

<MAiMAi>	
用例规范： <用例名称	日期： <11/12/2022

[公司名称]

# MAiMAi mobile application

[文档副标题]

Yihan Xiao, Zihan Gao, Rongchuan Hai

2022-12-11

<MAiMAi>	
用例规范： <用例名称	日期： <11/12/2022

## 目录

1. 导言	6
1.1 目的	6
1.2 范围	6
1.3 定义、缩略语和简称	6
1.4 参考文献	6
1.5 概述	6
2. 总体说明	7
3. 具体要求	7
3.1 功能性	7
3.1.1 用户注册系统	7
3.1.2 用户数据收集系统	7
3.1.3 销售信息上传系统	7
3.1.4 待售物品浏览系统	7
3.1.5 货币交易系统	8
3.1.6 通讯系统	8
3.2 可用性	8
3.2.1 系统应与现实世界的熟悉程度相匹配。	8
3.2.2 接口应符合内部标准和任何现有的外部标准。	8
3.2.3 系统状态的可见性应更加熟悉。	9
3.2.4 遵守用户控制和容错原则。	8
3.2.5 系统应检测错误并进行错误修复。	8
3.2.6 系统应具备交易记忆功能。	8
3.2.7 应用程序应灵活高效。	9
3.2.8 设计应符合项目的简洁性和一致性目标。	9
3.2.9 系统应支持多种主流语言。	8
3.2.10	系统将为残疾人提供用户界面和功能。 8
3.3 可靠性	8
3.3.1 初步设计软件可用率为 95	8
3.3.2 初步设计的平均无故障时间（MTBF）为 720 小时，此时需要操作员进行例行系统维护。	9
3.3.3 初步设计平均修复时间（MTTR）为 3 小时	9
3.3.4 初步设计最低分辨率为 960X640，326PPI	9
3.3.5 最高缺陷率预计为每 1000 行代码出现一个小缺陷（不影响程序运行的缺陷）。每 3,000 行代码中不超过一个中等缺陷（在一定程度上影响程序运行但不会导致程序崩溃的缺陷）。	9
3.3.6 整个程序中的重大错误（严重影响程序运行的错误，如导致程序崩溃、数据	

<MAiMAi>	
用例规范： <用例名称	日期： <11/12/2022

丢失等）不超过两个。	9
3.3.7 每次系统维护时都必须修补和修复上述所有错误。	9
3.4 性能	9
3.4.1 此应用程序必须与互联网互动，所有通信都必须通过网络服务器进行。	9
3.4.2 应用程序的初始加载时间取决于用户网络连接的强度，也取决于该应用程序的数据吞吐量。	9
3.4.3 为提供良好的用户体验，应用程序的内存磁盘占用不应超过客户端最终用户的 10%。	9
3.5 支持性	9
3.5.1 使用标准命名约定进行命名；请参阅程序中的注释。	9
3.5.2 需要 5 人小组每月进行 2 小时的例行维护。	9
3.5.3 可能需要额外的服务器内存和存储空间来存储用户数据和处理程序交互。	9
3.6 设计限制	9
3.6.1 标准开发工具：	9
3.6.2 网络产品：	10
3.7 在线用户文档和帮助 系统要求	10
3.8 外购部件	10
3.9 界面	10
3.9.1 用户界面	10
3.9.2 硬件接口	10
3.9.3 软件接口	10
3.9.4 通信接口	10
3.10 许可要求	10
3.11 法律、版权和其他声明	11
3.12 适用标准	11
4. 系统的用例模型	11
4.1 引言	11
4.2 概述	11
4.3 用例图	12
5. 演员名单	
5.1 销售商	10
5.2 买方	10
5.3 管理员	10
6. 用例清单	12
6.1 出售项目	12
6.2 购买项目	12
6.3 留下反馈意见	13
7. 简要说明	13

<MAiMAi>	
用例规范： <用例名称	日期： <11/12/2022

8. 演员简介	13
8.1 销售商	13
8.2 买方	13
8.3 管理员	13
9. 先决条件	13
10. 事件的基本流程	13
11. 替代流程	13
11.1 无效用户	13
11.2 上传交易的文本信息/图像缺失	14
12. 主要情景	14
12.1 MAiMAi 服务器维护中，无响应。	14
13. 后期条件	
13.1 顺利完成	14
13.2 故障条件	14
14. 特殊要求	14
15. 基本流程图	15
16. 简要说明	16
17. 演员简介	16
17.1 买方	16
17.2 管理员	16
18. 先决条件	16
19. 事件的基本流程	16
20. 替代流量	16
20.1 无效用户	16
20.2 上载交易的文本信息/图像缺失	16
21. 主要方案	16
21.1 MAiMAi 服务器正在维护。无响应。	16
22. 后期条件	17
22.1 顺利完成	17

<MAiMAi>	
用例规范： <用例名称	日期： <11/12/2022

22.2 故障情况	17
23. 单独要求	17
24. 基本流程图	17
25. 概念建筑	19
25.1 概念性架构	19
25.2 云架构	20
25.3 数据库	21
26. 高保真线框	24
27. 技术堆栈	32
27.1 前端	32
27.2 后台逻辑	33
27.3 数据库	33
27.4 DevOps	33
28. 工作分解结构	34
29. 项目时间表计划	36
30. 项目成本计划	36

<MAiMAi>	
用例规范： <用例名称	日期： <11/12/2022

# 软件要求规范

## 1. 引言

任何人都可以阅读这份文件，它将帮助开发人员和教师更好地理解项目的要求，并找出需要改进的地方。这意味着，学生和教职员工离开学校时，往往会留下许多不便带走的物品，包括对低年级学生可能有价值的学习用书。

移动应用程序“MAiMAi”的目的是让学生和教职员工通过一个标准化的安全平台交换、出售或购买他们不再使用的物品。目标用户有限制——只有学校的学生和教职员工——这不仅确保了物品的物美价廉，还减少了学生受骗的机会；同时提高了物品的使用率，减少了学术资源的浪费。MAiMAi 是一种典型的媒介，它有助于将学生不需要的物品从一所大学转移到另一所大学。在帮助一个群体避免浪费的同时，也为另一个群体提供了以合理价格获取资源的机会。

除了上述有关商业交易的价值外，我们的软件还能让用户与前辈建立联系，从而结识更多朋友。

### 1.1 目的

该项目的目的是为迈阿密大学的学生、教职员工提供一个可靠的二手物品交易平台。用户可以方便快捷地使用该应用程序发布出售和购买他人二手物品的信息。同时，应用程序上的个人信息将与学校账户关联，这需要得到隐私法的批准。该项目应能在手机上以移动应用程序的形式运行，主要使用 Python 和 C 语言编程，随后还将开发可在苹果设备上运行的版本。此外，该应用程序应通过 App Store 的审核。

### 1.2 范围

该项目软件主要包括一个购物平台，具有登录、通信和支付功能。通过专注于在线销售，卖家可以编辑销售商品的信息并在应用程序中发布。与此同时，买家可以在购买前随意浏览和查找商品信息。该应用程序力求获得迈阿密大学的批准，确保得到其支持和推广。理想情况下，可以战略性地利用该应用程序，通过用户特权和佣金赚取收入。

### 1.3 定义、缩略语和简称

UM	迈阿密大学
超文本标记语言	超文本标记语言

### 1.4 参考资料

### 1.5 概述

文件的其余部分包括一般说明、详细要求和用例模型。文件的第 2 部分是一般描述，包括产品功能和限制。第 3 部分给出了项目的具体功能要求、非功能要求和限制条件。该部分还讨论了软件平台接口。第 4 节是系统的用例模型。第 5 节和

<MAiMAi>	
用例规范： <用例名称	日期： <11/12/2022

第 6 节分别是参与者和用例列表。

## 2. 总体说明

该项目旨在帮助加州大学的学生、教职员工快速、方便地出售和购买二手物品。因此，它需要对用户进行身份验证、一定程度的身份验证以防止恶意或非法活动，并提供安全可靠的使用环境。MAiMAi 的用户必须是 UM 的学生或教职员工，软件必须同时适用于安卓和苹果手机。

该项目包括浏览产品、发布产品、用户间交流和付款等功能。主要功能是帮助用户出售和购买闲置物品，包括浏览产品、发布请求、发布产品、购买产品和用户交流。产品限制允许开源项目，开发人员数量限制在 3-5 人，不允许使用风险技术。假设和依赖条件是，如果学校不同意访问教职员工和学生数据系统来帮助验证用户信息，则使用电话号码进行绑定，一个电话号码只能绑定一个账户。如果用户未进行身份验证，则该账户会被公开标记为“未验证账户”。

## 3. 具体要求

由于我们的初步计划是设计一个移动应用程序，我们可能会使用 Html、CSS3 和特定的编程语言来进行用户界面布局。

### 3.1 功能性

本小节包含我们对校内二手物品交易平台的要求。这些需求是根据功能特性组织的。然后，可通过设想其突出的主动交互功能将其可视化，形成用例图和序列图，以便于理解和跟进。

#### 3.1.1 用户注册系统

系统应收集和保存用户的必要信息，进入学校系统进行身份验证需要高度安全。

#### 3.1.2 用户数据收集系统

系统应便于 7X24 小时实时数据收集，捕捉应用程序内所有用户访问和端到端通信互动。

#### 3.1.3 销售信息上传系统

系统应允许用户设定价格和交易信息，上传交易项目的显示图像，并将其转换为完整的视觉信息。

#### 3.1.4 待售物品浏览系统

系统应为用户提供浏览和搜索待售物品的功能，默认显示第一张上传的物品图片，缩略显示交易价格和交易信息，用户点击进入后显示完整的交易信息。

<MAiMAi>	
用例规范： <用例名称	日期： <11/12/2022

### 3.1.5 货币交易系统

系统应提供买家在确定要购买物品时通过银行卡支付资金、向第三方银行安全账户转账、确认交易完成后向卖家提交的账户转账、交易失败时退回原账户等功能。

### 3.1.6 通信系统

系统应为交易双方提供发送信息和图片的通信功能、信息记录的网络备份功能以及防止外部篡改的高安全性。

## 3.2 可用性

这部分包括所有影响可用性的要求。这些都是非功能性需求，也就是说这不是项目的首要目标，但如果我们能提供这些服务，客户就会非常高兴和满意。

### 3.2.1 快速培训时间

普通用户和高级用户的培训时间应在 10 分钟左右，以便了解应用程序的工作原理。系统应与现实世界相匹配。系统状态的可视性应更加熟悉。

### 3.2.2 图形用户界面

系统界面应符合内部标准和任何现有的外部标准。应用程序应灵活高效。设计应符合项目的简洁性和一致性目标。

### 3.2.3 用户控制和容错原则

将遵守用户控制和容错原则。系统应检测错误并进行错误修复。系统应具有事务记忆功能。

### 3.2.4 语言

该系统应支持多种主流语言。 。

### 3.2.5 为残疾人提供无障碍环境

该系统将为残疾人提供用户界面和功能。

## 3.3 可靠性

### 3.3.1 可用性

初步设计软件 可用性 百分比应为 99% 。



<MAiMAi>	
用例规范： <用例名称	日期： <11/12/2022

### 3.3.2 MTBF

初步设计的平均故障间隔时间应为 720 小时,届时需要操作人员对系统进行例行维护。

### 3.3.3 MTTR

初步设计的平均修复时间应为 3 小时 。

### 3.3.4 决议

初步设计的 分辨率至少应为 960X640 ， 326PPI 。

### 3.3.5 缺陷率

最高 缺陷率 应为每 1000 行代码中一个次要缺陷(不影响程序运行的缺陷)。每 3,000 行代码中不超过一个中等缺陷(在一定程度上影响程序进程但不会导致程序崩溃的缺陷)。

