

杭州电子科技大学

软件工程学院

# [杀手预约管理系统]

课程设计报告

专业：软件工程

班级：14108411、14108413

小组：杀手小组

指导老师：金洁洁

完成时间:2016-01-12

# 目录

- 0.项目相关说明
  - 0.1小组成员
  - 0.2计划安排
  - 0.3项目实际进展追踪表
- 1.系统介绍
- 2.系统需求分析
  - 2.1系统综合需求
  - 2.2数据流图
  - 2.3数据字典
- 3.系统模块划分
  - 3.1系统结构图
  - 3.2各模块功能概述
- 4.概念结构设计
- 5.逻辑结构设计
  - 5.1逻辑结构设计
  - 5.2模式优化
  - 5.3表关系图
- 6.物理设计
  - 6.1索引设计
  - 6.2表的创建
  - 6.3视图的创建
  - 6.4存储过程或触发器的创建
- 7课程总结
- 8参考文献
- 9致谢

# 0.项目相关说明

## 0.1小组成员

序号	学号	姓名	角色及任务	工作量
1	14108106	张若琦	小组长	25%
2	14108303	彭薇	组员	25%
3	14108302	梁冰	组员	25%
4	14108107	朱湘如	组员	25%

## 0.2计划安排

阶段名称	主要任务	计划完成时间	人员安排及任务
需求分析	充分思考，分析需求，确定主要的功能，画数据流图，做数据字典	2015-11-15	由四个组员共同讨论选题，与需求，主要交给组长整理并记录
概念设计	画E-R图	2015-12-10	一位同学主要做，其他同学负责修改
逻辑设计	将概念模型转换为关系模型	2015-12-15	交由一位同学负责，上一位写ER图的同学负责协助，其他同学修改
模式优化	规范关系模型，确定主键，外键	201-12-25	交由一位同学进行负责，其他同学进行修改
物理设计	索引设计 表的创建 视图的创建 存储过程或触发器的创建。	2016-01-05	全部组员共同参与

### 0.3项目实际进展跟踪表

阶段名称	计划完成时间	实际完成时间	情况说明与分析
需求分析	2015-11-15	2015-11-28	因为没有经验，加上找的例子也没人做过示例，素以比较麻烦。前期想的也不是很全面。导致进行后续工作的时候发生了全部重做的现象。
概念设计	2015-12-10	2015-12-15	E-R图总共修改了三次，第一次是经验不足导致的逻辑混乱，第二次将不需要存储的视图也存入，第三次经过老师的修改，终于得到最终版本。
逻辑设计	2015-12-15	2015-12-22	ER图的命名方式不是很好，导致逻辑设计比较麻烦，不过总体还不错。
模式优化	2015-12-25	2015-12-25	前期设计的时候，花费了很大的功夫，所以设计的还不错，达到了3NF，故没有花太多时间。
物理设计	2016-01-05	2016-01-05	较为简单，完成的不错。

# 1.系统介绍（完成者： 姓名 张若琦 学号14108106）

随着网络游戏的发展，人们对于游戏的需求也越来越大，现在几乎每个人都玩过网游。正式基于这样的一个大背景下，在看到了未来发展的潜力后，我们决定做一个关于杀手的预约管理系统，主要是满足一些用户在游戏中的一些恩仇需求，一级提供一个关于任务供求信息发布，并实现任务的接取的一个游戏类型的系统。

正是基于上述的这么一些需求，我们将系统分为主要的八大功能，分别为用户管理、任务信息管理、任务信息查询浏览功能、系统盈利查询功能、用户发布任务信息管理、任务交易管理、杀手榜排名管理以及安全管理。下面对几个功能做详细介绍：

用户管理：这是对于使用者的一个管理。主要功能模块又可细分为四个部分，主要由管理员管理、登录日志管理、会员管理、杀手管理。管理员管理主要是用来对管理员的修改和删除。登录日志管理主要是记录管理员进入后台管理的一些情况。会员管理主要是针对注册会员的信息的修改和删除管理。杀手管理主要是针对内部杀手的信息的修改和删除管理。

任务信息管理：对于任务信息的管理，主要分为两个部分：第一分部是主要对于任务的等级、悬赏金额、结束日期、特殊要求等相关信息的管理。第二部分是对于任务信息的修改和删除管理。

任务信息查询浏览功能：对于这个部分，主要以两种情况来体现，第一、给杀手提供搜索任务信息功能。第二、给用户来查询自己的任务信息功能。

系统盈利管理与浏览功能：对于系统的盈利，主要根据任务的悬赏金额以及任务的完成情况进行计算。

杀手个人盈利查管理：对于杀手的个人盈利情况，要根据每次任务的完成情况与任务悬赏金额进行计算。

用户发布任务信息管理：对此，我们也划分两块来实现。第一，会员能够登陆并发布自己相关的悬赏任务信息。第二、对于会员用户发布的信息进行等级匹配和分类管理。

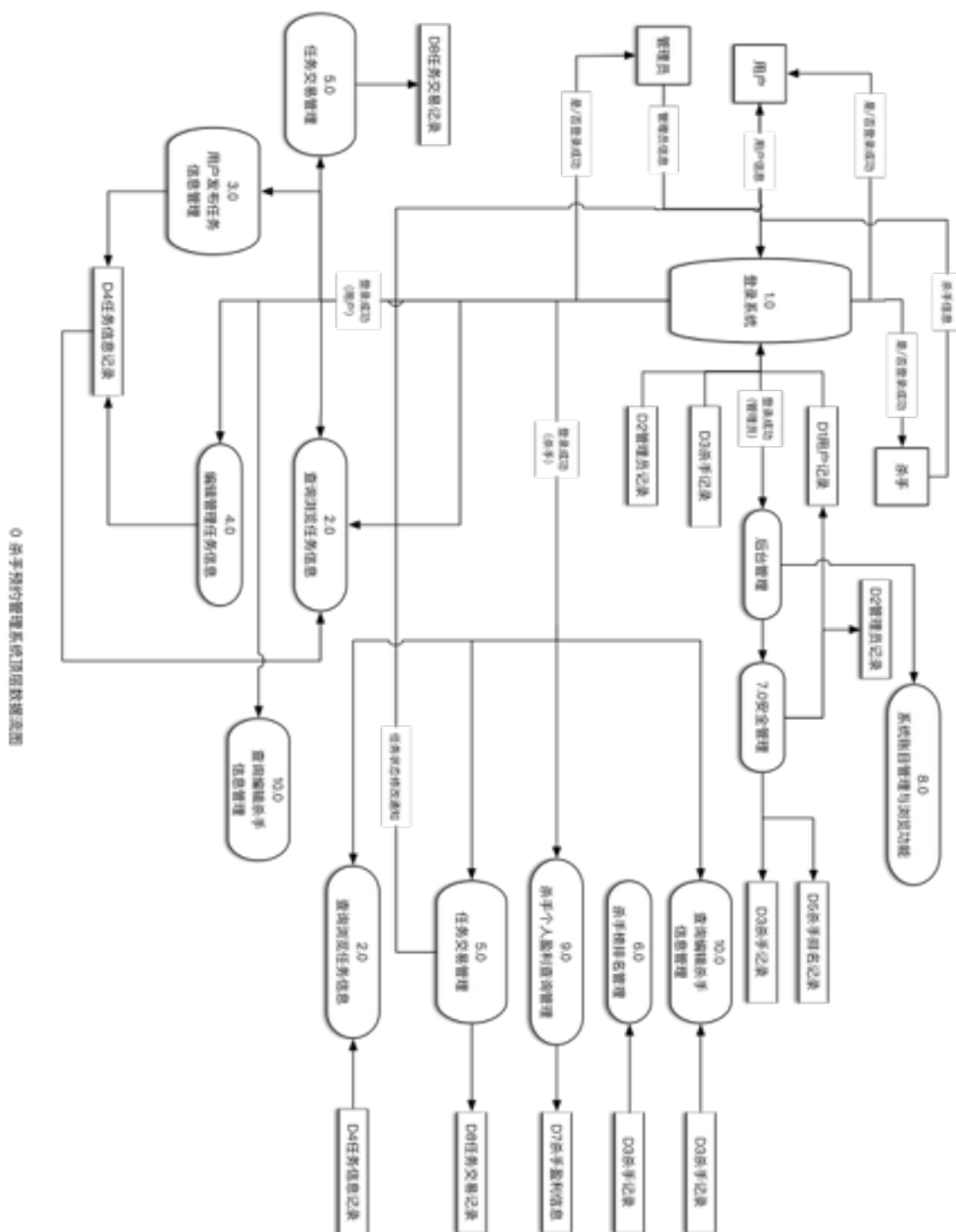
任务交易管理：对于网上交易模块主要是方便杀手领取任务的实现，进而对此进行的管理。

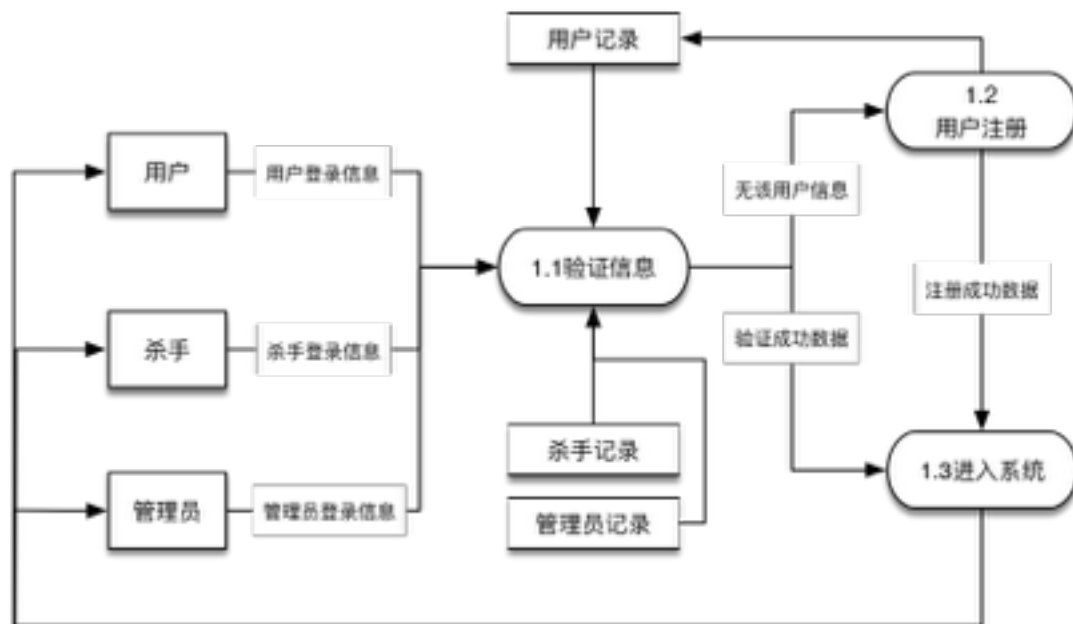
查询编辑杀手信息管理：对此，划分两个情况。第一、杀手或者用户查询杀手信息。第二、杀手对于自己的个人信息进行编辑。

