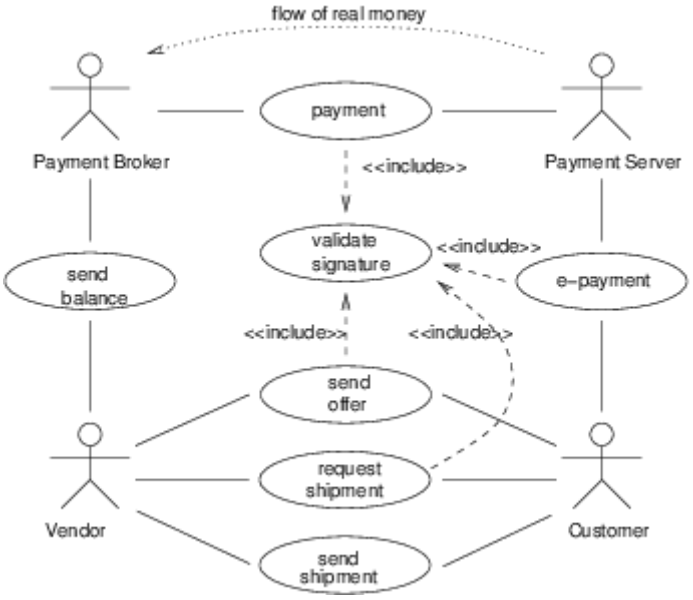
Case studies for Web interface Design

Payment Management System

UML Use Case Diagram

A payment management system is a software solution designed to streamline and automate the process of receiving and processing payments from customers or clients.

An electronic payment system, or e-payment system, is a digital method of conducting financial transactions over the internet. It facilitates the exchange of money or value between parties electronically, eliminating the need for physical currency or checks. E-payment systems have revolutionized the way businesses and consumers transact, offering convenience, security, and efficiency.



Payment Server is a specialized computer or software application that handles payment transactions between a customer and a merchant. It securely processes payment requests, verifies the authenticity of transactions, and communicates with payment gateways or processors to authorize and facilitate payments.

A payment broker is an intermediary entity that facilitates payment transactions between parties by connecting them to different payment networks or methods. Payment brokers often provide services such as aggregating multiple payment options, managing currency conversions, and ensuring compliance with regulatory requirements.

Payment refers to the transfer of money or value from one party (the payer) to another (the payee) in exchange for goods, services, or as a settlement of a debt. Payments can be made using various methods, including cash, checks, credit/debit cards, bank transfers, and digital wallets.

Sending balance refers to transferring funds or credit from one account or payment instrument to another. It could involve moving money between bank accounts, transferring funds from a digital wallet to a bank account, or allocating credit from a prepaid card to another account.

Checking balance involves verifying the amount of funds or credit available in an account or payment instrument. This can be done through various channels, such as online banking portals, mobile apps, ATMs, or by contacting the financial institution directly.

Sending shipment involves initiating the process of delivering goods or products to a recipient or customer. This typically involves arranging for transportation, packaging the items, and providing necessary documentation for shipping.

Requesting shipment involves soliciting or arranging for the delivery of goods from a supplier or vendor to a specified location. This may include specifying shipping preferences, delivery timelines, and any special instructions for handling the shipment.

Send Offer: Sending an offer involves presenting a proposal or invitation to engage in a transaction, such as purchasing goods or services, to a potential buyer or customer. Offers may include pricing details, product descriptions, discounts, or promotional incentives to encourage acceptance.

