**Focus**

**Control**

**REPORT FOR ASSIGNMENT6.2**

**71117324 施 展**

**71**11**7405 刘钰宵**

**71117402 张婧媛**

**71117408 梅洛瑜**

**2019/12/20**

**A SIMPLE PROGRAM FOR FOCUS CONTROLING**

目 录

1 系统设计………………………………………………………………**3**

1.1 项目概述…………………………………………………………**3**

1.2 项目需求………………………………………………………… **3**

1.2.1 原始需求……………………………………………………**3**

1.2.2 简单需求设计………………………………………………**4**

1.2.3 性能需求……………………………………………………**5**

1.3 项目实施计划及任务安排………………………………………**6**

1.4 系统架构设计……………………………………………………**6**

1.4.1 系统总体结构图……………………………………………**6**

1.4.2 软件功能结构设计…………………………………………**7**

1.4.3 业务逻辑……………………………………………………**7**

1.5 类设计……………………………………………………………**7**

1.5.1 数据访问层…………………………………………………**7**

1.5.2 业务逻辑层…………………………………………………**9**

2 数据库设计…………………………………………………………**12**

3 界面设计……………………………………………………………**14**

**1 系统概述**

* 1. **项目概述**

1. 工作内容：

该项目为一个专注注意力的安卓开发APP,用户通过安装APP，可以注册账号、登录。用户可以在主界面选择添加专注任务，并设置任务时长，计时类型，设置成功后，除去已设置信息，还会显示已完成时长。用户可以在此系统中设置个人信息，以及看全员排名。还可以通过专注时长获得的积分点(“燃点数”)在“精品商城”界面兑换物品。同时我们还提供了管理员系统，管理员可以进行数据观测，包括监控用户人数和数据库使用比例，制作和发布、管理推送等。

1. 条件和限制

项目开发平台:前端: Android studio

后端: pycharm

数据库管理系统:SQL Server 2015

开发人员:71117324 施展,71117402 张婧媛,71117405刘钰宵,71117408 梅洛瑜

已具备条件:项目架构基本完成，开发工具也齐备

未具备条件:暂无

项目预计完成日期:2010年1月10日前

1. 产品

产品名称:默燃

开发语言:python

1. 文档

项目需求分析报告，项目设计报告，数据库设计报告，项目源代码，网站功能说明书（帮助文档）

1. 运行环境

Android

* 1. **项目需求**

**1.2.1 原始需求**

**用户系统:**

用户可以注册账号、登录界面，如果用户忘记了之前的密码需要能提供将密码找回的机制，最后提供用户QQ登录、微信登录功能；

登录后进入主界面1；

主界面1是一个专注系统，有添加专注任务的按钮，每添加一个任务后主界面就会从上往下增加一栏，每一栏内容包括任务的名称、任务时长、计时类型和已经完成时长，重新编辑该栏信息（包括包括删除该栏），长按某一栏可以还有一个“开始”键。最后整个主界面1的最右侧有一个“燃度条”，这个燃度条需要能随着时间和用户实时变化，燃条变化的算法逻辑暂略。

点击任务添加的按钮后会跳转页面进入一个添加任务的界面，在这里用户可以自定义任务的名称、时长和计时类型，设置完成后会在主界面增加一栏并对于。

点击上面任务的某一栏的“开始”键后会跳转界面，进入专注模式的界面，界面上方是任务的名称和计时类型，中间是一个倒计时装置，下边有三个按钮，从左到右是“白噪音”（点击可选择并打开白噪音）、“暂停”（点击后弹出一个对话框，对话框包括一个倒计时、暂停告知和“继续计时”的按钮）、“终止”按钮。（弹出一个对话框，选择是否确定退出），且限制此时不能退出该界面。

在任务完成后弹出一个对话框，最上面是恭喜完成等语句，下面排列两个单向选择题，询问用户“专注时长比例”和“充实度”。最下面是“完成结束”的按钮，点击按钮进入主界面1.