杀手榜排名管理：通过两个部分进行实现：第一、系统对内部杀手进行管理，并分处等级与排名。第二、内部杀手可以查询浏览目前的杀手榜前十位和自己所处的排名。

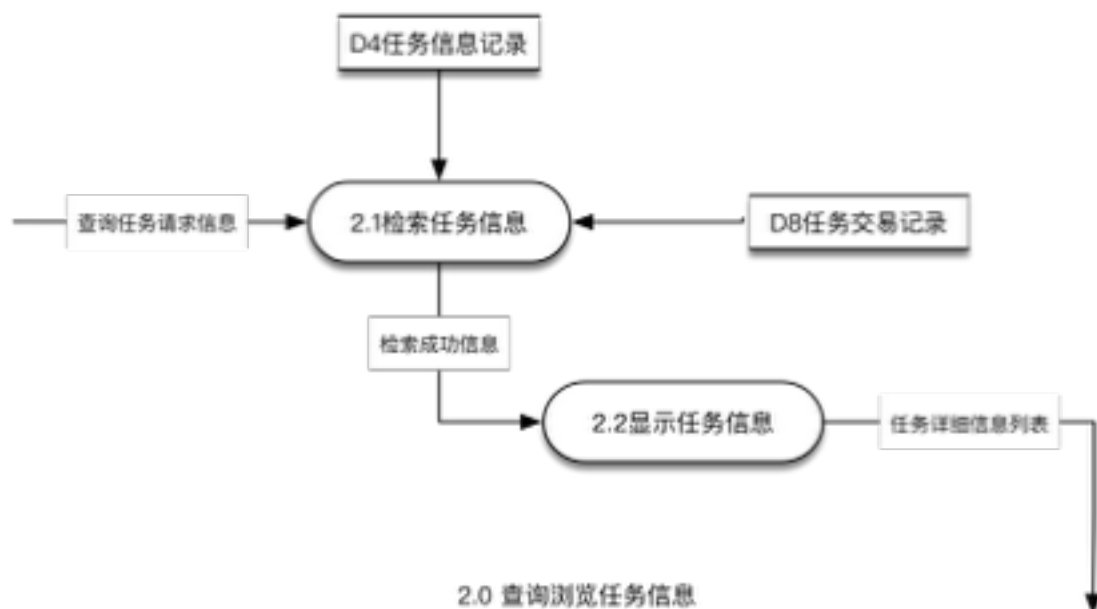
安全管理：对于用户在系统中的信息保密、维护的管理。

以上就是我们对于杀手预约管理系统的初步构思。

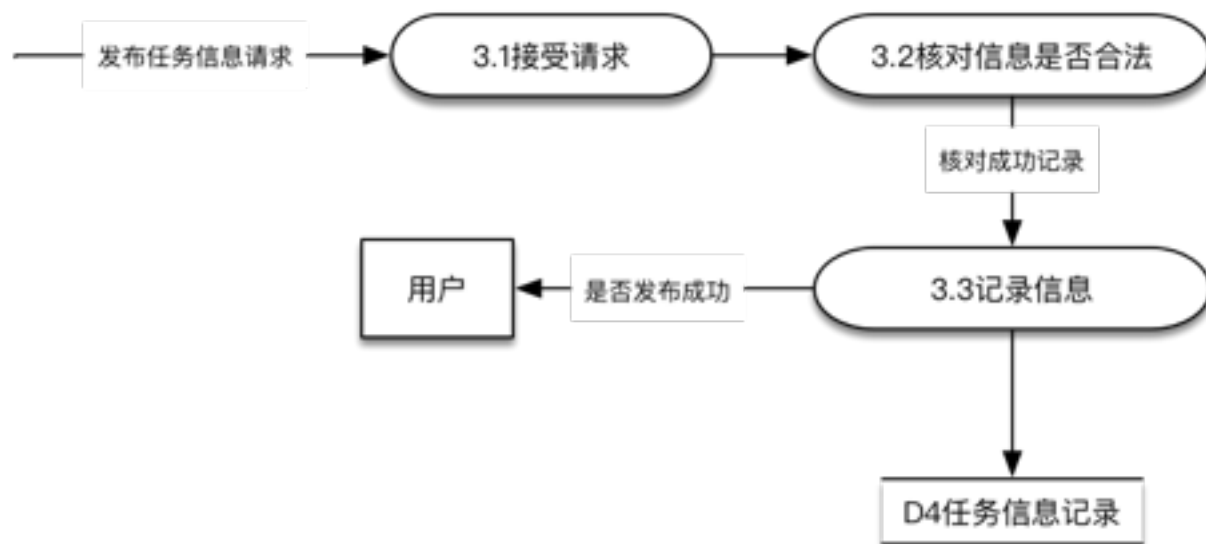




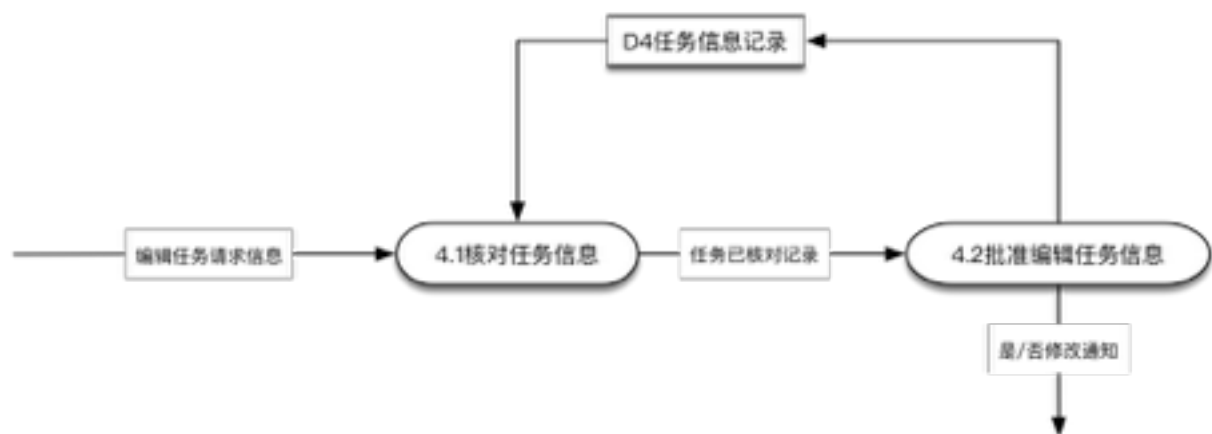
### 1.0 登录系统



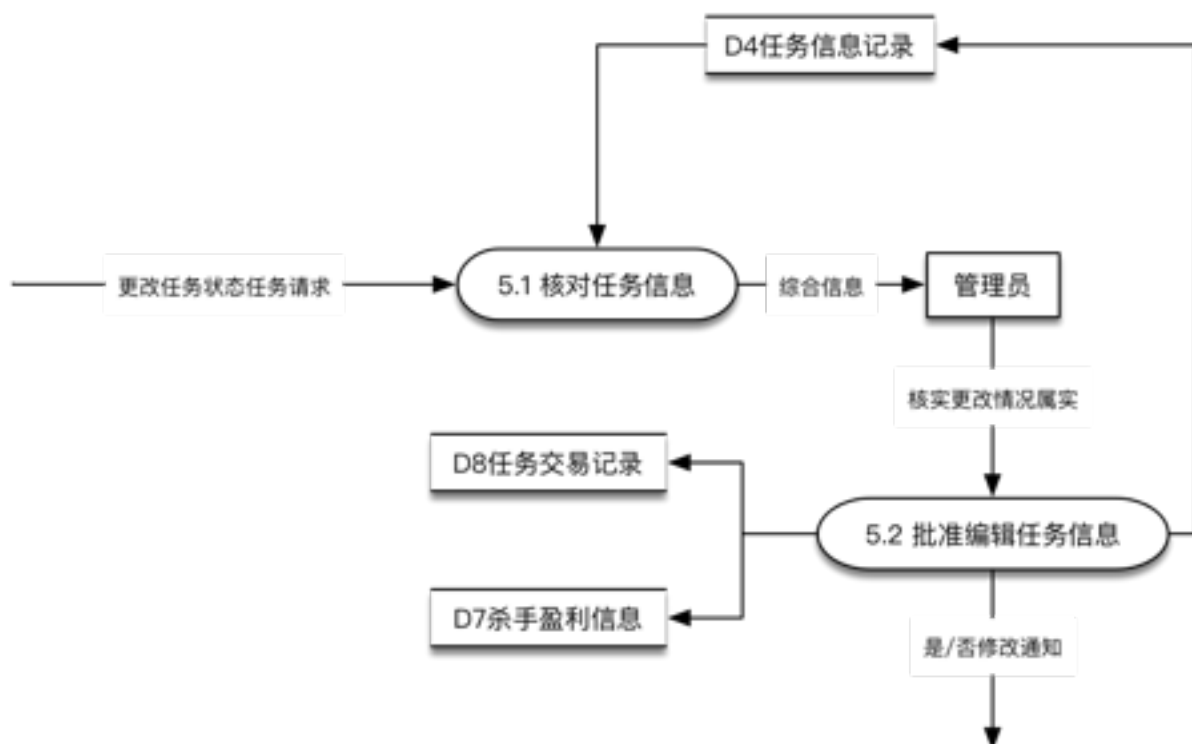
### 2.0 查询浏览任务信息



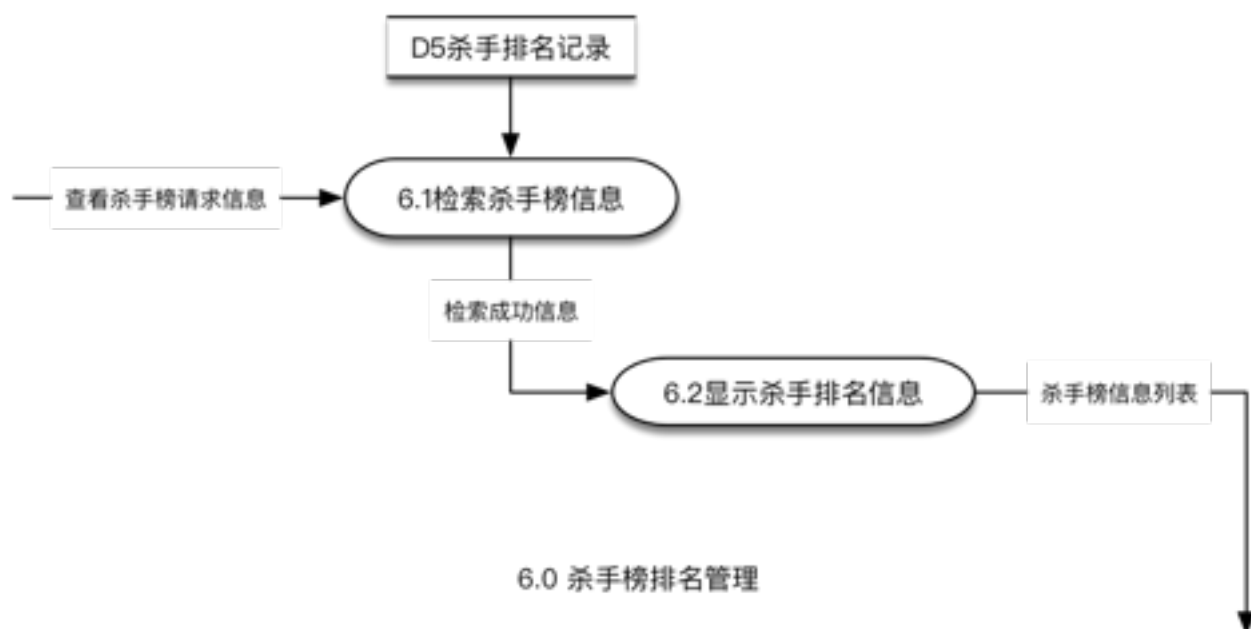
### 3.0 用户发布任务信息管理



4.0编辑管理任务信息

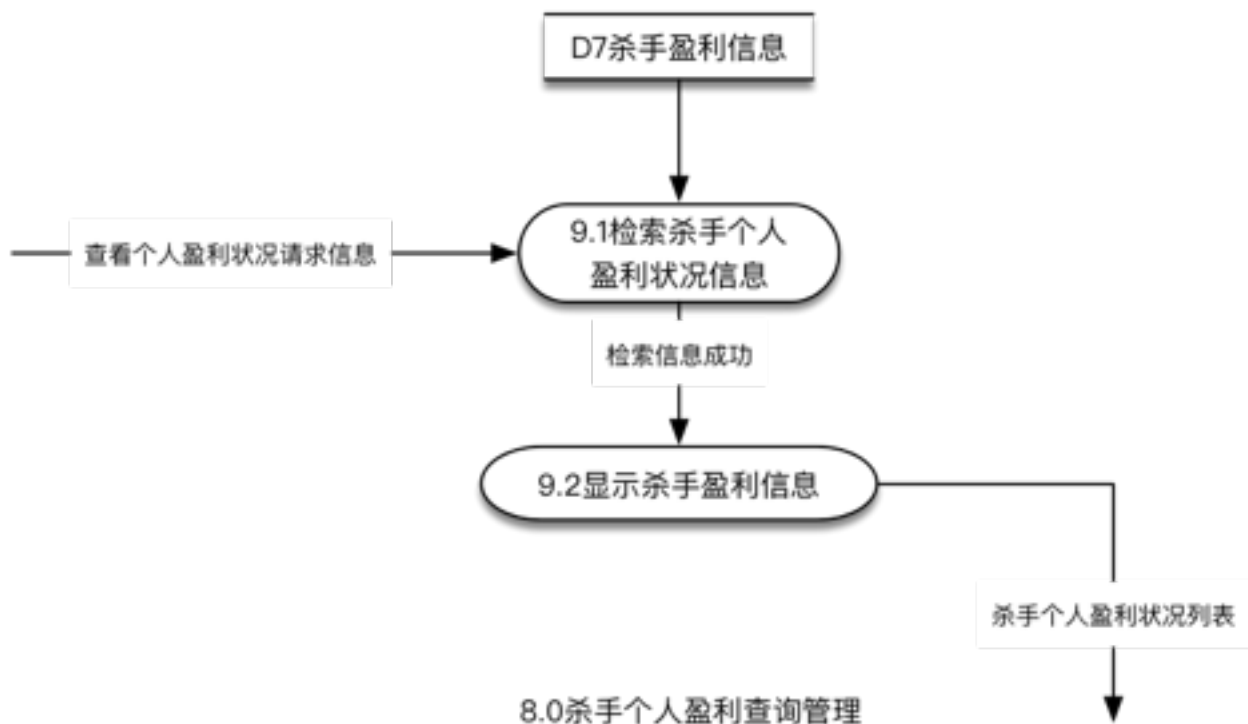