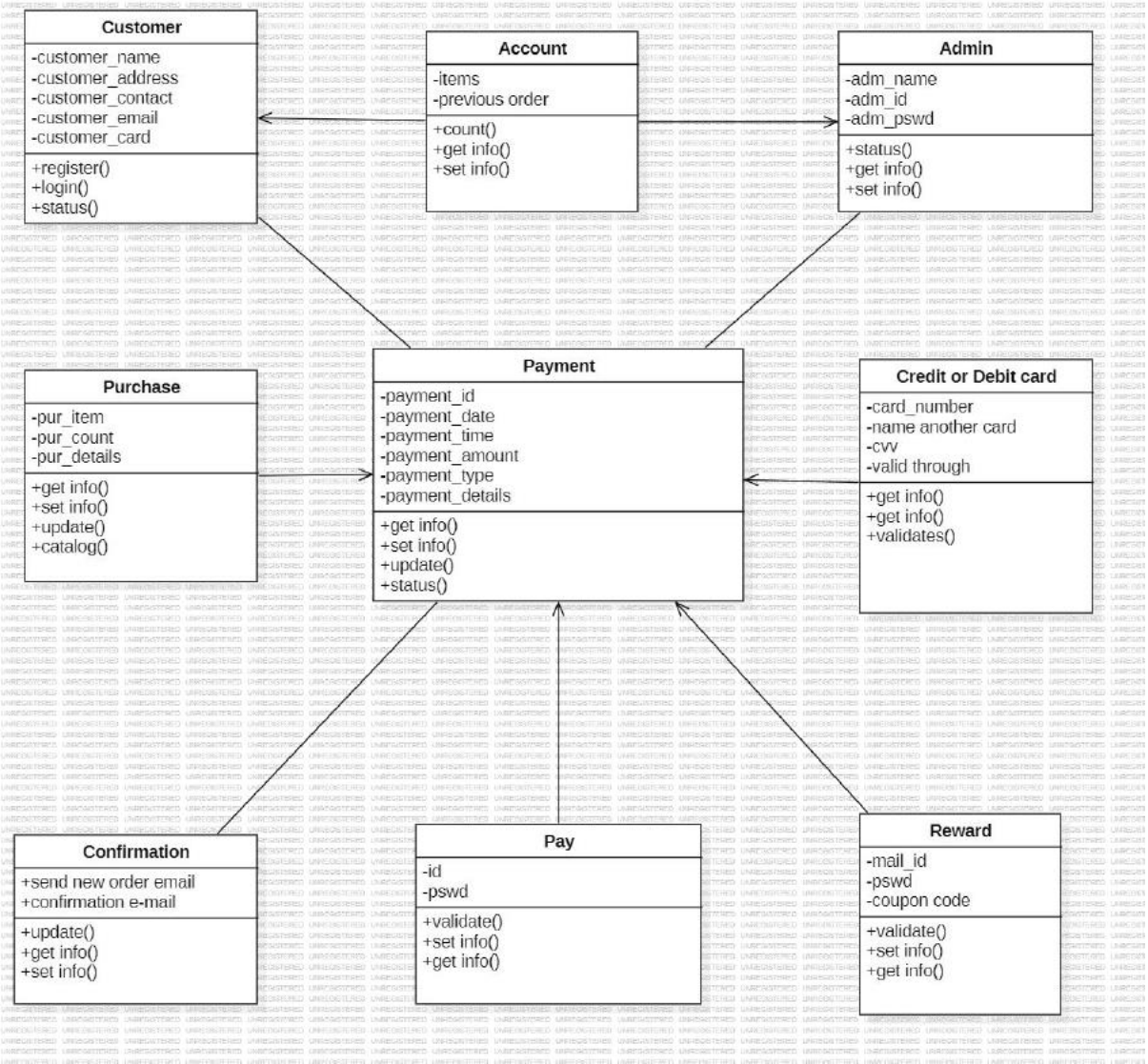
Valid Signature: A valid signature is a legally recognized mark or endorsement made by an individual to authenticate a document, agreement, or transaction. In the context of electronic transactions, a valid signature may take the form of a digital signature or an electronic mark that meets specific legal requirements for authentication and non-repudiation.

A vendor is a seller or supplier of goods or services, typically in a business-to-business (B2B) or business-to-consumer (B2C) context. Vendors offer products or services for sale, negotiate terms and pricing with buyers, and fulfill orders or contracts.

A customer is an individual or entity that purchases goods or services from a seller or provider in exchange for payment. Customers may be individuals, businesses, or organizations that engage in transactions to fulfill their needs or requirements. Building and maintaining positive relationships with customers is essential for business success and growth.

