

软件工程需求文档

项目名称：模拟商业竞拍交易大赛系统

组名：proj9_rua

组长：寇明阳 2015011318

组员：姚超 2015011319

黄权伟 2015011328

李忠明 2015011326

1、项目基本情况

1.1、项目背景

ASDAN，是一个非盈利性教育组织，得到英国教育部课程局（QCA）和英国高等院校招生服务中心（UCAS）授权，为全球学生提供素质教育认证和领导力课程开发。

ASDAN 举办的模拟商业竞赛是由学生组成的多支团队在线下展开的商业竞赛，完全仿真还原市场和公司运营的真实环境与决策过程，包含运营、竞拍交易和路演三大部分，赢取最多利润的公司成为冠军。

其中的 竞拍交易 部分需要一个独立的在线系统，供多支参赛团队实时参与竞赛，通过竞拍、生产、交易流程获取最多利润，打败竞争对手赢得比赛。

1.2、大赛规则

模拟商业竞拍交易大赛是现场举行的比赛，在大赛中有两个角色，分别是政府(赛事管理员)和参赛团队，所有团队都有相同的起始资金，团队的首要目标是修建最多的房子，第二个目标是获得更多的现金，房子是使用三种红绿蓝不同的原材料按照一定的比例建造的。

大赛分为多个回合，每个回合分为竞拍、生产、交易环节。其中竞拍环节的时间可以根据现场情况修改，而生产和交易两个环节是一起的，且具有固定的时间限制，下面分别介绍一下三个环节：

1、**竞拍**：政府拍卖三种生产建房原材料的机器，它可以用低成本生产原材料，比如可以用机器以 100 块的价格生产 10 个原材料，然后以 500 块的价格在交易环节卖出去，该机器只能用三次。

2、**生产**：拥有机器的公司以低成本生产原材料，生产出的原材料可以用来自己团队盖房子，也可以在交易环节与其他公司交易。

3、**交易**：各个团队可针对机器及原材料自行商讨价格，签订合同，完成交易，所有的交易都是现金交易，不能物物交换。其中合同需要双方都确认，机器的交易不会重置使用次数。

比赛结束后，根据房子及现金数量确定各支团队的排名结果。

1.3、预期用户

模拟商业竞拍交易大赛的赛事管理员和参赛团队。

1.4、项目概述

本项目系统可以分为 web 端、微信小程序端、服务器端（后端）

Web 端的前端采用了 vue 框架，使用了 vue-admin、iview 组件。

微信小程序则使用了微信官方推荐的 WeUI 样式库进行界面设计。

服务器端（后端）使用了 python 语言的 django 框架，使用了 channels 发送 websocket。

2、功能需求

2.1、web 端功能需求

Web 端为赛事的管理员使用，主要功能为管理所有赛事、控制一场比赛的进行以及比赛数据的统计和记录。

2.1.1、初始化

赛事管理员设置参赛人数、起始资金等比赛信息，然后创建一场新的比赛，给每个参赛团队准备登陆账号和密码。

2.1.2、拍卖

赛事管理员负责记录现场的拍卖结果并将数据发送给各队伍。

2.1.3、生产

进入生产/交易环节后赛事管理员发起倒计时并可浏览各订单情况。

2.1.4、结算

管理员端显示各队伍的信息。

2.1.5、可视化

比赛过程中提供单独的界面实时展示各个团队的资产信息以及生产/交易环节的倒计时。提供各种表格用于比赛的数据统计。

2.2、微信小程序端功能需求

微信小程序为用户参加模拟竞拍时的使用工具，支持用户生产、交易及信息查看等功能。

2.2.1、用户登录

参赛队员根据赛事方分配的账号密码登录，进行比赛。

2.2.2、拍卖

拍卖环节，用户可以实时得到全部拍卖信息、本队拍卖信息以及队伍信息查看。

2.2.3、生产

用户可以选择某种机器进行生产。

2.2.4、交易

用户可以选择一支队伍进行交易，同时可以对其他队伍的交易请求进行响应。

2.2.5、信息查看

用户可以查看本队或其他队伍详细的资产情况。

2.2.6、排名展示

用户可以实时得知当前的排名情况。

2.3、数据库和服务器端功能需求

服务器端的主要功能是响应来自 Web 端管理员发来的请求和手机 App 端发来的请求，并向手机 App 端主动发送部分信息，同时更新数据库内的相关数据。

按照不同的阶段，服务器端处理的请求如下：

2.3.1、Web 端初始化

管理员登录 web 端，获取所有比赛总览或一场比赛的信息

2.3.2、比赛初始化

该阶段的需求分为以下几部分：

- (1) 管理员从 web 端创建比赛，创建队伍并获取队伍总览。
- (2) 服务器端生成一场比赛、比赛中的队伍、用户名、密码，存入数据库。
- (3) 管理员创建比赛配置模板并保存为文件，读取比赛配置模板。
- (4) 用户从微信小程序端使用用户名和密码登录，修改昵称及密码。

