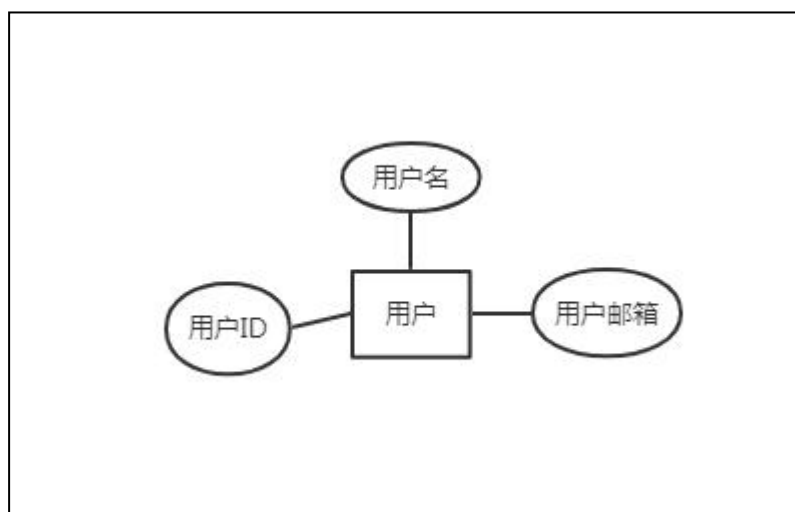


图 4.5 用户实体图

如图 4.6 所示，游戏实体包括了游戏 ID、游戏名称、价格、图片地址、游戏简介和

游戏类型属性。其中游戏 ID 属性是对象主键，每个游戏对象都有一个唯一的游戏 ID。游戏名称属性记录了游戏的名称，如果游戏的名字超过 45 个字符，就采用缩写形式存储。

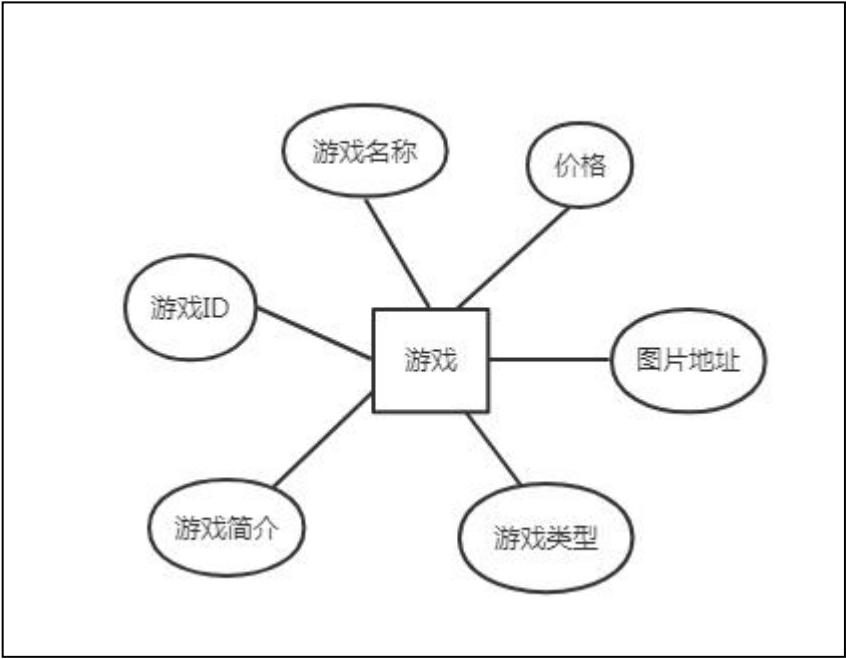


图 4.6 游戏实体图

如图 4.7 所示，订单实体包括了订单号、游戏 ID、用户 ID、购买时间、游戏序列号和序列号使用情况属性。其中订单号是订单对象的主键，每个订单都有一个唯一的订单 ID。游戏 ID 和用户 ID 是订单对象对游戏对象和用户对象的外键索引。游戏 ID 属性不能为空，用户 ID 属性则是在用户购买了该库存之后设置的。

