

---

## 可行性分析(研究)报告(FAR)

说明：

1. 《可行性分析(研究)报告》(FAR)是项目初期策划的结果，它分析了项目的要求、目标和环境；提出了几种可供选择的方案；并从技术、经济和法律各方面进行了可行性分析。可作为项目决策的依据。

2. FAR 也可以作为项目建议书、投标书等文件的基础。

---

# 目录

可行性分析(研究)报告(FAR).....	1
1 引言.....	3
1.1 标识.....	3
1.2 背景.....	3
1.3 项目概述.....	3
1.4 文档概述.....	4
2 引用文件.....	4
3 可行性分析的前提.....	4
3.1 项目的要求.....	4
3.2 项目的目标.....	5
3.3 项目的环境、条件、假定和限制.....	5
3.4 进行可行性分析的方法.....	6
4 可选的方案.....	7
4.1 原有方案的优缺点、局限性及存在的问题.....	7
4.1.1 优点.....	7
4.1.2 局限性及存在的问题.....	8
4.2 可重用的系统, 与要求之间的差距.....	8
4.3 可选择的系统方案 1.....	9
4.4 可选择的系统方案 2.....	9
4.5 选择最终方案的准则.....	9
5 所建议的系统.....	10
5.1 对所建议的系统的说明.....	10
5.2 数据流程和处理流程.....	11
5.3 影响(或要求).....	12
5.3.1 设备.....	12
5.3.2 软件.....	12
5.3.3 运行.....	12
5.3.4 开发.....	12
5.3.5 环境.....	13
5.3.6 经费.....	13
5.4 局限性.....	13
6 技术可行性(技术风险评价).....	13
7 法律可行性.....	14
8 用户使用可行性.....	14
9 其他与项目有关的问题.....	14
10 注解.....	14
附录.....	15

---

# 1 引言

## 1.1 标识

本文档适用于“校易通”（Campus TradeConnect）商品网上交易系统，版本号 V1.0，初次发布。

## 1.2 背景

随着校园内二手物品交易的日益增长，学生之间的物品流转需求日趋旺盛。传统的线下交易模式存在诸多不便，如信息不对称、交易效率低、信任度不足等问题。目前校内的二手交易仍依赖于社交平台、公告栏或线下摆摊，这种模式存在较大的局限性。例如，交易双方无法快速匹配合适的商品，缺乏系统性的监管和信用评估机制，导致欺诈风险增加。基于此，本项目的目标是创建一个高效、可信、易用的校园二手商品交易系统，利用现代 Web 技术改善交易体验，同时为学生提供更加规范化的交易渠道。

该系统旨在满足学生用户的需求，提供在线商品展示、购买、留言互动、订单管理等功能，同时设立管理员审核机制，确保交易环境的安全性和可信度。系统运行环境为基于 Web 的在线应用，可通过 PC 端和移动端访问，支持常见的现代浏览器。

## 1.3 项目概述

本项目的核心目标是构建一个高效、可靠、安全的二手商品交易平台，使学生可以通过互联网便捷地发布商品、浏览商品、购买商品，并完成交易过程。系统主要特点包括：

用户管理：用户注册、登录及留言功能。

商品管理：支持商品发布、图片展示、信息修改与删除。

审查管理：管理员对用户注册和商品信息进行审核。

购物功能：用户可查找商品并在线下单购物。

订单管理与配送：买卖方可管理订单状态，包括出货、收货等环节。

本系统的开发主要基于 Spring Boot + Vue 技术栈，后端采用 Spring Boot 框架进行开发，前端采用 Vue 进行页面渲染和交互，数据库采用 MySQL 存储系统数据，整体架构采用前后端分离的模式。开发工具主要使用 IntelliJ IDEA 进行

---

编码与调试。

项目目标用户为校园内的学生群体，开发团队由软件工程小组负责，包括系统的架构设计、功能开发、测试与后期维护，指导老师提供技术支持与项目监督。

本系统的运行环境主要为校园内部局域网，同时考虑未来的互联网部署方案，以适应更广泛的用户需求。

## 1.4 文档概述

**文档概述** 本文档用于详细描述“校易通”（Campus TradeConnect）商品网上交易系统的可行性分析，包括技术可行性、经济可行性、法律可行性及用户使用可行性等方面的研究。本报告将对系统的开发背景、功能需求、技术架构、成本效益、市场预测等进行全面评估，并提供多个可选方案，最终推荐最佳实施方案。

