**详细设计文档**

|  |  |
| --- | --- |
| **课程名称：** | **企业软件项目实训** |
| **项目名称：** | **Petshop** |
| **学生姓名：** | **队长：邹鹏宇**  **队员：翁焕滨、吴金泽、卢越兴、秦华** |
| **学生专业：** | **软件工程** |
| **开课学期：** | **2018-2019第二学期** |

**软件学院**

**2019年6月**

1. **需求概述**

本项目以Fisco-bcos为底层平台，设计并实现一个虚拟宠物交易商店的区块链应用。

在这个虚拟宠物交易商店中，有普通用户和管理员两种角色。普通用户主要可以查看所有在售的宠物列表、上架并售卖自己的宠物、购买其他用户上架的宠物等。而管理员主要负责为普通用户开户、仲裁用户之间的交易纠纷等。该应用为用户提供了一个便利的宠物购买和转卖的分布式市场。

1. **需求分析**

**2.1 用户角色分析**

**2.1.1 管理员**

（1）登录：使用管理员密钥登录，进入管理员界面。

（2）处理退货：收到用户的退货请求后，选择接受或拒绝该请求，并强制进行相关的转账等操作。

（3）监测市场信息：查看市场的交易信息列表、在售宠物列表。

**2.1.2 普通用户**

（1）注册账户：发出注册请求，收到自己的密钥。

（2）登录：使用注册时返回的密钥登录。

（3）创建宠物：每个账户最多可创建一个宠物，可为该宠物添加相关信息，且价格不可超过5000。

（4）修改宠物信息：修改自己账户中拥有的宠物的在售状态和其他属性。

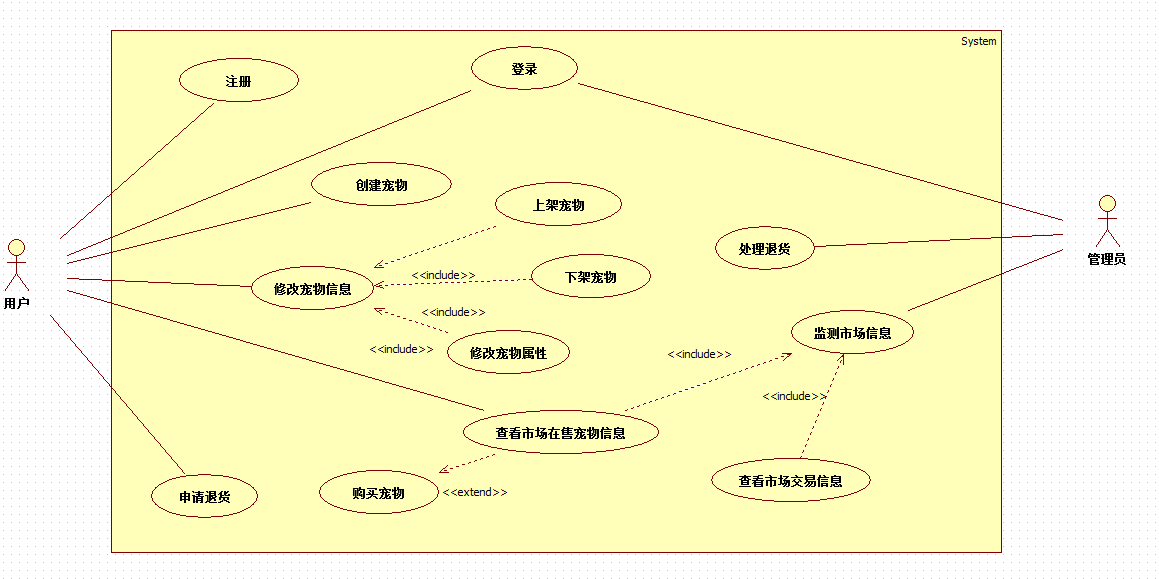
（5）查看市场在售宠物信息：查看市场中所有在售宠物的信息。

（6）购买宠物：购买市场中的在售宠物。

（7）申请退货：在购买交易发生后，向管理员发出退货申请。

**2.2 功能性需求**

**2.2.1 系统功能架构**



**2.2.2 具体功能**

**2.2.2.1 注册**

|  |  |
| --- | --- |
| 用例标识 | 1 |
| 用例名称 | 注册 |
| 参与者 | 未注册用户 |
| 简要说明 | 未注册用户可通过此功能完成注册 |
| 基本事件流 | 1.未注册用户进入首页  2.点击【注册】按钮  3.系统将返回该用户的私钥，并显示注册成功。 |
| 备选事件流 | 1.未注册用户进入首页  2.点击【注册】按钮跳转到注册页面  3.点击【取消】按钮返回首页  1.未注册用户进入首页  2.点击【用户登录】按钮跳转到登录页面  3.点击【立即注册】按钮跳转到注册页面  4.点击【注册】按钮  5.系统将返回该用户的私钥，并显示注册成功 |
