|  |
| --- |
| **C2C Marketplace** |
| Architecture Design Document |
|  |
|  |
| **<<date>>** |
| **Nikhil Gupta** |

This document is an Architecture Design Document for developing **C2C Marketplace**.

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Description |
| 0.1 | <<date>> | Nikhil Gupta | Initial document creation |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

[1. Overview 3](#_Toc516321189)

[2. Requirements 7](#_Toc516321190)

[2.1. Functional Requirements 7](#_Toc516321191)

[2.2. Non-functional Requirements 12](#_Toc516321192)

[2.3. Quality Attributes 13](#_Toc516321193)

[3. Architecture 14](#_Toc516321194)

[4. Modules 15](#_Toc516321195)

[Appendix 16](#_Toc516321196)

[A. Domain Model 17](#_Toc516321197)

[B. Quality Scenarios 18](#_Toc516321198)

[C. Quality Scenario Analysis 19](#_Toc516321199)

[D. Candidate Architectures 20](#_Toc516321200)

[E. Candidate Architecture Evaluation 21](#_Toc516321201)

[F. Architecture Design 22](#_Toc516321202)

[G. Architecture Evaluation(ATAM) 23](#_Toc516321203)

# Overview

// A1. System Definition

// C1-1. Is the defined system boundary clear?

// C1-2. Is there sufficient explanation of the system’s operation and business environment as business drivers?

* 1. **Introduction**

A Customer-to-Customer (C2C) Marketplace is a platform that allows users to **safely buy and sell** both used and new products. It provides a digital space where individuals can easily list their items and buyers can discover items they want using advanced search. The Marketplace should ensure that all the activities in the system are safe & secure.

* 1. **System Definition**

The purpose of this project is to design a Customer-to-Customer (C2C) Marketplace.

Below Figure 1 – depicts the System Boundary and how C2C Marketplace will interact with the outside Components.

* System will provide interface for Actors like Buyer and Seller.
* System will also interact with an external interface which is
  + External Payment Service Provider
  + External Notification Service

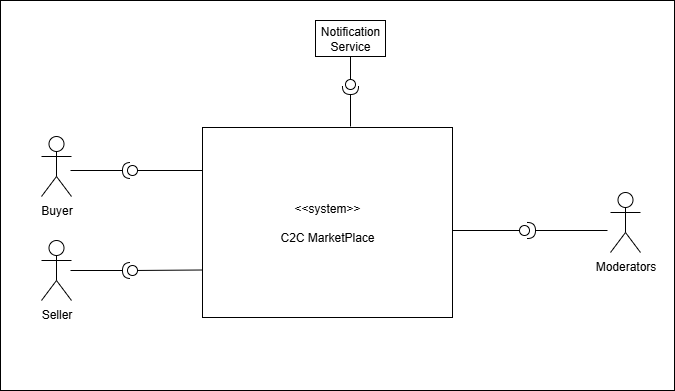


Figure 1a - **System boundary for C2C Marketplace**

* 1. **Business Context & Drivers**

~~Our marketplace serves buyers and sellers across regions, devices, and network conditions. Inventory is user-generated and long-tail, with uneven titles, images, and metadata; demand is spiky and intent can fade quickly. Shoppers expect instant, relevant answers and smooth handoffs from search to chat to checkout.~~

~~The peer-to-peer nature also introduces real risks—counterfeits, scams, off-platform payments—that must be contained without adding friction. Success therefore hinges on surfacing the right items fast, keeping interactions responsive, and building enough trust for people to complete the deal. These realities lead directly to the following business drivers:~~

Our marketplace connects buyers and sellers across many regions, devices, and internet speeds. Since listings are created by users and often cover niche items, the titles, photos, and details are inconsistent. Also, demand can rise suddenly, and shopper interest drops quickly. Shoppers expect instant, relevant results and a seamless handoff from search to chat to checkout.

The peer-to-peer model also creates real risks, including counterfeits, scams, and attempts to pay off the platfo

rm. We must reduce these risks without adding any friction. Our success depends on showing the right items quickly, keeping every interaction responsive, and building enough trust for people to complete their purchase. These realities lead to the following business drivers.

1. **~~Speed & Responsiveness:~~** ~~Keep search and checkout interactions fast; ensure new listings index quickly.~~
2. **Relevance & Recommendations:** Return highly relevant results and personalized recommendations to boost discovery and intent.
3. **Trust & Safety:** Protect buyers/sellers with strong fraud prevention, content moderation, secure payments, and fair disputes.
4. **Conversion:** Maximize search-to-purchase completion by removing friction across discovery, negotiation, checkout, and fulfillment.

# Requirements

## Functional Requirements

// A2. Functional Requirement Specification

// C2-1. Is there sufficient functional requirement specification to affect the system’s architecture?