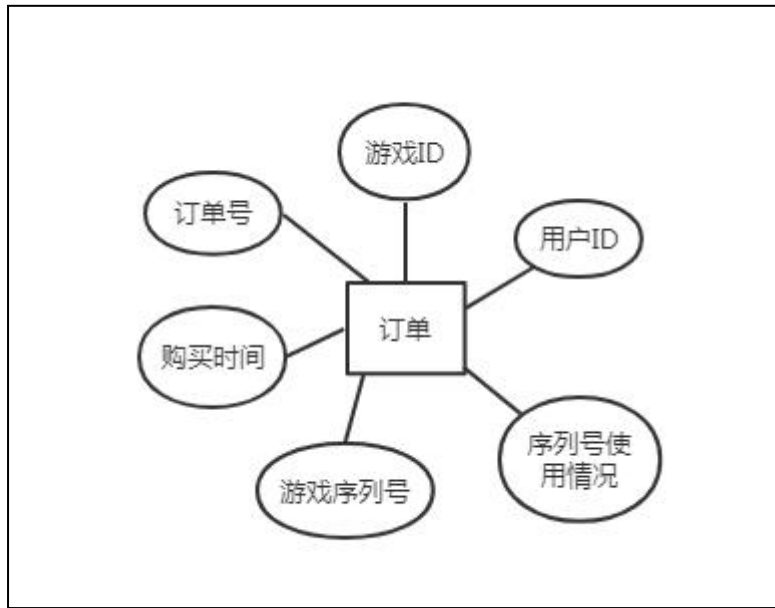


图 4.7 订单实体图

如图 4.8 所示，整个数据库共有三种实体类型，其中订单实体与用户实体和游戏实体都具有关联，当用户购买游戏时，用户信息才会注入到库存中，使之变成订单信息。

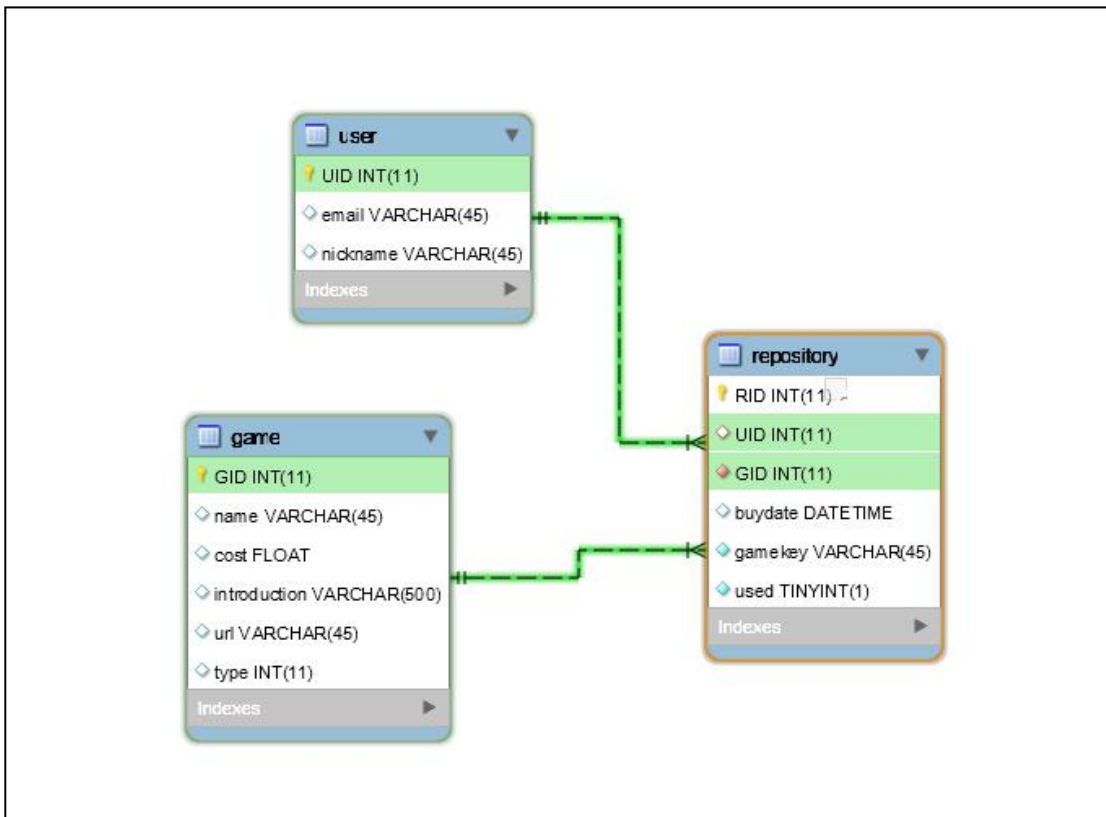


图 4.8 全局实体图

#### 4.4.2 数据库表设计

如表 4.1 所示，user 表的设计内容包含 3 个属性字段分别是用户 ID、用户邮箱以及用户名。其中用户 ID 是对象的主键，每个用户对象都有一个唯一的用户 ID。用户名记录了用户的昵称，该属性在一般情况下是唯一的。用户邮箱属性记录的是用户绑定的邮箱，用户可以修改自己绑定的邮箱。

**表 4.1 用户表设计**

字段名称	数据类型	主键	是否空	说明
uid	INT(11)	Y	N	主键
email	VARCHAR(45)	N	Y	用户邮箱（唯一）
nickname	VARCHAR(45)	N	N	用户名（唯一）

如表 4.2 所示，game 表设计内容包含 6 个属性字段游戏 ID、游戏名称、价格、简介、图片地址以及游戏类型。其中游戏 ID 属性是对象主键，每个游戏对象都有一个唯一的 game ID。游戏名称属性记录了游戏的名称，如果游戏的名字超过 45 个字符，就采用缩写的形式存储。

