

系统设计与数据库设计 汇报

卡

卡

颂

汇报小组：unity从入门到入土

目录

CONTENTS

01 功能模块和层次设计

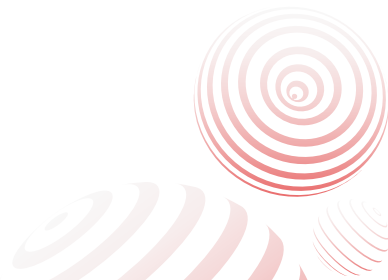
02 接口设计

03 E R 图与表结构

04 泳道图与数据流图

05 安全性和上节课剩余问题

本次报告主要说明游戏“卡卡颂”的系统结构与内部设计、数据库信息。主要介绍了系统的层次结构设计、功能模型和接口设计及验收标准，用于快速了解本项目的层次结构，有利于团队的开发与部署，给未来项目的实现提供了设计思路 and 具体标准。

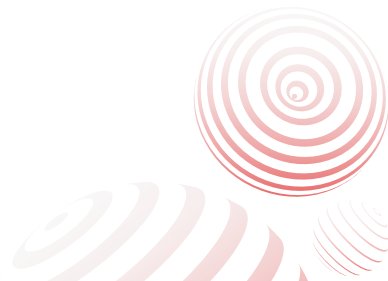
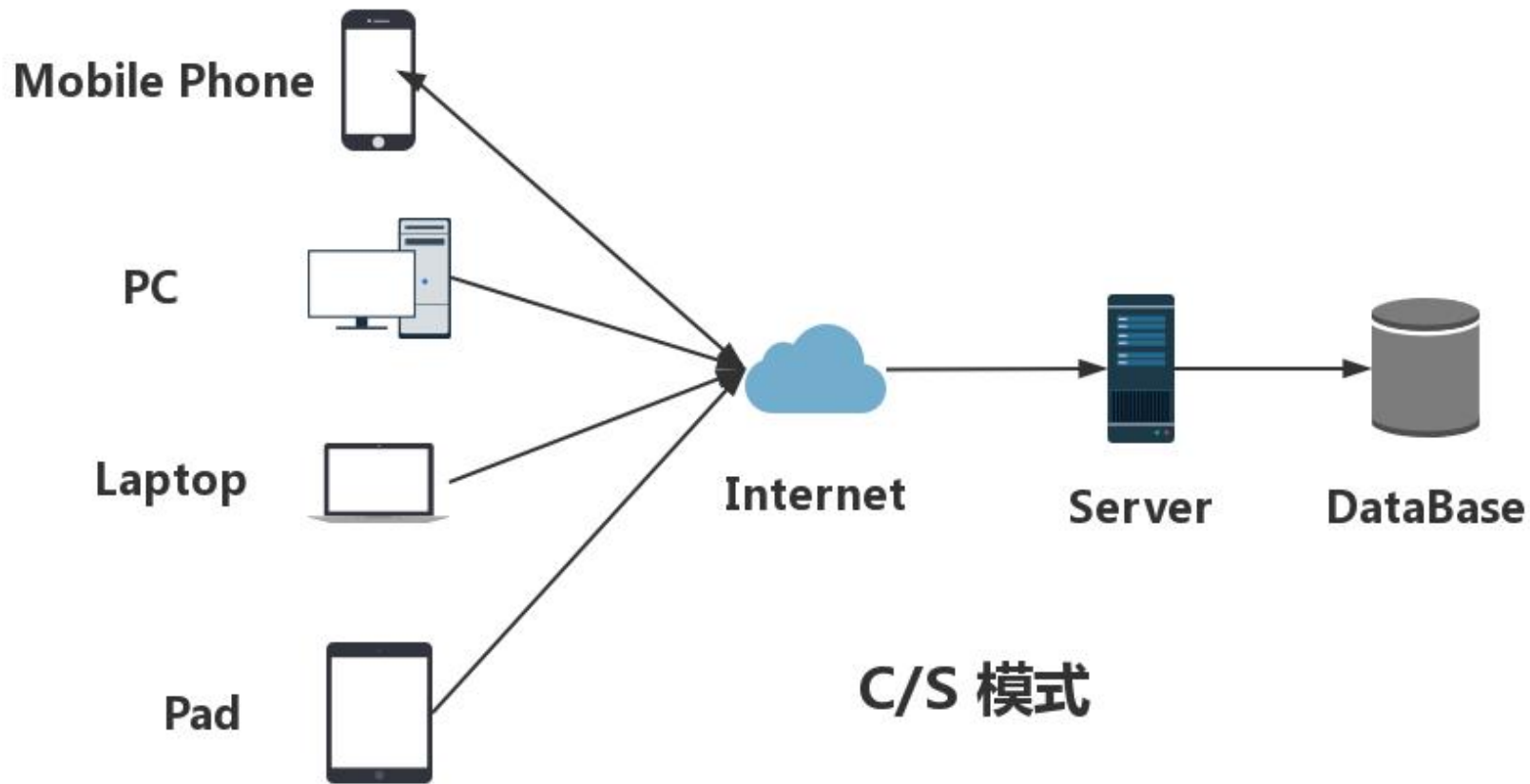