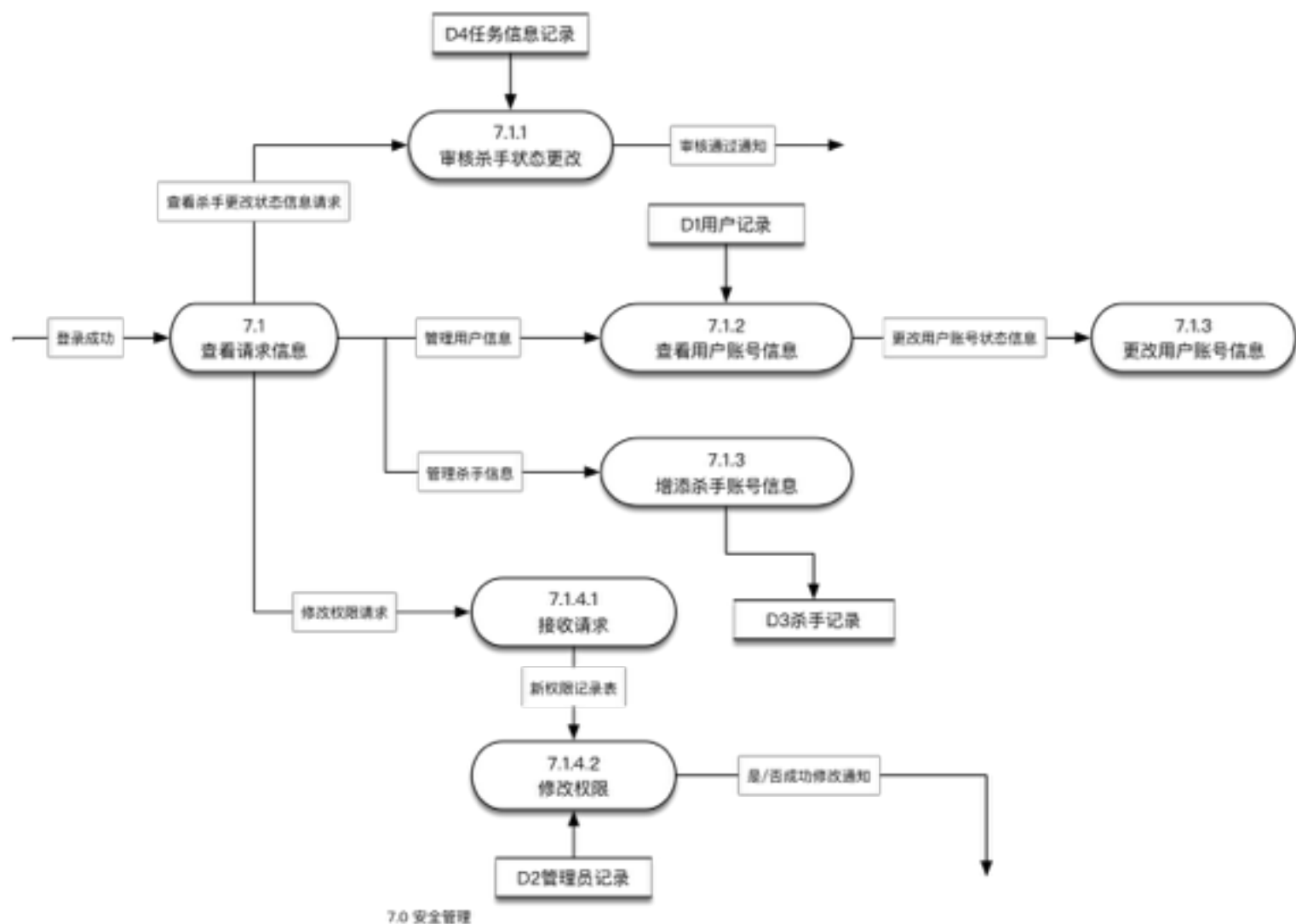


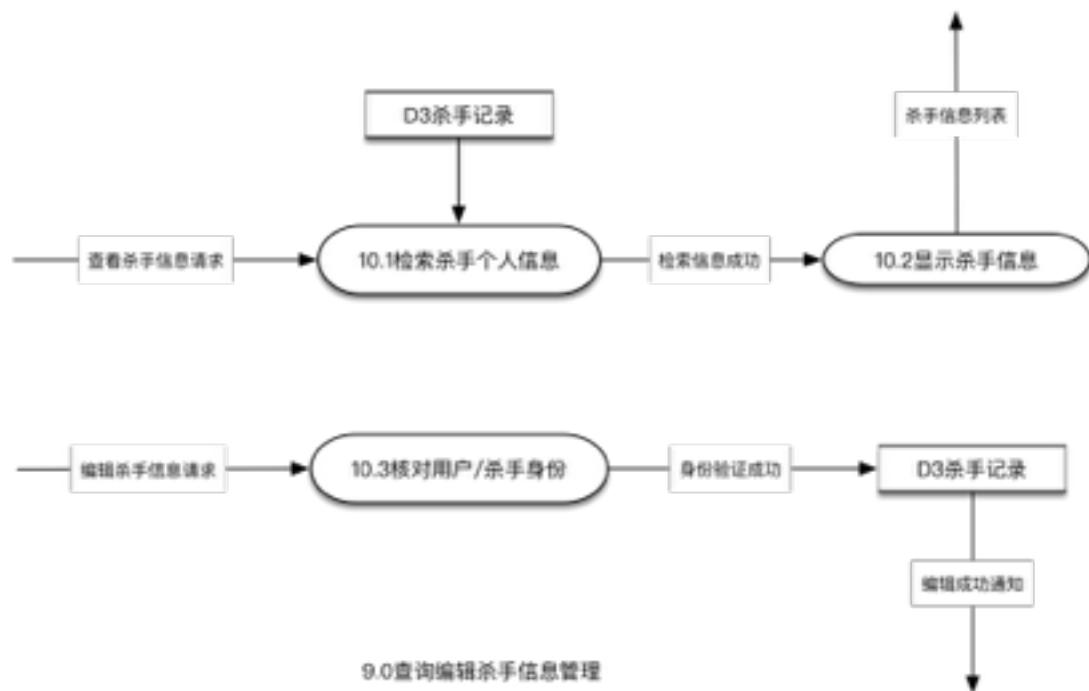
5.0任务交易管理



6.0 杀手榜排名管理







## 2.3 数据字典

### 2.3.1 主要的数据流定义

(完成者 姓名 朱湘如、张若琦 学号14108107、14108106)

- (1) 用户登录信息
  - 位置：用户→1.1
  - 定义：用户信息=账号（邮箱/手机号）+密码+头像
  - 说明：根据用户情况建立或更新用户信息
- (2) 杀手登录信息
  - 位置：杀手→1.1
  - 定义：杀手信息=账号（id）+密码+头像
  - 说明：根据杀手情况建立或更新杀手信息
- (3) 管理员登录信息
  - 位置：管理员→1.1
  - 定义：管理员信息=账号+密码
  - 说明：根据管理员的账号和密码鉴别管理员身份
- (4) 无该用户登录信息
  - 位置：1.1→1.2
  - 说明：没有该用户信息
- (5) 验证成功数据