|  |  |
| 异常事件流 | 1.未注册用户进入首页  2.点击【注册】按钮  3.显示：注册失败，请重试 |
| 前置条件 | 网络状态良好，用户已进入首页 |
| 后置条件 | 显示注册成功，并返回用户私钥 |
| 非功能性需求 |  |
| 相关业务数据 | 用户私钥 |
| 用例相关人 | 卢越兴 |
| 注释 | 用户需要将自己的私钥记录，用于以后登录 |

**2.2.2.2 登录**

|  |  |
| --- | --- |
| 用例标识 | 2 |
| 用例名称 | 登录 |
| 参与者 | 用户、管理员 |
| 简要说明 | 用户和管理员可通过此功能登录进入各自的页面 |
| 基本事件流 | 1.用户点击【登录】按钮  2.在密钥输入框中输入自己的密钥  3.点击【立即登录】按钮  4. 跳转到用户主页 |
| 备选事件流 | 1.管理员点击【登录】按钮  2.在密钥输入框中输入自己的密钥  3.点击【立即登录】按钮  4. 跳转到管理员主页 |
|  |  |
| 异常事件流 | 1. 点击【登录】按钮  2.在密钥输入框中输入自己的密钥  3.点击【立即登录】按钮  4. 系统显示“用户未注册” |
| 前置条件 | 网络状态良好，已进入登录页面 |
| 后置条件 | 跳转到用户或管理员首页 |
| 非功能性需求 |  |
| 相关业务数据 | 用户或管理员密钥 |
| 用例相关人 | 秦华 |
| 注释 |  |

**2.2.2.3 创建宠物**

|  |  |
| --- | --- |
| 用例标识 | 3 |
| 用例名称 | 创建宠物 |
| 参与者 | 未创建过宠物的用户 |
| 简要说明 | 每个新注册的用户拥有一次创建宠物的资格 |
| 基本事件流 | 1.用户点击【宠物仓库】，进入仓库界面  2.点击【创建宠物】按钮  3.跳转至创建宠物界面  4.输入宠物相关信息  5.点击【创建】按钮  6.显示创建成功并返回宠物仓库界面 |
| 备选事件流 | 1.用户点击【宠物仓库】，进入仓库界面  2.点击【创建宠物】按钮  3.显示已创建过宠物，不能再次创建 |
| 异常事件流 | 1.用户点击【宠物仓库】，进入仓库界面  2.点击【创建宠物】按钮  3.跳转至创建宠物界面  4.输入宠物相关信息  5.点击【创建】按钮  6.显示创建失败 |
| 前置条件 | 网络状态良好，且未创建过宠物 |
| 后置条件 | 显示创建成功，返回仓库界面 |
| 非功能性需求 |  |
| 相关业务数据 | 宠物信息 |
| 用例相关人 | 邹鹏宇 |
| 注释 |  |