主界面1最下面有一个滑钮，点击滑钮进入主界面2，主界面2上面是一个钟形倒计时装置，记录距离上次专注的时间。下面有四个大小相同的矩形方块组成，从上至下分别是“个人信息”、“数据报表”、“全员排名”、“精品好物”。

点击“个人信息”进入个人数据界面，包括头像、编号、账户、累积专注时长等信息，下面有一些修改密码功能。

点击“数据报表”进入数据报表界面，数据报表界面呈现一些简单的报表，包括每日专注时长统计图、每日专注时间段分布情况、月度统计情况、年度统计情况等等。该模块需要设置限制，每日只能查看一次，在被查看过后锁住该模块。

点击“排行榜”进入排行榜界面，用户可以看到全球用户的最高排名和自己的排名，用户除了自己以外都显示为匿名。该模块需要设置限制，每周只能查看一次，在被查看过后锁住该模块。

点击“精品好物”进入商城界面，界面最上面显示用户当前拥有的燃点数和预计本周增加的燃点数，燃点数由每天的燃度值决定、每周结算一次。下面是一个数字倒计时装置，限定用户使用该功能时间。再下面一些官方的推送条，每条推送条上显示图标、标题、概述、燃烧值。

点击推送条，进入推送界面，推送可由图片、文字、链接组成，用户观看时需要按照时间花费燃点数。

**管理员系统:**

主界面1是数据观测界面，依次呈现各类数据的实时值，包括用户人数、当前使用人数、数据库使用比例、以及一些数据图标等。

主界面2是推送的制作和发布界面，提供文本输入、链接插入、图表插入等功能，

主界面3是推送的管理界面，可以看到当前已经发布的所有推送及信息，可以进行重新编辑和删除等操作。