### 3.3.6 主要错误

在整个程序中,系统不得有超过两个 的重大错误 s(严重影响程序运行的错误,如导致程序崩溃、数据丢失等)。

### 3.3.7 维护

在每次系统维护(2 小时)期间,应修补和修复上述所有错误。

## 3.4 性能

3.4.1 该应用程序应基于互联网进行交互,所有通信都必须通过网络服务器进行。

3.4.2 应用程序的初始加载时间取决于用户网络连接的强度,也取决于该应用程序的数据吞吐量,而数据吞吐量则需要根据用户数量(最初指定为学校所有学生的数量)这一基数来确定。。

3.4.3 为提供良好的用户体验,应用程序的内存磁盘占用不应超过客户最终用户的 10%。

## 3.5 支持性

3.5.1 命名应使用标准命名规则;请参阅程序中的注释。。

3.5.2 该系统需要一个 5 人小组每月进行 2 小时的例行维护。。

3.5.3 需要额外的服务器内存和存储空间来存储用户数据和处理程序交互。。

## 3.6 设计限制

### 3.6.1 标准开发工具

系统应使用符合 Html、CSS3 和特定编程语言的标准网页开发工具构建。

<MAiMAi>	
用例规范： <用例名称	日期： <11/12/2022

### 3.6.2 网络产品：

计算机必须配备网络浏览器，如 Internet Explorer。加载产品的响应时间不得超过三分钟。

### 3.7 在线用户文档和帮助 系统要求

由于该程序一般面向受过高等教育的大学生，因此只需提供简明扼要的使用指南即可。需要一个错误反馈机制来解决系统维护过程中未发现的问题。

### 3.8 外购组件

不适用

### 3.9 接口

MaiMai 软件系统支持大多数接口，如用户接口、软件接口、硬件接口等。客户端和服务器之间的通信将使用 HTTP 协议。使用的协议应为 HTTP，端口号为 80。

#### 3.9.1 用户界面

软件的用户界面在设计之初应兼容所有手机系统，无论是安卓系统还是 iOS 系统。用户界面应使用任何工具或软件包来实现。

#### 3.9.2 硬件接口

由于应用程序基于互联网交互，因此连接互联网所需的所有硬件都应系统的硬件接口相同。如调制解调器、Wi-Fi、以太网交叉电缆等。

#### 3.9.3 软件界面

程序需要与买卖双方建立互动，以确保交易顺利进行。使用具有单一读取功能的可读界面。

#### 3.9.4 通信接口

MaiMai 系统将使用 HTTP 协议进行互联网通信，内网通信将通过 TCP/IP 协议套件进行。

### 3.10 许可要求

MAiMAi 授予用户在运行合法获得的操作系统的设备上安装和使用软件产品的权利。

MAiMAi 将授予用户非排他性的、不可再授权的、有限的许可，允许用户在一台设备上安装和使用一份软件产品，该实用程序受本许可协议的条款和条件约束。在适用

<MAiMAi>	
用例规范： <用例名称	日期： <11/12/2022

法律允许的范围内，不得以任何其他方式使用、复制、修改或分发本软件产品，也不得出租本软件产品。根据适用法律，软件产品的更新版本可自动下载和安装。更新版本旨在通过修复缺陷、增强功能、添加新的软件模块和新版本来改进、增强和进一步扩展 MAiMAi 的服务，并可能包括对本软件许可协议条款和条件的修订。用户同意在使用服务时接收此类更新版本。本许可协议适用于所有更新版本、升级版本以及未根据单独许可或其他协议分发的附加功能。

### 3.11 法律、版权和其他声明

软件产品受版权法、国际版权条约以及其他知识产权法律和条约的保护。MAiMAi 保留对软件产品的所有权利。软件产品归 MAiMAi 所有，受适用的版权、商标、商业秘密、专利和其他知识产权法律保护。在用户和 MAiMAi 之间，MAiMAi 拥有并将继续持有软件产品的所有权利、所有权和利益，包括适用知识产权法规定的相关知识产权。本协议不授予用户对软件产品的任何所有权，而仅授予用户根据本协议使用软件产品的可撤销的有限权利。最终解释权归版权所有者所有。

### 3.12 适用标准

它应符合行业标准。遵循 HTML、TCP/IP、SMTP、POP 和 FTP 协议。本许可协议受美国佛罗里达州法院法律管辖，并按其解释。

## 4. 系统用例模型

### 4.1 导言

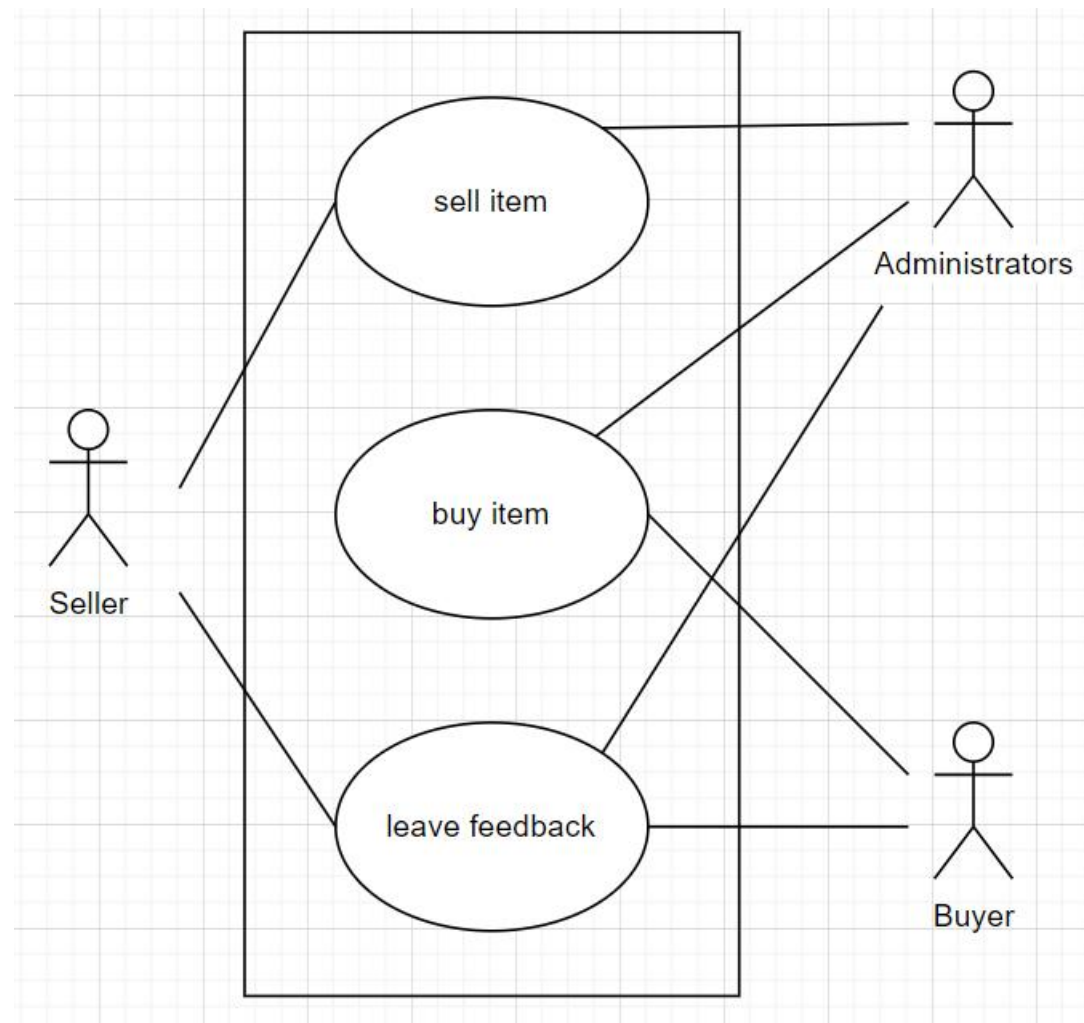
这是应用程序的用例模型。该模型有助于确定主要参与者和用例。

### 4.2 概述

MaiMai 是一款二手物品交易平台移动应用程序。该应用程序旨在帮助用户轻松、廉价、安全地出售和购买二手物品。它还有助于保护环境和减少废物的产生。

<MAiMAi>	
用例规范： <用例名称	日期： <11/12/2022

### 4.3 用例图



## 5. 演员名单

### 5.1 卖方

该演员代表一个拥有 MaiMai 帐户的人，其申请表上有需要出售的物品。

### 5.2 买方

这个角色代表一个拥有 MaiMai 帐户的人，他需要在应用程序上购买物品。

### 5.3 管理员

这个角色代表一个拥有 MaiMai 管理账户的人，他可以维护系统。

## 6. 用例列表

### 6.1 出售项目

本使用案例描述了 MaiMai 用户如何利用应用程序的框架向其他人出售自己的物品。

### 6.2 购买项目

本用例描述了 MaiMai 用户如何利用应用程序向他人购买物品。

<MAiMAi>	
用例规范： <用例名称	日期： <11/12/2022

### 6.3 留下反馈意见

本用例描述了 MAiMAi 用户在完成物品交易后，如何使用应用程序留下有关其满意度的反馈。

## 使用案例：销售产品

### 7. 简要说明

本使用示例将介绍卖家如何将其闲置物品放到 MAiMAi 平台上。

### 8. 演员简介

#### 8.1 卖方

#### 8.2 买方

#### 8.3 管理员

### 9. 先决条件

用户的手机已安装完整的 MAiMAi 应用程序并与网络连接。

服务器不在维护中。

用户需要将银行卡或其他支付方式与 MAiMAi 支付功能连接起来。

### 10. 事件的基本流程

1. 用例从卖方打开 MAiMAi 应用程序开始。
2. 用例：验证用户。
3. MAiMAi 显示客户端用户界面，显示最近交易的项目和功能选项。在这种情况下，用户将选择“出售项目”。
4. 客户端会显示一份表格，供卖方填写交易物品的相关信息。
5. 卖方填写交易文本信息并选择商品图片。
6. 卖方的账户信息、要交易的物品和物品的图片上传到 MAiMAi 的服务器。
7. 服务器根据卖方上传的信息生成项目用户界面，并显示给 MAiMAi 客户端。
8. 卖方被告知物品已成功上市。
9. 用例成功结束。