// C2-2. Is the relationship between use cases clear?

// C2-3. Is the division of use cases explicit?

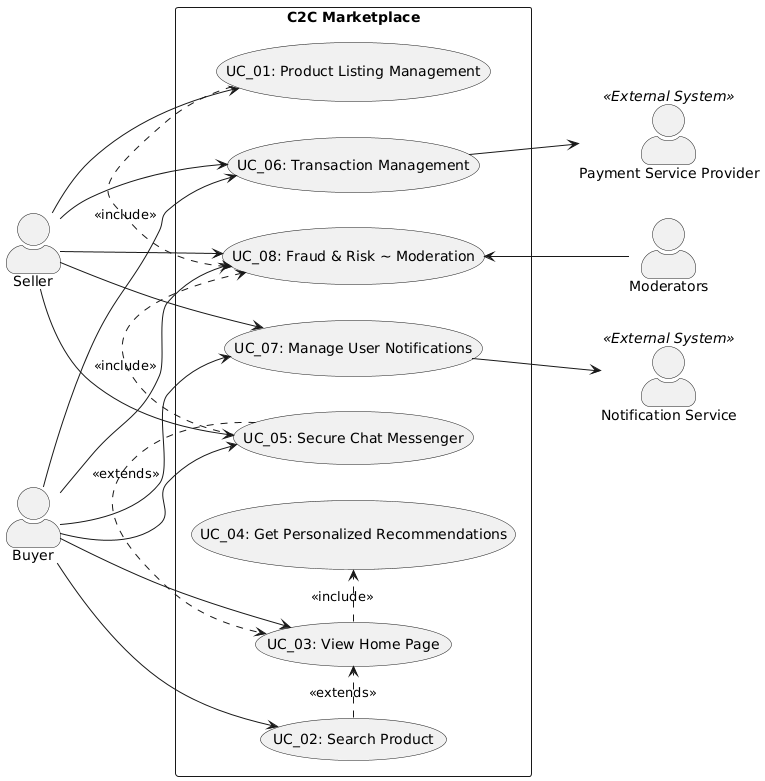


Figure 2: Use Case Diagram

Discussion Items :

1 chat & transaction(Seq & Status  – >payment ) now look similar

Checkpoints ( by what symbol to show external systems?

|  |  |
| --- | --- |
| **UC\_01** | Product Listing Management |
| Description | Seller can Create a new product listing & Update or Delete an existing product listing. |
| Actor | User (Seller) |
| Pre-condition | Seller is signed-in and authorized to make a new listing |
| Post-condition | New Listing is created in the system and changes are it is propagated to Search & recommendation Modules. |
| Basic Flow | 1. Seller uploads images and enters common item details like Category, Title, Description, Price, Location etc. (AF1)  2. Based on the Selected Category, System automatically shows additional fields related to the category selected by Seller.  3. The system runs content moderation checks to validate the filled data. (AF2)  4. The system opens the Preview Listing Page.  5. Seller validates the filled data and publishes the listing.  <Should we mention something about ads posting as well> |
| Additional Flow | AF1. The system auto saves drafts regularly and lets the seller restore the last saved version.  AF2. If policies are violated, the system explains the issue and offers fix options or an appeal to review. |

|  |  |
| --- | --- |
| **UC\_02** | Search Product |
| Description | Buyer can search for an Item by entering keywords & the system returns the results by using semantic search on entered Keywords and location. |
| Actor | User (Buyer) |
| Pre-condition | 1. Listings are already created and are indexed  2. User location are available |
| Post-condition | Results returned with categorical breakdowns |
| Basic Flow | 1. Buyer enters search keywords and queries  2. The system executes search query using all the inputs provided by user. (AF1)  3. The system re-ranks the results with personalization.  4. The system returns paginated results to the user. |
| Additional Flow | AF1. If No results are found system returns a default search result |

|  |  |
| --- | --- |
| **UC\_03** | Home Page |
| Description | User Opens the homepage of app or web interface, homepage is loaded along with recommendations for the user |
| Actor | User |
| Pre-condition | Internet Connection should be present. |
| Post-condition | System will load homepage along with the recommendation data |
| Basic Flow | 1. User opens the homepage of the application or web interface.  2. The system initializes the session context, checking for authentication, locale, and geo-location.  3. The system fetches the recommendation for this user (UC\_04) (AF1)  4. The system also fetches targeted advertisements or sponsored listings from sellers.  5. The system also fetches more data like trending items or recently viewed items.  6. The system combines and sends back all data needed to populate the homepage.  7. The system displays the final home Screen to the user |
| Additional Flow | AF1: If the user is new or not logged in, the system has no personalization data. Instead of a personalized feed, it will return a generic feed consisting of trending items, location-based bestsellers, or manually curated recommendations. |