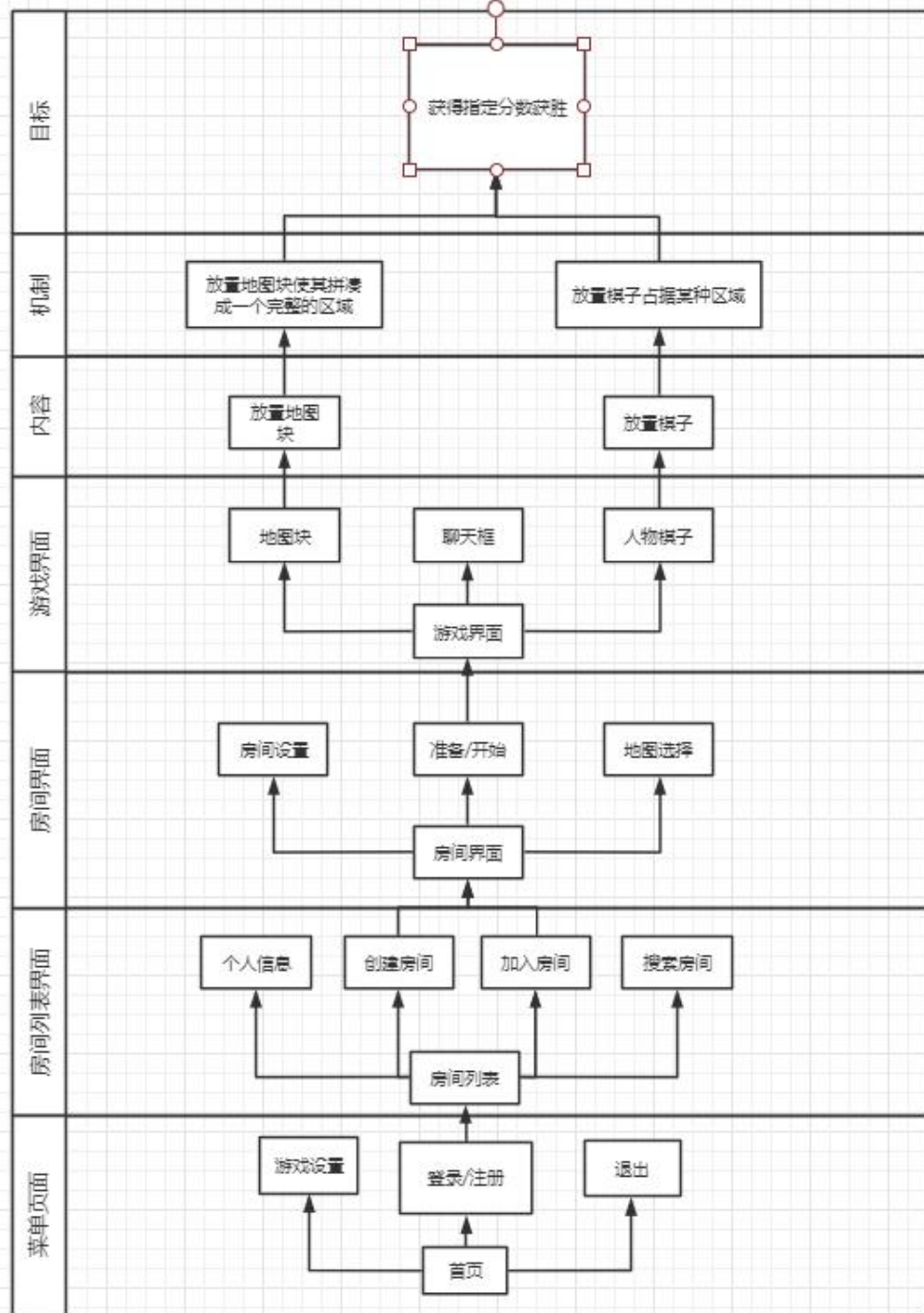
01

功能模块和层次设计

卡卡颂功能模块层次图







接口设计

02 接口设计

接口文档:

https://www.showdoc.com.cn/Carcassonne?page_id=6722817512873347



02 接口设计

注册

用户注册接口

简要描述

- 用户注册接口

请求URL

- `http://xx.com/offline/userRegister`

请求方式

- POST

参数

参数名	必选	类型	说明
accountNum	是	string	账号
password	是	string	密码
nickname	是	string	昵称
sex	否	string	性别
headPictureURL	是	string	头像资源地址

请求示例（未改）

```
{
  "error_code": 0,
  "data": {
    "uid": "1",
    "username": "12154545",
    "name": "吴系挂",
    "groupid": 2 ,
    "reg_time": "1436864169",
    "last_login_time": "0",
```

```
        login_time : 0 ,
    }
}
```

返回参数说明

参数名	类型	说明
code	int	状态码，返回200时为无异常，其余参考状态码说明
message	string	提示信息
token	string	该用户的临时签证
userInfo	JSONArray	用户的信息，以JSONArray的格式保存
accountNum	string	账号
nickname	string	昵称
sex	string	性别
level	string	段位
headPictureURL	string	头像URL
state	string	当前用户状态，有三种分别为wander、waitStart、playing，对应闲逛，房间中等待开始，进行游戏。注意在某种状态下才有权限请求对应状态提供的操作，比如wander下的用户只能请求“闲逛状态目录”的接口



查看个人信息



欢迎使用ShowDoc!

简要描述

- 查看个人信息

请求URL

- `http://xx.com/wander/getUserInfo`

请求方式

- POST

参数

参数名	必选	类型	说明
token	是	string	用户名

请求示例 (未改)

```
{
  "error_code": 0,
  "data": {
    "uid": "1",
    "token": "123456789"
```



欢迎使用ShowDoc!