### 11. 替代流量

#### 11.1 无效用户

如果在基本流程的第 2 步，卖方没有成功填写表格，那么用例就会以失败条件结束。

#### 11.2 上载交易的文本信息/图像缺失

如果卖方未能在表格中填写所有必要信息（即基本流程的第 5 步），例如，卖方仅上传了图片或仅填写了文本信息，则：

1. 客户提示：“缺少文字信息/图片，请务必填写完整”。
2. 用例恢复到步骤 5

<MAiMAi>	
用例规范: <用例名称	日期: <11/12/2022

# 12. 主要方案

12.1 MAiMAi 服务器维护中，无响应。

# 13. 后期条件

## 13.1 成功完成

用户可以在 MAiMAi 应用程序上成功上传自己的交易物品并向其他用户展示。

## 13.2 故障情况

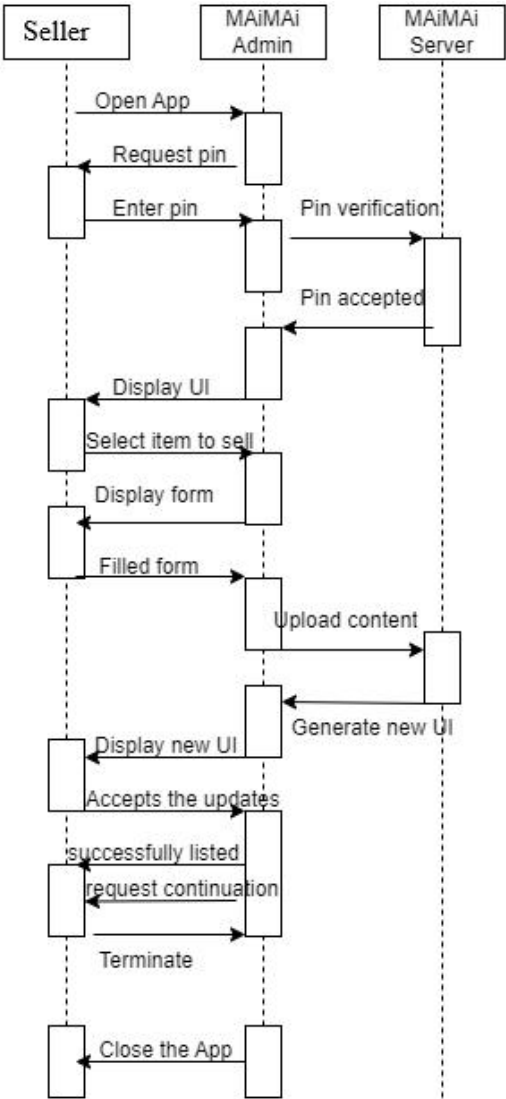
日志已相应更新。

# 14. 特殊要求

[SpReq:WC-1] MAiMAi 服务器将记录和备份用户上传的交易项目信息，包括上传文本图像的日期和时间等。

[SpReq:WC-2] 每个用户每天不能列出超过十个项目。

# 15. 基本流程图



<MAiMAi>	
用例规范： <用例名称	日期： <11/12/2022

## 使用案例：购买产品

### 16. 简要说明

本使用示例将介绍买家如何在 MAiMAi 平台上购买他们想要的产品。

### 17. 演员简介

#### 17.1 买方

#### 17.2 管理员

### 18. 先决条件

用户的手机已安装完整的 MAiMAi 应用程序并连接到网络。

MAiMAi 服务器不在维护中。

用户需要将银行卡与支付功能或其他支付方式连接起来。

### 19. 事件的基本流程

1. 买家打开 maimai 应用程序。
2. 用户登录自己的账户。
3. MAiMAi 显示客户端用户界面，显示最近交易的项目和功能选项。在本例中，用户将选择“购买项目”。
4. 买家可使用关键字、类别、使用时间等搜索所需物品。
5. 买方确认购买，向 MAiMAi 服务器提供地址和账户信息。
6. Maimai 应用程序通知卖家并发货。
7. 买方获得物流信息，收到货物，交易成功。

### 20. 替代流量

#### 20.1 无效用户

如果在基本流程的第 2 步，买方没有成功完成表单，那么用例就会以失败条件结束。

#### 20.2 上载交易的文本信息/图像缺失

买方：如果付款方式出现问题，无法完成付款

d 因为缺少送货地址信息。

1. 客户端提示“缺少文字信息/图片，请务必填写完整”。
2. 用例恢复到步骤 5
3. 客户提示：“请提供可使用的付款方式”。

### 21. 主要方案

- 21.1 MAiMAi 服务器正在维护。无响应。

### 22. 后期条件

- 22.1 成功完成

<MAiMAi>	
用例规范： <用例名称	日期： <11/12/2022

买方成功收到所购物品，表明交易完成，并可对购买进行评分。

## 22.2 故障情况

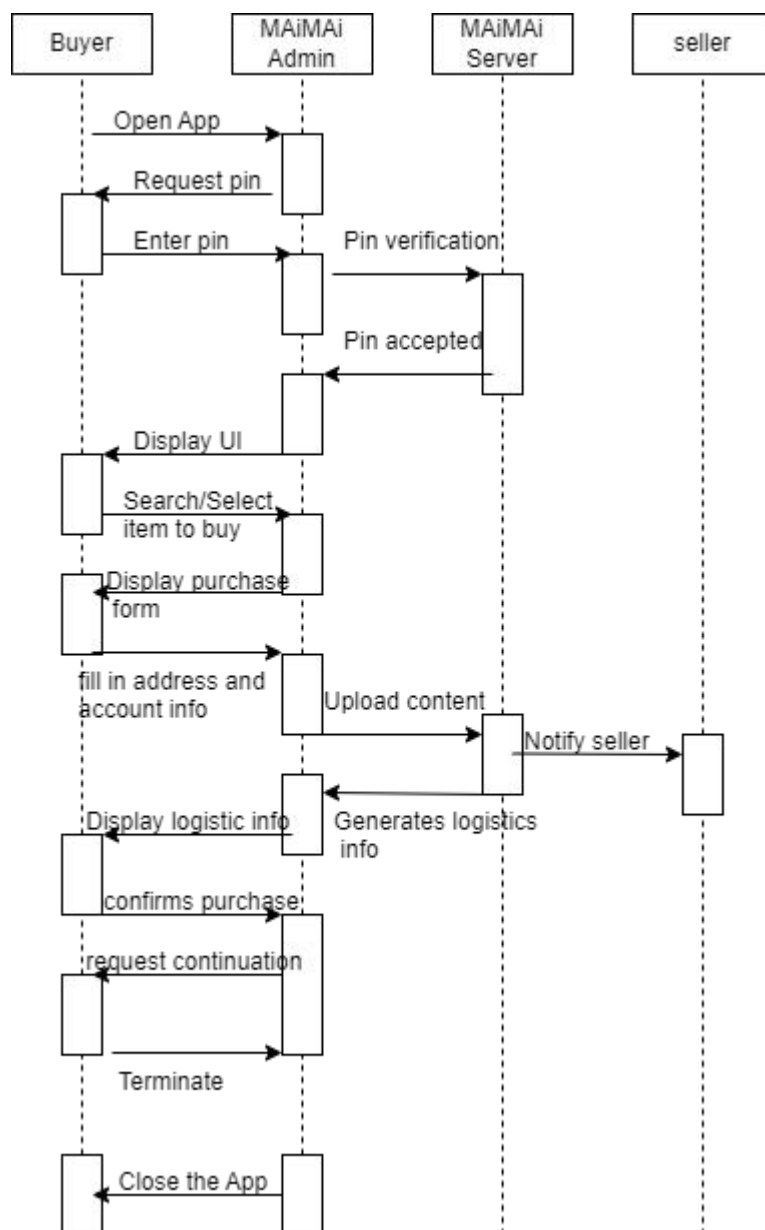
日志已相应更新。

## 23. 特殊要求

[SpReq:WC-1] MAiMAi 服务器将记录并备份用户上传的交易项目信息，包括上传文本图像的日期和时间等。

[SpReq:WC-2] 需要退货的买家需要让 MAiMAi 官员检查其物品是否完整，以避免损坏。

## 24. 基本流程图





<MAiMAi>	
用例规范： <用例名称	日期： <11/12/2022

## 25. 软件结构和数据库

### 25.1 概念架构：

适用于 MaiMai 应用程序的四种概念架构包括客户端-服务器风格、黑板风格、发布-订阅风格和基于事件的风格。

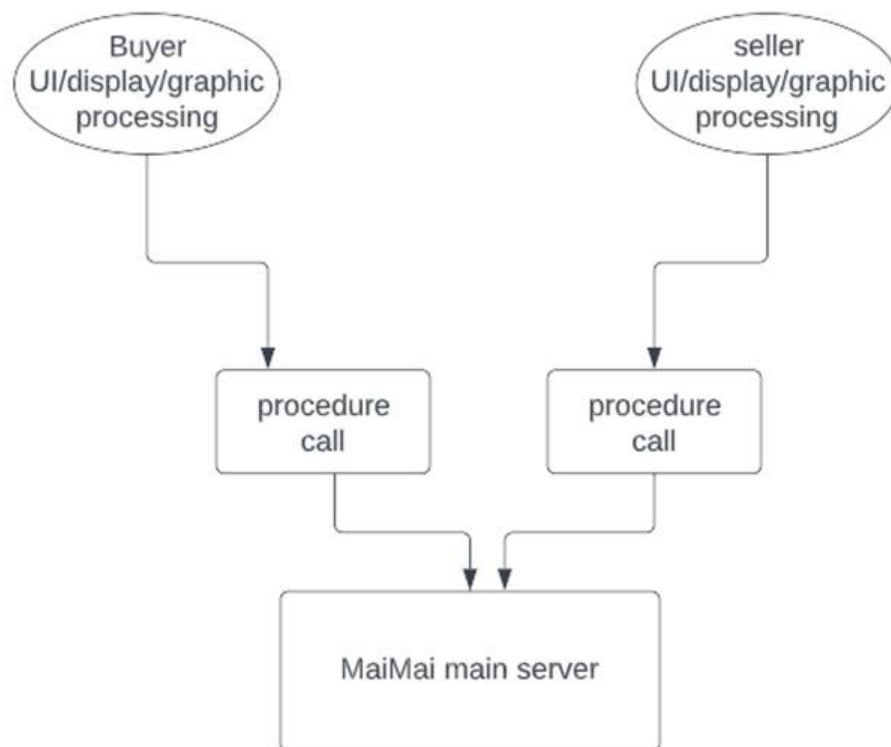
**客户端-服务器风格**是一种分层风格，它将每一层视为上一层的服务器和下一层的客户端，并且只有两层（de Lima Fontao, dos Santos, & Dias-Neto, 2015）。对于 MaiMai 移动应用程序，买卖双方搜索产品、下订单、上传产品和接收卖方订单时将使用线性分层。它功能强大，可为众多客户提供服务。