UML Class Diagram

In this e-payment system, each customer is assigned a unique identifier and is associated with precisely one account. The account is responsible for managing the customer's shopping cart and orders. Customers have the option to register as web users, granting them access to make purchases online. However, being a web user is not mandatory, as purchases can also be facilitated through phone orders or catalog purchases.



Customer: Represents an individual or entity engaging in transactions within the system. A customer is associated with personal information such as name, contact details, and payment preferences.

Account: Represents the account holder within the system. It may contain information such as account ID, balance, transaction history, and linked payment methods. An account is typically associated with one or more customers.

Admin: Represents an administrator or system manager who oversees and manages the system. Admins have privileges to monitor transactions, manage user accounts, and perform administrative tasks within the system.

Purchase: Represents a transaction initiated by a customer to buy goods or services. A purchase may include details such as item(s) purchased, quantity, price, and timestamp.

Payment: Represents the process of transferring funds from a customer's account to the merchant's account to complete a purchase. Payment may involve various methods such as credit/debit cards, digital wallets, bank transfers, or cash.

Credit or Debit Card: Represents a payment instrument used by customers to make purchases. It contains information such as card number, expiration date, cardholder name, and security code (CVV). Credit cards allow customers to borrow funds for purchases, while debit cards deduct funds directly from the customer's bank account.

Confirmation: Represents the acknowledgment or verification of a successful transaction. Confirmation may include details such as transaction ID, payment status, and delivery/shipping information. It serves as proof of the completed transaction for both the customer and the merchant.

Pay: Represents the action of initiating a payment transaction to complete a purchase. It involves verifying the payment details, authorizing the transaction, and updating the respective accounts' balances.

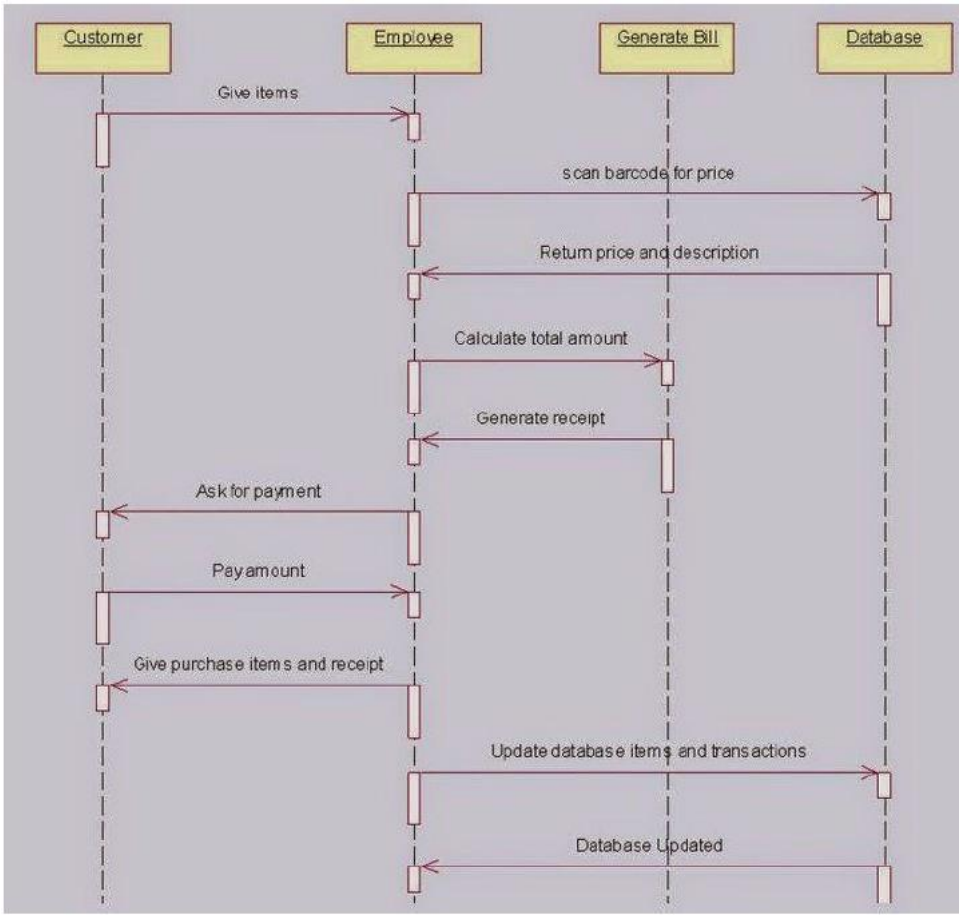
Reward: Represents incentives or benefits offered to customers as a reward for their loyalty or frequent purchases. Rewards may include discounts, cashback, loyalty points, or other promotional offers. They encourage repeat business and customer retention.

