WeChair系统设计说明书

文档修订控制

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 版本号 | 修订日期 | 修改概述 | 修订人 | 审批人 | 备注 |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

目录

1 引言

1.1 编写目的

此系统设计说明书将全面介绍WeChair小程序系统的设计方案，旨在给读者全面地分析此小程序的功能、架构及安全性的问题，为预期成果打下理论基础。面向读者为参与审核小程序的老师、专业投资者或投资公司、团队成员等专业性较强的个体或团体。

1.2 背景

近年来，随着高校招生规模的不断扩大、社会竞争的日益激烈，越来越多的学生为了应付各类考试而不断涌向图书馆看书备考，以提高自己的学习能力。图书馆为了更好地配合学校的教学工作，完善读者服务，面临着多方压力。尤其是学生占座问题，成为困扰高校图书馆的老大难问题。因此实现座位资源计算机网络化管理，对图书馆的座位进行合理调配，提高座位资源的利用率，这才是解决占座问题的关键。

“WeChair”是一款基于微信小程序的座位物联管理系统，主要原理是通过座位码物联图书馆每一张座位。用户仅需使用微信扫码直接获取每一张座位的使用信息，就可以知道该座位的使用情况。与传统技术相比，操作流程简单，不需要下载APP，不需要绑定公众号，不需要输入账号密码，不泄露个人隐私信息，交互双方座位信息同步，信息完全对等，直击占座问题要点。不仅有效的减少了学生的占座现象，同时也提高了图书馆资源利用率。以下将介绍“WeChair”的具体系统结构设计。

目前WeChair项目已经处于初级阶段，整体的设计、布局、架构已经准备就绪，前端、后端、数据库负责人都已经投入编写。此前在“软件工程”课堂上，团队成员有一起协同完成一个信息录入系统网站，这将给我们接下来的工作起到了不可或缺的作用，接来下我们将围绕系统设计说明书对我们的小程序进行制作与完善。由于我们是课程团队，可靠性和专业性可能达不到预期要求，但我们将努力做到最好，希望读者若发现不足能够及时提出宝贵意见。

1.3 参考资料

**参考文件：**

**技术标准：（微信小程序设计指南）**

https://developers.weixin.qq.com/miniprogram/design/#%E5%8F%8B%E5%A5%BD%E7%A4%BC%E8%B2%8C

**参考资料：**

[1]徐家骀，白忠建，吴磊，软件工程——理论与实践（第二版），[J].高等教育出版社，2009.10.

[2]栾跃，软件开发项目管理[M]，上海，上海交通大学出版社，2015，20-34.

[3]段琳琳，敏捷方法在需求工程中的研究与应用[D].湖南大学，2018.1.

**查阅资料：**

# [1]微信小程序设计规范总结：https://www.jianshu.com/p/78c93889410f

# [2]做一个微信小程序的完整流程：https://blog.csdn.net/bryant0917/article/details/82470433

# [3]小程序文档整理之——开放接口：https://blog.csdn.net/bobobocai/article/details/77651566

**[4]详解设计模式六大原则：**[**https://www.cnblogs.com/toutou/p/4870926.html**](https://www.cnblogs.com/toutou/p/4870926.html)

1.4 术语

列出本系统设计说明书中专门术语的定义和英文缩写词的原词组。

|  |  |  |
| --- | --- | --- |
| 术语名称 | 术语含义 | 备注 |
| 应用程序接口 | 预先定义的[函数](https://baike.baidu.com/item/%E5%87%BD%E6%95%B0" \t "/Users/jayjerry/Documents\\x/_blank)或指软件系统不同组成部分衔接的约定 | Application Programming Interface |
| 单一职责原则 | 应该有且仅有一个原因引起类的变更 | Single Responsibility Principle |
| 里氏替换原则 | 任何基类可以出现的地方，子类一定可以出现 | Liskov Substitution Principle |
| 依赖倒置原则 | 程序要依赖于抽象接口，不要依赖于具体实现 | Dependence Inversion Principle |
| 接口隔离原则 | 不出现臃肿的接口 | [Interface Segregation Principle，ISP](https://www.cnblogs.com/toutou/p/4870926.html#_label4) |
| 迪米特法则 | 一个软件实体应当尽可能少的与其他实体发生相互作用 | Law of Demeter, LOD |
| 开闭原则 | 模块应尽量在不修改原代码的情况下进行扩展 |  |
| WXML | 是框架设计的一套标签语言 | WeiXin Markup Language |
| WXSS | 一套样式语言，用于描述 WXML 的组件样式 | WeiXin Style Sheets |