**黑板风格**适用于 MaiMai 应用程序，因为它提供了一个集中的数据结构，所有客户、卖家和买家都可以轻松访问。

**发布-订阅风格**允许 MaiMai 系统决定向用户显示什么内容。例如，买家将注册为订阅者，以便系统显示适合其性格的产品，而买家将注册为发布者。系统将只向卖家显示向买家发货的订单。

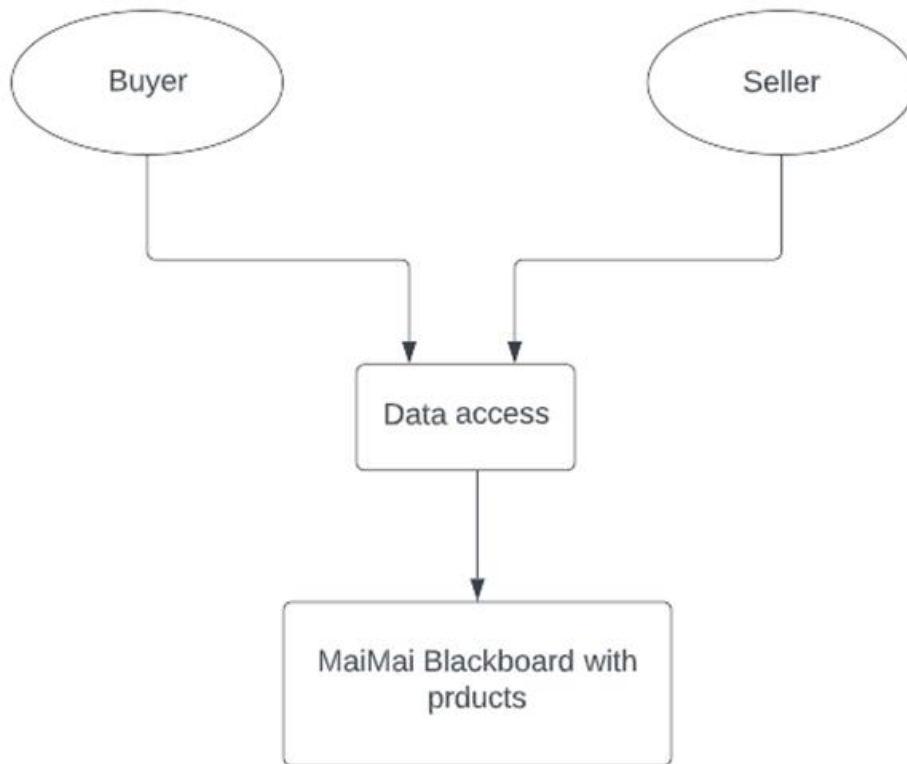
**基于事件的风格**涉及通过事件总线通信的各种组件，允许异步事件发射。由于 MaiMai 应用程序中有多个事件，如搜索产品和下订单，因此这种风格可以降低复杂性，易于使用（Gruhn & Köhler, 2006）。

#### i. 客户端-服务器

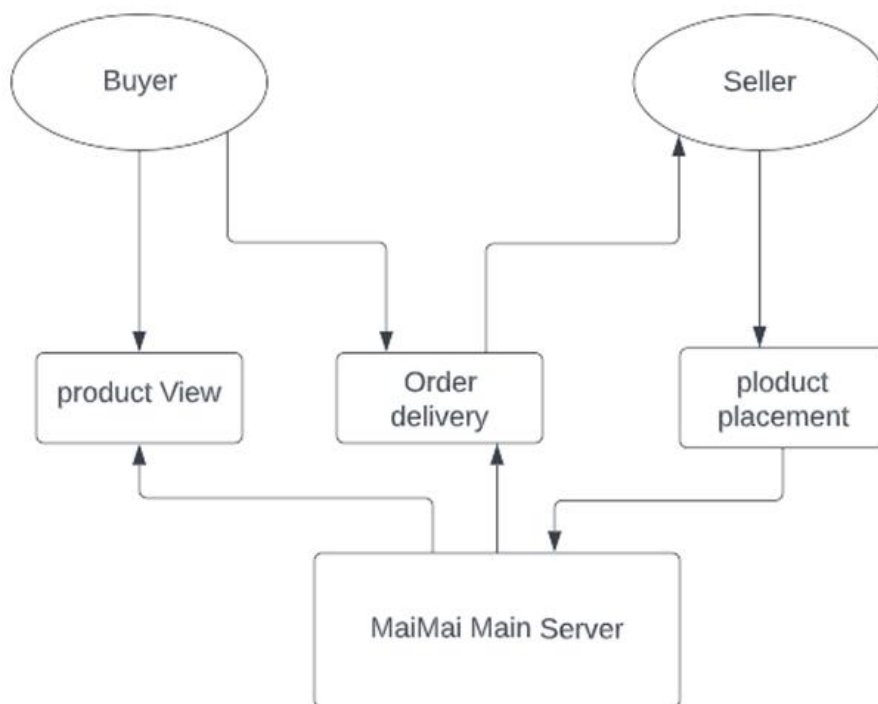


#### ii. 黑板风格

<MAiMAi>	
用例规范： <用例名称	日期： <11/12/2022

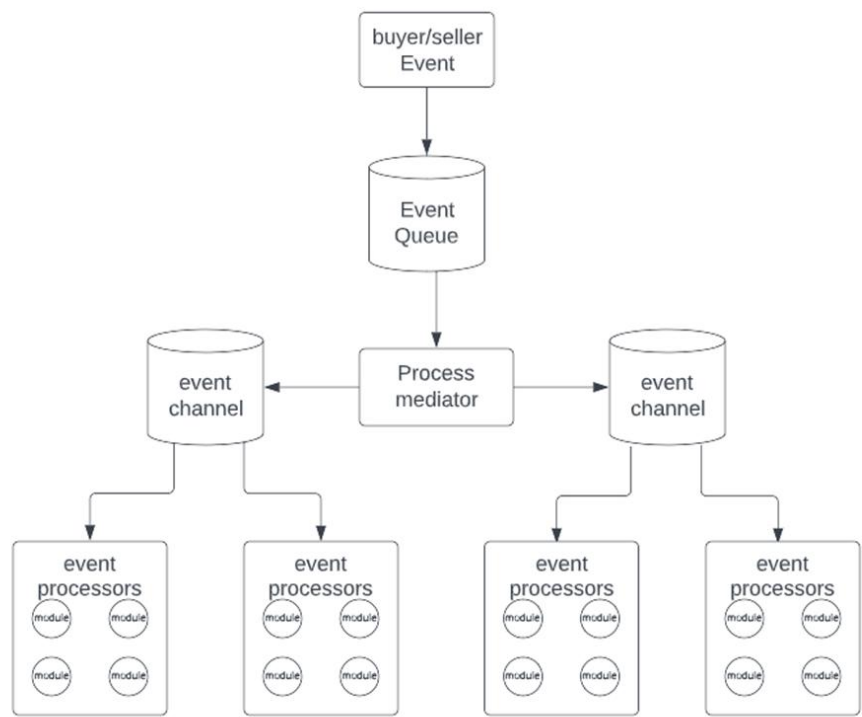


### iii. 发布-订阅方式



<MAiMai>	
用例规范： <用例名称	日期： <11/12/2022

iv. 基于活动的风格



25.2 云架构。

MaiMai 允许卖家在网站上发布产品。买家可以在网站上购买商家的商品。我们可能选择的基本服务模式是 PaaS，因此卖家也可以在 MaiMai 上创建自己的应用程序来展示他们的物品和产品。卖家可以使用 MaiMai 的 API 开发自己的应用程序。

我们选择的云应用架构是

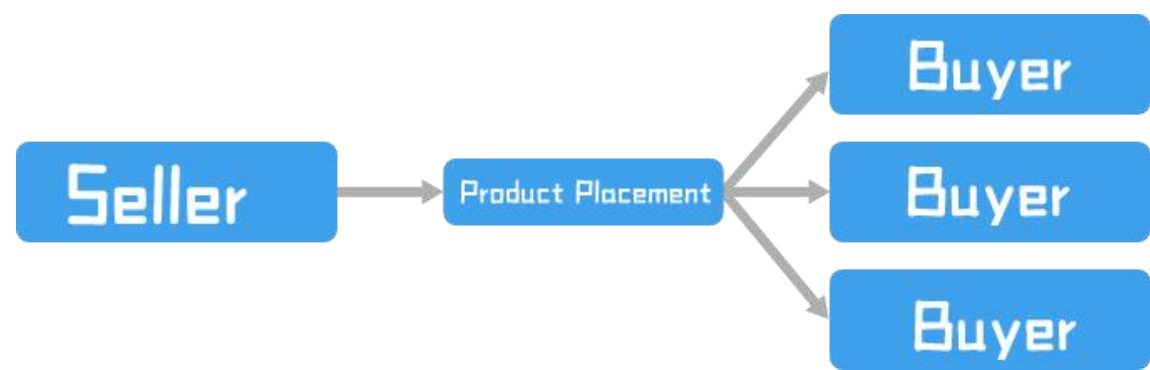
i. （可能）大数据

大数据云应用架构便于 MaiMai 系统使用。它有助于收集商品交易信息，进行实时数据分析或利用机器学习进行预测分析，了解买家的购物偏好，帮助卖家更好地销售商品，提高平台的交易量（Capilla 等，2020 年）。

ii. （主要）事件驱动

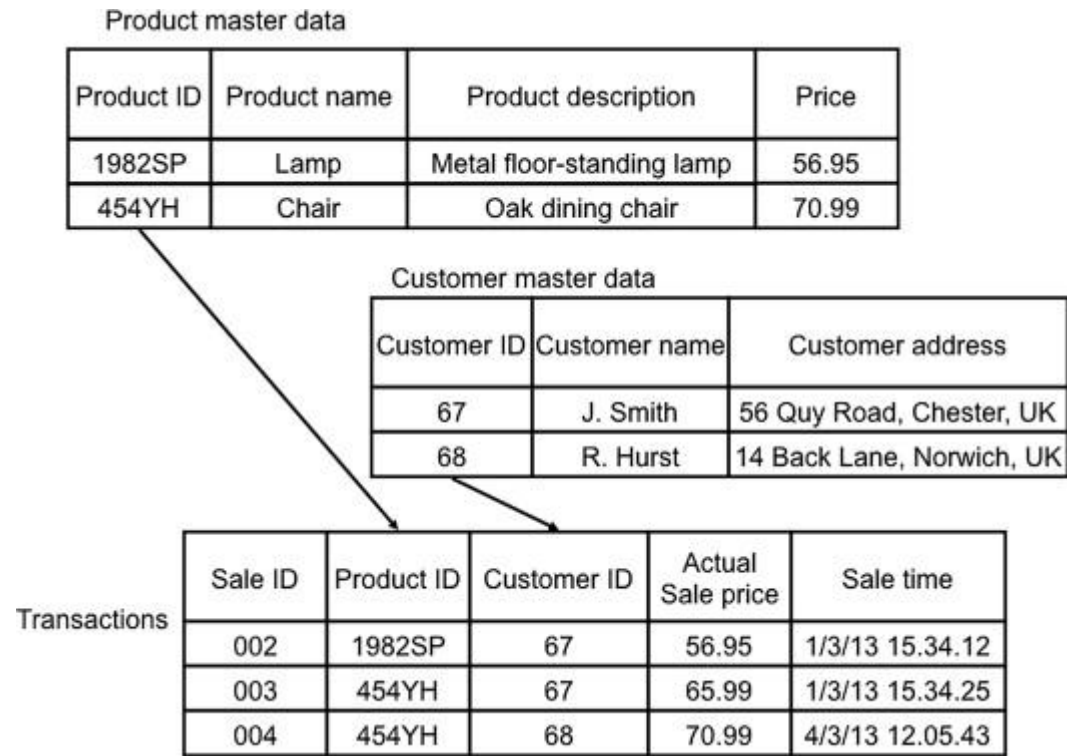
与 pub-sub 类似，当卖方（pub）发布二手物品交易时，买方（sub）可以立即收到该物品的最新信息。由于延迟时间较短，它允许实时发布物品并实时更新交易状态，从而避免了重复交易的错误。这种架构具有良好的可扩展性，因此可以同时处理许多发布的产品，并有能力存储任何产品信息。它还能轻松添加新用户。