位置：1.1→1.3

定义：验证成功数据=账号+密码

说明：通过账户和密码鉴别对象的身份，是否符合

(6) 注册成功数据

位置：1.2→1.3

定义：注册成功数据=账号（邮箱/手机号）+密码

说明：注册成功后根据用户信息直接进入系统

(7) 登陆成功

位置：1.3→2.0, 3.0, 4.0, 5.0, 10.0, 9.0, 7.1, 8.1

定义：用户信息+杀手信息+管理员信息

说明：根据用户信息，杀手信息，管理员信息进入用户，杀手，管理

员界面

(8) 查询任务请求信息

位置：2.0→2.1

请求信息=任务编号+用户/杀手id

说明：根据输入的信息查询任务信息

(9) 检索成功信息

位置：2.1→2.2

说明：成功检索到任务信息

(10) 任务详细信息列表

位置：2.2→用户/杀手

任务信息=任务编号+任务结束时间+任务内容+任务等级+任务状态+

悬赏金钱+是否付款

任务交易信息=任务交易时间+任务编号+接受任务的杀手信息

任务详细信息 = 任务信息+任务交易信息

说明：将任务信息呈现给用户或者杀手看

(11) 发布任务信息请求

位置：3.0→3.1

定义：任务信息请求=任务结束时间+任务内容+任务等级+任务状态+悬赏金钱

说明：用户发布自己想要发布的任务详情

(12) 核对成功记录

位置：3.2→3.3

说明：核对任务信息输入是否合法，然后录入

(13) 是否发布成功

位置：3.3→用户

定义：【是/否】

说明：向用户反馈任务是否发布成功

(14) 编辑任务请求信息

位置：4.0→4.1

定义：编辑任务请求信息 = 所编辑的任务编号+用户id

说明：根据所提供的任务编号查找任务信息，然后将更改后的信息录入

(15) 任务已核对记录

位置：4.1→4.2

定义：任务已核对记录 = 编辑任务请求信息+修改

说明：显示更改后的任务信息，方便用户核对

(16) 是/否修改通知

位置：4.2→用户

定义：【是/否】

说明：弹出窗口再次确认是否更改信息，确认则将新的任务信息覆盖原来的，否则这次操作无效，没有任何改变

(17) 更改任务状态请求

位置：5.0→5.1

定义：更改任务状态请求 = 任务编号+任务状态+任务等级+杀手id+杀手等级

说明：根据任务编号找到所要更改的任务信息与杀手信息，该修改信息是否正确（杀手是否够等级接受任务，任务目前状态是否可以被修改为该信息），并传给管理员

(18) 综合信息

位置：5.1→管理员

定义：综合信息 = 任务信息 + 杀手信息

说明：将更改后的任务信息反馈给管理员

(19) 核实更改情况属实

位置：管理员→5.2

说明：管理员核实任务状态是否属实

(20) 是/否修改通知

位置：5.2→杀手

定义：【是/否】

说明：弹出窗口再次确认是否更改任务信息状态，确认则将新的任务信息覆盖原来的，否则这次操作无效，没有任何改变

(21) 查看杀手榜请求信息

位置：6.0→6.1

定义：查看杀手榜请求信息 = 请求信息 + 杀手id

说明：请求查看杀手榜信息

(22) 检索成功信息

位置：6.1→6.2

定义：检索成功信息= 杀手榜前十id+该查询杀手是否包括在内

说明：查询杀手榜前十信息与该杀手所在排名信息，并显示查询结果

(23) 杀手榜信息列表

位置：6.2→杀手

定义：杀手榜信息列表 = 代号（昵称）+等级+武器+已完成任务数

说明：以列表的形式反馈杀手榜信息

(24) 查看杀手更改状态信息请求

位置：7.1→7.1.1

定义：杀手更改状态请求=杀手id+目前任务状态+更改任务状态+任务编号

说明：管理员接受并查看杀手发来的状态更改信息

(25) 审核通过通知

位置：7.1.1→数据库

说明：管理员审核通过，数据库录入更改后信息

(26) 管理用户请求

位置：7.1→7.1.2

定义：管理用户请求 = 用户账号+账号状态

说明：管理员可以管理用户账号的状态

(27) 管理杀手请求

位置：7.1→7.1.3

说明：管理员根据线下与杀手签订好的合同，对杀手信息进行填写

(28) 更改用户账号状态信息

位置：7.1.2→7.1.3

定义：更改用户账号状态请求信息 = 用户账号 + 更改后的状态

(29) 修改权限请求

位置：7.1→7.1.3.1

定义：修改权限请求=管理员账号+修改的权限

说明：管理员接受并查看修改权限请求

(30) 新权限记录表

位置：7.1.3.1→7.1.3.2

说明：将更改后的权限录入并覆盖原有的权限，并以表格的形式输出修改后的内

容

(31) 是/否成功修改通知

位置：7.1.3.2→管理员

定义：【是/否】

说明：弹出窗口再次向管理员确定是否修改

(32) 查看个人盈利状况请求信息

位置：9.0→9.1

定义：个人盈利状况=id+财富值

说明：根据杀手id查询杀手的财富值

(33) 检索信息成功

位置：9.1→9.2

说明：显示杀手的总财富值，以及每一个任务的金额

(34) 查看杀手信息请求

位置：用户→10.1

定义：查看杀手信息请求=id

说明：根据id查询杀手信息

(35) 检索信息成功

位置：10.1→10.2

说明：成功找到杀手信息

(36) 杀手信息列表

位置：10.2→用户

定义：杀手信息=账号（id）+密码+头像+代号（昵称）+等级+武器+目前状况+已完成任务数+总共可执行任务数

说明：将杀手信息以列表形式反馈给用户

(27) 编辑杀手信息请求

位置：杀手→10.3

定义：编辑杀手信息请求信息=头像+代号（昵称）+武器+风格特点

说明：杀手修改编辑自己的身份信息

(38) 身份验证成功

位置：10.3→数据库

定义：身份验证=账号（id）+密码

说明：杀手的账号（id）和密码正确，则将更改后的信息录入数据库

(39) 编辑成功通知

位置：数据库→杀手

说明：向杀手显示编辑修改成功后的信息

## 2.3.2数据储存定义（完成者：姓名 张若琦 学号14108106）

(1) 用户记录

输入：1.2，

输出：1.1，7.1.2，

数据结构：用户编号+账号+密码+头像+发布的任务编号+账号状态

存取方式：主要是检索处理，以随机检索为主

说明：编号具有唯一性和非空性，密码必须满8位，账号状态有“活跃”和“禁封”

