 **East West University**

**Semester:** Spring-2025

**Course Title:** Information System Analysis and Design

**Course Code:** CSE347 **Sec:** 05

# **PoshuPakhi: An E-Commerce Platform for Pet Products in Bangladesh**

**Group Name:** Team Alpha

**Group Members:**

|  |  |
| --- | --- |
| **Student Name** | **Student Id** |
| Sheikh Sarafat Hossain | 2022-3-60-109 |
| Rijia Parveen Raya | 2022-3-60-192 |
| Nipa Mridha | 2022-3-60-254 |
| Jubaiya Akter Shibly | 2022-3-60-255 |

**Submitted to-**

Md Sabbir Hossain

Lecturer

Department of Computer Science & Engineering East West University

**GitHub Link:** <https://github.com/shiekhsarafathossain/PoshuPakhi>

Table Of Contents

[**INTRODUCTION** 4](#_Toc198649337)

[**PURPOSE** 4](#_Toc198649338)

[**INTENDED AUDIENCE** 4](#_Toc198649339)

[**CONCLUSION** 4](#_Toc198649340)

[**INCEPTION OF POSHUPAKHI SYSTEM** 5](#_Toc198649341)

[**INTRODUCTION** 5](#_Toc198649342)

[**PROJECT INCEPTION OVERVIEW** 5](#_Toc198649343)

[**IDENTIFYING THE CLIENT** 5](#_Toc198649344)

[**ICEBREAKING** 6](#_Toc198649345)

[**IDENTIFYING THE STAKEHOLDERS** 6](#_Toc198649346)

[**MULTIPLE VIEWPOINTS OF STAKEHOLDERS** 6](#_Toc198649347)

[**CONCLUSION** 7](#_Toc198649348)

[**ELICITATION OF POSHUPAKHI SYSTEM** 8](#_Toc198649349)

[**COLLABORATIVE REQUIREMENTS GATHERING** 8](#_Toc198649350)

[**QUALITY FUNCTION DEPLOYMENT (QFD)** 8](#_Toc198649351)

[**Usage Scenario:** 10](#_Toc198649352)

[**PoshuPakhi Pet Shop Management System** 10](#_Toc198649353)

[**1) Account Management** 10](#_Toc198649354)

[**2) Ordering** 11](#_Toc198649355)

[**3) Order Tracking** 12](#_Toc198649356)

[**4) Payment** 12](#_Toc198649357)

[**5) Memo Generation** 13](#_Toc198649358)

[**6) Delivery** 13](#_Toc198649359)

[**7) Database Management** 13](#_Toc198649360)

[**9) Administrative Features** 14](#_Toc198649361)

[**Definition of Use Case** 15](#_Toc198649362)

[**Actors in PoshuPakhi System:** 15](#_Toc198649363)

[Use Case diagrams give the non-technical view of overall system. 16](#_Toc198649364)

[**Activity Diagram (Swimlane)** 22](#_Toc198649365)

[**PoshuPakhi Pet Shop Management System** 22](#_Toc198649366)

[**ER Diagram** 24](#_Toc198649367)

[**ER Diagram** 25](#_Toc198649368)

[**CLASS-Diagram** 25](#_Toc198649369)

[**Data Flow Diagram** 26](#_Toc198649370)

[**Sequence Diagram** 29](#_Toc198649371)

[**Component Diagram** 30](#_Toc198649372)

[**Deployment Diagram** 31](#_Toc198649373)

# **INTRODUCTION**

This chapter of this Software Requirements Specification (SRS) is to define the requirements for the project “PoshuPakhi: Online Pet Shop Management System**”.** It outlines the intended audience and provides the foundation for understanding the system’s scope and purpose. It serves as a guide for development, validation, and maintenance of the system.

## **PURPOSE**

This document describes the software requirements for the PoshuPakhi system. It outlines the functional, non-functional, and support requirements, and it serves as a baseline for development. This SRS ensures that all stakeholders—including users, developers, and testers—share a clear and common understanding of the system’s expected behavior. It will evolve as discussions with stakeholders refine and validate the requirements.