**表 4.2 游戏表设计**

字段名称	数据类型	主键	是否空	说明
gid	INT(11)	Y	N	主键
name	VARCHAR(45)	N	Y	游戏名（唯一）
cost	FLOAT	N	Y	价格
introduction	VARCHAR(500)	N	Y	简介
url	VARCHAR(45)	N	Y	图片地址
type	INT(11)	N	Y	游戏类型

如表 4.3 所示，repository 表的设计内容包含 6 个属性字段订单 ID、用户 ID、游戏 ID、购买时间、游戏序列号以及序列号使用情况。其中订单号是订单对象的主键，每个订单都有一个唯一的订单 ID。游戏 ID 和用户 ID 是订单对象对游戏对象和用户对象的外键索引。游戏 ID 属性不能为空，用户 ID 属性则是在用户购买了该库存之后设置的。购买时间顾名思义，也是在用户执行购买操作成功之后注入的信息。游戏序列号属性则是最重要的属性值，yoghurt 可以使用订单中的游戏序列号在游戏平台激活游戏。

**表 4.3 订单表设计**

字段名称	数据类型	主键	是否空	说明
RID	INT(11)	Y	N	主键
UID	INT(11)	N	Y	用户 ID（外键）
GID	INT(11)	N	N	游戏 ID（外键）
buydate	DATETIME	N	Y	购买时间
gamekey	VARCHAR(45)	N	N	游戏序列号（唯一）
used	TINYINT(1)	N	N	序列号使用情况



## 5 详细设计与实现

### 5.1 微信小程序的设计与实现

本设计的微信小程序部分是在微信开发者工具中进行开发的，微信小程序页面部分可以提供给微信用户查询和购买游戏，用户登录注册以及游戏订单查询功能。一个优秀项目的前端部分必定是非常精致、条理清晰、页面逻辑丝毫没有繁杂和混乱的。因此前端的设计要点有以下 3 点：简明扼要、条理清晰、层次分明。下面将结合项目的设计情况详细讲解系统的前端部分的设计。

#### 5.1.1 导航栏模块

导航栏模块是位于页面下方的导航栏，其中有 3 个按钮，分别是“游戏列表”、“订单管理”和“用户信息”。点击这三个按钮则会分别显示游戏显示模块、订单显示模块和用户登录模块。

在微信小程序中，需要将相关页面配置添加到 `app.json` 文件中。添加的页面配置必须要在 `pages` 属性中提前设置好，否则会无法启动小程序。另外，微信小程序对配置有导航栏的页面也有特别的限制。当开发者使用 `wx.navigateTo(Object object)` 接口时，无法跳转到配有有导航栏的页面，而且微信开发者工具调试器不会提示错误信息。

关键代码展示：

```
"tabBar": {
  "list": [
    {
      "pagePath": "pages/index/index",
      "text": "游戏列表"
    },
    {
      "pagePath": "pages/order/order",
      "text": "订单管理"
    }
  ]
}
```

```
{
  "pagePath": "pages/user/user",
  "text": "用户信息"
}
]
```

### 5.1.2 游戏显示模块

游戏显示模块主要分成两大部分：分类显示栏和游戏列表显示。

分类显示栏中有 6 个按钮，如图 5.1 所示，“全部”按钮调用了后台的显示全部游戏的接口，服务端会返回游戏列表，前端会将游戏列表设置到页面 Javascript 文件。由于微信小程序是采用动态渲染的形式进行页面的渲染，因此游戏列表显示栏会显示 Javascript 文件中的游戏列表信息。

关键代码展示：

```
<view class="item-content">
  <view class="box">
    <view class="item" wx:for="{{games}}" wx:key="{{item.gid}}"
data-id="{{item.gid}}" bindtap='gamedetail'>
      <image class="pic" src="{{item.url}}"></image>
      <text class="title">{{item.name}}</text>
      <text class="price">¥ {{item.cost}} 元</text>
    </view>
  </view>
</view>
```



图 5.1 全部游戏

另外的“FPS 类”、“棋牌类”、“赛车类”、“动作类”和“其它”按钮点击效果如图 5.2 所示，是预先设置在 Javascript 文件中的。当用户点击这些按钮时就会向服务端发送请求获取对应类别的游戏列表并且在 Javascript 文件中设置，从而令页面进行重新渲染。



图 5.2 FPS 类游戏

### 5.1.3 订单显示模块

当用户点击订单管理导航栏按钮时，页面会跳转到订单显示页面如图 5.3 所示。当用户没有进行登录时，页面会弹出提示“用户没有登录”，提示结束后页面自动跳转到用户登录界面。

关键代码展示：

```
<block wx:for="{{repositories}}">
    <view class="widget">
```

```
<view class="column">{{item.buyDate}}</view>
<view class="column">{{item.name}}</view>
<view class="column">
    <button class="btn" size='mini' type="primary" bindtap="onTap"
data-rid='{{item.rid}}' disabled="{{clickabled}}">发送邮件</button>
</view>
</view>
</block>
```

判断用户是否已经登录，需要在页面的 onshow 或者 onload 阶段进行判断用户是否已经登录。由于用户登录信息存放在 App.js 文件中，判断登录信息需要 getApp().globalData 中 user 对象是否存在。