简要描述

- 获取当前帧信息

请求URL

- `http://xx.com/playing/getFrameInfo`

请求方式

- POST

参数

参数名	必选	类型	说明
username	是	string	用户名
password	是	string	密码
name	否	string	昵称

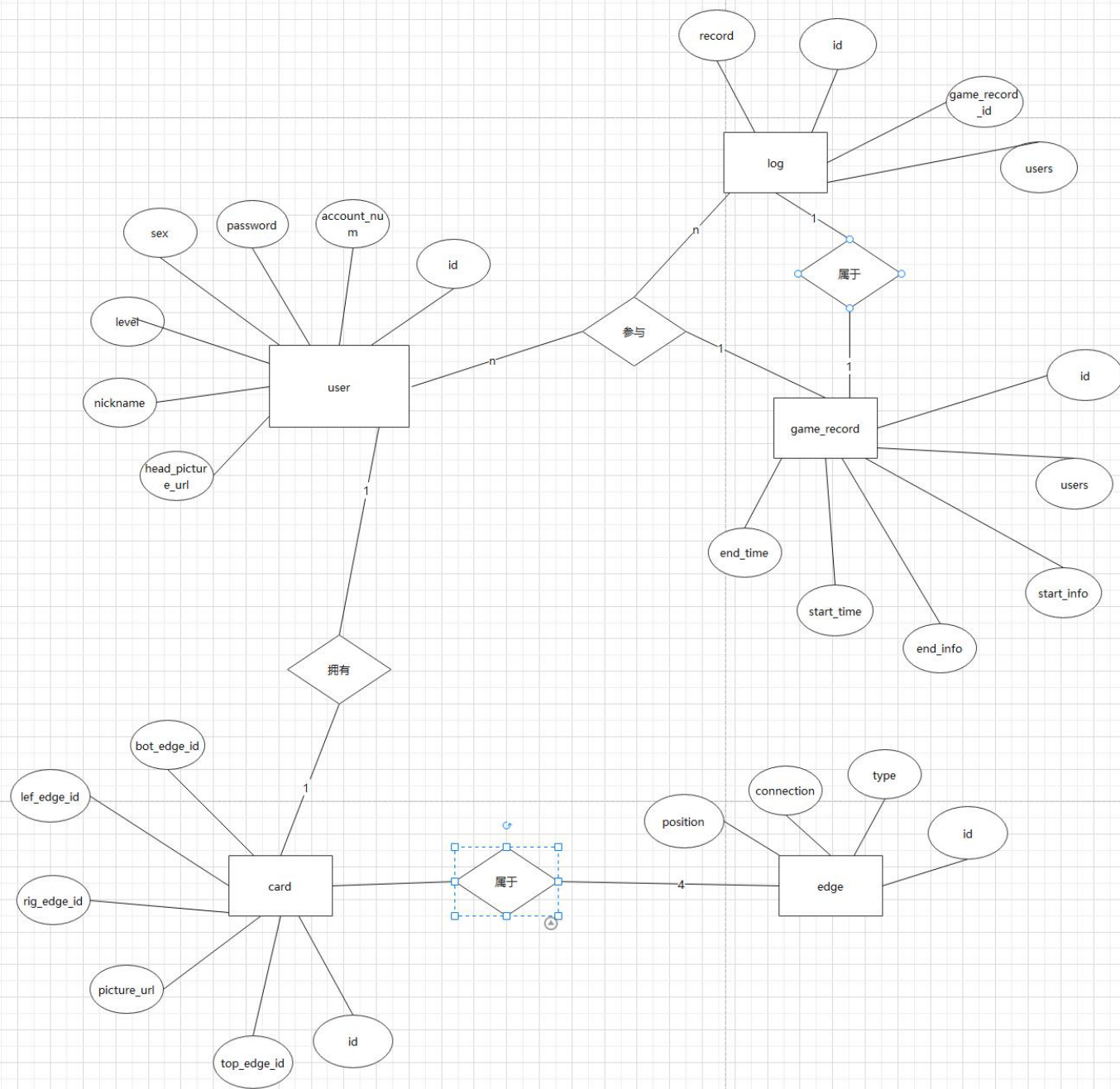
返回示例

```
{
  "error_code": 0,
  "data": {
    "uid": "1",
    "username": "12154545",
    "name": "吴系挂",
    "groupid": 2,
    "reg_time": "1436864169",
    "last_login_time": "0",
  }
}
```

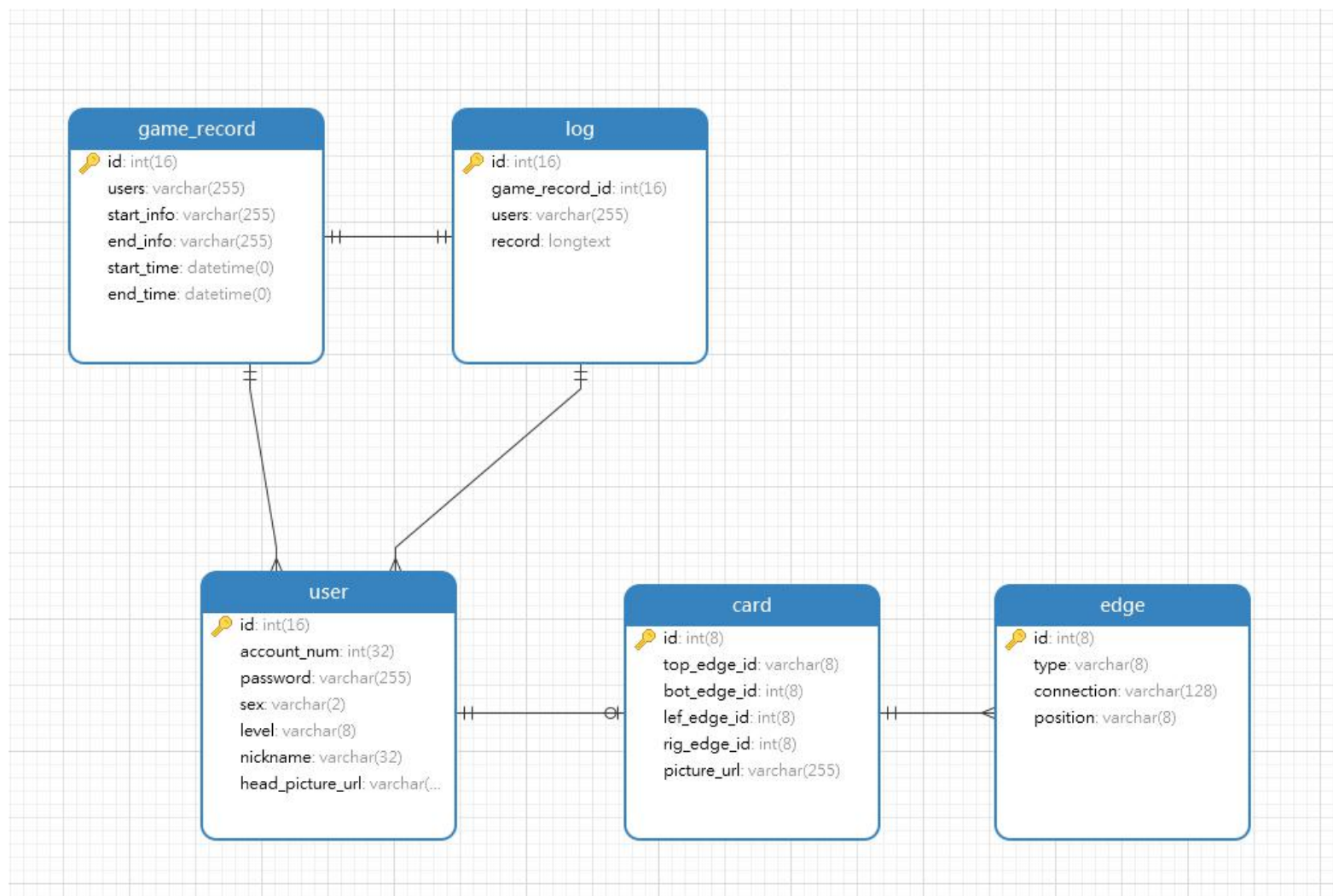


ER图与表结构

03 ER图



03 表结构 拓扑图



03 表结构

卡片基础信息表

字段	索引	外键	触发器	选项	注释	SQL 预览						
名						类型	长度	小数点	不是 null	虚拟	键	注释
id						int	8	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	card id 主键, 该表数据目前为只读
top_edge_id						varchar	8	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		上边的id
bot_edge_id						varchar	8	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		下边的id
lef_edge_id						varchar	8	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		左边的id
rig_edge_id						varchar	8	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		右边的id
picture_url						varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		该卡牌的图片资源 URL