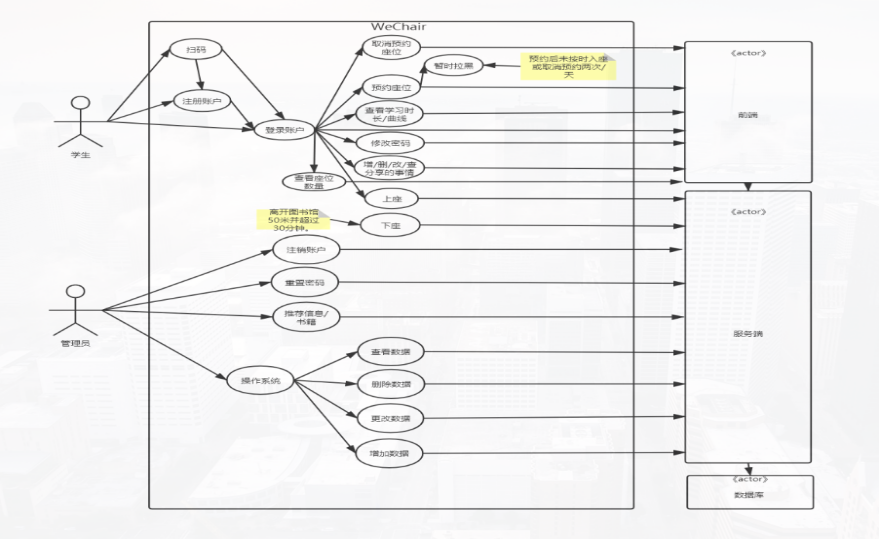
2 总体设计

2.1 系统运行环境

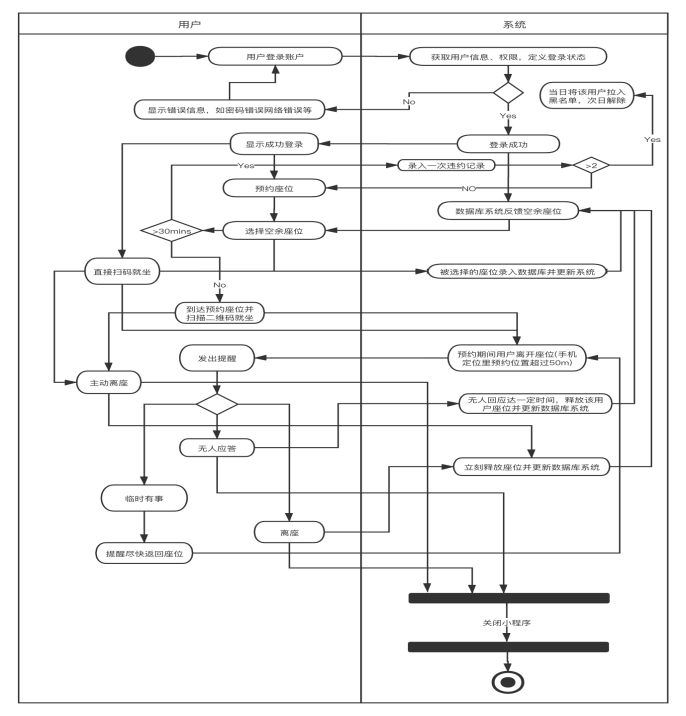
给出本软件系统运行所需的基本软/硬件环境、使用本软件系统典型用户的设备分布图及设备上相应软件配置。软件环境包括：操作系统、数据库、以及其它支撑软件；硬件环境包括：主机类型、网络类型、存储器容量、其它特殊设备。

2.2 系统设计模型

用例图（待修改，系统和数据库设计完成详细绘制）



活动图（待修改）



2.3 软件功能描述

描述被开发软件的功能，如有同等作用的文件（如已编写的《软件功能规格说明书》）则可直接在此引述该文件名及归档的部门即可。

2.4 软件结构

以图表形式给出软件子系统结构和模块组成结构。凡单独编写《系统设计报告》的子系统，其内部自用的模块不必列出。

2.5 技术路线

给出本软件系统在实现过程中所采用的技术路线、方法。对比较新的技术给予详细的阐述及必要的说明，在可能的情况下给出相应的技术参考资料。

2.6 外部接口

2.6.1 用户接口（界面原型）

描述该软件与用户的接口，包括用户输入输出界面样式、操作方式和界面间的转移关系。如具有等同作用的文件（如使用开发工具编写的界面原型程序等），本处仅需列出该文件名、保管部门即可。