## **INTENDED AUDIENCE**

This SRS is intended for:

* **Customers**: Pet owners browsing or purchasing items through the platform.
* **Shop Staff**: Employees managing inventory and preparing deliveries.
* **Administrators**: People responsible for full platform control and data oversight.
* **Project Managers**: Overseeing progress, deliverables, and system scope.
* **Developers**: Translating requirements into working features.
* **Testers**: Validating that requirements are met through rigorous testing.

Each group will use this SRS to ensure the final product meets both technical expectations and customer needs.

# **CONCLUSION**

This analysis of the audience helped us to focus on the users who will be using our analysis. This overall document will help each and every person related to this project to have a better idea about the project.

# **INCEPTION OF POSHUPAKHI SYSTEM**

This section covers the inception phase, where initial research and planning for the PoshuPakhi system were conducted.

# **INTRODUCTION**

**PoshuPakhi** is a fully online pet shop that offers pet food, grooming items, toys, and accessories. It provides a convenient eCommerce experience for pet owners who prefer to shop from home. Customers can browse freely, filter items by preference, place orders, and choose from multiple payment options. The platform also supports staff management and admin oversight for seamless operations.

## **PROJECT INCEPTION OVERVIEW**

The inception stage involved understanding the current challenges faced by pet shop owners and customers and determining how an online system could meet those needs. The main objective was to define core requirements and align expectations between users and developers.

To achieve this, we took the following steps:

* Identified the target clients and stakeholders.
* Conducted icebreaking conversations with pet shop owners and customers.
* Captured various stakeholder viewpoints.
* Defined initial system goals and limitations.

## **IDENTIFYING THE CLIENT**

The primary client is the **admin of PoshuPakhi**, responsible for overall business operations. **Customers** and **staff** are stakeholders who interact with the system for purchasing, managing stock, or fulfilling orders.

## **ICEBREAKING**

Icebreaking was essential in building trust with shop owners and understanding their existing workflows. We visited local pet shops, spoke with staff and customers, and used open-ended conversations to gather feedback about pain points, expectations, and desired features.

## **IDENTIFYING THE STAKEHOLDERS**

The main stakeholders include:

* **Customers** (pet owners)
* **Shop Staff**
* **Admins**
* **Delivery Partners** (indirectly impacted)

## **MULTIPLE VIEWPOINTS OF STAKEHOLDERS**

Different stakeholders have varying expectations:

**Customer Viewpoint**

* Easy-to-use online platform.
* Ability to view product details before logging in.
* Fast, secure checkout with mobile banking or cash on delivery.
* Real-time delivery tracking.
* Personalized product recommendations.
* Easy access to order history.

**Admin Viewpoint**

* Full control over product listings.
* Alerts when stock is low.
* Real-time sales and inventory reports.
* Supplier tracking and invoice generation.
* Secure transaction records.
* Insightful analytics to track business growth.

**Staff Viewpoint**

* Clear dashboard to view and process incoming orders.
* Inventory adjustment and shipment coordination.
* Easy status updates from packing to dispatch.

# **CONCLUSION**

The primary goal of this project is to model and design a software for the pet owners and ease order system and For admins to easily manage multiple orders and reservations. For these reasons, The software will be as simple as a customers can easily be able to use this and the managers can maintain it without any annoyance. The software will be designed in such a way as it takes very little time to manage. To make this software project successful, collaboration with stakeholders was a main priority that what they want, how the software will work, how it can be more convenient, how it will save time and energy, etc.

# **ELICITATION OF POSHUPAKHI SYSTEM**

After inception, we proceeded to elicit detailed system requirements. This process combined communication, analysis, and negotiation with stakeholders.

## **COLLABORATIVE REQUIREMENTS GATHERING**

We conducted multiple sessions with pet shop owners, staff, and customers to gather diverse insights. These meetings helped us refine both essential and advanced system features.

## **QUALITY FUNCTION DEPLOYMENT (QFD)**

QFD was used to translate user expectations into technical specifications. By applying QFD, we identified three categories of requirements:

**Normal Requirements**

* Users can register and log in to their accounts.
* Guest browsing without login.
* Add items to a shopping cart.
* Secure login verification and password recovery.
* Order confirmation memo generation.
* Real-time inventory updates post-purchase.
* Payment through mobile banking, cards, or cash on delivery.
* Staff dashboard to process and update orders.
* Admin dashboard for full control of products and reports.
* Notifications for low stock.
* Delivery tracking via courier integration.