此外，文档涉及的数据和信息需符合相关隐私保护条例，确保用户数据的安全性和保密性。本文档将作为软件工程与实践课程的作业之一，为后续的系统设计、开发和测试提供参考依据。

## 2 引用文件

1. 《软件工程原理与实践》，第三版，2023 年 6 月 1 日。
2. 《Web 应用开发技术标准》，第二版，2022 年 9 月 15 日。
3. 《数据隐私保护条例》，第一版，2021 年 12 月 10 日。
4. 《软件工程需求分析规范》，第四版，2023 年 3 月 22 日。
5. 《Spring Boot 开发指南》，第二版，2023 年 8 月 5 日。
6. 《Vue.js 前端开发手册》，第三版，2023 年 5 月 18 日。
7. 《MySQL 数据库优化实践》，第一版，2021 年 7 月 25 日。

## 3 可行性分析的前提

### 3.1 项目的要求

- （1）用户管理：主要包括注册、登录、留言等功能模块。
- （2）商品管理：主要包括发布在线产品展示功能（必须有图片的形式）以

---

及对商品信息的管理。

(3) 审查管理：主要包括管理员对注册的审查以及商品的审查。

(4) 用户购物：主要包括用户能对已经审核通过的商品进行在线产品查找功能，并且进行在线下订单购物功能。

(5) 配送：主要包括买家和卖家对订单的一系列操作：用户下订单后。卖家在发现后修改订单状态为出货，买家收到物品后能将订单状态修改为已收货，交易成功结束。

## 3.2 项目的目标

(1) 采用人机对话的操作方式，界面设计简洁，信息查询灵活方便，快捷准确，数据存储安全可靠。

(2) 全面展示商城内商品，并且在登录后才可购买，保证了交易安全。

(3) 可以实现网上商品的交易，达到网上购物的目的。

(4) 用户可以随时在网上查看自己的账户订单。

(5) 对用户输入的数据进行严格的数据检验，尽可能排除人为错误。

(6) 系统应极大限度地实现易维护性与易操作性。

## 3.3 项目的环境、条件、假定和限制

开发环境

软件平台：

操作系统：Windows 11 旗舰版；

数据库：MYSQL；

开发工具包：JDK 5.0；

JSP 服务器：Tomcat 5.0 以上；

浏览器：IE 7；

分辨率：最佳效果 1024×768。

关于数据库 MYSQL：

MySQL 是目前最流行的关系型数据库系统之一，在 2008 年以前由 MySQL AB 公司开发、发布和支持。MySQL 是一个开源的数据库服务器软件，可以自由从

---

Internet 下载和使用，同时，MySQL 是一个真正的多线程、多用户的 SQL 数据库服务器，基于客户/服务器结构实现，具有功能强大、使用简单、管理方便、查询速度快、安全可靠等优点。

关于 JDBC：

JDBC 是一种用于执行 SQL 语句的 Java API，可以为多种关系数据库提供统一访问，它由一组用 Java 语言编写的类和接口组成。JDBC 为工具/数据库开发人员提供了一个标准的 API，据此可以构建更高级的工具和接口，使数据库开发人员能够用纯 Java API 编写数据库应用程序。Java 数据库连接体系结构是用于 Java 应用程序连接数据库的标准方法。JDBC 对 Java 程序员而言是 API，对实现与数据库连接的服务提供商而言是接口模型。作为 API，JDBC 为程序开发提供标准的接口，并为数据库厂商及第三方中间件厂商实现与数据库的连接提供了标准方法。JDBC 使用已有的 SQL 标准并支持与其它数据库连接标准，如 ODBC 之间的桥接。JDBC 实现了所有这些面向标准的目标并且具有简单、严格类型定义且高性能实现的接口。JDBC 扩展了 Java 的功能。例如，用 Java 和 JDBC API 可以发布含有 applet 的网页，而该 applet 使用的信息可能来自远程数据库。企业也可以用 JDBC 通过 Intranet 将所有职员连到一个或多个内部数据库中（即使这些职员所用的计算机有 Windows、Macintosh 和 UNIX 等各种不同的操作系统）。随着越来越多的程序员开始使用 Java 编程语言，对从 Java 中便捷地访问数据库的要求也在日益增加。