图 5.3 订单管理页面

#### 5.1.4 用户登录

用户在用户信息界面(如图 5.4 所示)可以看到有两个按钮,分别是“用户登录”与“绑定邮箱”。用户点击登录按钮之后首先会先微信服务器发送请求,通过 `wx.getUserInfo()` 接口获取用户微信信息,紧接着小程序会将相关的用户信息发送至服务端进行用户校验。若用户在数据库已经注册过,则返回用户的详细信息,若没有用户记录,则会返回错误信息。

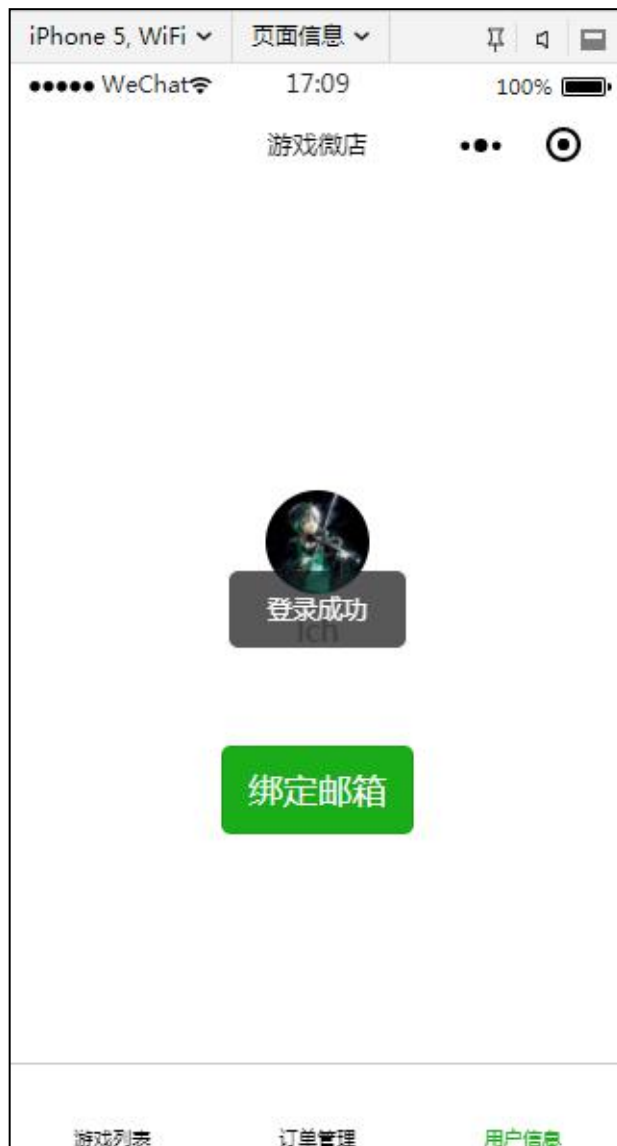


图 5.4 用户信息界面

### 5.1.5 用户修改邮箱

点击了用户信息页面的“绑定邮箱”按钮后，页面进行跳转到用户修改信息页面。用户按照页面中的提示进行输入用户邮箱，输入完成后点击绑定邮箱用户即可完成邮箱的绑定或修改。

该字段输入有输入校验，只有输入的字符串是正确的邮箱地址时，按钮才会亮起并进行绑定操作。



图 5.5 绑定邮箱界面

### 5.1.6 发送邮件

在订单管理界面，如果用户登录成功之后并且用户已经绑定了邮箱，则用户点击每条订单信息后的发送邮件按钮。服务端会读取用户的订单信息和用户邮箱，发送当前的订单信息到用户邮箱。发送成功之后前端会收到发送成功的回显（图 5.6）。

实现该功能的技术难点主要在数据库中订单信息和用户信息的联合查询并返回到服务器中，以及服务器中邮件发送模块内容填充。

其中订单信息的查询和用户信息的查询从数据库读取了两次数据。对服务端的稳定性和响应速度有一定的要求，而且返回到服务器中的数据是一个 Java 对象，需要将对象转变成邮件的内容。



邮件的发送需要使用 Spring Boot 推荐使用的 Thymeleaf 模版引擎。首先在服务端使用 Thymeleaf 写一个邮件的模版。通过模版引擎的动态填充数据功能完成邮件的内容，接着使用 Spring 的 mail 功能，将内容发送到用户的邮箱中（图 5.7）。



图 5.6 发送邮件界面



图 5.7 邮件接收示意图

## 5.2 后台管理系统的设计与实现

小程序后台管理系统本就是设计出来为管理员或者商家方便管理系统中的各项数据的，而本设计中的后台管理系统就包括了三大模块：用户管理模块、游戏管理模块以及订单管理模块。

### 5.2.1 整体结构

管理员打开管理员系统后显示的页面（图 5.8），主要分成三个部分：上部、导航栏组件以及页面主体。这里主要讲解上部和页面主体的设计与实现。

总体的结构采用的 Vuejs 官方提供的 container 布局。上部的设计参考了众多后台管理系统的设计，采用了简单明了的设计风格。令管理员可以专注在业务的管理上。