**Expected Requirements**

These requirements are intrinsic to the product or system and may be so elementary that the customer does not explicitly state them. Their absence will be a cause for significant dissatisfaction. Below the expected requirements for our project are briefly described: -

* Responsive interface.
* Accurate delivery time estimates.
* Ability to view and reorder past purchases.
* User profile management.
* Customer support through the platform.
* Data backup and redundancy.
* Real-time reporting and analytics.

**Exciting Requirements**

These requirements are for features that go beyond the customer's expectations and prove to be very satisfying when present. Following are some exciting requirements of our project:

* Personalized product suggestions based on order history.
* Promotion of frequently bought items on user homepage.
* Admin alerts for top-selling products.
* Loyalty discounts or reward system.
* Smart restock suggestions based on purchase patterns.

# **Usage Scenario:**

**PoshuPakhi Pet Shop Management System**

**PoshuPakhi** is a fully online eCommerce platform that simplifies the pet product shopping experience for pet owners. This system supports both the end users (customers) and the back-end operators (staff and administrators) by providing seamless functionality such as browsing, ordering, payment processing, inventory tracking, and personalized customer experience.

**1) Account Management**

**1.1) Create Account**

**User Perspective**:  
A user can browse the website without logging in. However, to place an order, they must register an account by providing:

* Full Name
* Mobile Number
* Email Address
* Password
* Shipping Address

After submitting the details, the user will receive an OTP on their mobile number or email. Entering this OTP completes the registration process.

**Admin Perspective**:  
Admins are assigned predefined credentials to access the admin dashboard. They do not need to register manually.

**1.2) Verification**

User credentials are verified via OTP before activation to ensure authenticity and prevent duplicate accounts.

**1.3) Update Account**

Users can update their account information, including:

* Mobile Number
* Email Address
* Password
* Shipping Address

**1.4) Password Recovery**

Users can recover forgotten passwords using:

* **Email**: A reset link is sent to their registered email.
* **Mobile**: An OTP is sent for verification and password reset.

**1.5) Login**

Users can log in using either their registered email or phone number along with the password. Admins log in with system-provided credentials.

**2) Ordering**

Only logged-in users can place orders. Customers can browse the product catalog, which includes pet food, grooming items, toys, and accessories. Each item includes price, stock status, description, and user reviews.

* Users add items to a virtual cart.
* Quantity can be modified before checkout.
* The system calculates the total cost and estimated delivery time.
* Payment options include:
  + Mobile Banking (bKash, Nagad)
  + Credit/Debit Card
  + Cash on Delivery (COD)

After successful payment or COD confirmation, the order is processed, and a confirmation memo is generated. Inventory is updated in real-time.

**3) Order Tracking**

Once an order is confirmed:

* A tracking ID is generated.
* Users can view real-time delivery progress through third-party logistics integration.
* Updates are sent via SMS and email.

**4) Payment**

**Online Payment:**

Integrated via **SSLCommerz**, which supports secure transactions using mobile banking, cards, and other gateways. Every transaction is logged in the admin database automatically.

**Cash on Delivery (COD):**

Users can opt to pay upon receiving the product. A small service charge may apply.

**Refund Policy:**

If an order is canceled before shipping, a full refund is processed within 3-4 business hours.

**5) Memo Generation**

For each confirmed order, a digital memo is generated containing:

* Order ID
* Product Details
* Quantity
* Total Price
* Estimated Delivery Time

Printed copies may be provided with the package upon delivery.

**6) Delivery**

**User Perspective:**

Users receive delivery updates through SMS/email. They can track orders and receive packages at their provided shipping address.

**Admin Perspective:**

Admins assign orders to delivery partners and update delivery status through the dashboard. The system uses integrated courier APIs for location tracking and confirmation.