综上，对于开发平台地选择：

1. 采用了 Internet 架构，Bowser/Server 模式。
2. Web 服务器采用 Tomcat5.0 服务器，数据库管理系统为 MySQL。
3. 开发工具使用 Dreamweaver 搭建系统页面框架，再用 Eclipse 在其中插入 JSP 语句，JavaBean 使用 Eclipse 编写。

限制：限制查询结果的条数。

## 3.4 进行可行性分析的方法

### 1. 逻辑经济可行性

网上购物系统自身具备的即时互动方式可以吸引很多购物者的注意力，这些购物网站能够随时让顾客参与购买、更方便、更详细、更安全。而且在这样的网站中，产品的管理是有秩序的，并将其进行科学化的分类，便于购买者查询。这

---

种网站的网页设计往往制作得更有指导性，并且更加美观，这些都将会吸引大批的购买者。

## 2. 技术可行性

基于 MYSQL 数据库技术实现

## 3. 系统功能需求可行性

通过对淘宝网，拍拍等网上购物的认识和分析，我对网上购物有了基本的了解，我觉得该系统至少应该具有以下的一些基本功能：

注册管理：用户可以自行注册账号。账号注册完成就成为系统用户，可以凭账号登陆系统。

商品浏览：显示商品详情，提供购买链接。浏览商品时不要求用户登录，但下订单前用户必须登录。

订单管理：只有登录的用户可下订单；用户可以查看自己的订单。管理员可以修改订单的状态，确认订单，删除过期订单。

商品管理：管理员可以增添商品种类；修改商品基本信息；删除商品过时商品。

注册用户管理：用户可以注册，管理员可以查询用户，并管理用户的详细信息，可做修改。

# 4 可选的方案

## 4.1 原有方案的优缺点、局限性及存在的问题

原有方案是以闲鱼为代表的 C2C 交易平台，能够部分满足学校学生之间商品交易需求。

### 4.1.1 优点

以闲鱼为代表的现有 C2C 平台，在多年市场更迭中已形成成熟的功能体系。其核心模块如商品发布、实时聊天、交易流程担保及信用评价体系，均经过大量用户的使用和反馈，及开发团队不断地版本更新和优化，功能完整性和稳定性达到行业领先水平。例如，用户可通过图文混排、视频上传等方式快速发布商品，同时平台内置的智能推荐算法能精准匹配买卖双方需求。在社区运营层面，类似

---

“鱼塘”的垂直兴趣社区有效提升了用户粘性，通过话题讨论、活动运营等方式构建了独特的社交化交易场景。

依托阿里巴巴集团的生态优势，闲鱼天然具备高用户信任度。支付宝的资金托管机制解决了 C2C 交易中最关键的信任难题，用户对支付安全、纠纷仲裁等环节的认可度显著高于新兴平台。同时，菜鸟物流体系的深度融合，使得商品流转效率大幅提升。在技术层面，闲鱼的基础设施支持千万级并发交易，经历过双十一等极端流量场景的考验，其弹性扩容能力与灾备机制值得借鉴。

### 4.1.2 局限性及存在的问题

尽管功能完善，现有平台的功能泛化特性难以满足垂直领域深度需求。例如，在高校二手书、二手商品交易场景中，用户需要与各个课程联动的教材匹配功能、校内自提点存取货等特色服务，而闲鱼的标准功能无法实现此类定制化需求。此外，平台采用“一刀切”的规则体系，导致特定用户群体（如企业级卖家）的操作体验受限。