|  |  |
| --- | --- |
| **UC\_04** | Personalized Recommendations |
| Description | This use case describes how the system provides proactive, context-aware product recommendations to users for various use cases |
| Actor | User (Buyer) |
| Pre-condition | 1. User is logged in. |
| Post-condition | A set of personalized and relevant items is displayed to the user. |
| Basic Flow | Scenario A: Recommendations on the Home Page  1. Buyer opens the home page.  2. The system generates a personalized feed by selecting and ranking relevant products based on user data and system policies.  3. The final, curated list of recommended items is then displayed to the user on the home page  Scenario B: Product Detail Page Recommendations  1. Buyer opens the product detail page of a specific item that he is interested in.  2. The system generates a recommendation of related Items for this item.  3. Related Items for this item is shown to the user. |
| Additional Flow | AF1. Cold start: For new users, since use trending near location or manually recommended items.  AF2. Latency guardrails: If the process of calculating the personalized recommendations takes too much time, simply return cached/trending; if that is also empty, suppress the widget. |

|  |  |
| --- | --- |
| **UC\_05** | Secure Chat Messenger |
| Description | Chat feature enables Buyers & Sellers to communicate via secure-in-app chat to make offers (or receive counter offers). |
| Actor | User (Buyer & Seller) |
| Pre-condition | 1. Buyer & Seller are authenticated.  2. A product listing must be active and available in the system. |
| Post-condition | 1. All messages are saved in the system for future reference and auditing  2. In case the offer is accepted mark the item as sold. |
| Basic Flow | 1. The Buyer navigates to a product listing and initiates a chat with the Seller by sending a message.  2. The Buyer or the Seller can send a formal offer that includes a specific price.  3. The user who receives the offer can choose to accept it, decline it, or send a counter-offer with a new price.  4. The system sends a notification to the other user for each new message, offer, or response  5. This flow (steps 2-4) continues until an offer is accepted or one of the users ends the conversation. |
| Additional Flow | AF1: Automated Content Filtering: If a message contains content that violates policies, such as requests for off-platform payments, the system automatically blocks the message to prevent scams.  AF2: Spam Prevention: The system monitors message frequency and will temporarily limit a user's ability to send messages if they are suspected of spam or abuse. |

|  |  |
| --- | --- |
| **UC\_06** | Transaction Management |
| Description | This use case describes the process of managing a transaction after a buyer decides to purchase an item |
| Actor | User (Buyer & Seller) |
| Pre-condition | 1. Buyer & Seller are authenticated.  2. A product listing must be active and available in the system. |
| Post-condition |  |
| Basic Flow | 1. Buyer submits a formal purchase request for an item.  2. System send a notification to the Seller informing them of the new purchase request.  3. System also immediately responds to the buyer, acknowledging that the request has been successfully submitted.  4. At a later time, the Seller reviews the purchase request and accepts it. (AF1)  5. System sends notification to the buyer to inform about seller’s action. |
| Additional Flow | AF1: If the Seller chooses to reject the purchase request, the system cancels the order and sends a notification to the Buyer informing them of the rejection. |

Discuss if this is through third party, need to change it accordingly.

|  |  |
| --- | --- |
| **UC\_07** | User Notifications |
| Description | This use case describes the system's event-driven notification feature, which sends timely and relevant alerts to users across multiple channels (push, in-app, email, SMS). |
| Actor | System (Notification Service), User (Buyer/Seller) |
| Pre-condition | 1. User account exists with verified contact details (email & contact number).  2. User notification preferences and locale stored.  3. Provider credentials configured |
| Post-condition | 1. The notification is sent (or scheduled) with delivery status recorded.  2. The system respects the user's preferences |
| Basic Flow | 1. An event occurs in another part of the system (e.g., an offer is received, a price drops).  2. Based on the event, the system determines the correct notification format and channel (e.g., email, SMS).  3. The system then customizes the message with specific details like the user's name and the item's price, translating it into the appropriate language.  4. The system checks the user's preferences, such as opt-in status, quiet hours, and frequency limits, before queuing the notification.  5. An external service provider sends the message via the selected channel (e.g., Email, SMS, Push Notification) to the user.  6. The system receives the status update (such as "delivered”, or "bounced”) from external service provider and records it. |
| Additional Flow | AF1: Retry & failover: If sending a notification fails due to a temporary external service provider issue, the system will automatically retry. |