**2.2.2.4 上架宠物**

|  |  |
| --- | --- |
| 用例标识 | 4 |
| 用例名称 | 上架宠物 |
| 参与者 | 用户 |
| 简要说明 | 用户可将自己仓库内的宠物标定价格并上架 |
| 基本事件流 | 1.用户点击【宠物仓库】，进入仓库界面  2.点击【创建宠物】按钮  3.跳转至创建宠物界面  4.输入宠物相关信息  5.点击【创建】按钮  6.显示创建成功并返回宠物仓库界面 |
| 备选事件流 | 1.用户点击【宠物仓库】，进入仓库界面  2.点击【创建宠物】按钮  3.显示已创建过宠物，不能再次创建 |
| 异常事件流 | 1.用户点击【宠物仓库】，进入仓库界面  2.点击【创建宠物】按钮  3.跳转至创建宠物界面  4.输入宠物相关信息  5.点击【创建】按钮  6.显示创建失败 |
| 前置条件 | 网络状态良好，且未创建过宠物 |
| 后置条件 | 显示创建成功，返回仓库界面 |
| 非功能性需求 |  |
| 相关业务数据 | 宠物信息 |
| 用例相关人 | 邹鹏宇 |
| 注释 | 此时宠物的价格处显示为0 |

**2.2.2.5 下架宠物**

|  |  |
| --- | --- |
| 用例标识 | 5 |
| 用例名称 | 下架宠物 |
| 参与者 | 用户 |
| 简要说明 | 用户可将自己正在售卖的宠物下架 |
| 基本事件流 | 1.用户点击【宠物仓库】，进入仓库界面  2.在目标宠物图标下点击【下架】按钮  3.在弹出来的提示框里点击【确定】  4.显示下架成功，宠物的价格又变回零 |
| 备选事件流 |  |
| 异常事件流 | 1.用户点击【宠物仓库】，进入仓库界面  2.在目标宠物图标下点击【下架】按钮  3.在弹出来的提示框里点击【确定】  4.提示【该宠物不在售】 |
| 前置条件 | 网络状态良好 |
| 后置条件 | 显示创建成功，返回仓库界面 |
| 非功能性需求 |  |
| 相关业务数据 | 宠物价格和在售状态 |
| 用例相关人 | 邹鹏宇 |
| 注释 |  |

**2.2.2.6 修改宠物信息**

|  |  |
| --- | --- |
| 用例标识 | 6 |
| 用例名称 | 修改宠物属性 |
| 参与者 | 用户 |
| 简要说明 | 用户修改在售宠物的描述或价格 |
| 基本事件流 | 1.用户点击【宠物仓库】，进入仓库界面  2.在目标宠物图标下点击【修改】按钮  3.在弹出的框内输入宠物信息  4.点击【确认】按钮  5.显示修改成功，返回仓库界面 |
| 备选事件流 | 1.用户点击【宠物仓库】，进入仓库界面  2.在目标宠物图标下点击【修改】按钮  3.点击【取消】按钮  4.返回宠物仓库 |
| 异常事件流 | 1.用户点击【宠物仓库】，进入仓库界面  2.在目标宠物图标下点击【修改】按钮  3.在弹出的框内输入宠物信息  4.点击【确认】按钮  5.提示网络错误 |
| 前置条件 | 网络状态良好 |
| 后置条件 | 显示修改成功，返回仓库界面 |
| 非功能性需求 |  |
| 相关业务数据 | 宠物名称、品种、图片、价格和描述 |
| 用例相关人 | 邹鹏宇 |
| 注释 |  |