同时，平台规则对商业模式的限制日益凸显，部分商品类目需缴纳高达 10% 的佣金，强制使用指定物流服务导致成本不可控，且营销推广费用逐年攀升形成利润挤压。

## 4.2 可重用的系统，与要求之间的差距

尽管闲鱼等现有平台提供了标准化的 C2C 交易流程，但在实际落地垂直场景时，其可复用能力与自建平台的定制化需求存在显著差距。

功能层面，闲鱼的标准化流程虽然覆盖了基础的发布、沟通、支付功能，但无法满足细分场景的深度需求。例如，针对高校场景的二手教材交易，需要增加“学科分类标签”功能，允许用户按专业、课程名称筛选商品。此外，校园场景对用户身份真实性要求更高，需集成“学生统一认证系统”，而闲鱼的通用身份认证无法满足这一需求。

成本方面，闲鱼按交易额抽成的模式（5%-10%）在业务规模扩大后成本压力剧增。例如，假设平台年交易额达 1 亿元，仅佣金支出就需 500 万-1000 万元，而我们的系统采用固定成本模式，仅需支付服务器费用和运营维护开销，可节省 80% 以上成本。

用户体验上，闲鱼的通用界面设计难以满足垂直用户群体的特殊诉求。我们的平台以学生群体为主，计划在 UI 上深度融合校园场景：商品详情页增加“相



---

关课程”模块，展示该教材对应的课程信息；甚至结合地理位置，在交易流程中提供地图显示自提点方位。

## 4.3 可选择的系统方案 1

网页版方案采用 vue 作为前端核心框架，后端选用 SpringBoot 构建分层清晰的 API 服务，结合 MySQL 存储结构化交易数据。

该方案的显著优势在于跨平台兼容性——用户无需下载安装包，浏览器输入网址即可使用，尤其适合初期快速验证市场。例如，学生群体在电脑、手机浏览器上均可无障碍访问。开发层面，团队只需维护一套代码，无需针对 iOS 和 Android 分别适配，预计节省 30% 的开发周期。版本迭代时，前端热更新机制可绕过应用商店审核流程，例如修复支付漏洞或调整 UI 按钮位置时，用户刷新页面即可生效，极大提升敏捷性。

然而，网页版存在明显功能短板。受限于浏览器环境，无法调用原生设备硬件能力——例如二手商品交易过程中，无法使用跳转支付功能等。

## 4.4 可选择的系统方案 2

原生 App 方案针对 iOS 和 Android 双端独立开发，iOS 端采用 SwiftUI+Combine 实现声明式 UI 与数据流管理，Android 端则基于 Kotlin+Jetpack Compose 构建现代化界面。后端服务与网页版保持一致，通过 RESTful API 提供商品、订单、用户等模块的数据交互，确保多端数据一致性。

原生应用的核心竞争力在于性能与设备的集成。例如，生物识别支付可调用 Face ID 或屏下指纹模块，比网页端的密码输入更安全便捷。消息推送系统能实现离线触达，提升用户打开率——数据显示，开启推送后日活留存率平均提高 25%。

但原生开发面临严峻的成本挑战。双端独立开发需多名专职工程师，人力成本倍增，且功能迭代需双端同步发版，应用商店审核机制可能延误更新等等大量碎片问题都需要考虑。

## 4.5 选择最终方案的准则

最终我们将选择网页版方案。

对于小团队而言，资源效率是首要考量。网页版的开发成本较低，且团队成员已有的前端基础可直接复用，而原生 App 则需额外投入时间掌握 iOS 和

---

Android 开发相关知识。维护成本上，原生应用面临应用商店审核变动的风险，可能因合规问题被迫回退版本，而网页端的热更新机制能快速修复漏洞，更适合我们团队。

而且学生群体对交易效率的需求远高于其他功能。网页版比 App 更快上线，通过二手群、校园墙进行宣传，在学期初等交易高峰能够更快更好地满足用户需求。

跨场景渗透能力是网页版的隐性优势。学生用户在 PC 端、手机端、平板端之间的多设备切换中，无需下载安装包，显著降低使用门槛。调研数据显示，87% 的学生更倾向通过浏览器直接访问服务类平台，而非主动搜索应用商店下载。

更重要的是，网页版可通过渐进式技术升级持续优化体验。当用户规模突破 5 万日活后，可针对性开发原生 App 核心模块，形成“网页版广泛获客+App 深度运营”的混合生态，使该项目能够长期运营下去。