页面主体默认是打开导航栏中 Home 选项的界面，该页面存放的是系统的公告。每当系统出现更新或者数据库有更新的时候，系统维护人员可以在此处放置系统公告，以提醒管理员需要注意。



图 5.8 整体结构图

### 5.2.2 导航栏组件

如图 5.9 所示，导航栏将后台管理系统的功能分成三大部分：用户管理、游戏管理以及库存管理。除此之外当然还有主页按钮。

其中导航栏默认是收起状态，点击一级标题可以展开或收起二级标题。当选项被选中时，字体会变为黄色。

导航栏是通过动态渲染实现的，渲染的信息从 `config` 文件加载。说到这里需要提及一个 `vue` 的组件：`vue-router`。该组件根据地址栏的修改动态渲染 `router-view`，从而实现导航栏的功能。

关键配置展示：

```
<el-menu-item index="home">HOME</el-menu-item>

  <el-submenu v-for="item in menu" :index="item.id" :key="item.id">
    <template slot="title">
      <span v-text="item.name"></span>
    </template>
    <el-menu-item-group class="over-hide" v-for="sub in
item.sub" :key="sub.componentName">
      <el-menu-item :index="sub.componentName"
v-text="sub.name"></el-menu-item>
```

```
</el-menu-item-group>
</el-submenu>
</el-menu>
```



图 5.9 导航栏结构图

### 5.2.3 单行查询表单

如图 5.10 所示,通过 Vuejs from 标签的 form-inline 属性设置表单为行内显示。此外,表单的请求地址和属性可以通过组件的参数传递动态产生,灵活性极高。

用户名	<input type="text" value="用户名"/>	<input type="button" value="查询"/>
-----	----------------------------------	-----------------------------------

图 5.10 单行查询表单

5.2.4 用户信息修改表单

如图 5.11 所示，该表单拥有三个字段，分别是：“id”，“用户名”以及“邮箱地址”。“ID”字段是不可操作的，“用户名”可以修改，但是可能会造成小程序用户无法登陆的情况。

id

1

用户名

lch

email

1010654308@qq.com

确认

图 5.11 用户信息修改表

5.2.5 游戏列表显示

在查询所有游戏界面，页面加载时向服务端发送请求获取游戏列表。  
游戏列表中的列名与数据库表中的游戏表设计属性相同（图 5.12）。

id	游戏名	价格	简介	url	type
1	雷神之锤	10	古老的pfs游戏	../image/ceMsrPRDccJm.jpg	1
2	Sleengster	1	已从steam中下架，但仍可使用	../image/qwg3s13gh34.jpg	0
7	E.Z	3.99	E.Z是一个简单构成的硬核平台游戏。游戏中100关的目标是从关卡的一边到达另一边，而每一次死亡都让你回到前面的关卡。	../image/2018-7-5-9423-E.jpg	4
8	炸弹塔防	3.5	完全不同的塔防类型游戏。它将塔防与益智元素结合起来，是一款难度较大的游戏。	../image/2018-7-5-7725-BombDefense.jpg	2

图 5.12 游戏列表显示

5.2.6 游戏信息修改表单

如图 5.13 所示，该表单中有 5 个字段：“id”、“游戏名”、“价格”、“简介”、“图片地

址”以及“游戏类型”。其中“id”属性不可被修改，“图片地址”这一项数据比较繁杂，这是因为数据存储图片地址时路径比较复杂。

id

1

游戏

雷神之锤

价格

10

简介

古老的pfs游戏

url

../image/ceMsrPRDccJm.jpg

type

1

确认

图 5.13 游戏信息修改表单

5.2.7 订单信息列表显示

如图 5.14 所示，当用户使用行内查询表单查询后，页面会显示出按照用户查询的订单信息或者按照游戏查询的库存信息。

RID	UID	GID	游戏名	购买时间	gamekey	used
1	1	1	雷神之锤	2019-03-30 23:19:15	eqwrq-wbsdf-ewqly	1
8	1	2	Sleengster	2019-03-31 23:13:40	WRTW-JTV9OW-CCDLD	1
9	2	9	神秘矿井	2019-02-11 23:14:00	TBKUB-CUY8W-311CW	1
10	1	22	Rento Fortune - Online	2018-11-03 23:14:21	请联系管理员	1

图 5.14 订单信息列表显示

5.2.8 订单信息修改表单

如图 5.15 所示，该表单拥有 7 个字段，分别是：“RID”、“UID”、“GID”、“游戏名”、

“购买时间”、“游戏序列号”以及“序列号使用情况”。“RID”、“GID”、“序列号使用情况”字段是不可操作的，其中“UID”和“购买时间”在用户购买后自动通过用户信息表进行注入，使库存变为了订单。

RID

1

UID

1

GID

1

name

雷神之锤

buydate

2019-03-30 23:19:15

gamekey

eqwrg-wbsdf-ewqly

used

1

确认

图 5.15 订单信息修改表单

## 6 运行与测试

### 6.1 测试目的

在系统进行正式上线之前，需要对系统进行一次全面的测试。从而找出系统中可能储存在的 BUG。这些程序漏洞包括程序设计是没有考虑到的异常情况，或者是程序在生产环境中可能会发生的未知问题。