**2.2.2.7 查看市场在售宠物**

|  |  |
| --- | --- |
| 用例标识 | 7 |
| 用例名称 | 查看市场在售宠物信息 |
| 参与者 | 用户、管理员 |
| 简要说明 | 用户、管理员可通过此功能查看市场所有在售宠物的详细信息 |
| 基本事件流 | 1.用户进入首页  2.点击一个在售宠物  3.跳转至宠物详情页面 |
| 备选事件流 | 1.管理员进入“在售宠物列表”页面  2.点击一个在售宠物  3.跳转至宠物详情页面 |
|  |  |
| 异常事件流 |  |
| 前置条件 | 网络状态良好，用户或管理员已登录，市场上有宠物在售 |
| 后置条件 | 显示宠物详情 |
| 非功能性需求 |  |
| 相关业务数据 | 在售宠物列表、宠物信息 |
| 用例相关人 | 秦华 |
| 注释 |  |

**2.2.2.8 购买宠物**

|  |  |
| --- | --- |
| 用例标识 | 8 |
| 用例名称 | 购买宠物 |
| 参与者 | 用户 |
| 简要说明 | 用户可在市场内的宠物列表内挑选宠物，并点击【购买】按钮 |
| 基本事件流 | 1.用户进入【市场】界面  2.点击宠物图片下方的【购买】按钮  3.提示将花费xx金额购买“xx”  4.点击确定  5.购买成功，返回市场界面 |
| 备选事件流 | 备选事件流一：  1.用户进入【市场】界面  2.点击一个在售宠物  3.跳转至宠物详情页面  4..点击宠物下方的【购买】按钮  5.提示将花费xx金额购买“xx”  6.点击确定  7.购买成功，返回市场界面  备选事件流二：  1.用户进入【市场】界面  2.点击一个在售宠物  3.跳转至宠物详情页面  4..点击宠物下方的【取消】按钮  5.返回【市场】界面  备选事件流三：  1.用户进入【市场】界面  2.点击宠物图片下方的【购买】按钮  3.提示将花费xx金额购买“xx”  4.点击确定  5.提示金额不足  备选事件流二：  1.用户进入【市场】界面  2.点击宠物图片下方的【购买】按钮  3.提示将花费xx金额购买“xx”  4.点击确定  5.提示该宠物已下架 |
|  |  |
| 异常事件流 |  |
| 前置条件 | 网络状态良好，用户或管理员已登录 |
| 后置条件 | 购买成功 |
| 非功能性需求 |  |
| 相关业务数据 | 用户余额，宠物 |
| 用例相关人 | 邹鹏宇 |
| 注释 |  |

**2.2.2.9申请退货**

|  |  |
| --- | --- |
| 用例标识 | 9 |
| 用例名称 | 申请退货 |
| 参与者 | 用户 |
| 简要说明 | 用户申请退货并，说明相关原因给管理员裁定 |
| 基本事件流 | 1.用户点击【订单】按钮  2.跳转至订单页面  3.在相应的订单后点击【申请退货】  4.在弹出框内填写理由  5.点击【提交】按钮  6.显示提交成功，订单后的【申请退款】变成【仲裁中】 |
| 备选事件流 | 1.用户点击【订单】按钮  2.跳转至订单页面  3.在相应的订单后点击【申请退货】  4.点击【取消】按钮  5.返回订单界面 |
|  |  |
| 异常事件流 |  |
| 前置条件 | 网络状态良好，用户或管理员已登录，有订单存在 |
| 后置条件 | 申请成功，返回订单界面 |
| 非功能性需求 |  |
| 相关业务数据 | 订单信息，订单状态 |
| 用例相关人 | 邹鹏宇 |
| 注释 |  |

**2.2.2.10 查看市场交易信息**

|  |  |
| --- | --- |
| 用例标识 |  |
| 用例名称 | 查看市场交易信息 |
| 参与者 | 管理员 |
| 简要说明 | 管理员可通过此功能查看市场中发生的所有交易信息 |
| 基本事件流 | 1.管理员进入“交易信息页面”  2.可看到所有市场中发生过的交易 |
| 备选事件流 | 1.管理员进入“交易信息页面”  2.交易列表为空 |
|  |  |
| 异常事件流 |  |
| 前置条件 | 网络状态良好，管理员已登录并进入交易信息页面 |
| 后置条件 | 展示交易列表 |
| 非功能性需求 |  |
| 相关业务数据 | 交易列表、交易详情 |
| 用例相关人 | 秦华 |
| 注释 |  |