## 5 所建议的系统

### 5.1 对所建议的系统的说明

#### 系统架构设计：

前端：Vue3 + Vite（轻量构建） + Element Plus（基础 UI 组件）

后端：SpringBoot（内嵌 Tomcat） + MyBatis（简化 ORM）

数据库：MySQL（本地部署） + H2 数据库（测试环境内存数据库）

图片存储：本地文件存储（开发环境） + Base64 编码（少量图片直接存数据库）

支付/物流：虚拟模拟（按钮点击直接完成状态变更，无需对接真实接口）

#### 核心功能模块：

##### 用户管理模块：

注册/登录：邮箱验证（通过 Mock 邮件服务），Session 或 JWT 简单鉴权

权限控制：通过注解区分普通用户和管理员

`(@PreAuthorize("hasRole('ADMIN')"))`

##### 商品管理模块：

商品发布：基础表单（标题、描述、价格） + 本地图片上传（限制 5MB 以内）

商品展示：列表分页查询（MySQL LIMIT 语句） + 关键词搜索（LIKE 模糊

---

匹配)

审核管理模块:

人工审核: 管理员后台直接操作 (无 AI 审核, 通过状态字段标记 is\_approved)

敏感词过滤: 本地敏感词库 (文本文件加载, 如 “枪支” “代考” 等校园违禁词)

交易流程模块:

购物车: MySQL 临时表存储 (关联用户 ID 和商品 ID)

订单状态: 简单状态流转 (前端按钮模拟 “支付成功” “确认收货”)

库存控制: 虚拟库存 (无真实扣减, 仅演示逻辑)

## 5.2 数据流程和处理流程

数据流程图:

graph TD

A[用户浏览器] -->|HTTP 请求|

B(Vue 前端) B -->|Axios API 调用|

C(SpringBoot Controller) C --> D[Service 层逻辑]

D -->|SQL 操作| E[(MySQL)]

E -->|返回数据| D

D --> C

C -->|JSON 响应| B

B -->|渲染页面| A

关键处理流程:

商品发布流程:

用户填写表单 → 前端校验必填字段 → 上传图片到服务器  
/static/upload 目录

后端接收数据 → 敏感词过滤 (遍历本地词库) → 插入数据库 goods 表 (状态默认为 “待审核”)

管理员登录后台 → 查看待审核商品 → 点击 “通过” 按钮更新状态为 “已上架”

---

订单模拟流程：

用户点击“立即购买” → 创建订单（orders 表插入记录，状态为“待支付”）

点击“模拟支付”按钮 → 后端直接修改订单状态为“已支付” → 前端跳转到订单详情页

卖家后台点击“发货” → 状态变更为“已发货” → 买家点击“确认收货”完成交易

## 5.3 影响(或要求)

### 5.3.1 设备

服务器：开发阶段使用个人笔记本电脑（Windows/Mac，8GB 内存）

客户端：Chrome/Firefox 最新版（无需移动端适配，PC 端优先）

### 5.3.2 软件

开发工具：

IntelliJ IDEA Community Edition（免费版） + VS Code（前端开发）

Navicat Lite（MySQL 可视化工具）或 DBeaver（开源替代）

### 5.3.3 运行

依赖环境：JDK17 + Node.js 16.x + MySQL 8.0（本地安装包）

网络要求：本地局域网访问

### 5.3.4 开发

技能要求：

前端：Vue3 基础语法 + 组件通信（Props/Emit）

后端：SpringBoot 基础 CRUD + MyBatis XML 配置

开发周期：5 人团队，预期 12 周（需求分析 2 周，编码 8 周，测试 2 周）

---

### 5.3.5 环境

测试数据：使用 Java Faker 库生成模拟数据（100 条商品/用户记录）

版本控制：GitHub 私有仓库（学生免费） + Commit 规范（Angular 风格）

### 5.3.6 经费

总成本：趋近于 0 元

开发工具：全部使用免费社区版或开源软件

服务器：本地运行，无需云服务费用

仅潜在成本：域名/IP（演示时可使用 localhost 或局域网 IP）

## 5.4 局限性

性能瓶颈：

高并发场景未优化（如 100+用户同时访问可能导致响应延迟）；

功能简化：

无真实支付/物流，交易流程依赖按钮模拟；

数据安全：

敏感信息（密码）加密但无定期密钥轮换机制；

扩展性限制：

架构未设计微服务，后期扩展需重构。

## 6 技术可行性(技术风险评价)

现有小组成员 5 人，开发工具：IntelliJ IDEA Community Edition（免费版）和 MySQL（免费开源）。服务器：初期可使用本地服务器或免费云服务（如 Heroku、Vercel）进行测试和部署。