通过测试，利用发现的 Bug 整理出一份详细的报告，能够为系统往后的改进发展提供一个标准和参考。也能为开发人员再次开发此类软件积累宝贵的经验。

### 6.2 测试方法

系统的测试方法可以按照不同方法进行分类。从测试者的角度出发，测试人员是否把关注点放在系统内部的程序实现则是可以分成黑盒测试和白盒测试，黑盒测试是指测试人员对软件的程序实现过程不关心，而是关注程序的最终实现结果，白盒测试则是程序的实现过程对测试人员透明，实现过程中出现的问题也会被发现。从系统开发的每个阶段来进行细分，可以大致分为单元测试、集成测试、确认测试、系统测试、验收测试等测试方法。

下面将会从系统开发阶段这个方面来对本设计的所涉及到的测试方法进行详细的说明。

#### 6.2.1 单元测试

单元测试是系统开发的第一个阶段，设计对程序模块进行校验，查看程序是否能够达预期中的结果，这里所说的程序模块是系统设计中最小的单位，也就是 Java 编程中的一个函数或者一整个类（class）。

进行单元测试的目的在于发现在每个程序模块中可能会出现某些错误。这就要求开发者在开发过程中对方法的功能实现与程序逻辑有清楚认识和把控。

在程序编译器中，Java 可以很容易地进行单元测试。这里就必须介绍 Junit 框架。它完美地与 Java 结合，能够为每个模块进行测试，还有专用的接口对程序某个阶段或者



整体的参数进行实时的监控，可以让开发者清楚知道如果程序出现了错误是从哪一个部分引发的。

### 6.2.2 集成测试

集成测试是系统开发的第二个阶段，是将已经通过了单元测试的程序模块连接在一起，查看之间的信息交流是否出现错误以及集成出来的程序是否可以完成预期中的结果。这个时候测试的主要关注点有 3 点。分别是在连接所有单元模块的时候需要关注模块之间的参数传递是否成功，参数是否会发生丢失；模块运行的时候是否会对其他模块产生无法预料的影响，造成程序出现错误；将多个单元模块结合起来后是否达到预期之中的结果，实现要求的功能。

在集成测试中，采用的是传统的在代码块中生成断点，在 `debug` 模式中运行代码的方式完成测试。

### 6.2.3 系统测试

系统测试是系统已经确认完成，确保每个部分都能够完成需求。然后将系统从开发环境部署到生产环境中。生产环境通常与开发环境有一定的差距，主要表现在系统的区别，软件环境比如说 `tomcat` 服务器的版本、`Java` 的版本等。将软件放置在生产环境中可以检测出软件是否与生产环境中的计算机硬件、某些驱动软件、数据在硬件中的储存方式与稳定性、维护人员是否能够完成维护工作等。

本项目实行系统测试的方式是依靠微信开发者工具的控制台显示信息和 `Google Chrome` 的开发者调试工具进行系统测试。

### 6.2.4 验收测试

验收测试是系统开发过程中的最后一个测试阶段，将系统交付给甲方，由甲方的测试人员来对系统的整体需求进行确认，确认系统是否完成了约定的需求。验收测试对于其他测试阶段来说，最大的不同就是以用户为主的有效性测试。测试用户可以从用户的角度对软件系统进行中肯的评判。测试中除了对软件系统的功能与性能之外，还会对软件系统的兼容性、界面的用户友好程度，可维护性、对程序错误的恢复能力等进行完整

地确认。

本项目的验收测试是将小程序展示给指导老师，还有推荐给周边的同学，听取测试用户的宝贵意见，对系统进行一系列的修改。

## 6.3 测试流程

本项目的测试部分为服务端的 API 接口，微信小程序和后台管理系统都会调用 API 进行功能的实现。

测试分为两部分：DAO 层的测试与 Service 层测试：

其中 DAO 层是服务端与数据连接，进行数据库操作的层次，进行的操作包括添加记录、查询信息、修改记录等。而 Service 层是进行业务逻辑处理的层次，是整个服务端编写的核心，也是重中之重。不仅封装了复数的 DAO 层操作，而且还天啊及了业务逻辑所需要的各种数据读写，邮件发送的服务。因此，测试流程将主要讲解在 Service 层测试的过程。

### 6.3.1 用户测试

该模块是用户在业务层可能出现的各种需求，根据需求分析可以知道，在该层的业务主要包括通过用户 ID 查询用户信息、通过用户名查询用户、添加新的用户、修改已经存在的用户信息。

#### 1、queryUserByID

(1) 功能：根据用户 ID 查找用户

(2) 输入：用户 ID 为 1

(3) 预期结果：获得数据库中 id 为 1 的用户信息

(4) 实际结果：获得（id: 1, 用户名: lch, 用户邮箱: “1010654308@qq.com”）的记录

#### 2、queryUserByName

(1) 功能：根据微信用户名查找用户

(2) 输入：用户 ID 为“lch”