2.6.2 软件接口

描述该软件与其它系统的软件接口，主要说明其它系统提供的编程接口和本系统向其它系统提供的编程接口，如具有等同作用的文件（如包含接口说明及注释的源文件程序头文件、接口帮助的超文本文件等），本处仅需列出该文件名、保管部门即可。

2.6.3 其它接口

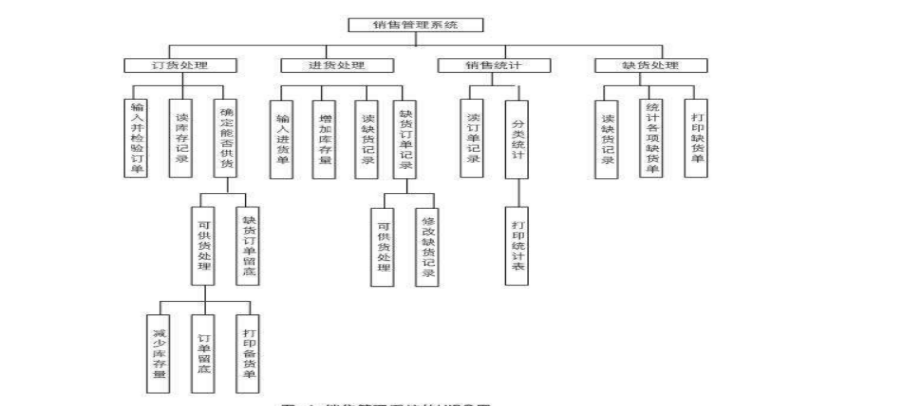
描述该软件与其它系统的其它类型接口，主要说明其它系统提供的和本系统向其它系统提供的硬件接口、通信协议及规程、数据交换格式等，如具有等同作用的文件，本处仅需列出该文件名、保管部门即可。

3 功能模块层次设计模块设计

3.1 功能

3.1.1 简述各模块的功能。

3.1.1功能模块层次设计+功能模块层次图（10′）

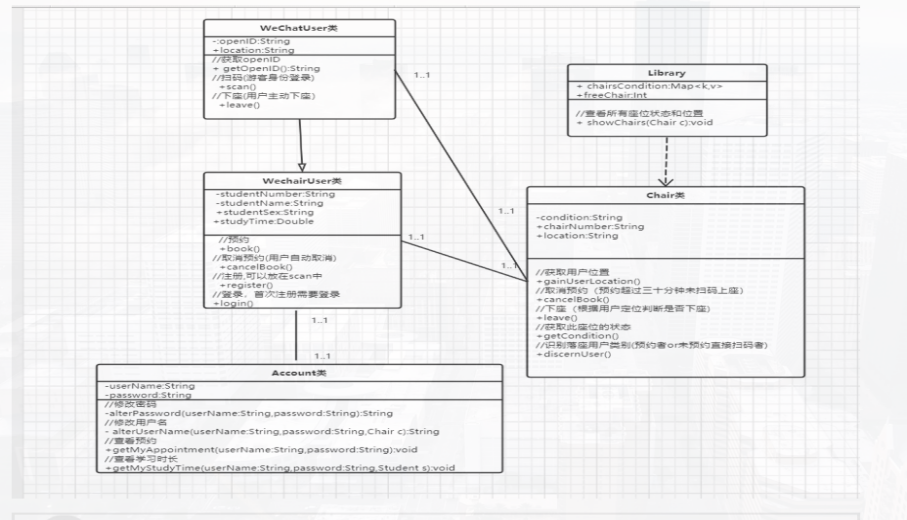


3.1.2设计类图（10′）

类图待修改：

<https://www.cnblogs.com/LangZXG/p/6208716.html>

<https://blog.csdn.net/loumoxiaozi/article/details/81036408>



六个模块每个模块画出较清晰的类图：

预约座位->闪座

学时排行

时间沙漏

预约

抢座(没有座位才能进行抢座)

我的

3.2 性能

3.2.1 说明对各模块的性能要求，包括精度、时间特性和处理速度。

3.3 内部接口

接口设计（10′）

<https://www.cnblogs.com/toutou/p/4870926.html>

[接口设计六大原则](https://www.cnblogs.com/sweet6/p/10851768.html)

一.单一职责原则

Single Responsibility Principle, 简称SRP。

定义：There should never be more than one reason for a class to change.