<MAiMAi>	
用例规范： <用例名称	日期： <11/12/2022



### 25.3 数据库

事务是一组具有以下特性的操作：原子性、一致性、隔离性和持久性（ACID）。对事务的支持可以开发新的应用程序，同时简化开发流程，使应用程序更加健壮(Weber, Gabriel, Lux, & Menke, 2022)。事务数据通常与组织的事务有关，包括产品销售或购买时获取的数据。主数据在不同的交易中被提及，例如客户、产品或卖方数据。



对于我们的应用程序“麦麦”来说，交易的特点可以体现在很多功能中。例如，当用户需要购买一件商品时，交易过程会经历以下步骤：检查账户是否有足够的钱支付费用。如果账户中有足够的钱，则从账户中扣除该金额，并将其记入“MaiMai”平台。卖方发送货物，平台记录数据。麦麦“把钱寄给卖家。

事务性数据将存储在搜索引擎数据库中，因为搜索引擎数据库的主要特点是能够快速存储信息和编制索引，并为搜索请求提供快速响应时间 Launio, E. (2019). 索引可以是多维的，支持对大量文本数据进行自由文本搜索。索引可以通过搜索引擎数据库触发的拉模式或推模式启动的外部应用代码来执行(Weber, Gabriel, Lux, & Menke, 2022)。

<MAiMAi>	
用例规范： <用例名称	日期： <11/12/2022

### 我们应用程序的用户数据

键/值存储。用户可以通过关键字索引找到所需的项目。单个键/值存储具有很强的可扩展性，因为数据存储可以在不同机器上的多个节点上有效地分发数据。

### 26. 高保真线框

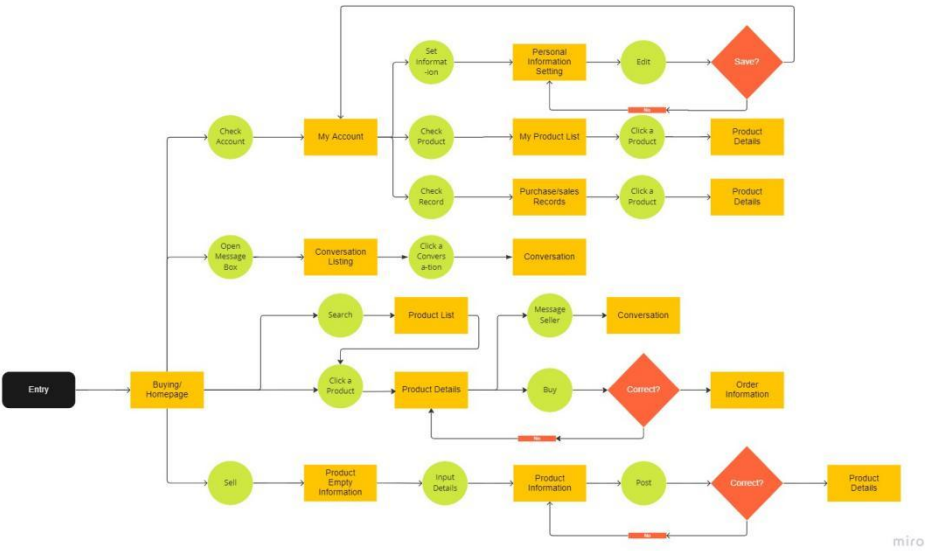


图 1：用户体验流程图

进入应用程序后，用户将看到主页，这也是购买页面。在该页面中，用户发布的所有产品都将按所选顺序和类别列出，方便用户选择和购买。进入该页面后，用户可以选择搜索或进入特定产品页面进行购买。在购买之前，程序会与用户进行确认，然后再确认订单。用户还可以选择与商家交流，进入对话页面。