(2) 管理员记录

输入：

输出：1.1，7.1.3.2，

数据结构：管理员编号+管理员账号+管理员密码

存取方式：主要是检索处理；以随机检索为主

说明：编号具有唯一性和非空性

### (3) 杀手记录

输入：10.3，

输出：1.1，10.1，

数据结构：杀手编号+账号（id）+头像+代号（昵称）+等级+武器+目前状况+已完成任务数+总共可执行任务数+财富值

存取方式：主要是检索处理；以随机检索为主

说明：编号和代号具有唯一性和非空性，等级由后台判断并输入

### (4) 任务信息记录

输入：3.3，4.2，5.2，

输出：2.1，4.1，5.1，7.1.1

数据结构：任务编号+任务开始时间+任务结束时间+任务内容+任务等级+任务状态+悬赏金额+是否付款+接受任务的杀手信息

存取方式：已更新操作为主：随机检索

说明：任务编号具有唯一性和非空性，任务内容和悬赏金钱具有非空性，任务等级由后台判断并输入、任务开始时间由用户提交时间为准

### (5) 杀手排名记录

输出：6.1，

数据结构：杀手编号+杀手id+代号+排名

存取方式：主要是检索处理；以随机检索为主

说明：杀手排名根据财富值进行从大到小的排序

### (6) 杀手盈利记录

输入：5.2

输出：9.1

数据结构：杀手编号+id+财富值

存取方式：已更新操作为主：随机检索

说明：杀手编号和id具有唯一性和非空性，财富值为每一个完成任务的悬赏金额的85%扣除失败任务的悬赏金额的5%

### (7) 任务交易记录

输入：5.2

输出：2.1

数据结构：任务交易时间+任务编号+杀手id+任务状态

存取方式：以更新操作为主：随机检索

说明：任务交易时间为杀手领取任务时间

### 2.3.3主要处理过程（完成者：姓名 彭薇 学号 14108303）

#### (1) 验证信息

输入：用户记录，杀手记录，管理员记录

输出：无该用户信息，验证成功数据

处理说明：通过用户信息，杀手信息，管理员信息表：检索用户是否符合信息表

#### (2) 用户注册

输入：无该用户信息

输出：DQ，注册成功数据

处理说明：实现用户信息的简历，增删改和维护功能

#### (3) 进入系统

输入：验证成功数据，注册成功数据

输出：是否验证成功，登入用户界面

处理说明：确认用户符合条件，进入用户系统界面

#### (4) 审核任务状态修改信息、任务修改信息

输入：D4，编辑任务信息请求、更改任务状态请求

输出：审核通过通知

处理说明：根据用户反馈的信息模糊查询任务信息，并审核是否合法。根据杀手反馈的信息模糊查询任务信息，并审核状态更改是否属实。

#### (5) 审核用户基本信息

输入：D1

输出：管理用户请求

处理说明：建立用户信息表，对用户编号：实现对用户基本信息的查看。根据用户模糊查询用户的信息。

#### (6) 增添杀手操作

输入：D3

输出：管理杀手请求

处理说明：建立用户信息表，根据线下合约信息进行填写以及对用户进行编号：实现对杀手基本信息的查看。根据杀手模糊查询杀手的的信息。

#### (7) 接受请求

输入：修改权限请求

输出：新权限记录表

处理说明：建立权限记录表，实现对用户的权限功能

#### (8) 检索任务信息

输入：查询任务信息请求，修改任务信息请求