UML Sequence Diagram

UML sequence diagrams model the flow of logic within the system in a visual manner, enabling the user both to document and validate the logic, and are commonly used for both analysis and design purposes. Sequence diagrams are the most popular UML artifact for dynamic modelling, which focuses on identifying the behaviour within your system. Sequence diagrams, along with class diagrams and physical data models, are the most important design-level models for modern application development.

Sequence diagrams are typically used to model:

1. **Usage scenarios**. A usage scenario is a description of a potential way the system is used. The logic of a usage scenario may be part of a use case, perhaps an alternate course. It may also be an entire pass through a use case, such as the logic described by the basic course of action or a portion of the basic course of action, plus one or more alternate scenarios. The logic of a usage scenario may also be passed through the logic contained in several use cases. For example, a student enrolled in university, and then immediately enrolled in three seminars.
2. **The logic of methods**. Sequence diagrams can be used to explore the logic of a complex operation, function, or procedure. One way to think of sequence diagrams, particularly highly detailed diagrams, is as visual object code.
3. **The logic of services**. A service is effectively a high-level method, often one that can be invoked by a wide variety of clients. This includes web-services as well as business transactions implemented by a variety of technologies such as CICS/COBOLor CORBA-compliant object request brokers (ORBs).



UML Activity Diagram

UML activity diagrams are typically used for business process modelling, for modelling the logic captured by a single use case or usage scenario, or for modelling the detailed logic of a business rule. Although UML activity diagrams could potentially model the internal logic of a complex operation it would be far better to simply rewrite the operation so that it is simple enough that you don‘t require an activity diagram. In many ways, UML activity diagrams are the object-oriented equivalent of flow charts and data flow diagrams (DFDs) from structured development.

Let‘s start by describing the basic notation :

● **Initial node**. The filled-in circle is the starting point of the diagram. An initial node isn‘t required although it does make it significantly easier to read the diagram.

● **Activity final node**. The filled circle with a border is the ending point. An activity diagram can have zero or more activity final nodes.

● **Activity**. The rounded rectangles represent activities that occur. An activity may be physical, such as Inspect Forms, or electronic, such as Display Create Student Screen.

● **Flow/edge**. The arrows on the diagram. Although there is a subtle difference between flows and edges , never a practical purpose for the difference .

● **Fork**. A black bar with one flow going into it and several leaving it. This denotes the beginning of parallel activity.

● **Join**. A black bar with several flows entering it and one leaving it. All flows going into the join must reach it before processing may continue. This denotes the end of parallel processing.

Condition. Text such as [Incorrect Form] on a flow, defines a guard which must be evaluate to true in order to traverse the node.

● **Decision**. A diamond with one flow entering and several leaving. The flows leaving include conditions, although some modelers will not indicate the conditions if it is obvious.

● **Merge**. A diamond with several flows entering and one leaving. The implication is that one or more incoming flows must reach this point until processing continues, based on any guards on the outgoing flow.