用户还可以通过主页访问销售页面、我的账户页面和留言信箱页面。

在销售页面，用户可以通过填写物品信息快速发布想要销售的物品，应用程序还会在发布前与用户确认，以确保发布信息的准确性。

在账户页面，用户可以轻松访问个人信息页面并进行修改。用户还可以查看自己的买卖记录和正在出售的物品。

在消息框页面，用户可以继续交流并查看聊天记录。

<MAiMAi>	
用例规范: <用例名称	日期: <11/12/2022

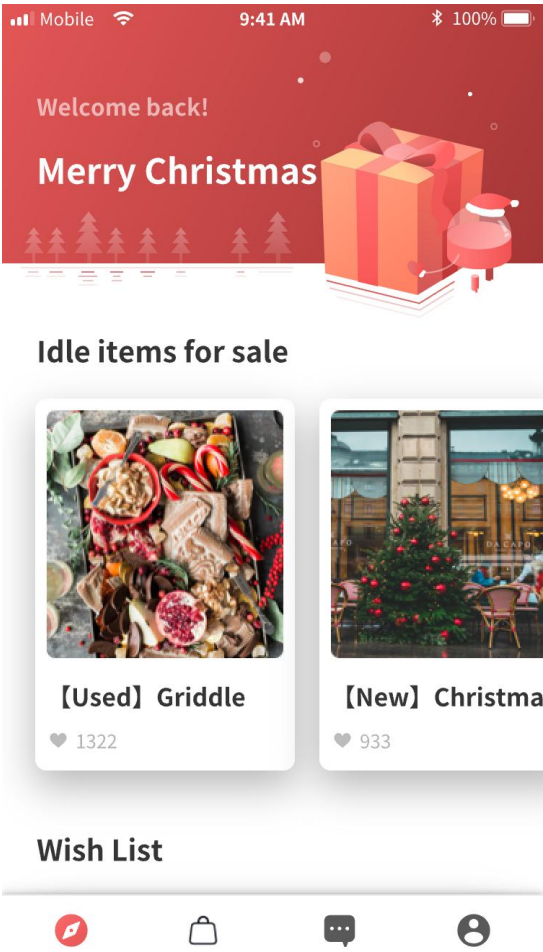


图 2: 主页

<MAiMAi>	
用例规范： <用例名称	日期： <11/12/2022

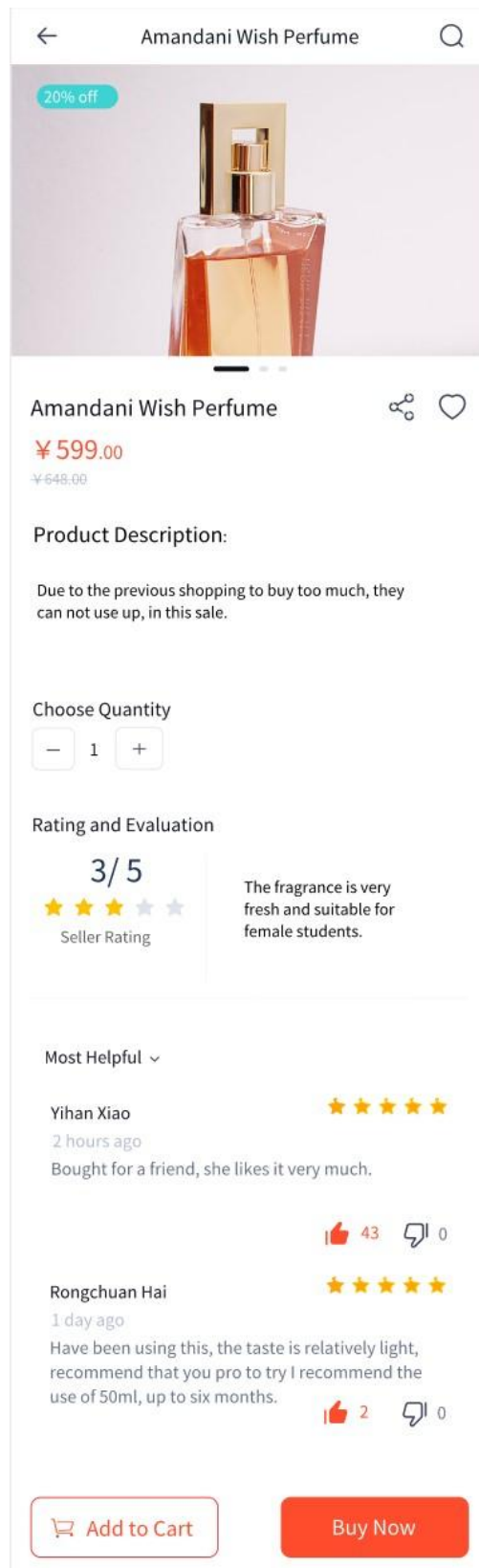


图 3：产品信息

<MAiMAi>	
用例规范: <用例名称	日期: <11/12/2022

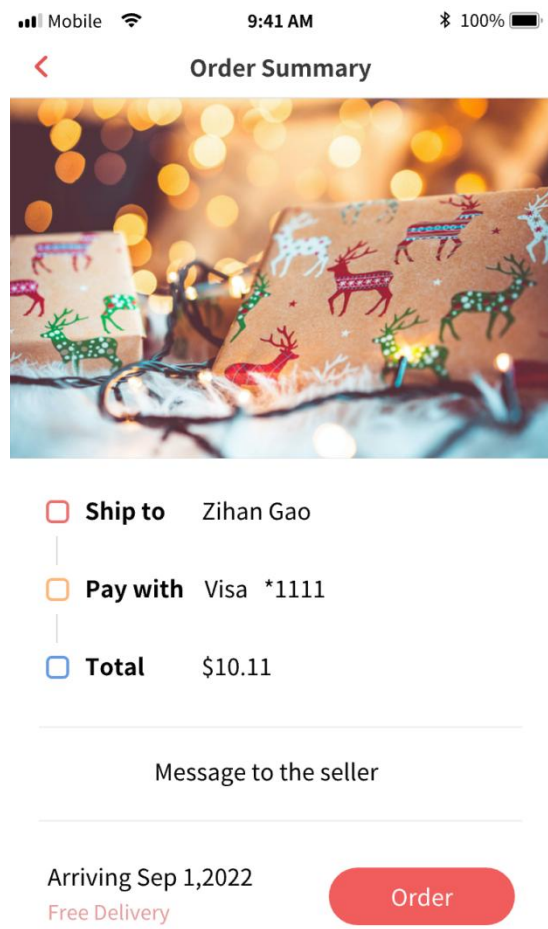


图 4: 订单摘要



<MAiMAi>	
用例规范： <用例名称	日期： <11/12/2022

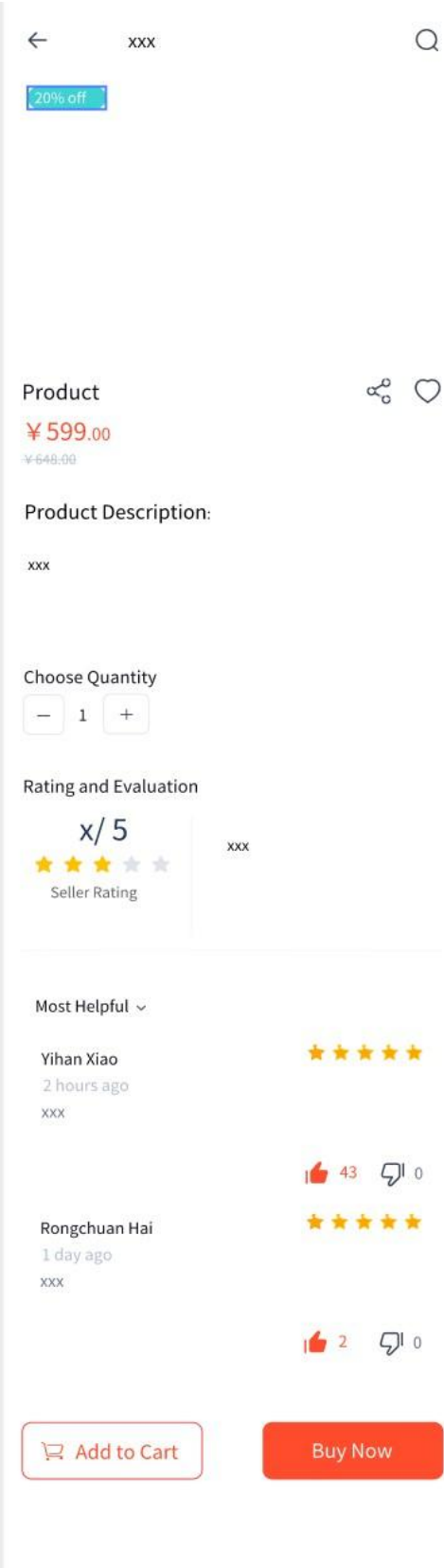


图 5：产品空信息

<MAiMAi>	
用例规范: <用例名称	日期: <11/12/2022

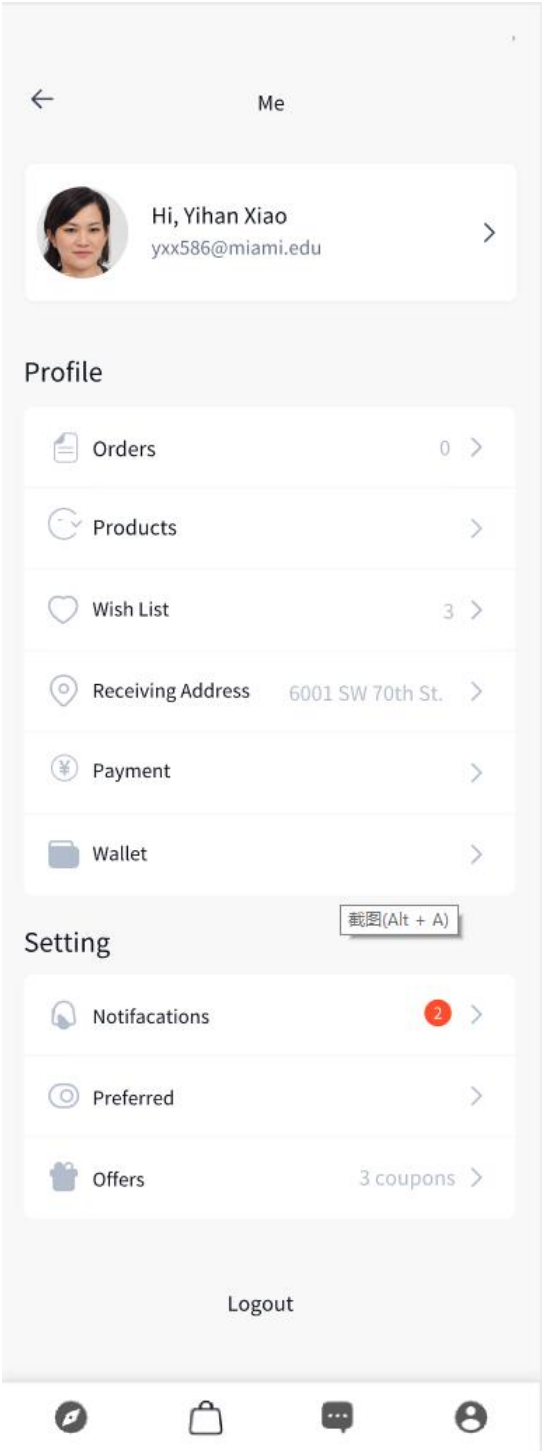


图 6: 我的账户

<MAiMAi>	
用例规范: <用例名称	日期: <11/12/2022

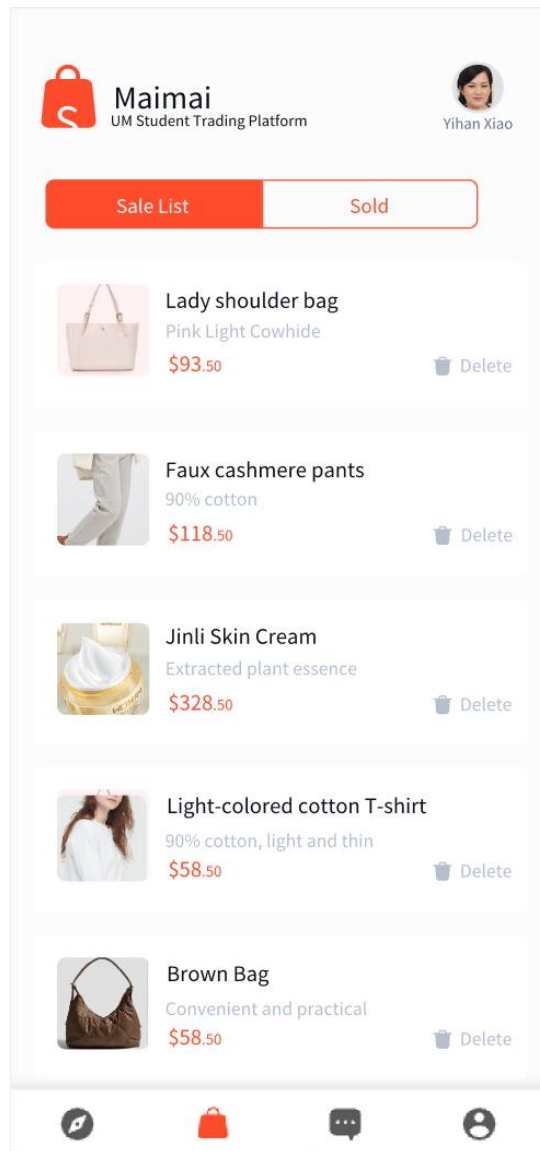


图 7：销售状况

<MAiMAi>	
用例规范: <用例名称	日期: <11/12/2022

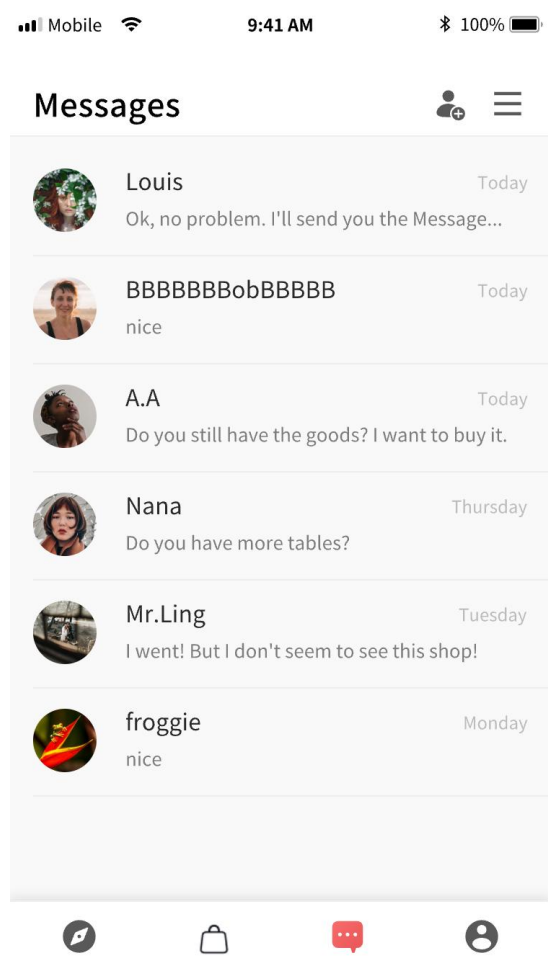


图 8: 信息框

<MAiMAi>	
用例规范: <用例名称	日期: <11/12/2022

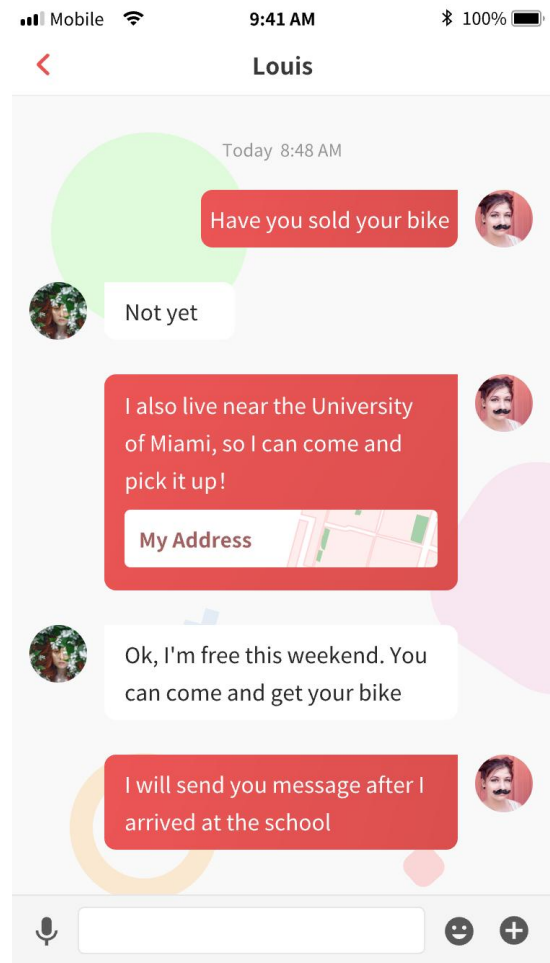


图 9：对话

## 27. 技术堆栈

### 27.1 前端

- a. JavaScript 轻量级、解释型、面向对象语言，具有一流的功能
- b. jQuery: jQuery 是一个跨平台 JavaScript 库，旨在简化 HTML 的客户端脚本编写。它是技术堆栈中 JavaScript UI 库类别中的一个工具。
- c. PHP: PHP: 快速、灵活、实用，从你的博客到世界上最流行的网站，PHP 为一切提供动力。PHP 是技术堆栈中语言类的一种工具。
- d. HTML5HTML5 是互联网的核心技术标记语言，用于构建和展示万维网内容。
- e. NGINX: nginx [engine x] 是一个 HTTP 和反向代理服务器，也是一个邮件代理服务器，由 Igor Sysoev 编写。NGINX 是技术堆栈中网络服务器类别的一个工具。
- f. Bootstrap: Bootstrap 是最流行的 HTML、CSS 和 JS 框架，用于开发响应性强、移动优先的网络项目。