**1.2.2 简单需求设计**

默燃在功能方面可分为两个区域，分别是：普通用户功能区域、管理员功能区域。

1. 普通用户区域

主要功能是使用户可以设置专注任务

* 用户登录
* 找回密码
* 使用专注功能
* 查看自己的长期专注信息
* 设置和修改个人信息
* 接收一些好的文章和推荐
* 查看自己的专注情况所在排名

等等

1. 管理员功能区域

本区域主要提供了针对管理员的功能，规定了管理员的具体权限

* 查看各种后台的数据和相关图表，了解软件的运行状况
* 进行文章推送的管理。

等等

**1.2.3 非功能性需求**

1.响应时间：用户在使用排行榜、高级报表时反馈等待时间短 在95％的情况下，一般时段响应时间不超过1秒，高峰时段不超过3秒。

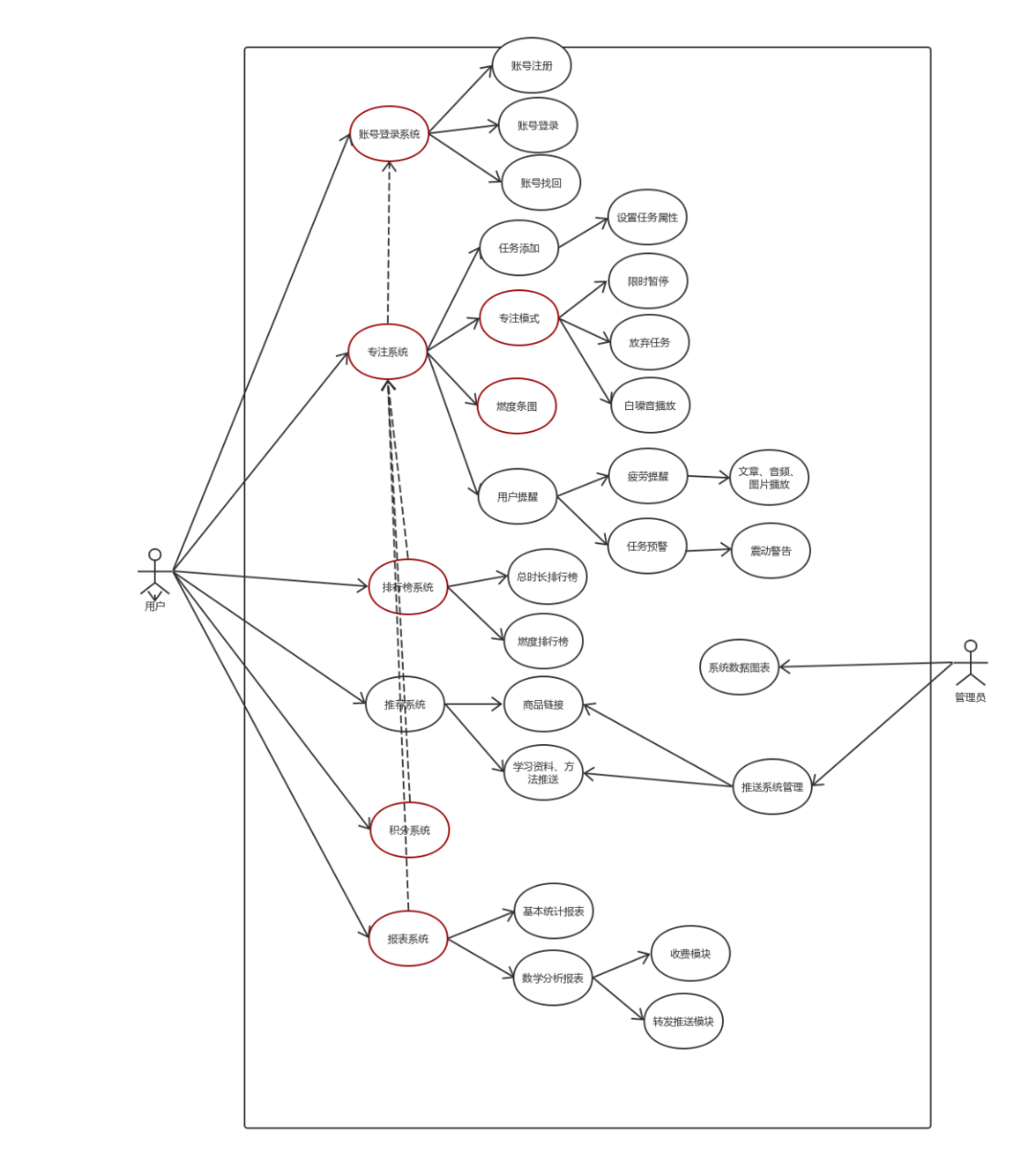
2.系统容量：服务器能够容纳至少一万用户的数据量 服务器容量达到500GB

3.兼容性：用户能在安卓手机、windows电脑上运行同一个账号开发Android和Windows系统两个版本

4.信息安全性：用户的私人信息不会泄露，系统只能使用用户给定权限内的功能、数据传输进行加密

5.数据安全性：用户的使用数据不会丢失采用本地存储和服务器存储双备份

**需求分析用例图如下所示：**



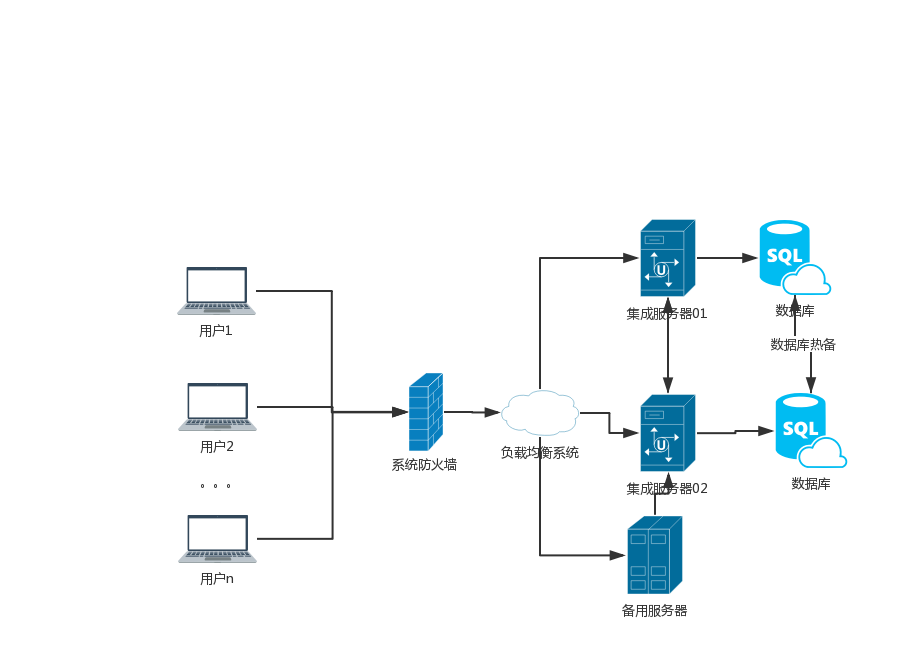
* 1. **项目实施计划及任务安排**

|  |  |  |  |