校内二手交易平台的功能相对简单，技术难度适中，适合学生团队开发。但是如果团队成员缺乏某些技术（如 Spring Boot 或前端框架），可能需要额外学习时间，可以通过在线课程或者邀请老师进行指导。如果需要高性能服务器，可申请学校合作的云服务（如华为云平台等）。效益上，平台可以为校内学生提供便利的二手交易服务，提升资源利用率，同时项目经验可以提升团队成员的技术能力。

---

最后此工程的技术可行性，无需额外成本，项目难度适中，适合学生团队开发

## 7 法律可行性

对于用户个人方面，系统需遵守《著作权法》、《个人信息保护法》、《网络安全法》等，确保用户的隐私安全，实施数据加密，防止个人信息泄露；确保用户上传的图片或者视频不会侵犯他人的版权，或者发布违禁品或者虚假信息诈骗等，需严格审查，设置举报功能如有问题需要下架等

## 8 用户使用可行性

平台主要面向校内学生，用户单位为学校（或学生组织），行政管理相对灵活，但需符合学校的管理制度。平台需获得学校相关部门的批准和支持，尤其是涉及学生数据、交易行为的管理，同时需要制定明确的用户行为规范、交易规则和纠纷处理机制，并纳入学校的管理框架。工作制度可分为平时系统维护的技术组、商品审核和解决纠纷的监督组。

## 9 其他与项目有关的问题

如果项目开发并运行后，可能会面临用户需求增加，比如拍卖、租聘、私人定制等，同时平台官方也要推出活动福利等增强用户粘性，并根据大数据分析引入智能推荐系统，根据用户行为推荐商品等。技术上可能有前端技术更新（如从 Vue.js 2 升级到 Vue.js 3）或者后端框架升级（如从 Spring Boot 2 升级到 Spring Boot 3）。若以后需要更大的收费服务器用来支持，需要收取一定的平台手续费。用户体验上也可能不满足于网页，需要开发如微信小程序独立移动端应用。用时为了方便学校内的其他留学生使用，需要有不同的语言提供以及翻译功能

## 10 注解

前端：用户直接交互的界面部分，包括网页和移动端界面

---

后端：处理业务逻辑、数据存储和交互的服务器端部分

API：应用程序编程接口，用于前后端之间的数据交互

ORM：对象关系映射，将数据库表映射为程序中的对象，简化数据库操作

RESTful API：一种基于 HTTP 协议的 API 设计风格，使用 GET、POST、PUT、DELETE 等方法进行数据操作

响应式设计：网页或应用界面能够根据设备屏幕尺寸自动调整布局

技术债务：由于快速开发或技术选择不当导致的后续维护成本增加

负载均衡：将用户请求分配到多个服务器，以提高系统性能和可靠性

分布式架构：将系统拆分为多个独立模块，部署在不同的服务器上，以提高扩展性和容错性

CI/CD：Continuous Integration/Continuous Deployment（持续集成/持续部署）

HTML：HyperText Markup Language（超文本标记语言）

CSS：Cascading Style Sheets（层叠样式表）

JavaScript：一种用于网页交互的脚本语言

UI：User Interface（用户界面）

UX：User Experience（用户体验）

Vue.js：一种用于构建用户界面的 JavaScript 框架

Spring Boot：一种基于 Java 的后端开发框架

SwiftUI：苹果公司推出的用于构建 iOS 应用界面的框架

Kotlin：一种用于 Android 开发的现代编程语言

Jetpack Compose：Android 的现代 UI 工具包

H2：一种轻量级的内存数据库，常用于测试环境

Base64：一种将二进制数据编码为文本的编码方式

## 附录

暂无