**7) Database Management**

**Admin Database:**

* Stores product inventory, supplier details, sales reports, and all user transactions.
* Alerts are triggered for low stock.
* Includes delivery logs and payment records.

**User Database:**

* Contains personal information, past orders, and transaction history.
* Updates made by users are mirrored in the admin database to maintain consistency.

**8) Additional Features**

* **Personalized Suggestions**: Frequently purchased or browsed items are shown on the homepage.
* **Top Products Display**: Based on user behavior and admin analytics.
* **Live Stock Alerts**: Items marked as "Out of Stock" are hidden from order options.
* **Responsive Design**: Platform works on mobile and desktop.
* **Customer Support Chat**: Users can contact support directly from the website.

**9) Administrative Features**

**9.1) Product Management:**

Admins can:

* Add/edit/delete product listings
* Set prices, stock quantities, and categories
* View top-selling products and low-stock items

**9.2) Inventory Management:**

* Daily updates on stock levels
* Real-time deduction on confirmed orders
* Notification system for reordering from suppliers

# 

# **Definition of Use Case**

A Use Case captures a contract that describes the system behavior under various conditions as the system responds to a request from one of its stakeholders. In essence, a Use Case tells a stylized story about how an end user interacts with the system under a specific set of circumstances. A Use Case diagram simply describes a story using corresponding actors who perform important roles in the story and makes the story understandable for the users. The first step in writing a Use Case is to define that set of “actors” that will be involved in the story. Actors are the different people that use the system or product within the context of the function and behavior that is to be described. Actors represent the roles that people play as the system operators. Every user has one or more goals when using the system. Before building the diagram or detailing the use cases, the key actors of the system must be identified.

## **Actors in PoshuPakhi System:**

**Primary Actors**

Primary actors are the main users who directly interact with the PoshuPakhi platform to fulfill their goals:

* **Customer**:
  + Browses products, creates an account, places orders, tracks deliveries, and makes payments.
* **Admin**:
  + Manages product listings, tracks inventory, monitors sales reports, and oversees system operations.
* **Staff**:
  + Processes orders, updates inventory, handles packaging, and coordinates with delivery partners.

**Secondary Actors**

Secondary actors support the operations by providing external services or data but do not interact with the system directly in the same way as primary actors:

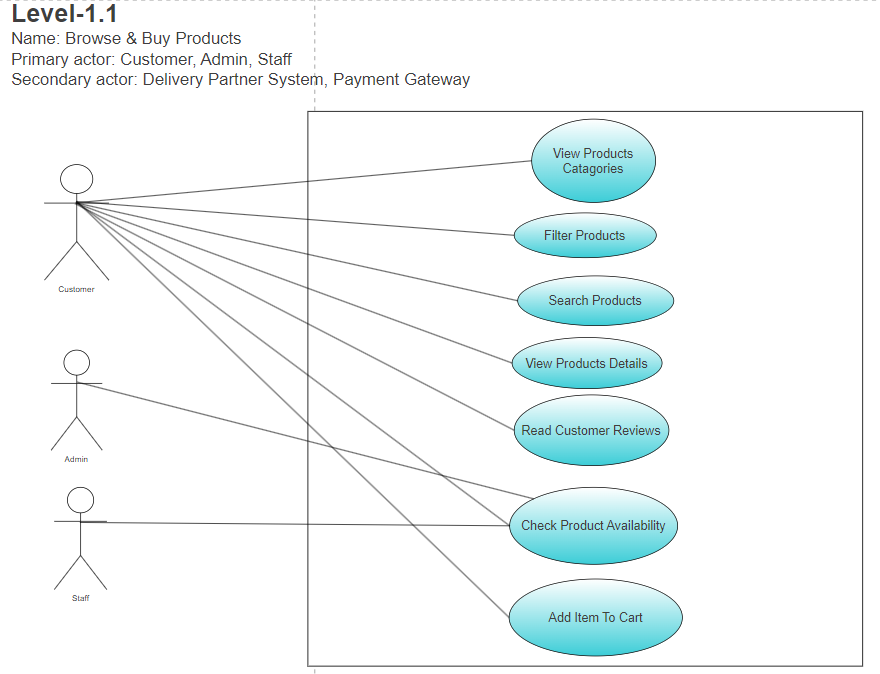
* **Delivery Partner System**:
  + Facilitates order delivery and updates tracking information. Integrated with the system to provide real-time delivery status to users.
* **Supplier System**:
  + Supplies product restocks to the admin; used for managing purchase orders and updating inventory availability.
* **Payment Gateway (e.g., Bkash/ Nagad)**:
  + Handles secure transactions, sends payment confirmation to users, and updates the admin database with transaction logs.