- g. GitGit 是一个免费开源的分布式版本控制系统，旨在快速高效地处理从小型到超大型的所有项目。Git 属于技术堆栈中的版本控制系统类别。

<MAiMAi>	
用例规范: <用例名称	日期: <11/12/2022

## 27.2 后台逻辑

- h. JavaScript 轻量级、解释型、面向对象的语言，具有一流的功能。
- i. ES6: ECMAScript 2015 的目标包括为大型应用程序、库创建以及将 ECMAScript 用作其他语言的编译目标提供更好的支持。它的一些主要改进包括模块、类声明、词块范围、迭代器和生成器、异步编程承诺、重构模式和适当的尾部调用。
- j. RedisRedis 是一种开源（BSD 许可）内存数据结构存储，可用作数据库、缓存和消息代理。Redis 提供字符串、哈希值、列表、集合、范围查询排序集合、位图、超日志、地理空间索引和流等数据结构。
- k. ExpressJS: Express 是一个最小且灵活的 node.js 网络应用程序框架，为构建单页面、多页面和混合网络应用程序提供了一套强大的功能。
- l. AWS Lambda: AWS Lambda 是一种计算服务，可根据事件运行您的代码，并自动为您管理底层计算资源。您可以使用 AWS Lambda 以自定义逻辑扩展其他 AWS 服务，或创建自己的后端服务，以 AWS 的规模、性能和安全性运行。

## 27.3 数据库

- m. sql-server:sql-server 是技术栈 Terraform 包类别中的一个工具。
- n. MySQLMySQL 软件提供了一个非常快速、多线程、多用户和强大的 SQL（结构化查询语言）数据库服务器。MySQL 服务器适用于关键任务、大负荷生产系统，也适用于嵌入到大规模部署的软件中。

## 27.4 DevOps

- o. DockerDocker Platform 是业界领先的容器平台，可实现持续、高速的创新，使企业能够无缝地构建和共享任何应用程序（从传统应用程序到下一代应用程序），并在任何地方安全地运行它们。Docker 是技术堆栈虚拟机平台和容器类别中的一种工具。
- p. Jenkins 可扩展的开源持续集成服务器简而言之，Jenkins CI 是领先的开源持续集成服务器。它采用 Java 构建，提供 300 多个插件，支持几乎所有项目的构建和测试。Jenkins 是技术栈中的持续集成类工具。Jenkins 是一款开源工具，在 GitHub 上拥有 20K 个星级和 7.8K 个 GitHub 分叉。
- q. TeamCityTeamCity 是面向专业人士的终极持续集成工具。TeamCity 是一款面向专业开发人员、构建工程师和 DevOps 的用户友好型持续集成（CI）服务器。它易于设置，对小型团队和开源项目免费。TeamCity 是技术栈中的持续集成类工具。

<MAiMAi>	
用例规范：<用例名称>	日期：<11/12/2022

## 28. 工作分解结构

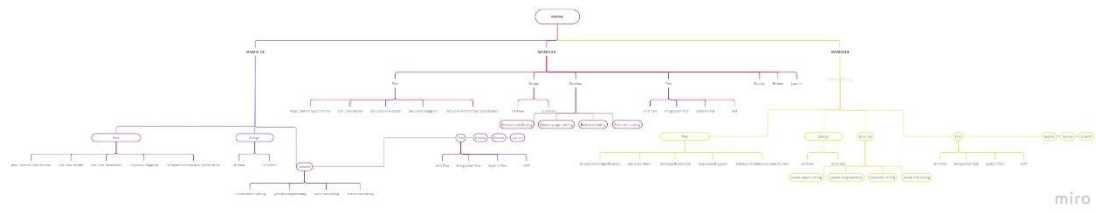
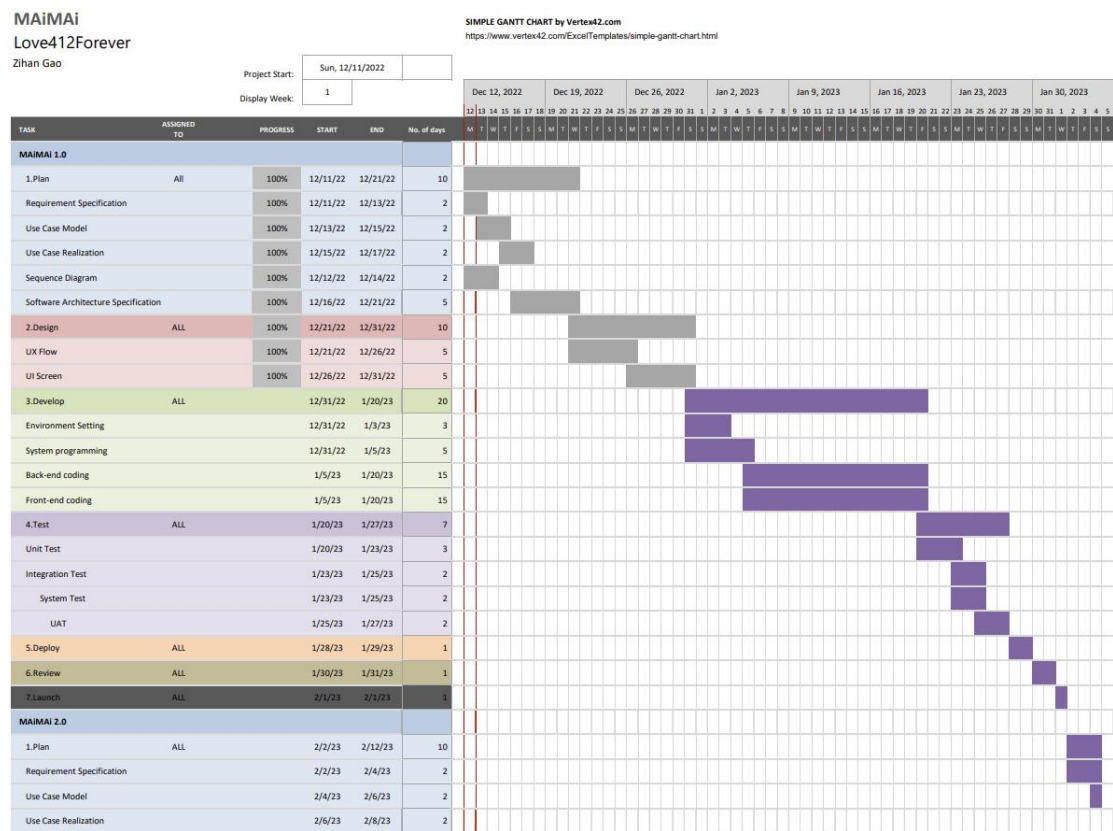
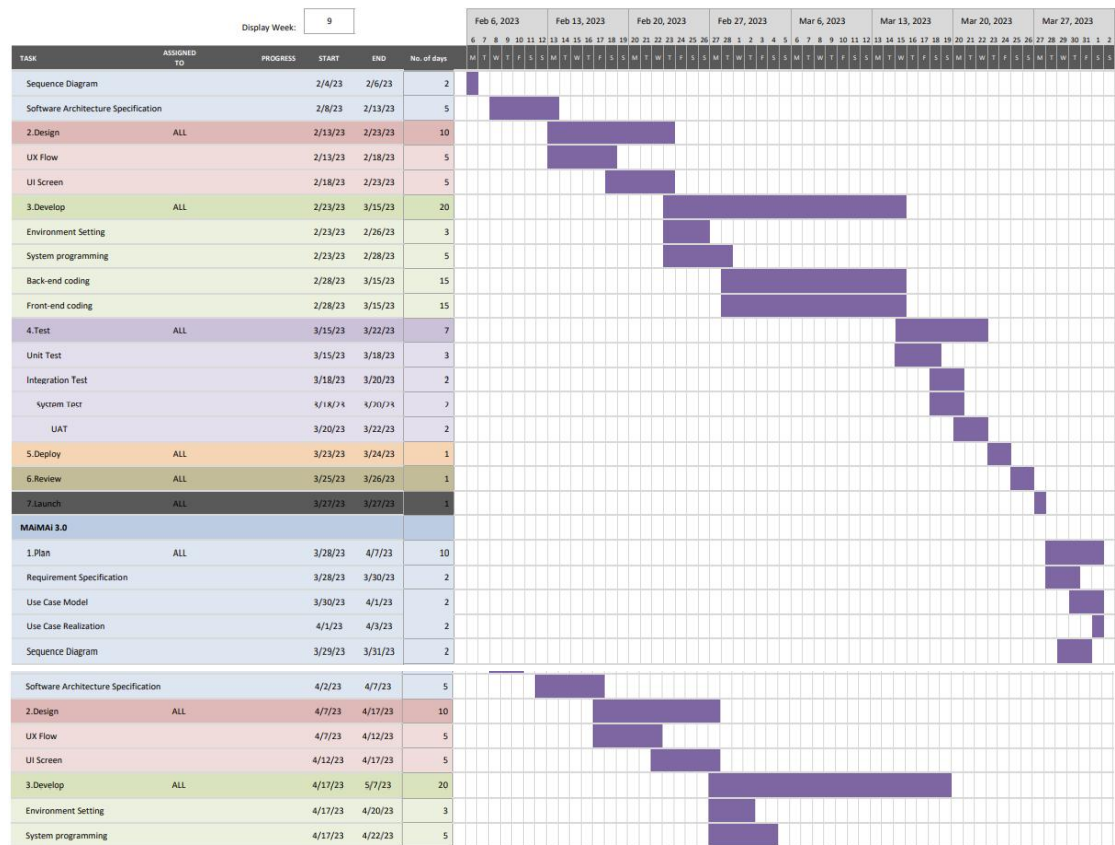