应该有且仅有一个原因引起类的变更。

职责的划分？单一的定义和级别？

应该根据实际业务情况而定。关注变化点。

实际使用时，类很难做到职责单一，但是接口的职责应该尽量单一。

二.里氏替换原则

Liskov Substitution Principle, 简称LSP。

定义：Functions that use pointers or references to base classes must be able to use objects of derived classes without knowing it.

(所有引用基类的地方必须能透明地使用其子类的对象)

里氏替换原则为良好的继承定义了一个规范：

1.子类必须完全实现父类的方法

2.子类可以有自己的个性（属性和方法）。

3.覆盖或实现父类的方法时输入参数可以被放大。

4.覆写或实现父类的方法时输出结果可以被缩小。

注：在类中调用其他类时务必要使用父类或接口，如果不能使用父类或接口，则说明类的设计已经违背了ＬＳＰ原则。

三.依赖倒置原则

Dependence Inversion Principle, 简称DIP

定义：High level modules should not depend upon low level modules.Both should depend upon abstractions.Abstractions should not depend upon details.Details should depend upon abstractions.

翻译过来，包含三层含义：

1.高层模块不应该依赖低层模块，两者都应该依赖其抽象。

2.抽象不应该依赖细节。

3.细节应该依赖抽象。

精简的定义： 面向接口编程。

Test-Driven Development 测试驱动开发是依赖倒置原则的最好体现。

测试驱动开发要求先写测试类，测试通过才写实现类，这就要求你要先想接口定义。

依赖的三种写法：

1.构造函数传递依赖对象。

2.Setter方法传递依赖对象。

3.接口声明依赖对象。

最佳实践：  
1.每个类尽量都有接口或抽象类，或者抽象类和接口两者都具备。  
2.变量的表面类型尽量是接口或抽象类。  
3.任何类都不应该从具体类派生。  
4.尽量不要覆写基类的方法。  
5.结合里氏替换原则使用。

四.接口隔离原则：  
接口--这里指用interface关键字定义的接口。  
定义：  
1.Clients should not be forced to depend upon interfaces that they don't use.(客户端不应该依赖它不需要的接口)  
2.The dependency of one class to anther one should depend on the smallest possible interface.(类间的依赖关系应该建立在最小的接口上)

概括：建立单一接口，不要建立臃肿庞大的接口。

通俗来讲：接口尽量细化，同时接口中的方法尽量少。

如何细化？细化到什么程序？

没有统一的标准，应根据业务合理细分，适合业务才是重点。

保证接口的纯结性：  
1.接口要尽量小。  
2.接口要高内聚。  
3.定制服务。  
4.接口的设计是有限度的。

最佳实践：  
1.一个接口只服务于一个子模块或业务逻辑。  
2.通过业务逻辑压缩接口中的public方法，接口时常去回顾，尽量让接口达到“满身筋骨肉”，而不是“肥嘟嘟”的一大堆方法。  
3.已经被污染了的接口，尽量去修改，若变更的风险较大，则采用适配器模式进行转化处理。  
4.了解环境，拒绝盲从。每个项目或产品都有特定的环境因素，不要盲从大师的设计，要根据业务逻辑进行最好的接口设计。

五.迪米特法则  
Law of Demeter, LOD。又称最少知识原则（Least Knowledge Principle, LKP）。  
通俗来讲：一个类应该对自己需要耦合或调用的类知道得最少，你（被耦合或调用的类）的内部是如何复杂都和我没有关系，那是你的事情，我就调用你提供的public方法，其他一概不关心。

低耦合要求：  
1.只和朋友交流  
朋友类：出现在成员变量、方法的输入输出参数中的类。方法体内部的类不属于朋友类。  
2.朋友间也是有距离的  
迪米特法则要求类“羞涩”一点，尽量不要对外公布太多的public方法和非静态的public变量，尽量内敛，多使用private、package-private、protected等访问权限。  
3.是自己的就是自己的  
如果一个方法放在本类中，既不增加类间关系，也对本类不产生负面影响，就放置在本类中。  
4.谨慎使用Serializable