| --- | --- | --- | --- |
| 过程 | 时间 | 目标 | 任务分配 |
| 需求阶段 | 2019/12/18 | 完成需求分析文档 | 普通用户基本功能：刘钰宵、张婧媛  管理员系统功能：梅洛瑜、施展  扩展功能(如商城)：梅洛瑜、施展  整合：刘钰宵、张婧媛 |
| 构思阶段 | 2019/12/19-2019-12/21 | 实现用例图和场景说明文档 | 普通用户基本功能：刘钰宵、张婧媛  管理员系统功能&扩展功能：梅洛瑜、施展 |
| 设计阶段 | 2019/12/22-2019/12/24 | 实现数据库、系统和安卓端、网页端设计 | 数据库一同设计  安卓端网页等均一同设计  扩展功能和管理员系统：施展、梅洛瑜 |
| 实现阶段 | 2019/12/25-2019/12/30 | 完成页面、APP编码以及功能编码 | 普通用户功能区域：施展、刘钰宵  管理员系统&代码整合：梅洛瑜、张婧媛  前端开发:张婧媛、刘钰宵 |
| 稳定阶段 | 2019/12/31-2020/1/3 | 简单测试 | 所有人员交替测试所有功能 |
| 部署阶段 | 2020/1/4 | 完善系统、发布 | 一起完善系统 |

* 1. **系统架构设计**

在需求分析中，我们确定了系统的功能模块，包括登录和注册功能，浏览功能，设置专注任务功能和管理员功能。用户可以通过下载APP，使用服务器上的类库和数据库，实现专注任务的设置和浏览。系统的界面应简单易懂，用户可以方便地操作。