2.3.3、拍卖

该阶段的需求分为以下几部分：

- (1) 管理员从 web 端发送拍卖信息，存入数据库。
- (2) 用户在微信小程序端从数据库中获取拍卖信息。

2.3.4、生产交易

该阶段的需求分为以下几部分：

- (1) 管理员从 web 端设置倒计时，从数据库中获取订单情况。
- (2) 用户在微信小程序端使用机器进行生产相应材料，保存到数据库中
- (3) 用户在微信小程序中发起交易请求，服务器端将交易信息发给另外一个队伍。
- (4) 用户在微信小程序端中接受或拒绝交易请求，服务器端对数据库进行相应操作。
- (5) 用户在微信小程序中获取倒计时。

2.3.5、结算

管理员或用户获取排名情况和每个队伍详细的资产情况。

2.3.6、在各个阶段都会收到的请求

在比赛的每个阶段都有的需求分为以下几部分：

- (1) 用户在微信小程序端查询排名情况。
- (2) 用户在微信小程序端查询本队或其他队伍详细的资产情况。

3、泳道图及 UML 用例图

3.1、UML 用例图

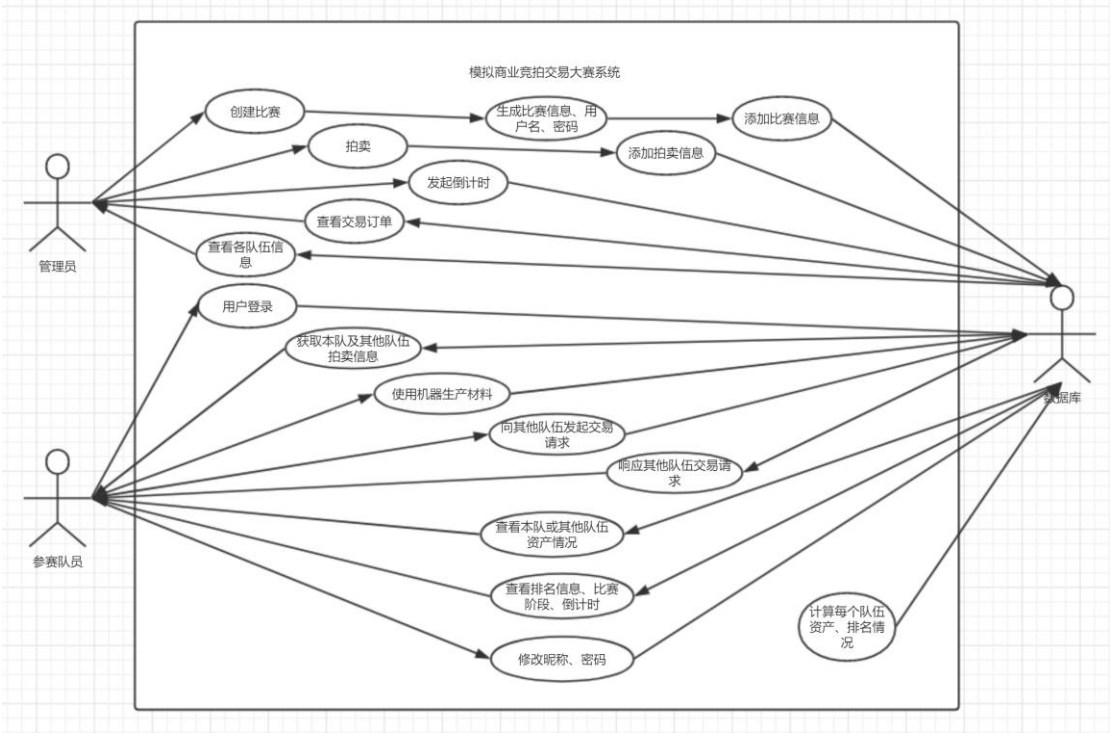


图 0 UML 用例图

3.2、泳道图

按照比赛的各个阶段，可以得到每个阶段的泳道图。

3.2.1、初始化

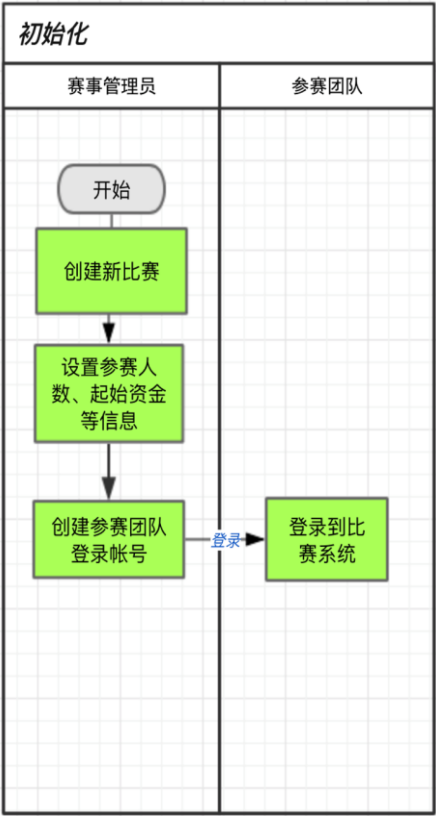


图 1.初始化