## Use Case diagrams give the non-technical view of overall system.

A diagram of a person's relationship

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A diagram of a network

AI-generated content may be incorrect.

A diagram of a product

AI-generated content may be incorrect.

A diagram of a payment method

AI-generated content may be incorrect.

A diagram of a delivery system

AI-generated content may be incorrect.

A diagram of a company

AI-generated content may be incorrect.

A diagram of a delivery order

AI-generated content may be incorrect.A diagram of a delivery system

AI-generated content may be incorrect.

# **Activity Diagram (Swimlane)**

**PoshuPakhi Pet Shop Management System**

**Definition of Activity Diagram**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency

**Swimlane Diagram Definition :**

A swimlane diagram is a type of flowchart that delineates who does what in a process. Using the metaphor of lanes in a pool, a swimlane diagram provides clarity and accountability by placing process steps within the horizontal or vertical “swimlanes” of a particular employee, work group or department. It shows connections, communication and handoffs between these lanes, and it can serve to highlight waste, redundancy and inefficiency in a process.

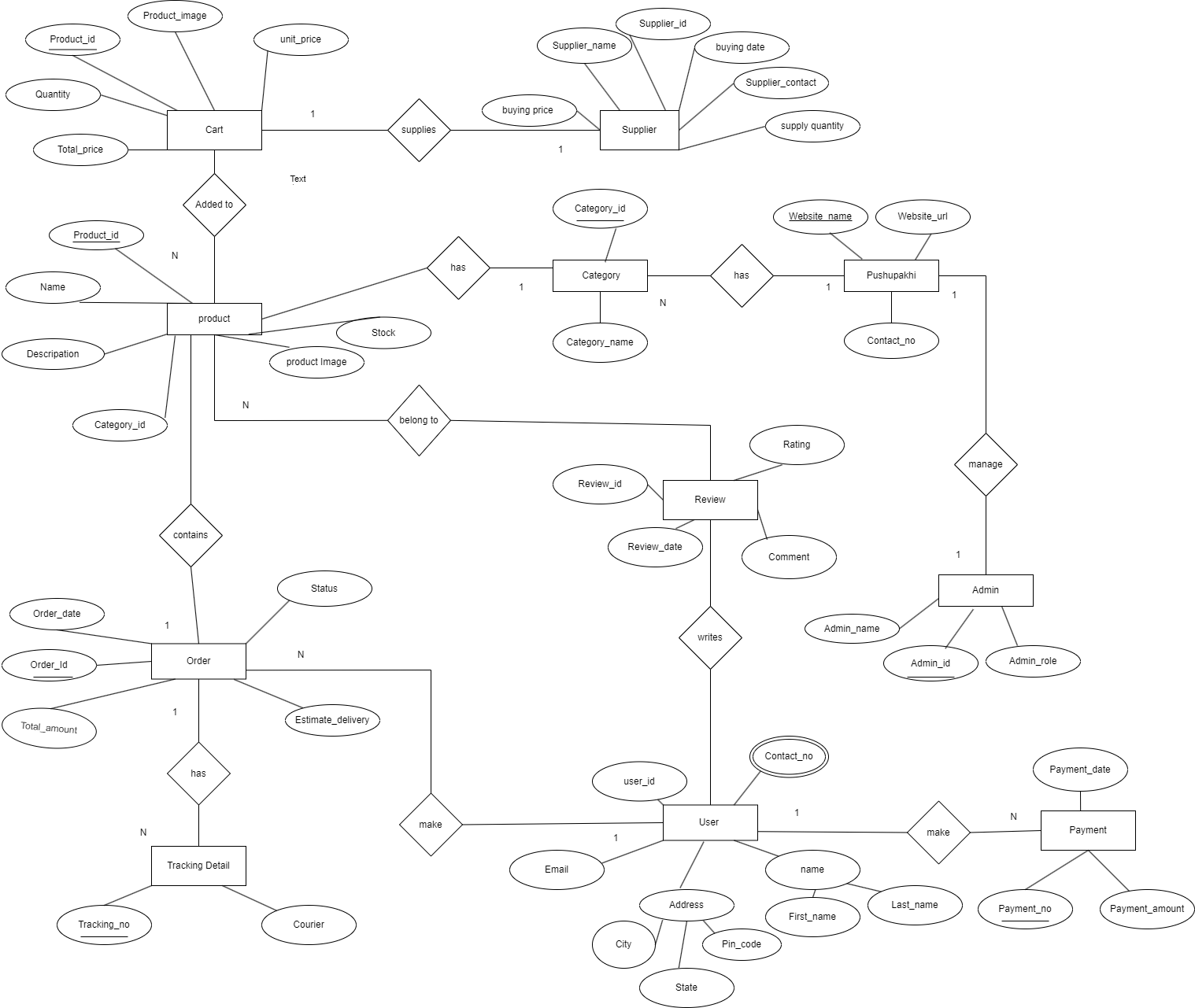
A screenshot of a computer screen

AI-generated content may be incorrect.

# **ER Diagram**

**Definition of ER Diagram**

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities”

such as people, objects or concepts relate to each other within a system. within a system. **ER Diagram**

# **CLASS-Diagram**

**Class Diagram concept:**

Class-based modeling represents the objects that the system will manipulate, the

operations that will be applied to the objects, relationships between the objects and the

collaborations that occur between the classes that are defined.

**A black background with white squares

AI-generated content may be incorrect.Class Diagram**

# **Data Flow Diagram**

**Definition of DFD Diagram:**

A data flow diagram (DFD) maps out the flow of information for any process or system.

A data flow diagram can dive into progressively more detail by using levels and layers, zeroing in on a particular piece.  DFD levels are numbered 0, 1 or 2.

* DFD Level 0 is also called a Context Diagram. It’s a basic overview of the whole system or process being analyzed or modeled.
* DFD Level 1 provides a more detailed breakout of pieces of the Context Level Diagram. It highlights and breaks down the high-level process of the Context Diagram into its subprocesses.
* DFD Level 2 then goes one step deeper into parts of Level 1 to reach the necessary level of detail about the system’s functioning.