图 1：工作分解结构图

## 29. 项目时间表计划



<MAiMAi>	
用例规范: <用例名称	日期: <11/12/2022



Page 2 of 3

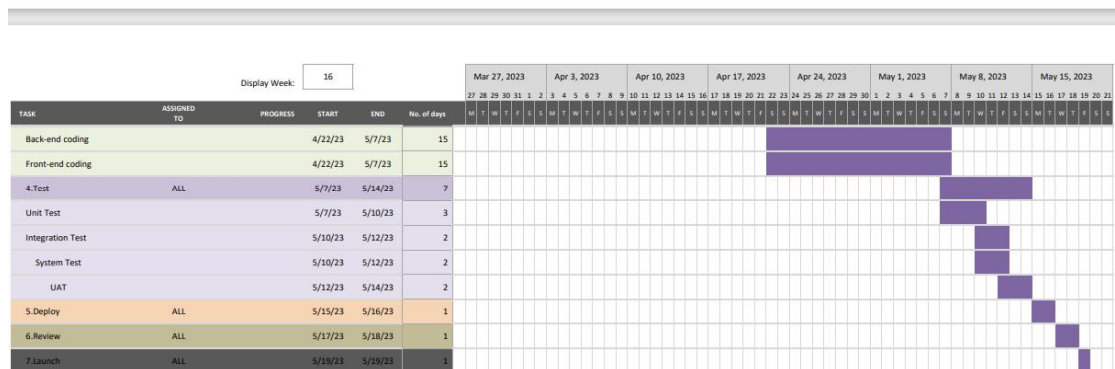


图 2: 项目时间表计划



<MAiMAi>	
用例规范： <用例名称	日期： <11/12/2022

### 30. 项目成本计划

Project Lead: Zihan Gao							BUDGET	ACTUAL	Under(Over)
Start Date: Sun, 12/11/2022							Total \$	\$	\$
							7,484.31	7,465.00	19.31
WBS	Task	Labor		Materials		Fixed	Budget	Actual	Under(Over)
		Hrs	Rate	Units	\$/Unit	Costs			
1	MAiMAi 1.0						\$ 2,177.69	\$ 2,205.00	\$ (27.31)
1.1	Plan	1.2	\$12.50	5.0	\$2.89	\$51.06	80.37	150.00	(69.63)
1.1.1	Requirement Specification	10.6	\$11.25	6.3	\$1.79	\$250.60	381.16	400.00	(18.84)
1.1.2	Use Case Model	9.1	\$11.35	4.8	\$2.30	\$48.99	114.23	100.00	14.23
1.1.3	Use Case Realization	9.9	\$12.31	5.5	\$1.70	\$50.31	131.15	140.00	(8.85)
1.1.4	Sequence Diagram	8.3	\$11.76	6.0	\$2.45	\$20.30	112.31	120.00	(7.69)
1.1.5	Software Architecture Specification	15.4	\$13.45	3.9	\$2.14	\$20.30	215.44	230.00	(14.56)
1.2	Design	1.0	\$14.87	9.9	\$4.95	\$4.50	63.73	65.00	(1.27)
1.2.1	UX Flow	12.6	\$14.45	10.5	\$5.06	\$44.98	280.38	250.00	30.38
1.2.2	UI Screen	13.5	\$13.65	10.7	\$5.73	\$46.80	292.10	300.00	(7.90)
1.3	Develop	1.1	\$17.64	11.1	\$5.36	\$47.00	125.90	100.00	25.90
1.3.1	Environment setting	18.4	\$16.35	9.9	\$8.09	\$43.00	380.93	350.00	30.93
1.3.2	System programming	31.0	\$15.89	10.0	\$7.77	\$44.50	570.29	500.00	70.29
1.3.3	Back-end coding	91.4	\$17.01	10.3	\$8.03	\$45.50	1,637.42	1,700.00	(62.58)
1.3.4	Front-end coding	89.7	\$16.57	11.4	\$8.30	\$44.25	1,580.95	1,500.00	80.95
1.4	Test	1.2	\$15.37	5.5	\$11.97	\$61.75	146.03	100.00	46.03
1.4.1	Unit Test	17.9	\$15.80	13.4	\$3.68	\$60.70	392.83	400.00	(7.17)
1.4.2	Integration Test	12.1	\$17.78	15.6	\$4.03	\$58.00	336.01	330.00	6.01
1.4.3	System Test	12.4	\$16.75	14.6	\$4.40	\$62.00	333.94	300.00	33.94
1.4.4	UAR	11.9	\$16.00	14.3	\$4.84	\$61.50	321.11	300.00	21.11
1.5	S.Deploy	7.9	\$15.05	13.9	\$31.66	\$49.70	608.67	500.00	108.67
1.6	6.Review	8.2	\$13.96	15.0	\$33.04	\$40.00	650.07	500.00	150.07
2	MAiMAi 2.0						\$ 2,177.69	\$ 2,205.00	\$ (27.31)
2.1	Plan	1.2	\$12.50	5.0	\$2.89	\$51.06	80.37	150.00	(69.63)
2.1.1	Requirement Specification	10.6	\$11.25	6.3	\$1.79	\$250.60	381.16	400.00	(18.84)
2.1.2	Use Case Model	9.1	\$11.35	4.8	\$2.30	\$48.99	114.23	100.00	14.23
2.1.3	Use Case Realization	9.9	\$12.31	5.5	\$1.70	\$50.31	131.15	140.00	(8.85)
2.1.4	Sequence Diagram	8.3	\$11.76	6.0	\$2.45	\$20.30	112.31	120.00	(7.69)
2.1.5	Software Architecture Specification	15.4	\$13.45	3.9	\$2.14	\$20.30	215.44	230.00	(14.56)
2.2	Design	1.0	\$14.87	9.9	\$4.95	\$4.50	63.73	65.00	(1.27)
2.2.1	UX Flow	12.6	\$14.45	10.5	\$5.06	\$44.98	280.38	250.00	30.38
2.2.2	UI Screen	13.5	\$13.65	10.7	\$5.73	\$46.80	292.10	300.00	(7.90)
2.3	Develop	1.1	\$17.64	11.1	\$5.36	\$47.00	125.90	100.00	25.90
2.3.1	Environment setting	18.4	\$16.35	9.9	\$8.09	\$43.00	380.93	350.00	30.93
2.3.2	System programming	31.0	\$15.89	10.0	\$7.77	\$44.50	570.29	500.00	70.29
2.3.3	Back-end coding	91.4	\$17.01	10.3	\$8.03	\$45.50	1,637.42	1,700.00	(62.58)
2.3.4	Front-end coding	89.7	\$16.57	11.4	\$8.30	\$44.25	1,580.95	1,500.00	80.95
2.4	Test	1.2	\$15.37	5.5	\$11.97	\$61.75	146.03	100.00	46.03
2.4.1	Unit Test	17.9	\$15.80	13.4	\$3.68	\$60.70	392.83	400.00	(7.17)
2.4.2	Integration Test	12.1	\$17.78	15.6	\$4.03	\$58.00	336.01	330.00	6.01
2.4.3	System Test	12.4	\$16.75	14.6	\$4.40	\$62.00	333.94	300.00	33.94
2.4.4	UAR	11.9	\$16.00	14.3	\$4.84	\$61.50	321.11	300.00	21.11
2.5	S.Deploy	7.9	\$15.05	13.9	\$31.66	\$49.70	608.67	500.00	108.67
2.6	6.Review	8.2	\$13.96	15.0	\$33.04	\$40.00	650.07	500.00	150.07
3	MAiMAi 3.0						\$ 2,177.69	\$ 2,205.00	\$ (27.31)
3.1	Plan	1.2	\$12.50	5.0	\$2.89	\$51.06	80.37	150.00	(69.63)
3.1.1	Requirement Specification	10.6	\$11.25	6.3	\$1.79	\$250.60	381.16	400.00	(18.84)
3.1.2	Use Case Model	9.1	\$11.35	4.8	\$2.30	\$48.99	114.23	100.00	14.23
3.1.3	Use Case Realization	9.9	\$12.31	5.5	\$1.70	\$50.31	131.15	140.00	(8.85)
3.1.4	Sequence Diagram	8.3	\$11.76	6.0	\$2.45	\$20.30	112.31	120.00	(7.69)
3.1.5	Software Architecture Specification	15.4	\$13.45	3.9	\$2.14	\$20.30	215.44	230.00	(14.56)
3.2	Design	1.0	\$14.87	9.9	\$4.95	\$4.50	63.73	65.00	(1.27)
3.2.1	UX Flow	12.6	\$14.45	10.5	\$5.06	\$44.98	280.38	250.00	30.38
3.2.2	UI Screen	13.5	\$13.65	10.7	\$5.73	\$46.80	292.10	300.00	(7.90)
3.3	Develop	1.1	\$17.64	11.1	\$5.36	\$47.00	125.90	100.00	25.90
3.3.1	Environment setting	18.4	\$16.35	9.9	\$8.09	\$43.00	380.93	350.00	30.93

图 3：项目成本计划

<MAiMAi>	
用例规范： <用例名称	日期： <11/12/2022

<MAiMAi>	
用例规范: <用例名称	日期: <11/12/2022

## 参考资料

Capilla, R., Kazman, R., Romera, C., & Carrillo, C. (2020). 软件架构中的可用性影响:

移动应用程序案例研究。《软件: 实践与经验》, 50 (12), 2145-2168。

Gruhn, V., & Köhler, A. (2006). 根据业务需求调整移动应用的软件架构。In *WISM*.

De Lima Fontao, A., dos Santos, R. P., & Dias-Neto, A. C. (2015, July). 移动软件生态

系统 (mseco): 系统映射研究。In *2015 IEEE 39th Annual Computer Software and*

*Applications Conference* (Vol. 2, pp. 653-658). IEEE.

Launio, E. (2019). 纸浆供应链的数字化与交易信息流分析》。

Weber, P., Gabriel, R., Lux, T., & Menke, K. (2022). 数据库系统。In *Basics in Business*

*Informatics* (pp. 123-150). Springer Vieweg, Wiesbaden.

亚历山大-博雷克, ... 菲利普-伍德, 《全面信息风险管理》, 2014 年。

我们的流程设计工具: <https://miro.com>

线框设计工具: <https://js.design/js.design>  
<https://js.design/f/CHQPii?p=9C797875>  
<https://js.design/f/wMe8ji?p=ksmNJUCfgt>

技术栈参考: <https://stackshare.io/stacks>