卡片边类型表

字段	索引	外键	触发器	选项	注释	SQL 预览					
名					类型	长度	小数点	不是 null	虚拟	键	注释
▶ id					int	8	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	边id 目前为只读
type					varchar	8	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		边的类型Grass: 草地, Road: 路, City: 城
connection					varchar	128	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		连接情况, 表示除了边自身, 其他可能的连接,
position					varchar	8	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		top, bot, lef, rig 分别表示上下左右


03 表结构

对局结果记录表

字段	索引	外键	触发器	选项	注释	SQL 预览						
名						类型	长度	小数点	不是 null	虚拟	键	注释
▶ id						int	16	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	🔑 1	对局结果记录 (战绩记录)
users						varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		参与的玩家, JSON格式保存, 示例: {"user1":3,
start_info						varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		开始对局时, 各种参数, 玩家信息等, 格式待定
end_info						varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		结束对局时的结算情况, 以及玩家信息
start_time						datetime	0	0	<input type="checkbox"/>	<input type="checkbox"/>		开始时间的时间戳
end_time						datetime	0	0	<input type="checkbox"/>	<input type="checkbox"/>		结束时间的时间戳

03 表结构

对局日志记录表

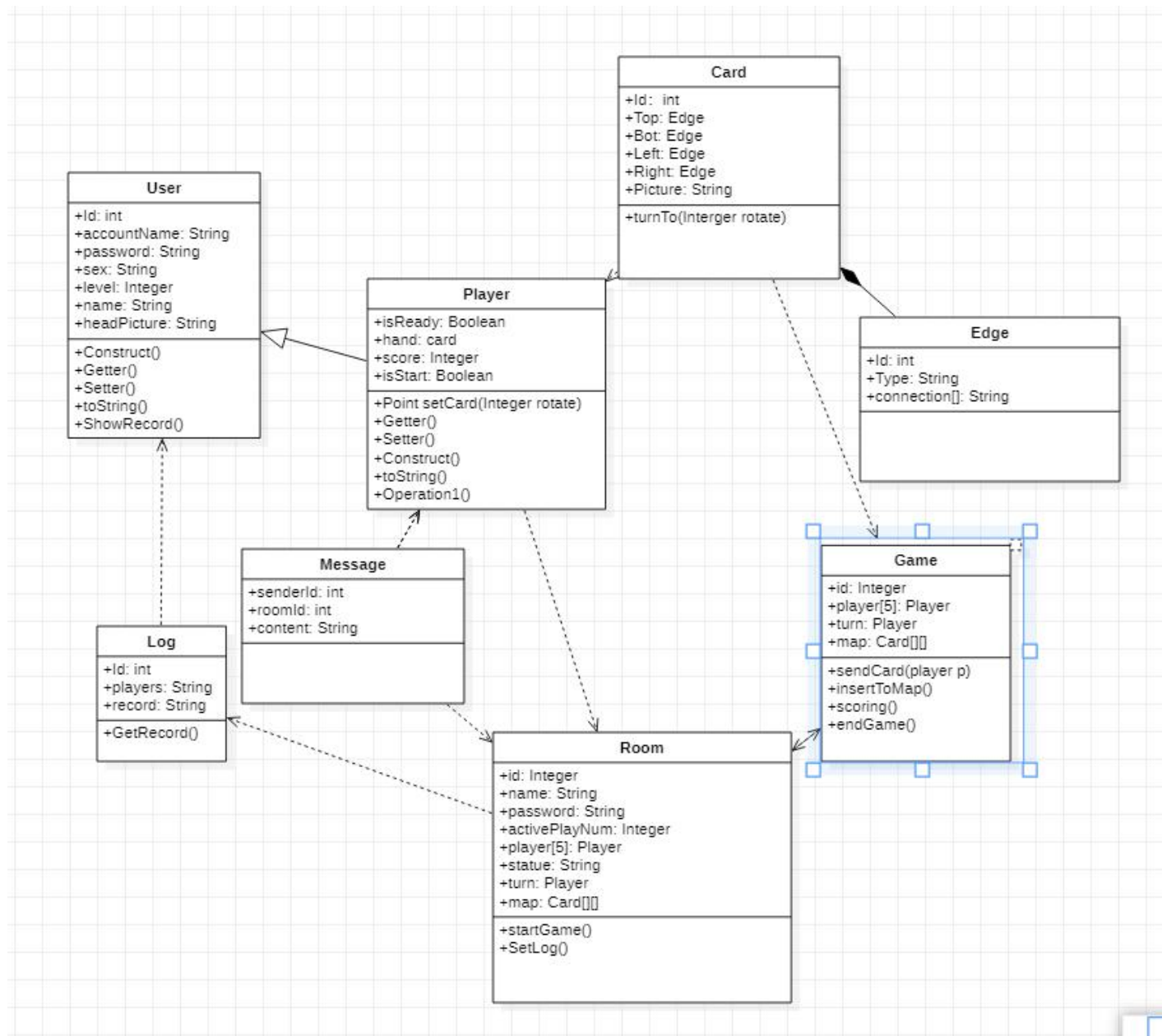
字段	索引	外键	触发器	选项	注释	SQL 预览					
名					类型	长度	小数点	不是 null	虚拟	键	注释
▶ id					int	16	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	对局日志的id
game_record_id					int	16	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		对应的战绩id
users					varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		参与玩家，与game_record 的users 一致，为方便查看
record					longtext	0	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		日志，本局对战的所有操作

用户表

字段	索引	外键	触发器	选项	注释	SQL 预览					
名					类型	长度	小数点	不是 null	虚拟	键	注释
id					int	16	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	 1	用户id
account_num					varchar	32	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		账号，打算采用手机号，邮箱
password					varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		密码，加密保存
sex					varchar	2	0	<input type="checkbox"/>	<input type="checkbox"/>		性别，“男”，“女”
level					varchar	8	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		等级Lv 0 ~ Lv 10
nickname					varchar	32	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		昵称
▶ head_picture_url					varchar	255	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>		头像图片的URL