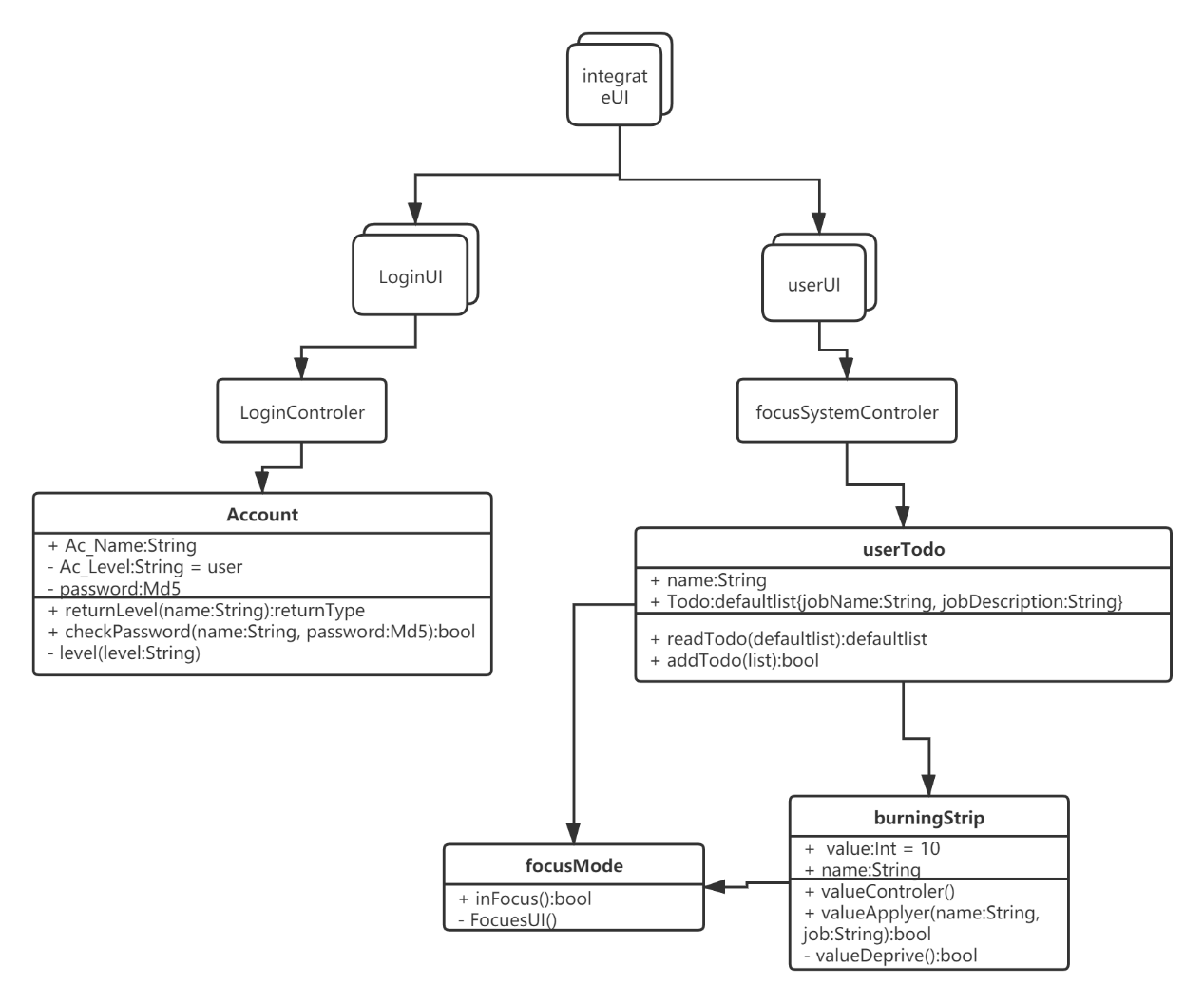
**1.4.1 系统总体结构图**

****

**1.4.2 软件功能结构设计**

|  |
| --- |
| 用户界面 |
| 业务逻辑类库 |
| 数据访问层 |
| 数据库 |

**1.4.3 业务逻辑**



* 1. **类设计**

**1.5.1 数据访问层**

数据访问层只考虑基本数据的最简单的增删改查，而不考虑业务信息。

Commodities,User,Orders,UserToDo,UserJob为相应的表单的增删改查操作模块。

**Commodities**

|  |  |  |  |
| --- | --- | --- | --- |
| 函数名 | 参数 | 返回值 | 说明 |
| getCommodityData() | 无 | Commodities表中元素的元组 | SELCECT \* FROM COMMODITIES |
| deleteItem() | CommodityId | 影响行数 | 按照主键删除书籍 |
| insertItem() | 所有属性 | 影响行数 | 插入一件商品 |
| selectItem() | CommodityId | 一本书的各属性元组 | 按主键查找商品 |
| upDateItem() | 所有属性 | 影响行数 | 按主键更新商品信息 |
| upDateItemActive() | CommodityId,active | 影响行数 | 更改商品是否下架 |
| upDateItemNum() | CommodityId,num | 影响行数 | 更改商品数量 |
| upDateItemPrice() | CommodityId,price | 影响行数 | 更改商品价格 |
| upDateItemSellNum() | CommodityId,num | 影响行数 | 更改销售量 |

**User**

|  |  |  |  |
| --- | --- | --- | --- |
| 函数名 | 参数 | 返回值 | 说明 |
| getUserData() | 无 | User表中元素的元组 | SELCECT \* FROM USER |
| deleteUser() | userAccount | 影响行数 | 按照主键删除用户 |
| insertUser() | 所有属性 | 影响行数 | 新建一个用户 |
| selectUser() | userAccount | 该用户的各属性元组 | 按主键查找用户 |
| upDateUser() | 所有属性 | 影响行数 | 按主键更新用户所有信息 |
| select() | 某一属性 | 返回选出的用户信息 | 按照某一属性降序查询 |