(3) 预期结果：获得数据库汇总 id 为 1 的用户信息

(4) 实际结果：获得（id: 1, 用户名: lch, 用户邮箱: “1010654308@qq.com”）的记录

### 3、insertUser

(1) 功能：添加新的用户记录。所需要数据包括（用户名，用户邮箱）

(2) 输入：一个用户对象（用户名: “xxxx”, 用户邮箱“xxxx.qq.com”）

(3) 预期结果：插入成功，数据库返回收到操作影响数据数量到 DAO 层，Service 层判断数量是否大于等于 1

(4) 实际结果：插入成功，数据返回“1”，与预计的结果相同

### 4、updateUser

(1) 功能：修改用户信息要求参数为一个完整的用户对象

(2) 输入：用户对象（id: 1, 用户名: “lch”, 用户邮箱: “123123123@qq.com”）

(3) 预期结果：修改用户信息成功，数据库返回收到操作影响数据数量到 DAO 层，Service 层判断数量是否大于等于 1

(4) 实际结果：修改用户信息成功，数据返回“1”，与预计的结果相同

## 6.3.2 游戏测试

该模块是游戏对象在业务层可能出现的各种需求，根据需求分析可以知道，在该层的业务主要包括查询所有游戏、通过游戏 ID 查询游戏信息、通过游戏类型查询游戏信息、添加新的游戏信息、修改已经存在的游戏信息。

### 1、queryGames

(1) 功能：查询所有游戏记录

(2) 输入：不需要输入数据

(3) 预期结果：获得所有游戏信息，返回一个 List 对象

(4) 实际结果：获得一个 List 对象，对象大小和数据库记录数量相同

### 2、queryGameByID

(1) 功能：根据游戏 ID 查找游戏记录

(2) 输入：游戏 ID 为 1

(3) 预期结果：获得数据库中游戏 id 为 1 的游戏信息

(4) 实际结果：获得雷神之锤游戏的详细信息与预期结果相同

### 3、queryGameByType

(1) 功能：根据游戏类型（type）查找所有游戏记录

(2) 输入：游戏类型 type 为 1

(3) 预期结果：获得数据库中游戏 type 为 1 的所有游戏信息，返回一个 List 对象

(4) 实际结果：获得数据库中游戏 type 为 1 的所有游戏信息，返回一个 List 对象，

List 对象大小和数据库记录数量相同。

### 4、updateGame

(1) 功能：修改游戏信息，要求参数为一个 Game 对象

(2) 输入：首先获取 ID 为 7 的游戏 Game 对象，将价格提高一元

(3) 预期结果：若修改成功，则返回 true

(4) 实际结果：修改成功，返回 true，并且查看数据库中信息已经发生了修改

### 5、insertGame

(1) 功能：添加新的游戏信息，要求参数为一个 Game 对象

(2) 输入：首先获取 ID 为 2 的游戏 Game 对象作为模版，修改游戏名称为“测试游戏”

(3) 预期结果：若添加成功，则返回 true

(4) 实际结果：添加成功，返回 true，并且查看数据库中信息已经发生了修改，微信小程序中也可以查询到新的游戏

## 6.3.3 订单测试

该模块是订单对象在业务层可能出现的各种需求，根据需求分析可以知道，在该层的业务主要包括通过订单 ID 查询订单信息、通过用户查询订单信息、通过游戏查询库存信息、将库存转换为订单信息、将订单转换成库存信息（系统出错管理员进行回滚）、添加新的库存、发送订单信息给用户。