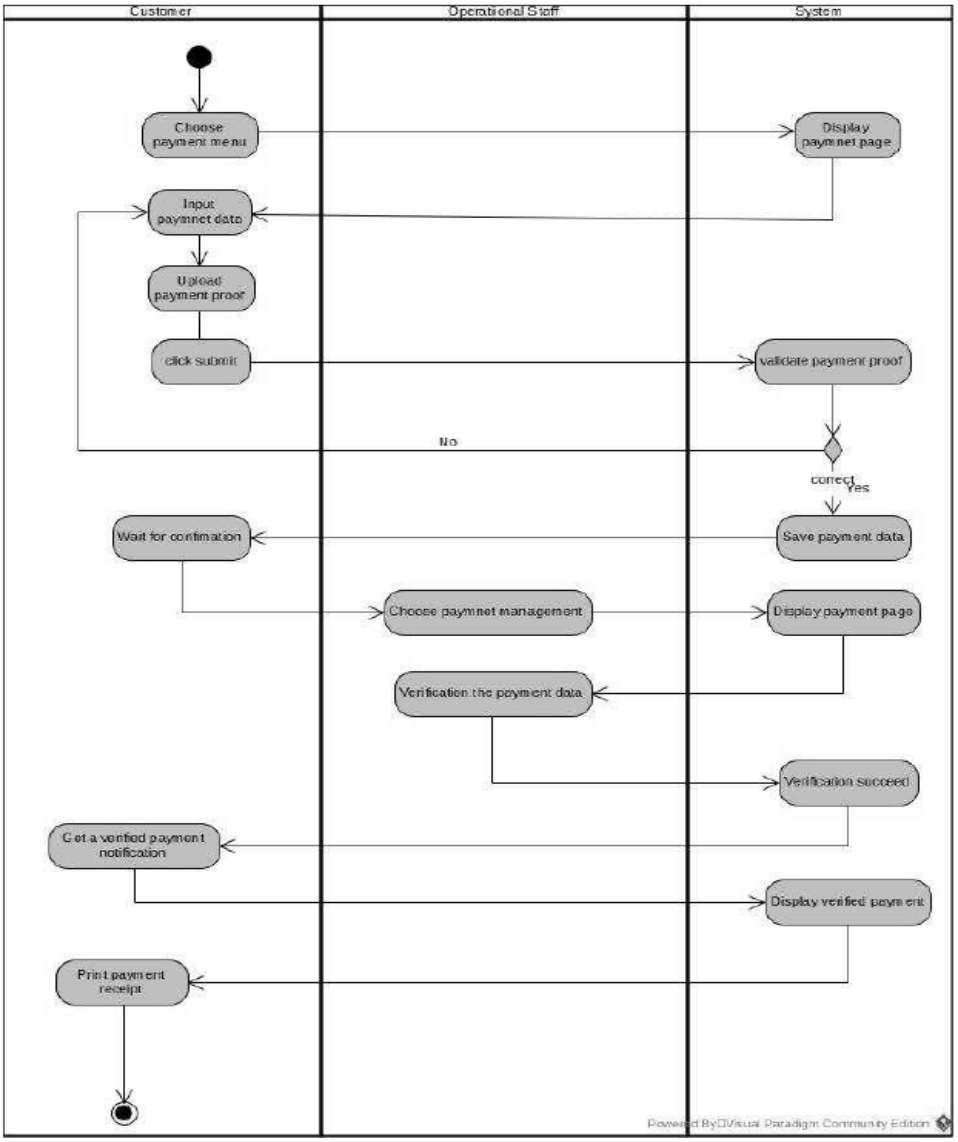
● **Partition**. If figure is organized into three partitions, it is also called swim lanes, indicating who/what is performing the activities (either the Applicant, Registrar, or System).

● **Sub-activity indicator**. The rake in the bottom corner of an activity, such as in the Apply to University activity, indicates that the activity is described by a more finely detailed activity diagram.

● **Flow final**. The circle with the X through it. This indicates that the process stops at this point.

Guidelines associated with drawing an Activity diagram:

1.General Guidelines
2.Activities
3.Decision Points
4.Guards
5.Parallel Activities
6.Swimlane Guidelines
7.Action-Object Guidelines


Customer
Operational Staff
System
Choose payment menu
Display paymnet page
Input paymnet data
Upload payment proof
click submit
validate payment proof
No
correct
Yes
Wait for confimation
Save payment data
Choose paymnet management
Display payment page
Verification the payment data
Verification succeed
Get a verified payment notification
Display verified payment
Print payment receipt
Powered By Visual Paradigm Community Edition