**Orders**

|  |  |  |  |
| --- | --- | --- | --- |
| 函数名 | 参数 | 返回值 | 说明 |
| getOrderData() | 无 | Orders表中元素的元组 | SELCECT \* FROM ORDERS |
| deleteOrder() | OrderId | 影响行数 | 按照主键删除订单记录 |
| insertOrder) | 所有属性 | 影响行数 | 新建一个订单 |
| selectOrder() | OrderId | 该订单的各属性元组 | 按主键查找订单 |
| upDateOrderState'() | OrderId,OrderIState | 影响行数 | 按主键更新订单状态 |

**UserToDo**

|  |  |  |  |
| --- | --- | --- | --- |
| 函数名 | 参数 | 返回值 | 说明 |
| getUserToDoData() | 无 | UserToDo表中元素的元组 | SELCECT \* FROM USERTODO |
| getUserToDoDataByUser() | userAccount | UserToDo表中该用户元素的元组 | 某一用户的待办任务 |
| deleteUserToDo() | TodoNo | 影响行数 | 按照主键删除待办任务 |
| insertUserToDo() | 所有属性 | 影响行数 | 新建一个待办任务 |
| selectUserToDo() | TodoNo | 该待办任务的各属性元组 | 按主键查找待办任务 |
| upDateUserToDo() | TodoNo,TodoAlreadyTime | 影响行数 | 按主键更新已完成时间 |

**UserJob**

|  |  |  |  |
| --- | --- | --- | --- |
| 函数名 | 参数 | 返回值 | 说明 |
| getUserJobData() | 无 | UserJob表中元素的元组 | SELCECT \* FROM USERJOB |
| getUserJobDataByUser() | userAccount | UserJob表中该用户元素的元组 | 某一用户的任务记录 |
| deleteUserJob() | JobNo | 影响行数 | 按照主键删除任务记录 |
| deleteUserJobByUser() | userAccount | 影响行数 | 删除某用户的任务记录 |
| insertUserJob() | 所有属性 | 影响行数 | 新建一个任务记录 |
| selectUserJob() | JobNo | 该任务记录的各属性元组 | 按主键查找任务记录 |
| upDateUserJob() | TodoNo,TodoAlreadyTime | 影响行数 | 按主键更新已完成时间 |
| selectUserJobByTime() | userAccount,date | 查找到的任务列表 | 某用户某时间后的所有任务记录 |

**1.5.2 业务逻辑层**

利用数据访问层提供的函数，处理其结果，得到业务需要的数据。

**CommoditiesBLL**

|  |  |  |  |
| --- | --- | --- | --- |
| 函数名 | 参数 | 返回值 | 说明 |
| addItem() | 商品信息 | 影响行数 | 增加一个商品，返回0表示添加失败 |
| BestSellers() | 无 | 销售量前5的商品信息 | 热销商品列表 |
| getItemData() | CommodityId | 该商品所有信息 | 查询商品详情 |
| deleteItem() | CommodityId | 影响行数 | 删除商品 |
| upDateItem() | 所有属性 | 影响行数 | 按主键更新商品信息 |
| getAll() | 无 | 所有商品信息 | 查看所有商品的信息 |
| upDateItemActive() | CommodityId,active | 影响行数 | 更改商品是否下架 |
| upDateItemNum() | CommodityId,num | 影响行数 | 更改商品数量 |
| upDateItemPrice() | CommodityId,price | 影响行数 | 更改商品价格 |
| sellItem() | CommodityId | 影响行数 | 指定商品数量-1，销售额+1 |

**UserBLL**

所有的用户操作

|  |  |  |  |
| --- | --- | --- | --- |
| 函数名 | 参数 | 返回值 | 说明 |
| getAll() | 无 | 所有用户信息 | SELCECT \* FROM USER |
| deleteUser() | userAccount | 影响行数 | 按照主键删除用户 |
| addUser() | 所有属性 | 影响行数 | 新建一个用户 |
| selectUser() | userAccount | 该用户的各属性元组 | 按主键查找用户 |
| upDateUser() | 所有属性 | 影响行数 | 按主键更新用户所有信息 |
| getSort() | 无 | 按照时长和燃度排序的用户列表 | 全体用户排名 |
| getTable() | userAccount | 计算出的需要的图表信息 | 根据学习信息计算生成图表的必要信息 |