输出：检索成功信息



(10) 显示任务信息

输入：检索成功信息，D4

输出：任务信息列表，所要编辑的任务信息，任务交易订单

处理说明：建立任务交易订单，根据任务交易订单编号来确定交易信息，实现任务信息的增删改的功能并建立任务信息列表。

### 3 系统模块划分

#### 3.1系统结构图（完成者：姓名 张若琦 学号14108106）



#### 3.2各模块功能概述（完成者：姓名 张若琦 学号14108106）

**登录系统模块：**主要由管理员管理、登录日志管理、杀手管理、用户管理

**用户发布任务信息模块：**主要用与用户发布自己的需求任务

**查询浏览任务信息模块：**主要用于杀手和用户查询浏览自己的任务以及用户进行近期的任务查询

**杀手榜排名模块：**主要用于定期给杀手进行排名

**查询编辑杀手信息管理：**主要用于用户、杀手查询杀手信息以及编辑杀手信息

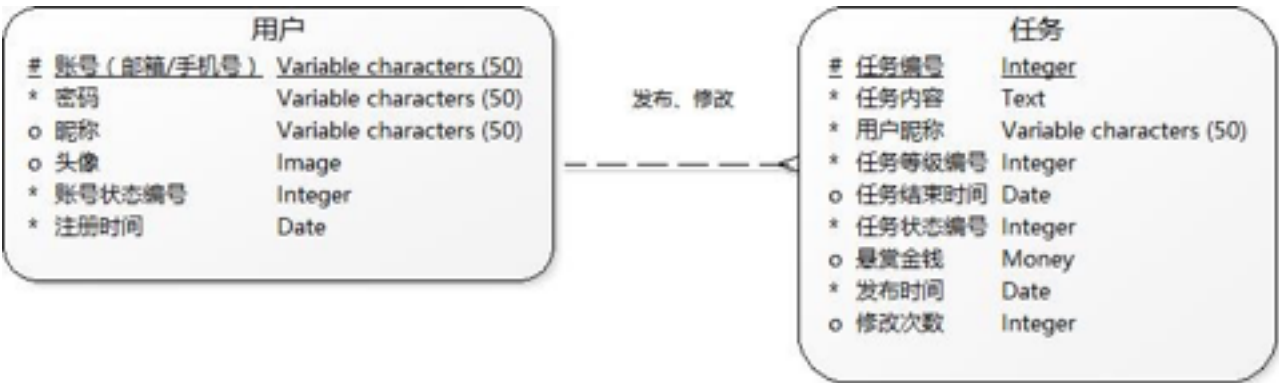
**编辑管理任务模块：**主要用于杀手对于任务的内容进行管理

**任务交易模块：**为有需求的用户提供交易功能

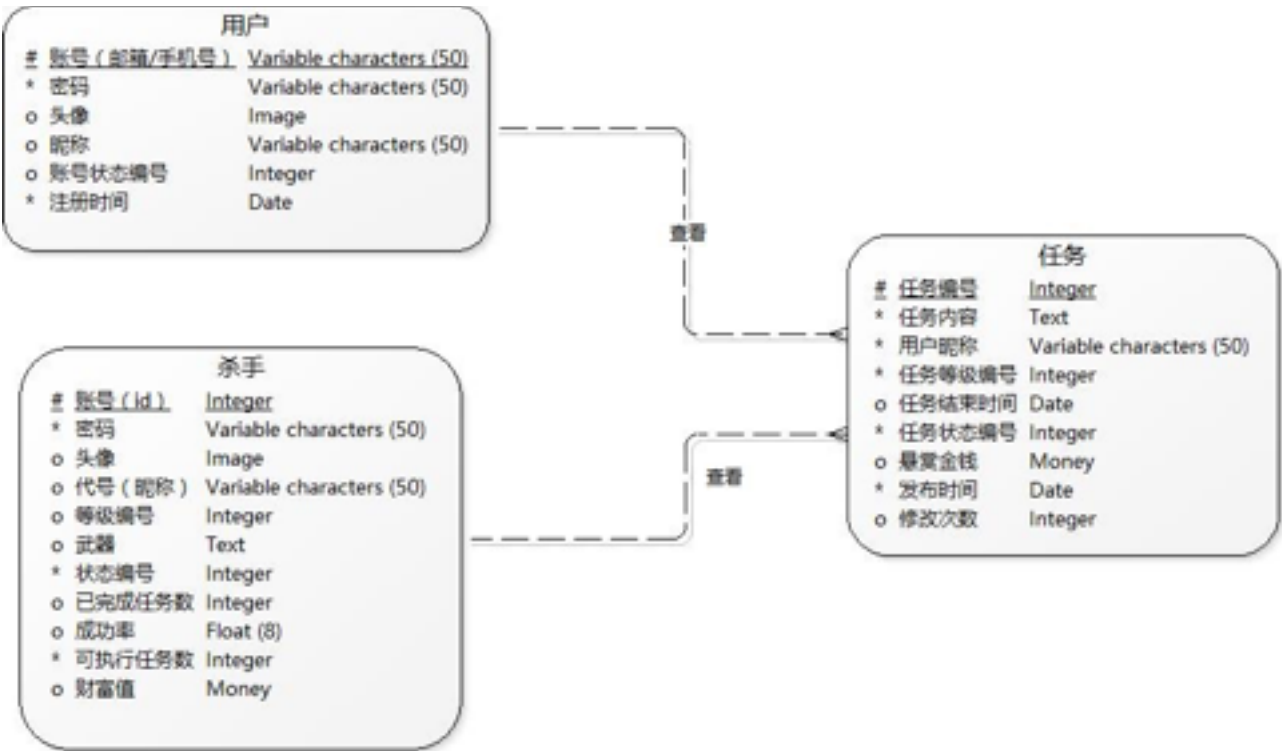
**杀手个人盈利查询模块：**查询杀手每次任务的盈利状况和总盈利状况

**安全管理模块：**保护用户和杀手的隐私以及账户安全

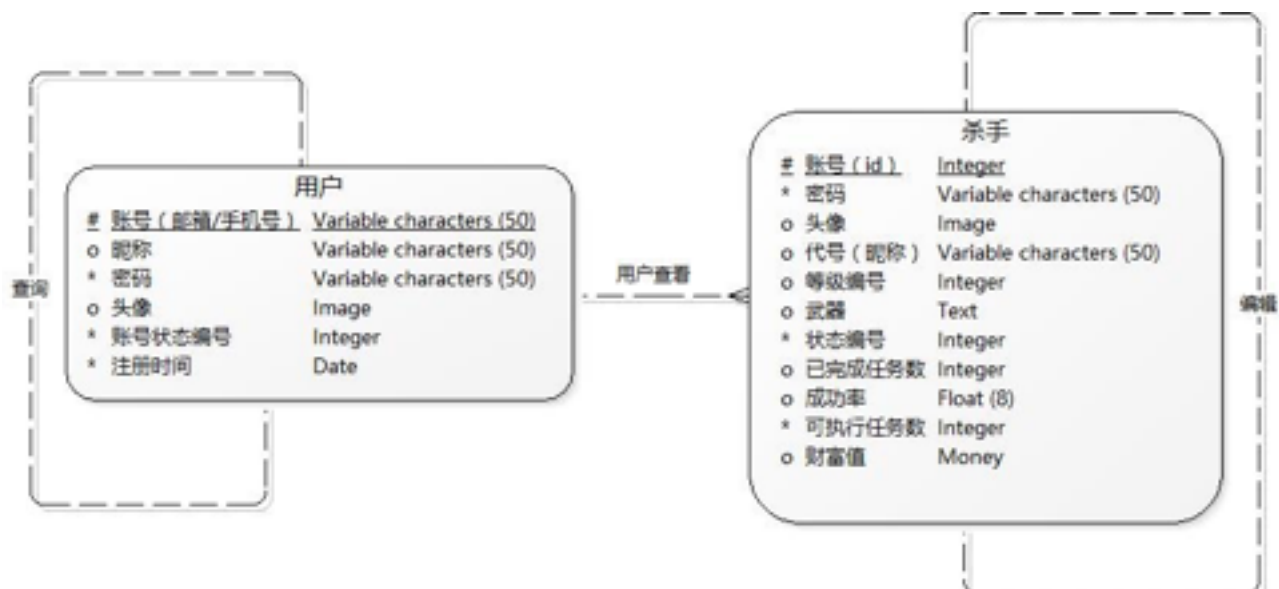
4 概念结构设计（完成者：姓名 梁冰 学号14108302）



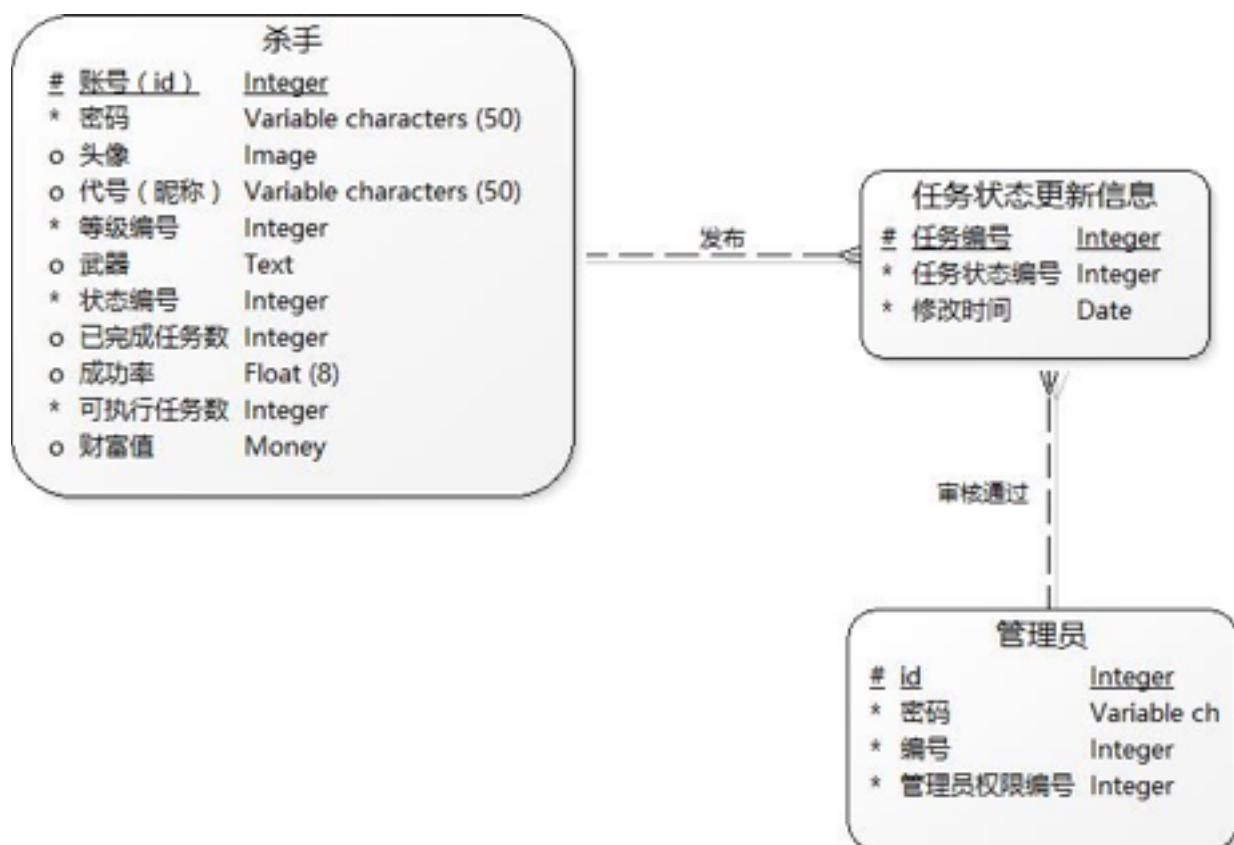
1.0用户发布修改任务管理分E-R图



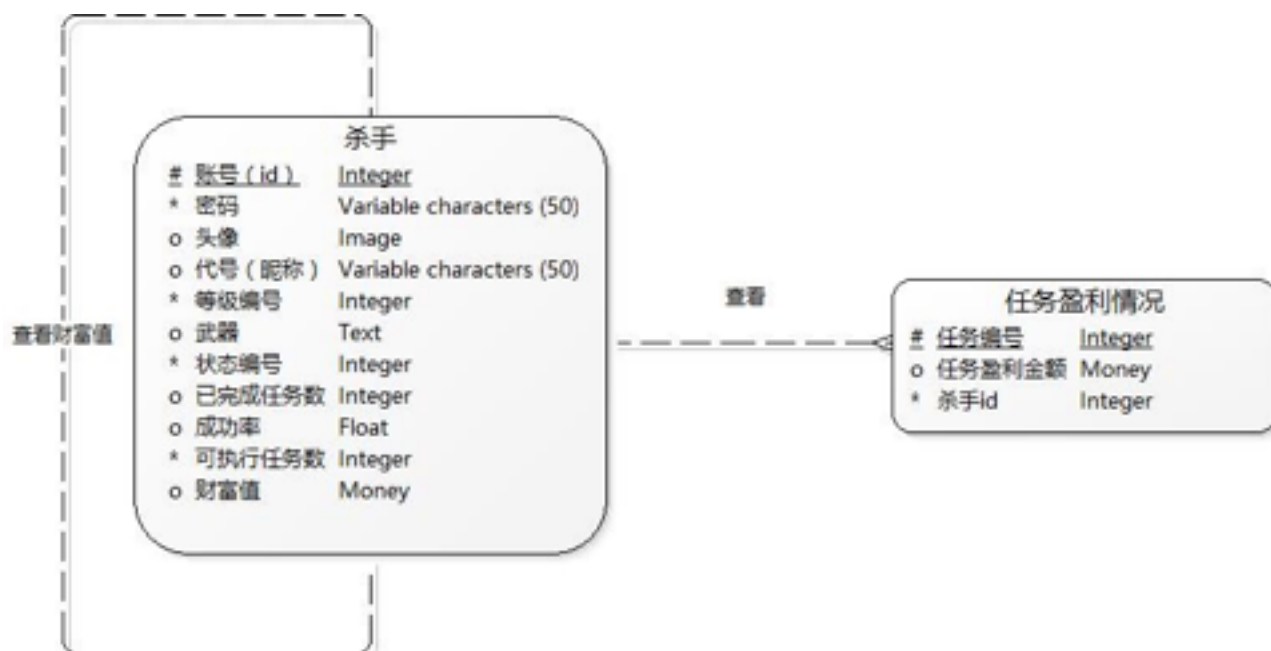
2.0查看任务信息分E-R图



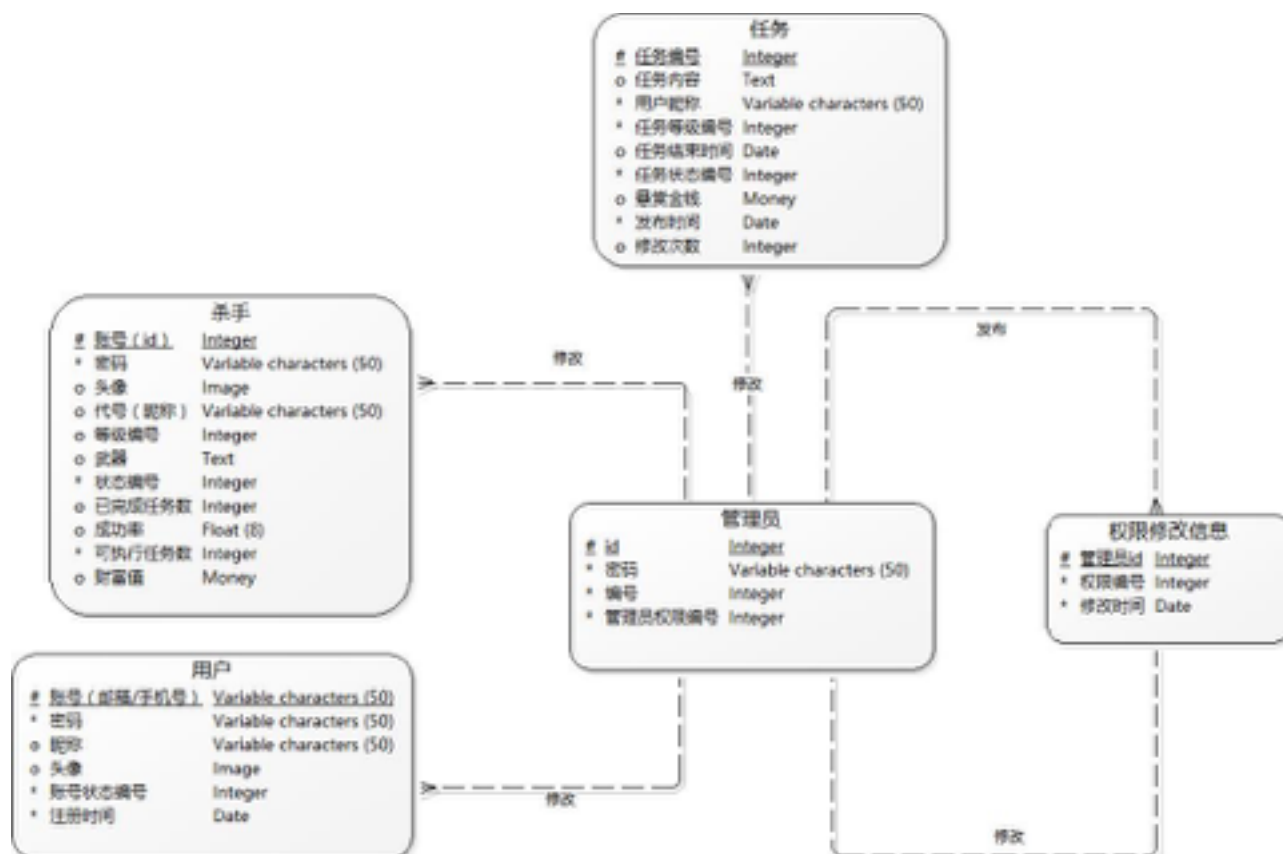
3.0用户查询、杀手编辑信息管理分E-R图



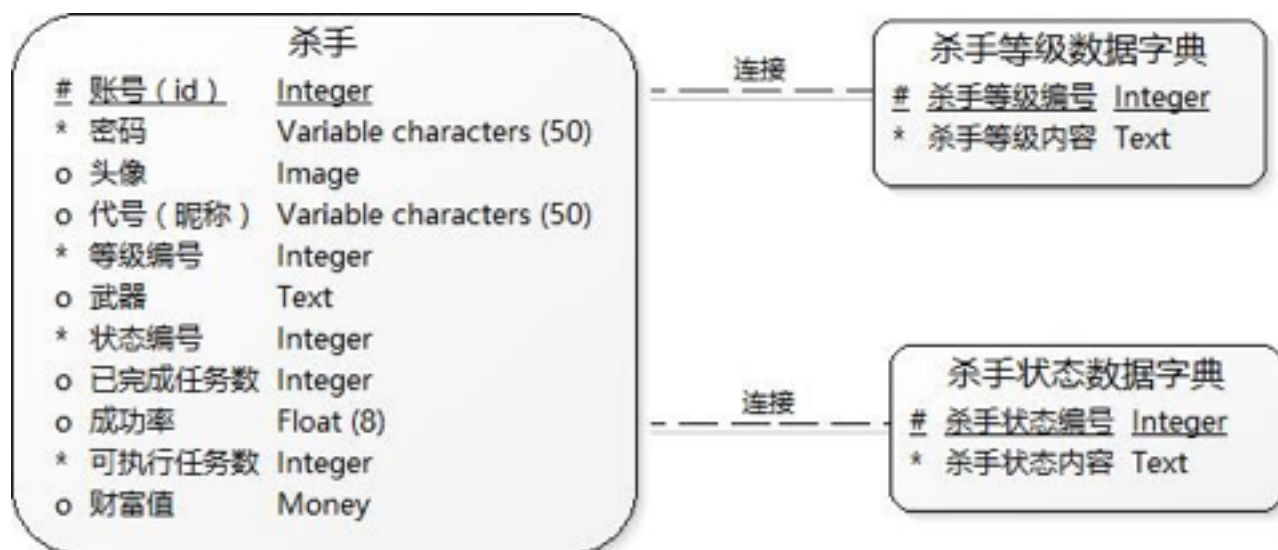
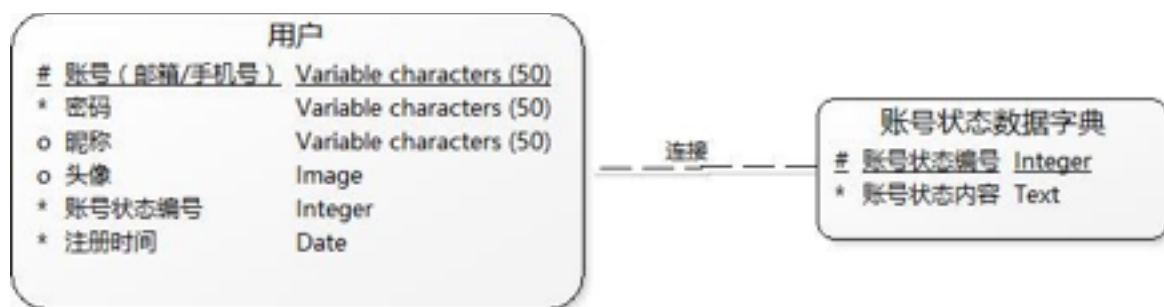
4.0杀手任务交易管理分E-R图



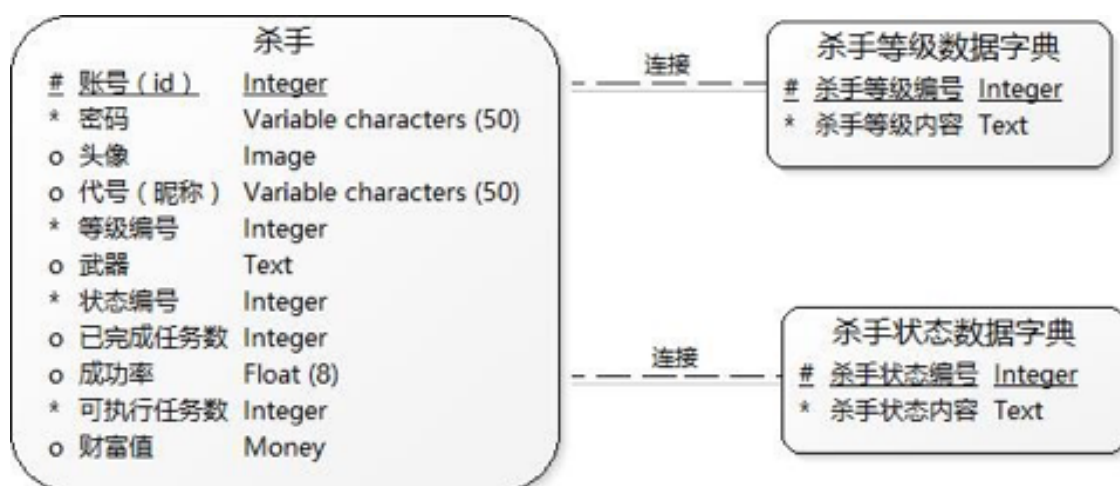
5.0杀手个人盈利查询管理分E-R图



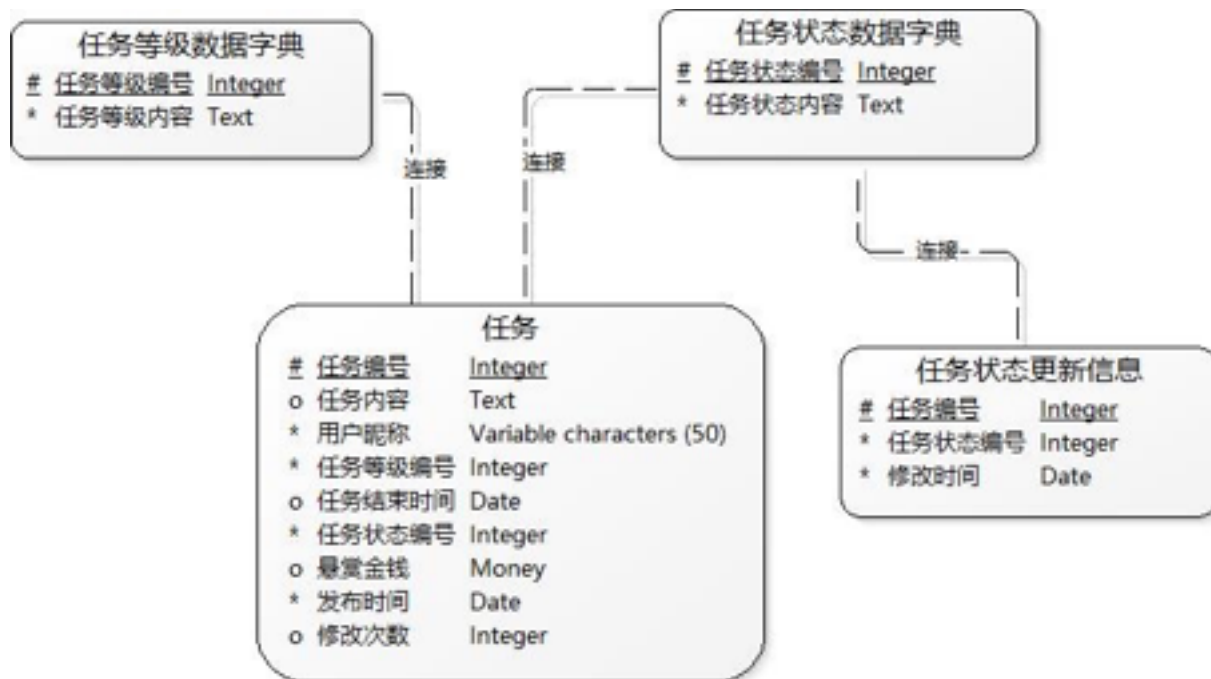
6.0管理员安全管理分E-R图



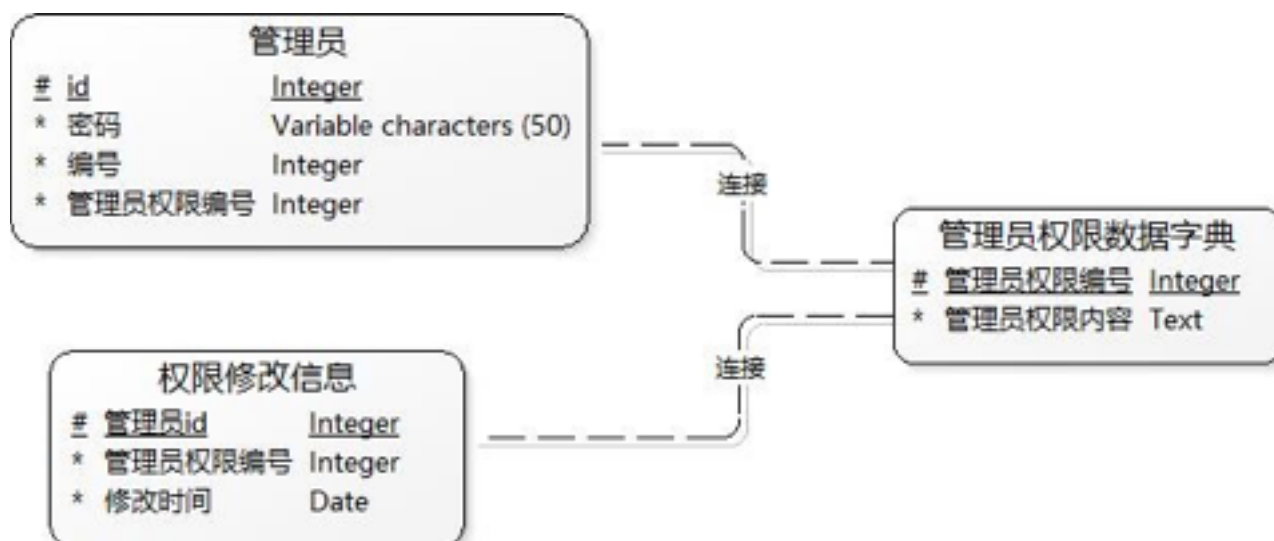
7.0用户数据字典



8.0杀手数据字典



## 9.0任务数据字典



## 10.0管理员数据字典





## 5 逻辑结构设计

### 5.1逻辑结构设计（完成者： 姓名 彭薇 学号 14108303）

用户(USER)				
字段名	数据类型	长度	说明	描述
UserID	Varchar	50	不空，主键	用户账号，为邮箱或手机号
Password	Varchar	50	不空	用户密码
Name	Varchar	50		用户昵称
Photo	Image			用户头像
IDStatusNum	Integer		不空	账号状态编号
regTime	Date		不空	注册时间