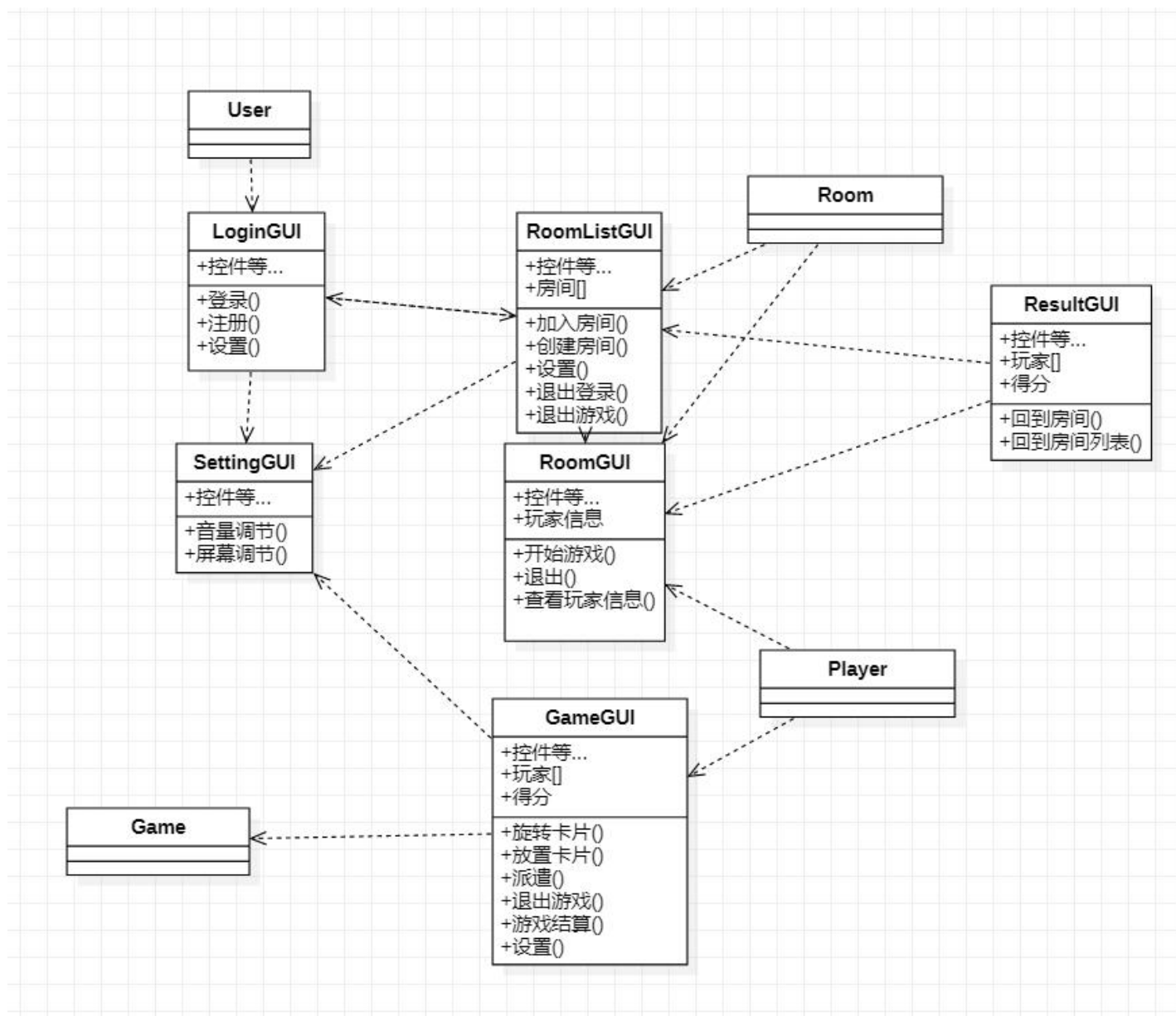
03 类图

游戏逻辑类



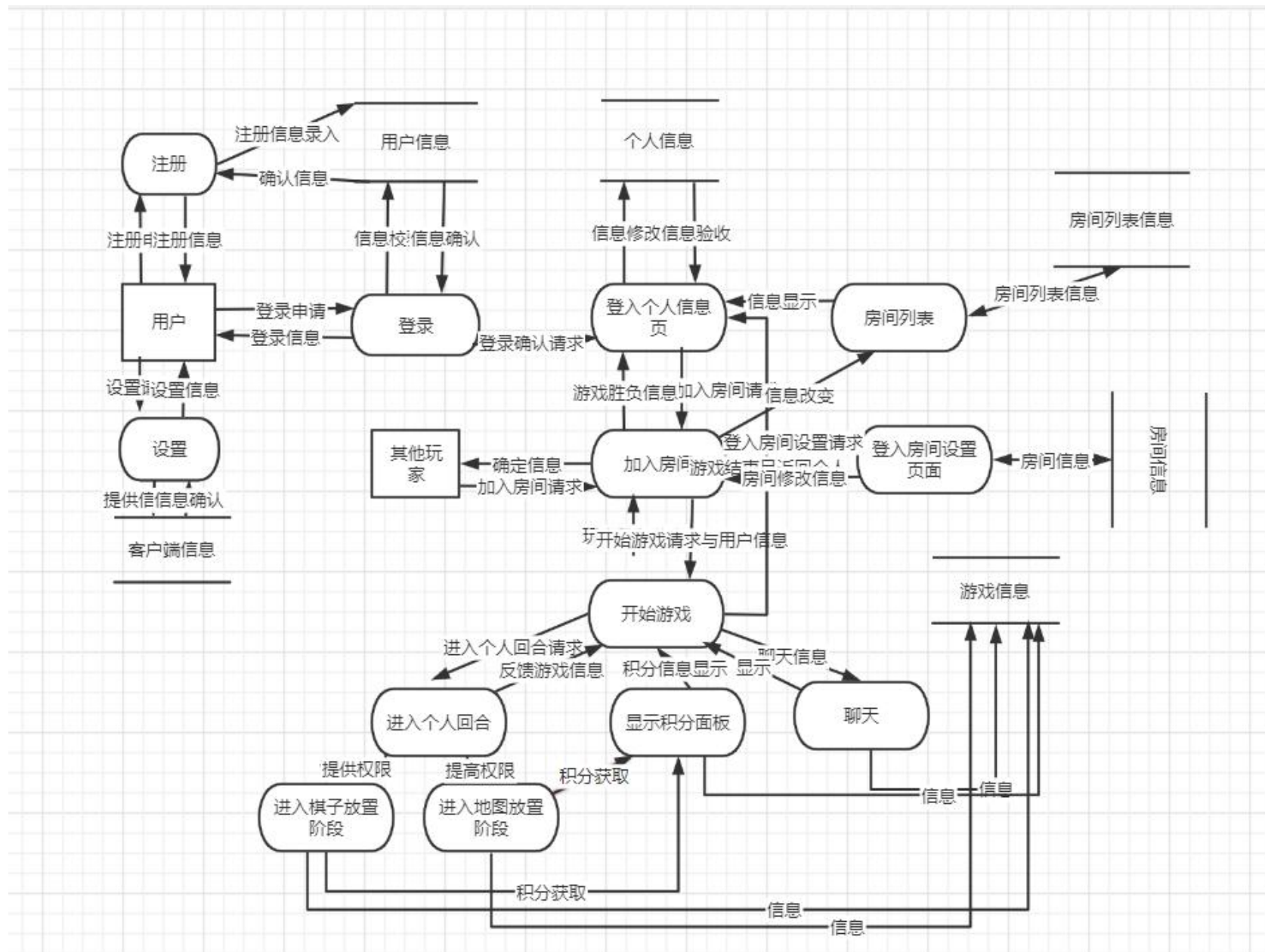
03 类图

界面类



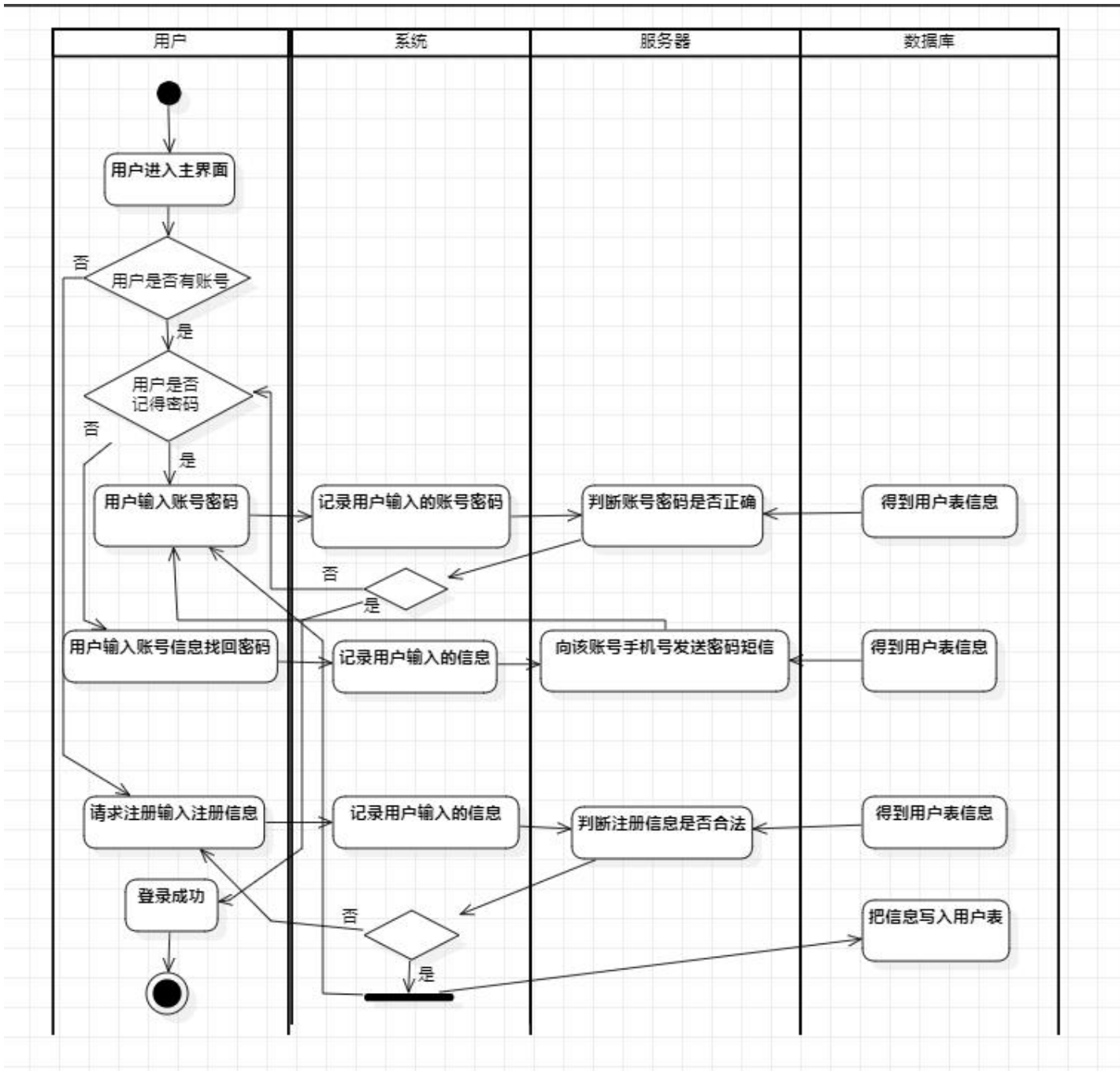