### 1、queryByID

(1) 功能：根据订单 ID 查找订单记录

(2) 输入：订单 ID 为 1

(3) 预期结果：获得数据库中订单 id 为 1 的订单信息

(4) 实际结果：获得数据库中订单 id 为 1 的订单信息

## 2、queryByUser

(1) 功能：根据用户 ID 查找订单记录

(2) 输入：用户 ID 为 1

(3) 预期结果：获得数据库中所有用户 id 为 1 的订单信息

(4) 实际结果：获得数据库中所有用户 id 为 1 的订单信息，与数据库中记录数量相

同

## 3、queryByGame

(1) 功能：根据游戏 ID 查找订单记录

(2) 输入：游戏 ID 为 2

(3) 预期结果：获得数据库中所有游戏 id 为 2 的订单信息

(4) 实际结果：获得数据库中所有游戏 id 为 2 的订单信息，与数据库中记录数量相

同

## 4、updateRepository

(1) 功能：用户购买游戏时，将没有使用的库存通过信息注入变为订单数据

(2) 输入：用户 ID 为 1，游戏 ID 为 2（表示用户 1 购买了游戏 2）

(3) 预期结果：若修改成功，则返回 true

(4) 实际结果：修改成功，返回 true，并且查看数据库中信息已经发生了修改

## 5、remakeRepository

(1) 功能：用户购买游戏后如果想进行退款或者购买过程中出现了问题，可以回滚订单信息

(2) 输入：订单号 8（该订单已经被使用）

(3) 预期结果：若修改成功，则返回 true

(4) 实际结果：修改成功，返回 true，并且查看数据库中信息已经发生了修改

## 6、insertRepository

(1) 功能：为游戏添加库存，要求参数包括（GID，游戏序列号 gamekey）

(2) 输入：使用未被使用的 RID 为 2 的库存作为模版。修改了 gamekey 作为测试数

据

(3) 预期结果：若添加成功，则返回 `true`

(4) 实际结果：添加成功，返回 `true`，并且查看数据库中信息已经发生了修改，微信小程序中也可以查询到新的订单记录

#### 7、sendRepository

(1) 功能：发送订单信息，从订单的 `UID` 查找用户信息，发送到用户信息中的邮箱

(2) 输入：订单号 8（已经被使用的订单）

(3) 预期结果：用户邮箱可以看到新的邮件，邮件中的游戏序列号可以被激活

(4) 实际结果：用户邮箱可以看到新的邮件，邮件中的游戏序列号可以被激活

### 6.4 测试反馈

编写起始代码越好，它会产生错误的错误就越少，那么就相信所做的测试是彻底的。系统测试旨在以有序的方式检测潜在的麻烦位置。同样，问题最有可能发生在边界，可以手动或以编程方式检查。它是自动测试的理想选择。回归测试检查程序是否可以生成与以前相同的输出。经过一些小改动后进行测试是一种很好的技术，可以帮助定位问题的范围，因为新问题通常出现在新代码中。

测试和故障排除通常被认为是一个阶段，实际上它们根本不是一回事。简而言之，当程序员已经知道程序有问题时，就会执行故障排除。测试是在认为程序正常运行时，并且在已经知道程序出现问题时进行故障排除。

## 结论

此项目是一个完整的前后端共同开发的项目，通过开发本毕业设计项目。在开发的过程中，全面掌握了 JAVA 的基础知识和编程技巧、前端开发的工作流程等。服务端开发过程中犯过很多大大小小的错误，这些错误很多是对技术与项目缺乏全面的、详细的认识。这个错误都能为以后提升自己的开发能力提供了宝贵的经验。

在开发过程中，还深刻的意识到在系统开发之前需求分析，可行性分析等分析工作十分重要。它会严重影响到后续系统开发中是否会有无法预料的结果，造成整个项目开发进度受到影响，甚至是无法完成项目。因此，在项目开发之前，不应急于编程。而是应该对项目的各种需求进行分析，对如何实现这些功能找到可用的、符合项目发展周期的有效技术，然后才是开始编写程序代码。

另外，也觉得在这个项目的设计与开发上走了很多弯路。主要是因为与软件开发工作的联系很少，缺乏应用经验，缺乏对前后端结合开发项目的经验，未来还需要更多的努力。

这个设计本身的影响远未结束。通过自己的学习和努力以及通过教师的指导和教育，不仅大大提高了解决实际问题的知识水平和能力。还深刻地认识到，将自己所想所学一一实现的时候，则必须面对在这个过程中可能会遇到的种种困难。

该项目设计不免会有本人无法认识到的错误以及需要改进的地方，希望各位师长、技术前辈提出宝贵的意见。

## 参考文献

- [1] 李健,田立锋,郭瑞.Java SSH 框架在 Java Web 中的应用[J].电脑与电信,2016(03):69-72.
- [2] Gupta A . Java EE 7 精粹[M]. 人民邮电出版社, 2015:37-40.
- [3] TimLindholm. The Java virtual machine specification[M]. 电子工业出版社, 2016:46-51.
- [4] PaulDeck. Spring MVC 学习指南[M]. 人民邮电出版社, 2015:67-60.
- [5] 刘红卫. 微信小程序应用探析 [J]. 无线互联科技, 2016(23):11-12+40.
- [6] Gutierrez F . Spring Boot [M]. Spring Boot Messaging. Apress, 2017:73-75.
- [7] 徐雯,高建华.基于 Spring MVC 及 MyBatis 的 Web 应用框架研究[J].微型电脑应用,2012,28(07):1-4+10.
- [8] 朱亚兴,余爱民,王夷.基于 Redis+MySQL+MongoDB 存储架构应用[J].微型机与应用,2014,33(13):3-5+9.
- [9] 麦冬, 陈涛, 梁宗湾. 轻量级响应式框架 Vue.js 应用分析[J]. 信息与电脑(理论版), 2017(7):58-59.
- [10] 李胜.基于 JavaScript 的面向对象语言特性及其程序设计[J].科技与创新,2019(06):112-113+116.



## 致 谢

本系统的设计和论文的编写是在指导教师张海笑老师的亲切关怀和悉心指导下完成的。

在题目的选择上感谢张海笑老师给予的引导和帮助；在系统的完成过程中，非常感谢张海笑老师的指导，对系统功能的引导，逻辑的纠正，张海笑老师给予了极大的帮助；在论文的完成阶段，更是感谢张海笑老师的耐心指导和修正。在此向张海笑老师表示深深的感谢和崇高的敬意。

最后再次感谢在开发系统和编写毕业论文过程中给予本人帮助的所有人以及提供参考文献的所有人，感谢大家的帮助和支持。