杀手(KILLER)				
字段名	数据类型	长度	说明	描述
KillerID	Integer		不空，主键	杀手账号，由后台录入
Password	Varchar	50	不空	杀手密码
Photo	Image			杀手头像
Name	Varchar	50	不空	杀手代号（昵称）
KRankNum	Integer		不空	杀手等级编号
Weapon	Varchar	30		杀手使用的武器
KStatusNum	Integer		不空	杀手状态编号
Done	Integer			杀手已完成任务数
SuccessRate	Float	8		杀手成功率
Executable	Integer		不空	可执行任务数
Money	Money			财富值

任务(TASK)				
字段名	数据类型	长度	说明	描述
TaskNum	Integer		不空，主键	任务编号
TaskCon	Text		不空	任务内容
UserID	Varchar	50	不空，外键	用户账号
TRankNum	Integer		不空	任务等级编号
EndTime	Date			任务结束时间
TStatusNum	Integer		不空	任务状态编号
Tmoney	Money			悬赏金额
StartTime	Date		不空	发布时间
ChangeTimes	Integer			修改次数



管理员(ADMIN)				
字段名	数据类型	长度	说明	描述
AID	Integer		不空，主键	管理员ID，由后台录入
Password	Varchar	50	不空	管理员密码
Num	Integer		不空	管理员编号
LimitNum	Integer		不空	管理员权限编号

账号状态数据字典(IDStatus)				
字段名	数据类型	长度	说明	描述
IDStatusNum	Integer		不空，外键	账号状态编号
IDStatusCon	Varchar	30	不空	账号状态内容

任务等级数据字典(TaskRank)				
字段名	数据类型	长度	说明	描述
TRankNum	Integer		不空，外键	任务等级编号
TRankCon	Varchar	30	不空	任务等级内容

任务状态数据字典(TaskStatus)				
字段名	数据类型	长度	说明	描述
TStatusNum	Integer		不空，外键	任务状态编号
TStatusCon	Varchar	30	不空	任务状态内容

任务盈利情况(TaskProfit)				
字段名	数据类型	长度	说明	描述
KillerID	Integer		不空，外键	杀手账号
TaskNum	Integer		不空，外键	任务编号
TaskMoney	Money		不空	任务盈利金额

杀手等级数据字典(KillerRank)				
字段名	数据类型	长度	说明	描述
KRankNum	Integer		不空	杀手等级编号
KRankCon	Varchar	30	不空	杀手等级内容

杀手状态数据字典(KillerStatus)				
字段名	数据类型	长度	说明	描述
KStatusNum	Integer		不空	杀手状态编号
KStatusCon	Varchar	30	不空	杀手状态内容

英雄榜(RankList)				
字段名	数据类型	长度	说明	描述
Name	Varchar	50	不空	杀手代号（昵称）
Photo	Image			杀手头像
Rank	Integer		不空，主键	名次
Money	Money			财富值

任务状态更新记录(TaskUpdate)				
字段名	数据类型	长度	说明	描述
TaskNum	Integer		不空，外键	任务编号
TStatusNum	Integer		不空	任务状态编号
UpdateTime	Date		不空	修改时间

权限修改记录(AuthorityModify)				
字段名	数据类型	长度	说明	描述
AID	Integer		不空，外键	管理员ID
LimitNum	Integer		不空	管理员权限编号
ChangeTime	Date		不空	修改时间

管理员权限数据字典(AdminLimit)				
字段名	数据类型	长度	说明	描述
LimitNum	Integer		不空	管理员权限编号
LimitCon	Varchar	50	不空	管理员权限内容

**表格信息汇总：**

用户（用户账号，用户密码，用户昵称，用户头像，账号状态编号，注册时间）

杀手（杀手账号，杀手密码，杀手头像，杀手代号，杀手等级编号，杀手使用的武器，杀手状态编号，杀手已完成任务数，杀手成功率，可执行任务数，财富值）

任务（任务编号，任务内容，用户账号，任务等级编号，任务结束时间，任务状态编号，悬赏金额，发布时间，修改次数）

管理员（管理员ID，管理员密码，管理员编号，管理员权限编号）

账号状态数据字典（账号状态编号，账号状态内容）

任务等级数据（任务等级编号，任务等级内容）

任务状态数据字典（任务状态编号，任务状态内容）

任务盈利情况（杀手账号，任务编号，任务盈利金额）

杀手等级数据字典（杀手等级编号，杀手等级内容）

杀手状态数据字典（杀手状态编号，杀手状态内容）

英雄榜（杀手代号，杀手头像，名次，财富值）

任务状态更新记录（任务编号，任务状态编号，修改时间）

权限修改记录（管理员ID，管理员权限编号，修改时间）

管理员权限数据字典（管理员权限编号，管理员权限内容）

注意：下划线为主键

## **5.2模式优化（完成者：姓名 彭薇 学号 14108303）**

在用户表中不存在部分函数依赖和传递函数依赖，所以达到了3NF。

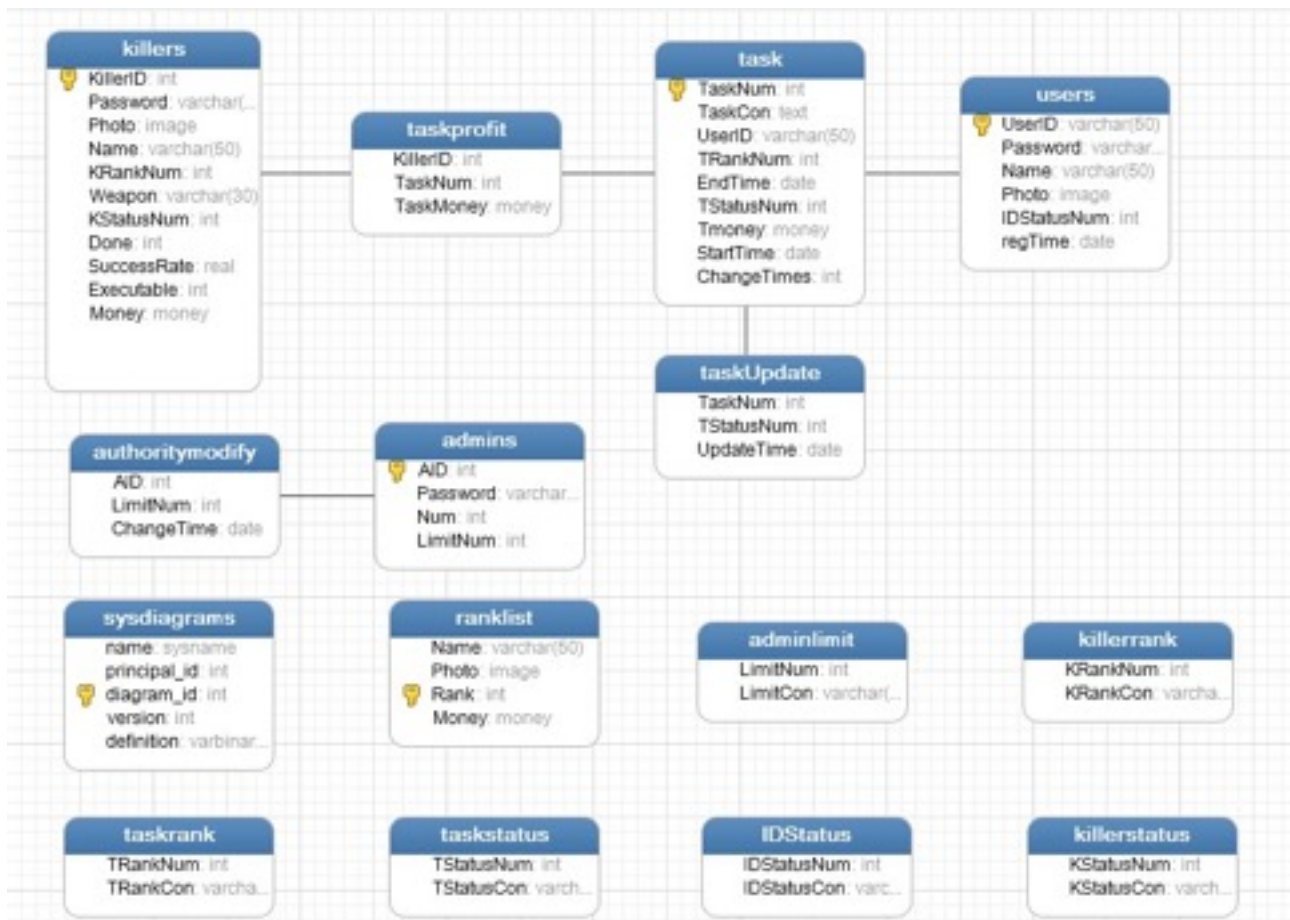
在杀手表中不存在部分函数依赖和传递函数依赖，所以达到了3NF。

在任务表中不存在部分函数依赖和传递函数依赖，所以达到了3NF。

在管理员表中不存在部分函数依赖和传递函数依赖，所以达到了3NF。

同样，在英雄榜表中不存在部分函数依赖和传递函数依赖，所以均达到了3NF。

## **5.3表关系图（完成者：姓名 彭薇 学号 14108303）**



## 6 物理设计

### 6.1索引（完成者：姓名 朱湘如 学号 14108107）

create Unique Index NID ON task(TaskNum)

create Unique Index TID ON taskupdate(TaskNum)

create Unique Index FID ON taskprofit(TaskNum)

go

### 6.2表的创建(完成者：姓名 朱湘如 学号 14108107)

create database killer

go

use killer

go

create table users(

```
UserID    varchar(50) not null primary key,  
Password  varchar(50) not null,  
Name      varchar(50) ,  
Photo     image,  
IDStatusNum  integer not null,  
regTime   date   not null,  
)  
go
```

```
create table killers(  
KillerID integer not null primary key,  
Password varchar(50) not null,  
Photo     image,  
Name      varchar(50) not null,  
KRankNum   integer not null,  
Weapon     varchar(30),  
KStatusNum integer not null,  
Done       integer,  
SuccessRate float(8),  
Executable integer not null,  
Money      money,  
)  
go
```

```
create table task(  
TaskNum      integer not null primary key,  
TaskCon      text not null,  
UserID       varchar(50) not null references users(UserID),  
TRankNum     integer not null,  
EndTime      date,  
TStatusNum   integer not null,  
Tmoney       money,  
StartTime    date not null,  
ChangeTimes  integer,  
)  
go
```

```

create table admins(
AID          Integer      not null primary key,
Password     Varchar(50) not null,
Num          Integer      not null,
LimitNum     Integer      not null,

)
go

create table IDStatus(
IDStatusNum   Integer not null,
IDStatusCon   Varchar(30) not null,
)
go

create table taskrank(
TRankNum      Integer      not null,
TRankCon      Varchar(30) not null,
)
go

create table taskstatus(
TStatusNum    Integer      not null,
TStatusCon    Varchar(30) not null,
)
go

create table taskprofit(
KillerID Integer not null references killers(KillerID),
TaskNum      Integer      not null references task(TaskNum),
TaskMoney    Money        not null,
)
go

create table killerrank(
KRankNum      Integer      not null,
KRankCon      Varchar(30) not null,
)
go

create table killerstatus(