泳道图与数据流图

04 数据流图



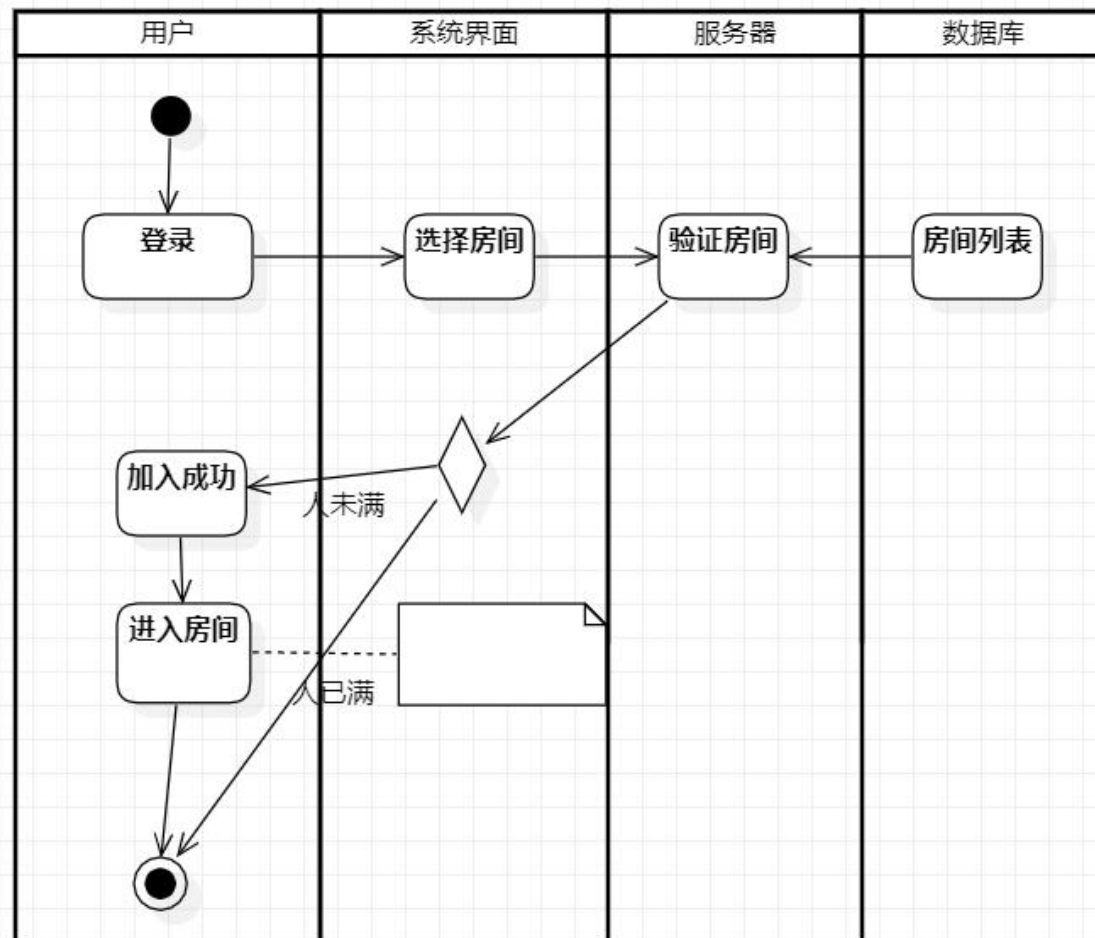
04 泳道图

用户登录泳道图



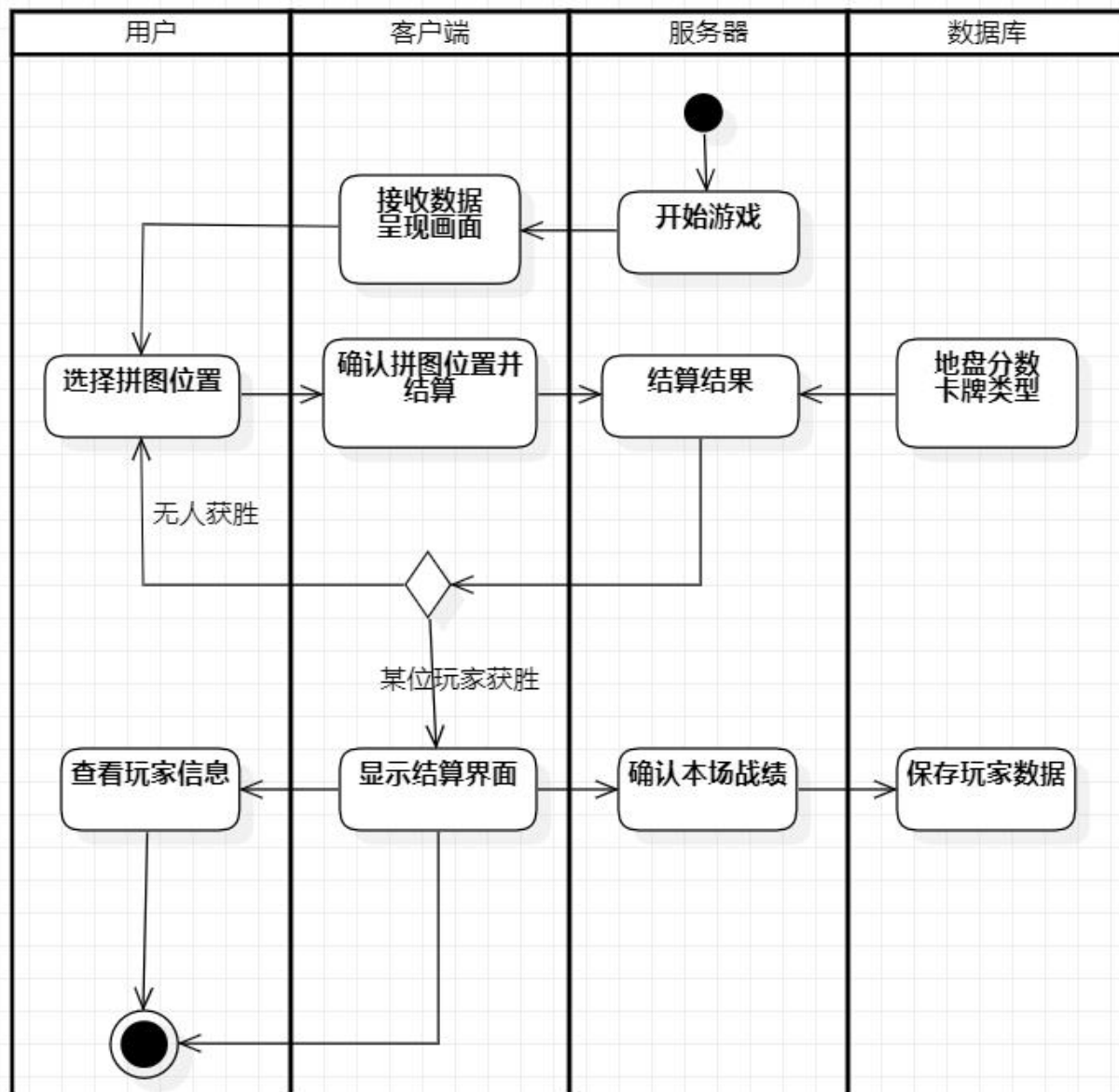
04 泳道图

加入房间泳道图



04 泳道图

游戏泳道图



05

安全性和上节课剩余问题

1.登录用户爆破

为提高数据库强度，防止被人恶意侵入端口，获取数据库信息。

措施：**mysql**限制访问**IP**，这个利用**alter user**语句就可以做到，**root**账户尽量只给**localhost**权限，慎用**user@%**；