**2.2.2.11 处理退货**

|  |  |
| --- | --- |
| 用例标识 | 11 |
| 用例名称 | 处理退货 |
| 参与者 | 管理员 |
| 简要说明 | 管理员可通过此功能处理用户的退货请求 |
| 基本事件流 | 1.管理员进入“退货信息”页面  2.点击【同意】按钮  3.显示“已同意退货申请” |
| 备选事件流 | 1.管理员进入“退货信息”页面  2.点击【拒绝】按钮  3.显示“已拒绝退货申请” |
|  |  |
| 异常事件流 |  |
| 前置条件 | 网络状态良好，管理员已登录，有未处理的退货请求 |
| 后置条件 | 相关用户的宠物列表和账户余额已按管理员的处理操作发生相应变动 |
| 非功能性需求 |  |
| 相关业务数据 | 退货信息列表、用户账户余额、用户宠物列表 |
| 用例相关人 | 秦华 |
| 注释 |  |

**2.3 非功能性需求**

**2.3.1 安全需求**

1）严格权限访问控制，用户在经过身份认证后，只能访问其权限范围内的数据，只能进行其权限范围内的操作。

（2）不同的用户具有不同的身份和权限，需要在用户身份真实可信的前提下，提供可信的授权管理服务，保护数据不被非法/越权访问和篡改，要确保数据的机密性和完整性。

（3）能经受来自互联网的一般性恶意攻击。如病毒（包括木马）攻击、口令猜测攻击、黑客入侵等。

**2.3.2 可靠性**

系统的可靠性也是很重要的一方面。本系统的可靠性主要体现在两个方面:

1.数据可靠性：可靠性主要体现在数据的可靠存储和防止数据丢失。

2.操作可靠性：通过功能的正确实现和硬件的合理配置，保证系统能顺利和准确的完成用户所要求的特定任务。

具体体现在：

（1）因软件系统的失效而造成不能完成业务的概率要小于5%

（2）在1,000,000次交易中，最多出现1次需要重新启动系统的情况

（3）对输入有提示，数据有检查，防止数据异常。

**2.3.3 性能**

在95％的情况下，一般时段响应时间不超过5秒，高峰时段不超过8秒。

在推荐配置环境下：登录响应时间在5秒内，刷新市场列表响应时间在5秒内，购买操作或退款操作响应时间秒5内，修改宠物信息响应时间3秒内。

在非高峰时间管理员查看整个市场的交易记录，可以在5秒内得到搜索结果。

**2.3.4 可用性**

系统的可操作性要求：系统具有友好的用户交互方式和高效合理的功能配给及组织形式。本系统对可操作性的设计要求主要可概括为以下几个方面：

1.功能定义明确，区分清晰：对系统的主要功能有明确的定义，并且在界面及系统内部有着明确的区分，使得用户可以非常便捷的识别和使用相关功能等。

2.操作直接，关联性充分：在用户的功能实现成本上，要尽量降低用户实现特定功能所需的点击量和复杂程度。主要通过两点来实现，一方面对特定功能，要尽量减少功能实现的层深，扩展功能树的广度而降低深度。另一方面，要使得相关功能的关联性尽量充分，使得用户可以通过功能之间的关联链接来便捷的从一种功能链接到相关的功能。

1. **系统设计**
2. **逻辑部署方案**
3. **物理部署方案**

**项目详细设计文档需包含如下内容：**

需求概述

需求分析

用户角色分析

管理员

参与的业务

用户

功能性需求

新建用户

非功能性需求

安全性

隐私性

系统设计

新建用户

智能合约

界面

测试

逻辑部署方案

物理部署方案

注意：

1. 正文部分（标题行用小四号字加粗，正文内容用小四号字）
2. 图表清晰

**正式上交本设计文档时，请删除本页。**