**OrdersBLL**

|  |  |  |  |
| --- | --- | --- | --- |
| 函数名 | 参数 | 返回值 | 说明 |
| getAll() | 无 | Orders表中元素的元组 | SELCECT \* FROM ORDERS |
| deleteOrder() | OrderId | 影响行数 | 按照主键删除订单记录 |
| addOrder() | 所有属性 | 影响行数 | 新建一个订单 |
| selectOrder() | OrderId | 该订单的各属性元组 | 按主键查找订单 |
| upDateOrderState() | OrderId,OrderIState | 影响行数 | 按主键更新订单状态 |
| getOrdersByAccount() | userAccount | 某用户的所有订单 | 查询用户的历史订单 |
| getOrdersByAccountState() | userAccount,state | 某用户某状态的所有订单 | 查询用户某状态的订单记录 |

**UserToDoBLL**

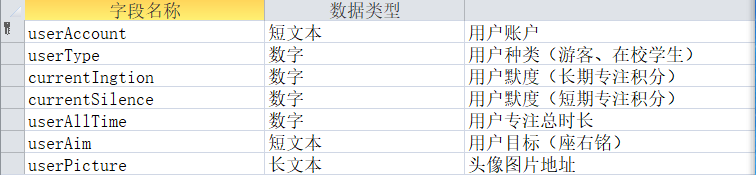
|  |  |  |  |
| --- | --- | --- | --- |
| 函数名 | 参数 | 返回值 | 说明 |
| getAll() | 无 | UserToDo表中元素的元组 | SELCECT \* FROM USERTODO |
| getAllByUser() | userAccount | UserToDo表中某用户的所有待办任务 | 查询某用户所有待办任务 |
| deleteUserToDo() | TodoNo | 影响行数 | 按照主键删除待办任务 |
| addUserToDo() | 所有属性 | 影响行数 | 新建一个待办任务 |
| selectUserToDo() | TodoNo | 该待办任务的各属性元组 | 按主键查找待办任务 |
| upDateUserToDo() | TodoNo,TodoAlreadyTime | 影响行数 | 按主键更新已完成时间 |

**UserJob**

|  |  |  |  |
| --- | --- | --- | --- |
| 函数名 | 参数 | 返回值 | 说明 |
| getUserJobData() | 无 | UserJob表中元素的元组 | SELCECT \* FROM USERJOB |
| getAllByUser() | userAccount | UserJob表中某用户的所有待办任务 | 查询某用户所有任务记录 |
| deleteUserJob() | JobNo | 影响行数 | 按照主键删除任务记录 |
| deleteUserJobByUser() | userAccount | 影响行数 | 删除某用户所有任务记录 |
| addUserJob() | 所有属性 | 影响行数 | 新建一个任务记录 |
| selectUserJob() | JobNo | 该任务记录的各属性元组 | 按主键查找任务记录 |
| upDateUserJob() | TodoNo,TodoAlreadyTime | 影响行数 | 按主键更新已完成时间 |
| selectUserJobByTime() | userAccount,date | 查找到的任务列表 | 某用户某时间后的所有任务记录 |

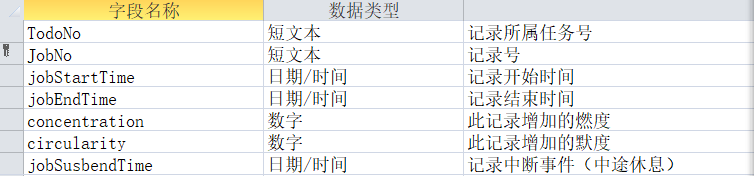
1. **数据库设计**

**User:(用户)**

****

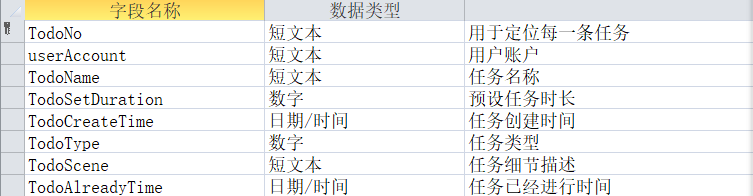
属性都是原子的，有唯一主键，不存在对非主键的依赖，没有传递依赖，因而是符合 3NF 的。