```

```

KStatusNum    Integer    not null,
KStatusCon    Varchar(30) not null,
)
go
create table ranklist(
Name    Varchar(50) not null,
Photo    Image,
Rank    Integer    not null primary key,
Money    Money,
)
go

create table taskUpdate(
TaskNum        Integer not null references task(TaskNum),
TStatusNum    Integer not null,
UpdateTime    Date    not null,
)
go

create table authoritymodify(
AID            Integer not null references admins(AID),
LimitNum        Integer not null,
ChangeTime    Date    not null,
)
go

create table adminlimit(
LimitNum        Integer    not null,
LimitCon        Varchar(50) not null,
)
go

```

## 6.3视图的创建（完成者：姓名 梁冰 学号14108302）

--账目视图

```
create View view_profit
```



```
as
select taskprofit.TaskNum,TaskCon,TaskMoney,EndTime,KillerID,UserID
from taskprofit,task
where taskprofit.TaskNum=task.Tmoney
with check option
```

--任务视图

```
create View view_task
as
select task.TaskNum,TaskCon,TMoney,StartTime,EndTime,changetimes,
TRankCon,TStatusCon,userID,KillerID
from task,TaskRank,TaskStatus,TaskProfit
where task.TRankNum=TaskRank.TRankNum
and task.TStatusNum=taskstatus.TStatusNum
and task.TaskNum=taskprofit.TaskNum
with check option
```

## 6.4存储过程或触发器的创建

### 6.4.1存储过程的创建（完成者 姓名：张若琦 学号：14108106）

--TASK表的储存过程创建代码：

```
CREATE PROCEDURE TASK_PRO
@N INT
AS
SELECT * FROM TASK
WHERE TaskNum = @N
GO
```

--TaskProfit表储存过程创建代码：

```
CREATE PROCEDURE TaskProfit_PRO
```

```
@N INT
```

```
AS
```

```
SELECT * FROM TaskProfit
```

```
WHERE TaskNum = @N
```

```
GO
```

--TaskUpdate表的储存过程创建代码：

```
CREATE PROCEDURE TaskUpdate_PRO
```

```
@N INT
```

```
AS
```

```
SELECT * FROM TaskUpdate
```

```
WHERE TaskNum = @N
```

```
GO
```

#### 6.4.2触发—触发器的创建：（完成者 姓名：彭薇 学号：14108303）

-- 当杀手财富值达到一定值时，杀手等级应及时更新。

--以下为killers表的Update触发器：

```
create trigger tg_rank on killers
```

```
for update
```

```
as
```

```
    declare @newMoney money
```

```
    select @newMoney = Money from inserted;
```

```
    if (@newMoney between 0 and 500)
```

```
        begin
```

```
            update killers set KRankNum=1
```

```
        end
```

```
    else if (@newMoney between 500 and 2000)
```

```
begin
    update killers set KRankNum=2
end
else if (@newMoney between 2000 and 5000)
begin
    update killers set KRankNum=3
end
else if (@newMoney between 5000 and 12000)
begin
    update killers set KRankNum=4
end
else if (@newMoney between 12000 and 25000)
begin
    update killers set KRankNum=5
end
else if (@newMoney between 25000 and 50000)
begin
    update killers set KRankNum=6
end
else if (@newMoney between 50000 and 100000)
begin
    update killers set KRankNum=7
end
else if (@newMoney between 100000 and 200000)
begin
    update killers set KRankNum=8
end
else if (@newMoney between 200000 and 500000)
begin
    update killers set KRankNum=9
end
else if (@newMoney between 500000 and 1000000)
begin
    update killers set KRankNum=10
end
```

go

--根据任务金额，相应划分任务等级

--以下为task表的Update触发器：

create trigger tg\_TaskRank on task

for update

as

declare @newMoney money

select @newMoney = Tmoney from inserted;

if (@newMoney between 0 and 15)

begin

update task set TRankNum=1

end

else if (@newMoney between 15 and 50)

begin

update task set TRankNum=2

end

else if (@newMoney between 50 and 200)

begin

update task set TRankNum=3

end

else if (@newMoney between 200 and 400)

begin

update task set TRankNum=4

end

else if (@newMoney between 400 and 800)

begin

update task set TRankNum=5

end

else if (@newMoney between 800 and 1500)

begin

update task set TRankNum=6

end

else if (@newMoney between 1500 and 3000)

begin

update task set TRankNum=7

```
        end
    else if (@newMoney between 3000 and 5000)
        begin
            update task set TRankNum=8
        end
    else if (@newMoney between 5000 and 10000)
        begin
            update task set TRankNum=9
        end
    else if (@newMoney between 10000 and 50000)
        begin
            update task set TRankNum=10
        end
    end
```

## 7 课程总结

### 数据库课程总结：张若琦

首先，感触最深的是最后想想原本只需要一个星期或者两个学期的事情，我们竟然用了一个学期。当然，不可避免的是还在学习过程中，但是有时候做到后面再回过头对前面进行修改时真的浪费了大量的时间。如果一开始下手的时候过于轻快，或者像这次基本无法进行市场调查的时候，就忙着开发，会给后面的工作带来很大麻烦，甚至推盘重来。所以，刚开始的时候，我们浪费了很大的时间，最后专门抽出了一个周末，重新做需求分析，才赶上进度。

第二，沟通也很重要。计划总是不全面的，所以在设计时，如果遇到困难，组员之间要及时沟通，才可以及时修改进度表。并且，作为学生，我们做的东西是不全面的，要利用好每节课的上机时间，及时与老师沟通，听取建议，及时修改是做好的关键。

第三，数据字典的使用很重要。很多时候你存储的内容不一定是真正的内容而是编号，比如管理员的权限管理。用编号去代替文字，再查询数据字典得到真正的权限。

在做数据库设计的过程中，也更正了许多自己对数据库的理解错误。比如，在ER图中每一个你看到的表格，都是一个实体，其实很多东西就没必要表现出来，那些你可以调用到的，做成视图就好，没必要作为实体等等。

不过有些地方我还是有些疑惑，比如在修改数据库的时候，要求达到第三范式，可是最近在做java web项目的时候，发现其实有时候数据冗余会减少数据库的访问，减少连表查询。也许随着磁盘容量的增加以及云数据库的使用，冗余也不是不可以，数据库是为了项目服务的，项目实现快很重要。

### 数据库个人感想：朱湘如

数据库程序设计这门课，我们通过分组进行小组项目，不像短学期的小项目一样，是个人的，这个更加看重的是小组成员之间的交流沟通与协作。因为我们组的都是一个寝室的人，我们的交流更加方便。所我们的主题是杀手预约系统，虽然不是老师给的那几个选题之一，网上也找不到相应的模板参照，但是我们依靠对这个内容的兴趣和突破天际的脑洞，我觉得我们前期的需求分析做的还是可以的。后来在做概念设计的时候，因为第一次接触，没有经验。我们采取了数据流图，数据字典，E-R让不同的人来，而不是分模块，让每个人负责自己那部分模块的数据流图，数据字典，E-R图。导致我们做数据字典和E-R图的时候，还要花很多心思去理解上一位画流图的想法，有时候还会出现理解错误的时候，导致效率低下。这个教训我们记住了，也算是学到了一点吧。数据库课程设计这门课，我们的进度跟着数据库的理论来的，基本老师上了什么内容，我们回去就可以在杀手预约系统里面进行操作。这样的话，我们既可以及时的巩固学过的知识，还有助于更加好的理解老师教过的内容。毕竟，只听老师说，自己不进行上手操作，是完全没有效果的。只有经过自己实践所习得的知识才是真正属于自己的，刻在脑子里的，不会忘记的太快。

### 数据库感想：梁冰

从大一的c语言、java、html课程等，数据库让我对编程有了更深刻和更加立体的理解。在这次的数据库小组设计实践中，我们充分发挥了自己的想象力，组长提出做“杀手管理系统”，虽然标题乍一看非常的标新立异，但本质上和其他管理系统没有区别，我们每个人参考样例的PPT，由组长分配任务，一开始我也是一头雾水，不知道从什么地方开始下手。但是随着数据库学习的深入和

组长数据流程图的完善，参考着网上的教程，一步一步的摸索，不断的完善自己的任务，有疑义的地方和其他人讨论修改。在多次修改后，终于得到了比较满意的结果。

在这次的设计实践中，我还明白了团队合作的重要性，从数据流图到数据字典，再到我负责的ER图，每个人的工作都有承上启下的作用，无论哪个单拎出来作业都是十分困难的，只有小组成员的合作努力才使这次设计实践变得不那么困难。不仅如此，当我对某些方面有疑问和不安的时候，和其他成员讨论后，很多问题都迎刃而解，有了其他成员，才让这次的项目顺利的完成下来。

这次的小组设计实践，让我学到了很多，特别是对数据库的操作上，光是靠书本和老师口头上的讲解，你以为自己懂了，但还是有很多隐藏在角落里一开始发现不了的问题，只有自己实际操作过，才能确实的明白到底这个作业是怎么样。

## 数据库设计感想：彭薇

本学期经过数据库课程设计的学习，我们小组完成了杀手预约管理系统的数据库设计过程。最开始的定题，我们在养猪管理系统、神盾局管理系统和杀手预约管理系统中抉择，最终选择了杀手预约管理系统。定题之后，需求分析耗费了不少时间，但也算顺利完成。其中，张若琦和我负责数据流图的工作。完成数据流图后，才对整个流程有了初步的了解，其中也修改过几次进行完善。系统模块划分部分轻松完成后，终于到了概念结构设计，也就是画E-R图的部分，也是很关键的一部分。E—R图的设计我感觉完成的不是很成功，总有些不足，但也找不出具体是哪里存在问题。之后的逻辑结构设计本应该可以由Powerdesigner直接导出，但是不知为何导出失败，只能手打。包括后面的表关系图，也是在Navicat中导入截取得到。（感觉Powerdesigner非常不好用，再也不会用这个软件= =）最后按照逻辑结构设计的表格用sql语句完成物理设计。整个课程设计也就终于完成了。

整个过程中，感受最深的是团队合作的重要性。一个项目中的任何一步，都有可能对后面的结果或任务产生影响。而在实际工作中，一个项目一般，都不太可能由一个人单独完成。这就涉及到多人合作。在团队合作中，一定要加强沟通。并不是只要负责好自己的分内事情就足够，也要了解其他人的任务进度、完成方式等。这样一来，既可以了解到别人的结果对我的任务的传递过程以及其中的影响，从而追究来龙去脉；又可以了解自己完成的任务对别人会产生怎样的影响，从而在自己完成任务时知道要注意哪些地方，以免给其他人造成不必要的影响。只有这样，项目的合作才能完成的更好。

## 8 致谢

感谢金洁洁老师一直以来对我们课程设计的关注与修改！