六.开闭原则  
Software entities like classes, modules and functions should be open for extension but closed for modifications.（一个软件实体如类、模块和函数应该对扩展开放，对修改关闭）

软件实体包括以下几个部分：  
1.项目和软件产品中按照一定的逻辑规则划分的模块。  
2.抽象和类。  
3.方法。

变化的三种类型：  
1.逻辑变化  
2.子模块变化  
3.可见视图变化

3.3.1 说明各模块之间接口，包括：

3.3.1.1 各模块的用途、使用(调用)方式；

3.3.1.2 说明其每个子程序（方法，Method）、特性（Property）和反馈的事件（Event），相应的输入、输出项说明；

3.3.1.3 指明各模块内含的各类外部接口，并列出这些外部接口说明材料名。

4 数据库设计

4.1 数据库模型

此处需要给出数据库设计模型，建议使用Power Designer等工具。

4.2 ER分析+表结构设计

<https://www.cnblogs.com/wyq178/p/8549715.html>

<https://blog.csdn.net/fighteryang/article/details/82848505>

4.3 安全保密设计

说明在数据库的设计中，如何通过区分不同的访问者、不同的访问类型和不同的数据对象，进行分别对待而获得的数据库安全保密设计考虑。

5 系统安全和权限设计

5.1 系统安全

<https://blog.csdn.net/qappleh/article/details/80485197>

<https://blog.csdn.net/weixin_34221073/article/details/89652071>

<https://blog.csdn.net/ljw499356212/article/details/81055141>

5.1.1 系统安全控制

5.1.2 用户身份鉴别

5.1.3 用户对系统的访问权限和范围

5.2 数据安全

5.2.1 数据用户身份鉴别

5.2.2 访问主体、访问对象的控制策略和实现方法

5.2.3 数据加密方法

5.3 备份与恢复

5.3.1 系统备份

5.3.2 数据备份

5.3.3 系统恢复

5.3.4 数据恢复

5.4 出错处理

5.4.1 列举出错情况

5.4.2 出错的形式、含义、处理方法

5.4.3 出错失效后的后备措施

5.5 权限设计

6系统接口设计

UI界面



外部接口

小程序登录模块与微信后台的接口

授权接口wx.getSetting

获取登录码的接口wx.login

获取用户信息的接口wx.getUserInfo

小程序扫码访问手机摄像头或相册接口:

wx.scanCode({

//调用成功则放回调用结果

success (res) {

//res.data能够获取座位的号码

}

})

小程序获取用户地理位置接口：

wx.getSetting({

success(res) {

if (!res.authSetting['scope.userLocation']) {

wx.authorize({

scope: 'scope.userLocation',

success () {

// 用户已经同意小程序使用定位，后续调用 wx.getLocation 接口不会弹窗询问

wx.getLocation({

type: 'wgs84',

success (res) {

//获取用户的地理位置

const latitude = res.latitude

const longitude = res.longitude

const speed = res.speed

const accuracy = res.accuracy

}

})

}

})

}

}

})

内部接口：

登录验证接口：

www\login\_check.jsp

在验证登录时，调用该接口，传入的参数为用户唯一标识userid和session会话维持

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 长度 | 备注 |
| Userid | String | 255字节 | 必填 |
| Session | String | 255字节 | 必填 |

如果用户存在且会话维持码正确返回111，如果用户存在但还是会话维持码不正确返回112，如果用户不存在则返回113。

实名认证验证接口:

在进行实名认证验证时，调用该接口，传入的参数为用户唯一标识userid

www\identify\_judeg.jsp

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 长度 | 备注 |
| Userid | String | 255字节 | 必填 |

如果用户未验证且会话维持码正确返回131，如果用户验证后但还是会话维持码不正确返回132。

座位状态判断接口:

www\seat\_judeg.jsp

该接口为座位状态判断接口，传入的参数为座位号、用户唯一标识userid

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 长度 | 备注 |
| Seat\_num | Int | 11字节 | 必填 |
| Userid | String | 255字节 | 必填 |

再扫二维码的情况下，如果座位是自己预约的或自己进行暂离的，则返回120，如果座位被预约了，但是不是自己预约的返回121，如果座位的状态是暂离的，则返回122。在不是扫码的情况下，已经被预约返回123。