**UserJob:(用户完成的任务)**

****

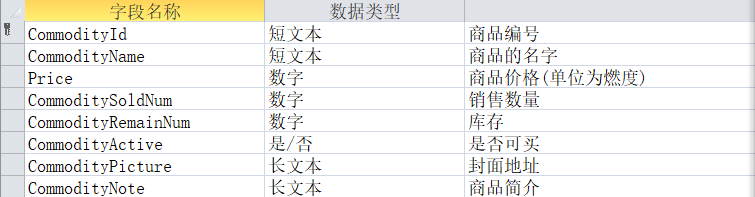
属性都是原子的，有唯一主键，不存在对非主键的依赖，没有传递依赖，因而是符合 3NF 的。

**UserToDo:(用户创建的待完成任务)**

****

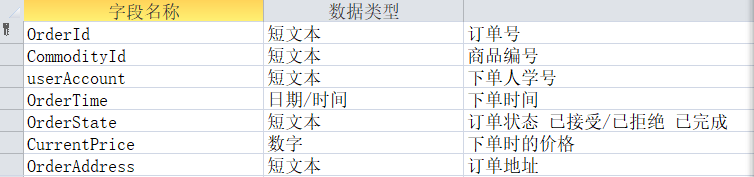
属性都是原子的，有唯一主键，不存在对非主键的依赖，没有传递依赖，因而是符合 3NF 的。

**Commodities:(商品)**

****

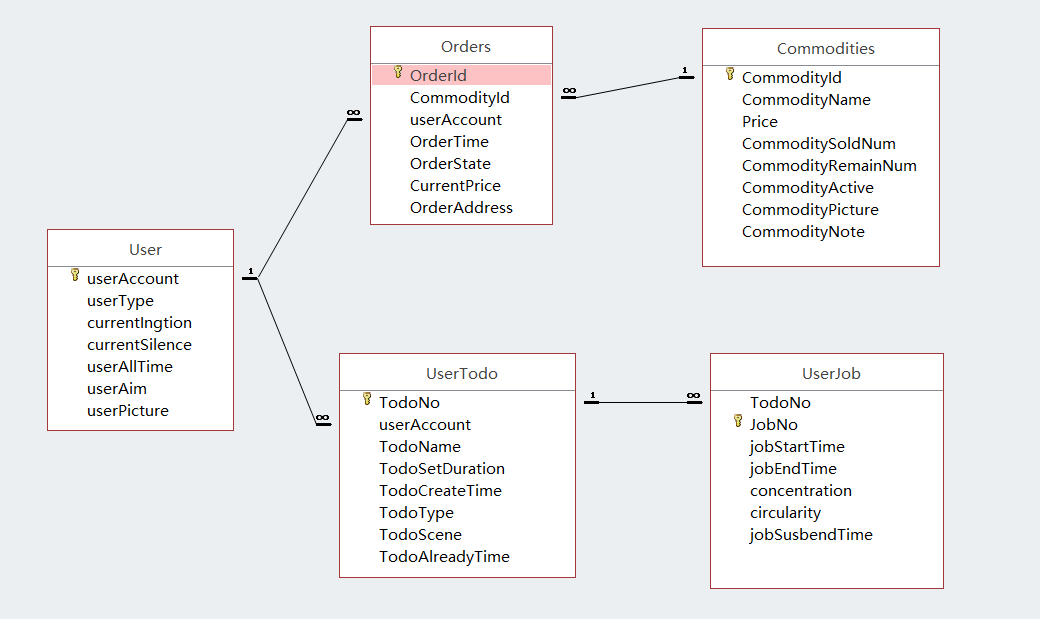
属性都是原子的，有唯一主键，不存在对非主键的依赖，没有传递依赖，因而是符合 3NF 的。

**Orders:(订单)**

****

属性都是原子的，有唯一主键，不存在对非主键的依赖，没有传递依赖，因而是符合 3NF 的。

商品价格依赖于商品号，而价格可能有改动，所以订单价格只依赖于订单号。



1. **界面设计**