3.2.2、拍卖

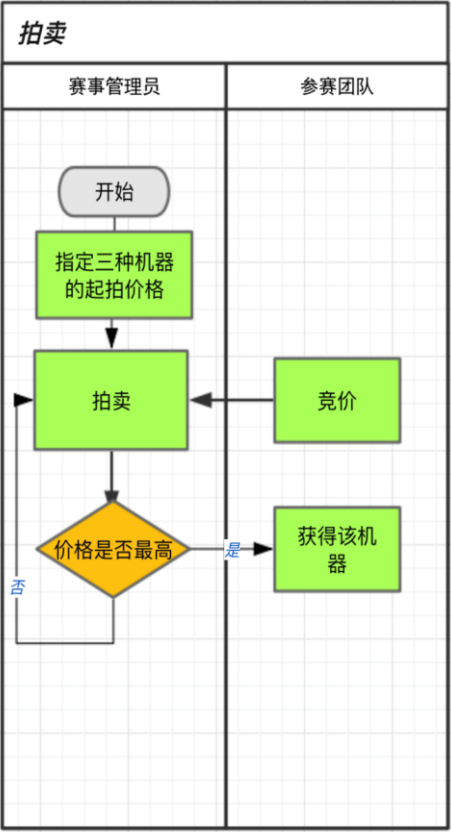


图 2.拍卖

3.2.3、生产

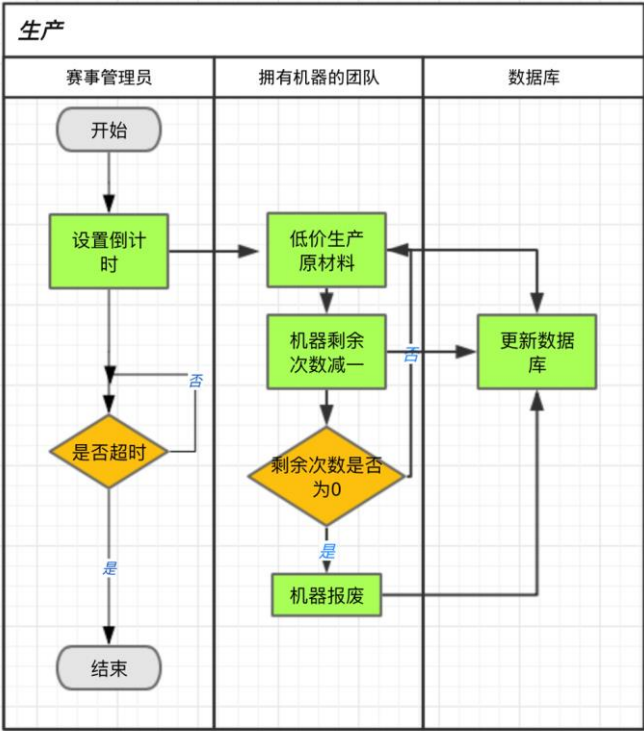


图 3.生产

3.2.4、交易

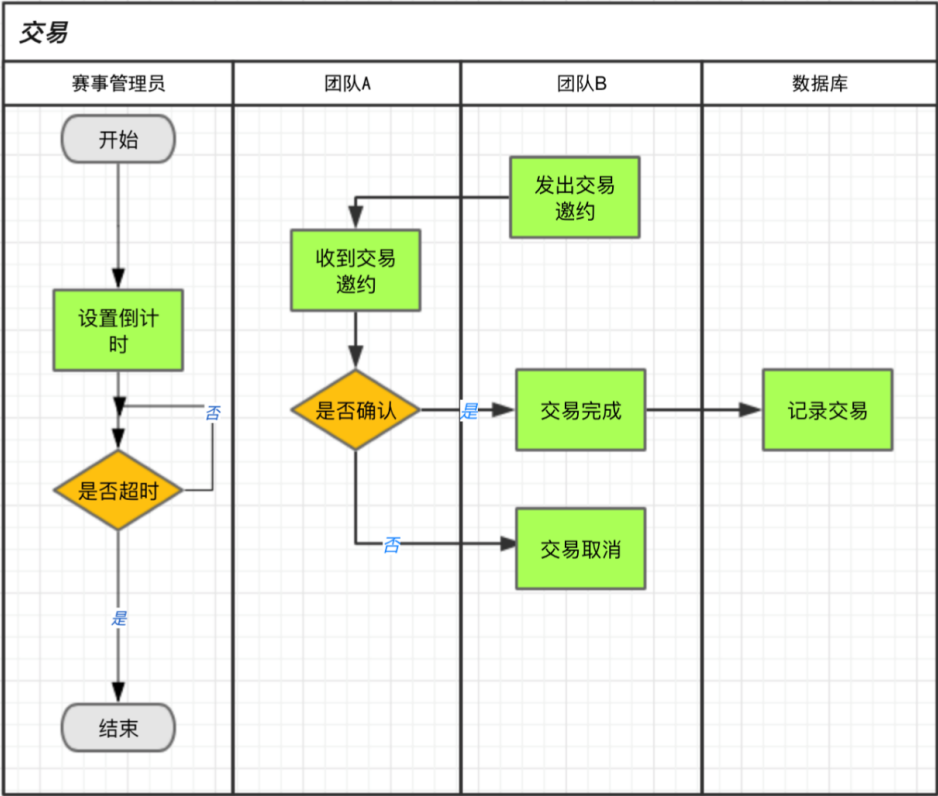


图 4.交易

3.2.5、结算

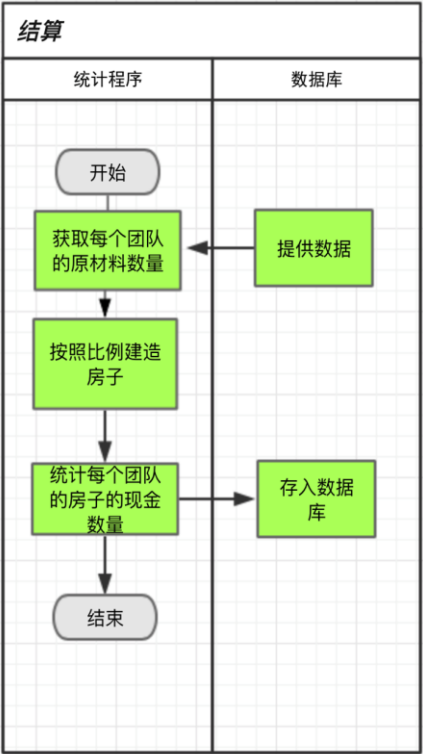


图 5.结算

3.2.6、可视化

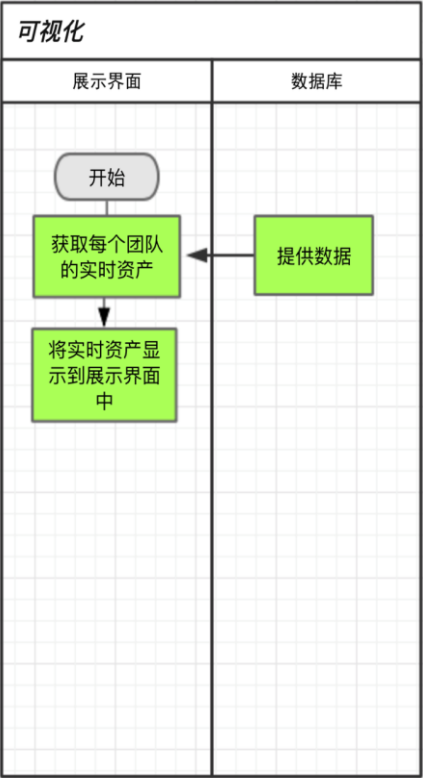


图 6.可视化