|  |  |
| --- | --- |
| **UC\_08** | Fraud & Risk ~ Moderation |
| Description | This use case describes the proactive and reactive measures the system takes to ensure trust and safety on the platform. It covers the automated and manual moderation of user-generated content (listings, messages) , fraud detection during transactions, and the process for handling user-reported issues and disputes. |
| Actor | System (Automated Services), Moderators, User (Buyer & Seller) |
| Pre-condition | A user action has occurred that requires a safety check, such as submitting a new listing, sending a message, or a user reporting a piece of content. |
| Post-condition | The content or user action is either approved, rejected with a reason, or has been escalated for manual review. |
| Basic Flow | Scenario A: Proactive Content Moderation (New Listing) [UC\_01]  1. A seller posts product details for listing a new product.  2. The system's automated services scan the content (text and images) for policy violations.  3. If the content is clearly safe, it is approved and published automatically. (AF1)  Scenario B: Reactive Moderation (User Report)  1. A buyer or seller reports a piece of content (e.g., a suspicious message, a counterfeit item) or another user to support the platform's trust and safety goals.  2. The system creates a case and adds it to the moderation review queue.  3. A Moderator reviews the case, including the reported content and the users' histories.  4. The Moderator takes an appropriate action, such as removing the listing, warning the user, or suspending the account.  5. The system notifies the user who made the report about the outcome. |
| Additional Flow | AF1: If a policy violation is detected, the content is automatically rejected and the system notifies the seller with the specific issue and offers fix options.  AF2: Appeals Process: A user whose content was rejected by the automated system can submit an appeal, which is then routed to a human Moderator for a final review. |

## Non-functional Requirements

// A5. Quality Requirement Specification

// C5-2. Is the specification of quality requirements appropriate?

// C5-3. Is quality requirement measurable?

// C5-4. Is the allowance of non-functional requirement clear?

|  |  |  |
| --- | --- | --- |
| **NFR\_01** |  |  |
| Description |  | |
| Environment |  | |
| Stimulus |  | |
| Response |  | |
| Measure |  | |
| Allowance |  | |

## Quality Attributes

// A5. Quality Requirement Specification

// C5-2. Is the specification of quality requirements appropriate?

// C5-3. Is quality requirement measurable?

|  |  |  |
| --- | --- | --- |
| **QA\_01** |  |  |
| Description |  | |
| Environment |  | |
| Stimulus |  | |
| Response |  | |
| Measure |  | |

# Architecture

// A8. Architecture Documentation

// C8-1. Is allocation of processes, etc. appropriate? (deployment)

// C8-2. Is grouping appropriate in terms of components? (component & connector)

// C8-3. Is the description of the system architecture appropriate?

# Modules

// A9. Module Specification

// C9-1. Is component specification sufficient to develop?

// C9-2. Is grouping appropriate in terms of module?

// C9-3. Is it appropriate to design dependencies between modules?

// C9-4. Is the work assignment appropriate?

Appendix

[A. Domain Model 17](#_Toc516321204)

[B. Quality Scenarios 18](#_Toc516321205)

[C. Quality Scenario Analysis 19](#_Toc516321206)

[D. Candidate Architectures 20](#_Toc516321207)

[E. Candidate Architecture Evaluation 21](#_Toc516321208)

[F. Architecture Design 22](#_Toc516321209)

[G. Architecture Evaluation(ATAM) 23](#_Toc516321210)

1. Domain Model

// A3. Domain Model Design

// C3-1. Is domain model sufficiently sub-divided?

// C3-2. Does domain model reflect architecture decisions?

1. Quality Scenarios

// A4. Quality Scenario Elicitation

// C4-1. Is there sufficient scenario elicitation affecting the architecture?

// C4-2. Is there sufficient review of the quality related to performance?

// C4-3. Is there sufficient review of the quality related to modifiability?

1. Quality Scenario Analysis

// A5. Quality Requirement Specification

// C5-1. Is quality scenario analysis appropriate? (evidence)

1. Candidate Architectures

// A6. Candidate Architecture Design

// C6-1. Are quality analysis and solution candidate appropriate?

// C6-2. Are performance analysis and solution candidate appropriate?

// C6-3. Are modifiability analysis and solution candidate appropriate?

1. Candidate Architecture Evaluation

// A7. Architecture Design

// C7-1. Is comparison analysis of colliding candidates appropriate? (evidence)

// C7-2. Is there sufficient complement of the selected candidate?

1. Final Architecture

// A7. Architecture Design

// C7-3. Is there right integration into the final architecture?

// C7-4. Is there appropriate risk management of the final architecture?

1. Architecture Evaluation(ATAM)

// A10. Architecture Evaluation

// C10-1. Are there sufficient quality scenarios evaluating architecture?

// C10-2. Are there sufficient architectural decisions identified?

// C10-3. Is the analysis of design decisions appropriate? (evidence)

// C10-4. Are the mitigation plans to the risk factors appropriate?