**Use Case Diagrams**

# 目录

Use Case Diagrams ........................ 2

1. Customer Use Cases ........................ 3

2. Service Provider Use Cases ........................ 4

3. Admin Use Cases ........................ 5

4. System-Level Use Cases ........................ 6

*[在Microsoft Word中，您可以右键点击上方目录区域并选择'更新域'来自动生成准确的页码]*

# **Use Case Diagrams**

This document provides a set of Use Case diagrams for the ArtisanConnect platform, structured according to the principles of software engineering. Use cases are illustrated from the perspective of each primary actor—Customer, Service Provider, and Admin—to ensure clarity and detail. The diagrams use the correct UML notation for use case diagrams.

## **1. Customer Use Cases**

This diagram illustrates the primary interactions a **Customer** has with the system, from discovering a service to providing feedback after completion.

![](data:None;base64,)

*mermaid chart*

**Description of Customer Use Cases:**

• **Register Account / Login / Logout**: The customer must be able to create an account, log in to and log out from the system.

• **Reset Password**: The customer can request a password reset if they forget their credentials.

• **Search for Service**: The customer can search for services based on categories (as seen in app/\_components/CategoryList.jsx).

• **Filter Search Results**: This is an extension of the search, allowing users to refine results by criteria like location or rating.

• **View Provider Profile**: The customer can view detailed information about a service provider.

• **View Service Portfolio**: Included in the profile view, this allows the customer to see examples of past work.

• **Select Date & Time**: As part of the booking process, the customer selects a suitable date and time from the provider’s calendar (app/(routes)/details/\_components/BookingSection.jsx).

• **Confirm Booking**: The customer finalizes the booking after selecting a time slot.

• **Make Payment**: This is an integral part of confirming the booking, handled via Stripe integration.

• **Manage Bookings**: The customer can view their booking history (app/(routes)/mybooking/page.jsx).

• **Cancel Booking**: The customer can cancel a pending or confirmed booking.

• **Communicate with Provider**: Customers can chat with providers in real-time.

• **Write Review**: After a service is complete, the customer can rate and review the provider.

• **Manage Notifications**: Customers can control how they receive notifications about their bookings and messages.

## **2. Service Provider Use Cases**

This diagram shows the core functions available to a **Service Provider** for managing their business on the platform.

![](data:None;base64,)

*mermaid chart*

**Description of Service Provider Use Cases:**

• **Register Account / Login / Logout**: The provider must be able to create an account, log in to and log out from the system.

• **Reset Password**: The provider can request a password reset if they forget their credentials.

• **Manage Business Profile**: This is the central use case for providers, allowing them to control how their business is presented. It’s a comprehensive process captured in app/\_components/ProviderRegistrationForm.jsx.

• **Define Services & Pricing**: Included in profile management, where providers detail what they offer.

• **Upload Credentials**: A critical step where providers upload documents like ghana\_card and business\_license for verification.

• **Submit for Approval**: After filling out their profile and uploading documents, providers submit their application for review.

• **Manage Portfolio**: Providers can upload and manage “before and after” photos of their work.

• **Manage Availability**: Providers can set their working hours and block off times, which is reflected in the booking calendar.

• **View & Respond to Bookings**: Providers have a dashboard to see all incoming and confirmed bookings.

• **Communicate with Customer**: Providers can use the chat system to talk to customers.

• **Track Earnings**: A dashboard is available for providers to see their financial performance.

• **Receive Booking Notifications**: Providers get alerts when new bookings are made.

• **Generate Service Quotes**: Providers can create and send quotes to potential customers.

• **View Analytics Dashboard**: Providers can access insights about their business performance.

• **Manage Work Progress**: Providers can update the status of ongoing work and track completion.

## **3. Admin Use Cases**

This diagram outlines the key responsibilities of a platform **Admin**, focusing on oversight, quality control, and platform management.

![](data:None;base64,)

*mermaid chart*

**Description of Admin Use Cases:**

• **Login / Logout**: The admin must be able to securely log in to and log out from the system.

• **Reset Password**: The admin can request a password reset if they forget their credentials.

• **Review Provider Application**: The admin’s core function for onboarding new providers, detailed in app/(routes)/admin/\_components/BusinessApprovalSystem.jsx.

• **View Submitted Documents**: As part of the review, the admin must be able to view the documents uploaded by the provider to verify their legitimacy.

• **Approve/Reject Application**: After reviewing the application, the admin can change the business’s status from PENDING to APPROVED or REJECTED.

• **Manage User Accounts**: The admin has oversight over all users on the platform.

• **Moderate Reviews**: The admin can review and, if necessary, remove reviews that violate platform policies.

• **View Platform Analytics**: The admin has access to a dashboard displaying key metrics about the platform’s health and growth.

• **Configure Platform Settings**: The admin can adjust system-wide settings and parameters.

• **Manage Service Categories**: The admin can add, edit, or remove service categories.

• **Monitor Platform Health**: The admin can view system performance metrics and status.

• **View Revenue Analytics**: The admin can analyze platform revenue and financial performance.

• **Manage Role-Based Access**: The admin can control which features are accessible to different user roles.

• **Handle Disputes**: The admin can mediate and resolve disputes between customers and providers.

## **4. System-Level Use Cases**

This diagram shows system-level use cases that involve multiple actors or automated processes.

![](data:None;base64,)

*mermaid chart*

**Description of System-Level Use Cases:**

• **Email Notifications**: The system sends automated email notifications for various events.

• **Payment Processing**: The system handles secure payment transactions.

• **Role-Based Access Control**: The system enforces access restrictions based on user roles.

• **Document Verification**: The system facilitates the verification of provider credentials.

• **Data Backup**: The system performs regular backups of all critical data.

• **Real-time Chat**: The system enables instant messaging between customers and providers.

• **Search Indexing**: The system maintains search indexes for efficient service discovery.

• **Analytics Generation**: The system collects and processes data for analytics dashboards.