**Data Flow Diagram:**

**A diagram of a blue circle with black text

AI-generated content may be incorrect.**

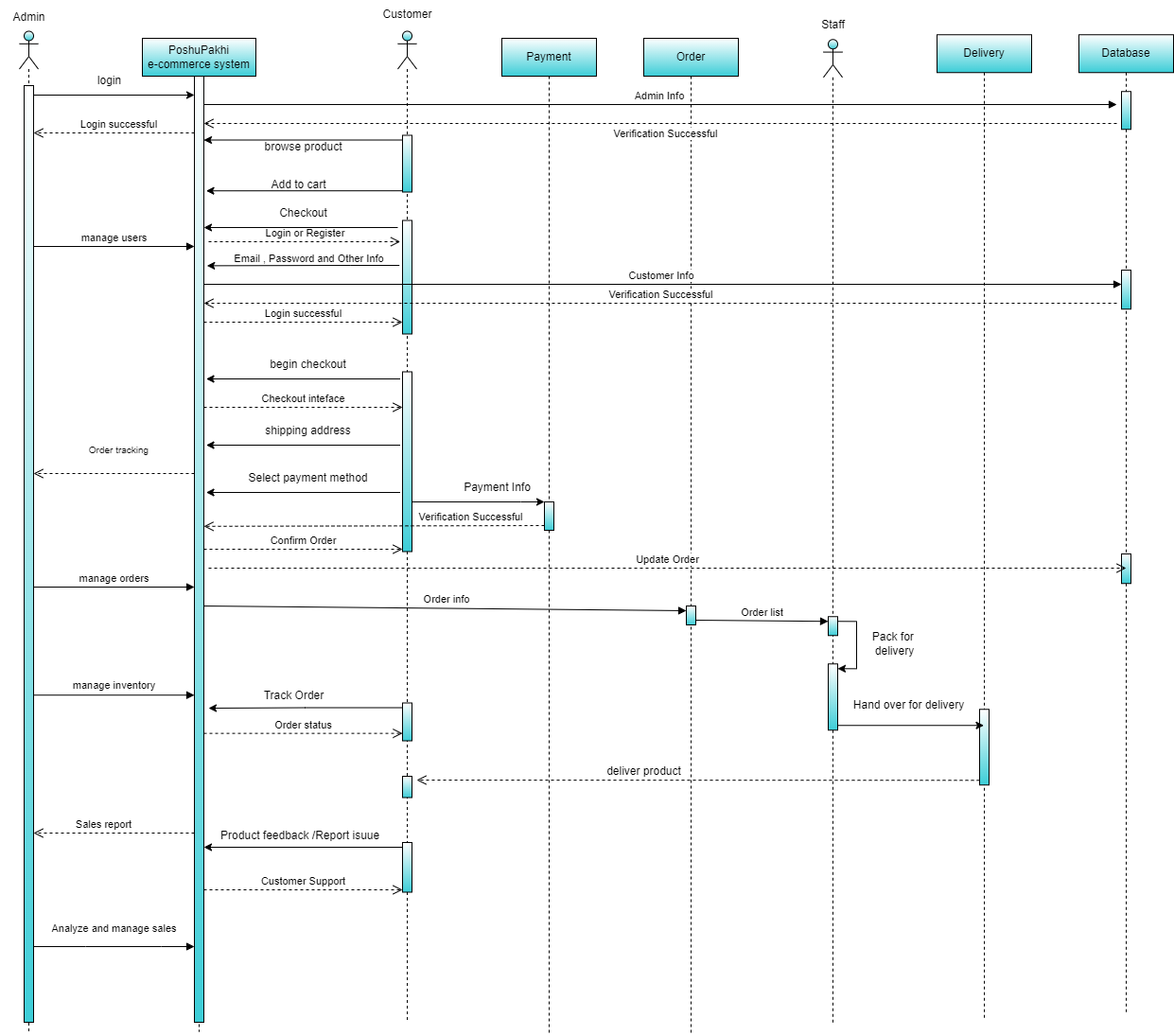
**A diagram of a software system

AI-generated content may be incorrect.A diagram of a customer

AI-generated content may be incorrect.**

# **Sequence Diagram**

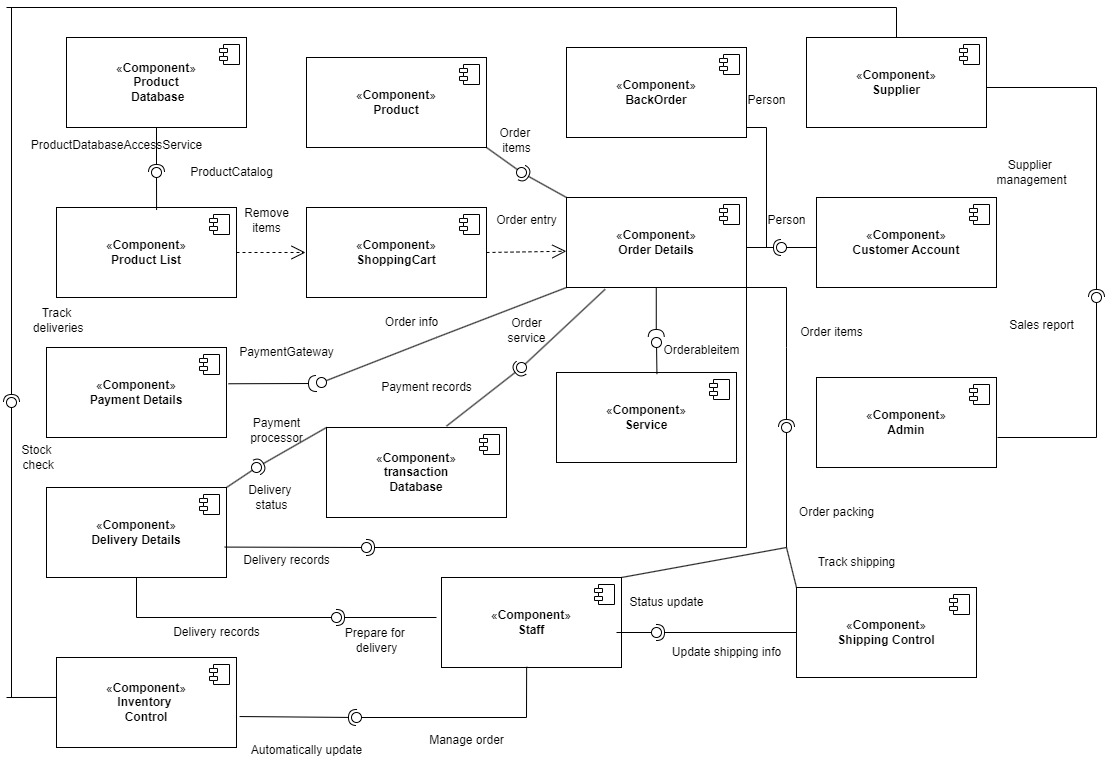
**Definition of Sequence Diagram:**

****A sequence diagram is a visual representation of the interactions between objects in a system over time. It focuses on the order in which messages are exchanged between objects, showing how different parts of a system interact during a specific scenario.

# **Component Diagram**

**Definition of Component Diagram:**

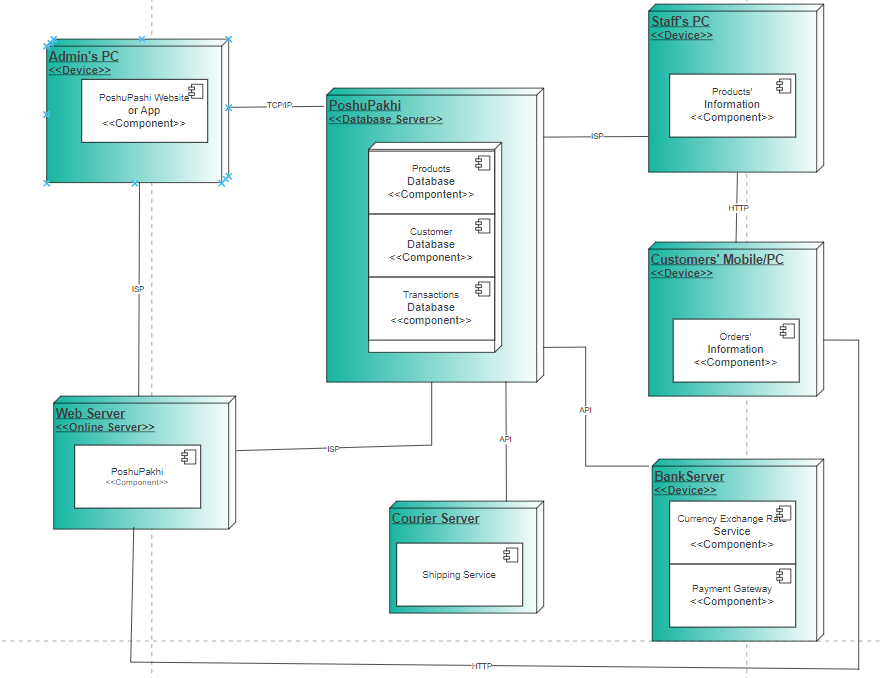
A component diagram is a type of structural diagram that visualizes the physical organization and relationships of components within a software system. It focuses on how components are assembled to form larger systems, highlighting their dependencies and interactions.

**Component Diagram:**

# **Deployment Diagram**

**Definition of Deployment Diagram:**

A deployment diagram is a visual representation of the physical structure of a system, showing how software components (artifacts) are allocated to hardware components (nodes). It depicts the static deployment view of the system architecture, outlining how the system is physically arranged for execution, operation, and scaling.

**Deployment Diagram:**