不要使用弱密码，尽量采用字母+数字+特殊字符的密码；

服务器使用放暴力破解服务，可以针对非法访问恶意操作进行锁**IP**防御。

2.服务器端口漏洞

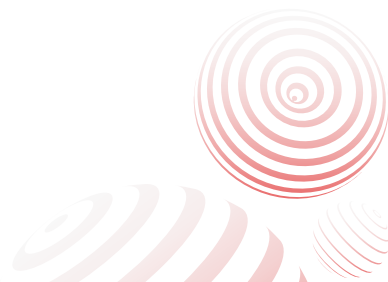
通过**nmap**很容易扫描到服务器暴露的端口，比如**139**端口外露。

因此，本项目开启防火墙，且只开放有需要的端口，敏感端口如**139,445**不外露。

3.操作系统漏洞

window等系统作为服务器安全性差，漏洞过多。

因此，本项目使用**Linux**作为服务器，并采取防火墙，提高服务器操作系统的安全性。



4.用户加密

客户端返回的数据使用非对称（**RSA**）算法传到服务器，使用 **MD5+**盐摘要算法保护用户的密码，进行哈希散列，再保存到数据库。这样可以有效应对彩虹表破解法。

彩虹表(**Rainbow Tables**) 就是一个庞大的、针对各种可能的字母组合预先计算好的哈希值的集合，不一定是针对**MD5**算法的，各种算法的都有，有了它可以快速的破解各类密码。越是复杂的密码，需要的彩虹表就越大，现在主流的彩虹表都是**100G**以上。

5.DDOS

DDoS高防（新**BGP**）和**DDoS**高防（国际）都支持为阿里云**ECS**或非阿里云服务器提供**DDoS**防护，也都支持防护通过阿里云域名服务或第三方域名服务平台开通的域名。



1.类图绘制的过于简单，有些方法的位置也有问题？

answer:我们在逻辑类的基础上添加了界面类，让类的内容更加充实。还有上次傅老师说的游戏类应该从房间类独立出来，我们也重新修改过了。

2.安全性的理解有错误，写的只是一些简单的东西，没有加密和提高系统安全性健壮性的方法。

answer: 我们这次使用非对称算法将数据从客户端传入服务器并使用**md5+**盐摘要算法保护用户的密码，进行哈希散列，再保存到数据库。为防止**DDOS**，我们的阿里云服务器也用**DDOS**防护。

3.如果一个用户恶意不进行游戏（在**5**分钟的自主操作之内，进行一次有效操作，但是不结束自己的操作回合），如何采取有效的方法结束游戏？

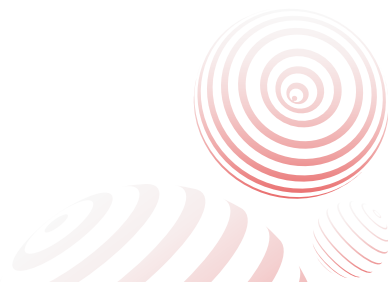
使用计数器，在规定的时间内不完成所有操作，即视为该玩家放弃抽卡和放置棋子，将这枚卡片交给下一玩家使用。



学号	工作内容	贡献度
041801206	数据库设计说明书的编写 接口文档的制作	15
221801402	数据库设计说明书的编写	8
221801403	系统设计说明书的编写 数据流图 ppt制作 宣讲	11
221801407	系统设计说明书的编写	9
221801409	系统设计说明书的编写	8
221801417	系统设计说明书的编写	8
221801418	数据库设计说明书的编写 泳道图	11
221801419	数据库设计说明书的编写 泳道图 分工与计划安排	11
221801433	系统设计说明书的编写 泳道图 类图改进	11
221801434	数据库设计说明书的编写	8

工作流程:

- 1.在网上找到有关说明书的参考，并制定第一次的说明书目录。
- 2.根据团队作业的要求开展线下讨论会，并分配好工作。
- 3.各成员完成自己有关部分的文档编写。
- 4.完成对上次需求分析作业的改进，并开会改进文档部分的结构与具体内容。
- 5.绘制泳道图与数据流图
- 6.编写PPT，制定下一阶段的工作计划与人员安排，提前开始学习有关技术。



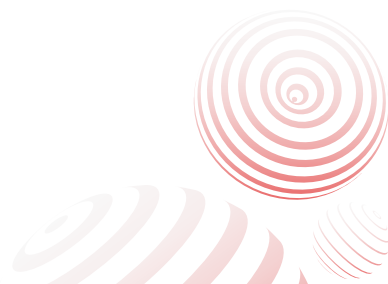
05 预期安排

A	B	C	D
	任务安排	开始时间	结束时间
数据库和系统设计	数据库和系统设计, 学习unity	4月18日	4月22日
	整合和ppt制作准备答辩, 学习unity	4月23日	4月24日
α版本开发	游戏的基础界面与跳转的制作	4月25日	4月27日
	游戏逻辑实现	4月27日	5月6日
	服务器部署与数据库连接	4月27日	5月6日
	整合和制作ppt准备答辩	5月7日	5月8日
软件评测	对软件进行各方面测试, 总结需要改进的问题	5月7日	5月14日
	对总结出的问题提出具体的改进措施和方案	5月15日	5月21日
β版本开发	尝试着手改进测评阶段发现的问题	5月22日	5月31日
	制作游戏官网	5月22日	5月31日
	对初步改进的结果第二次测评	6月1日	6月7日
	对第二次测评的结果进行进一步改进开发	6月8日	6月15日
	整合和总结	6月16日	6月18日
总结	个人技术的总结和项目开发总体过程总结	6月19日	



05 组员分工

学号	姓名	分工安排
048101206	陈明煜	数据库设计与游戏服务器部署, 提供接口和其他方面技术支持
221801402	普润寒	美工, 负责主界面与游戏界面的设计, 部分角色的原创设计和部分页面制作
221801403	胡绪豪	项目经理, 负责编写博客与答辩, 总体功能测试与总结问题
221801407	黄鹏程	后端, 完成房间列表与房间内部的逻辑开发与测试
221801409	洪汉杰	后端, 完成登录注册功能的逻辑开发与测试
221801417	黄雕	美工, 负责房间列表与房间内部界面的设计与部分页面制作
221801418	林子坤	前端, 完成部分游戏界面的开发, 分工安排
221801419	殷思忆	前端, 完成部分游戏界面的开发, 同时负责官网的服务器部署
221801433	甘淳钰	后端, 设计总体系统结构, 完成游戏内逻辑开发与测试
221801434	杨杰	前端, 完成部分游戏界面的开发, 前后端衔接与测试





2021

谢谢观看

W O R K S U M M A R Y B U S I N E S S R E P O R T

某某公司产品运营中心

汇报人：@51PPT模板网 时间：2021年4月11日

