



Understanding Our E-Commerce System Architecture

A Visual Guide to Our Application's Data and Process Flow

Authors: Mahdi Mohammadzadeh, Ali Ghasempour, Ali Arabi



The Evolution of System Diagrams

- Early Beginnings:
- Diagrams have been used since ancient times, with early applications in architectural plans and engineering blueprints.

Leonardo da Vinci: Renaissance, Innovation, Visualization

- Notable for his detailed anatomical and machine diagrams in the 15th century, showcasing the intersection of art, science, and technology.





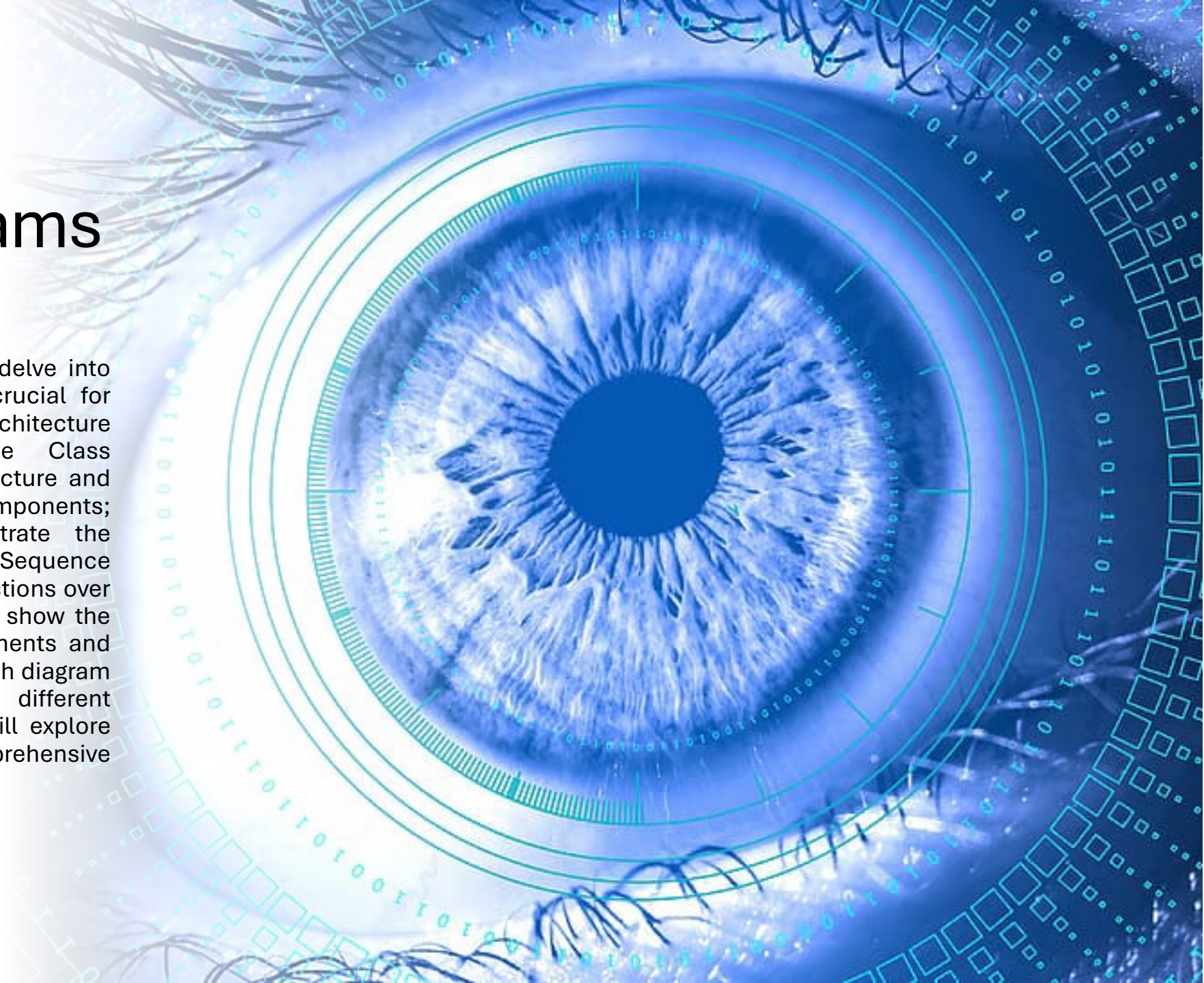
THREE AMIGOS

- These three individuals, often referred to as the "Three Amigos," are primarily responsible for the development of the Unified Modeling Language (UML). Their work in the 1990s combined elements from their respective methodologies:
- **Grady Booch** developed the Booch method, which provided techniques for object-oriented analysis and design.
- **Ivar Jacobson** introduced Objectory, which is the first commercially available software engineering process, emphasizing use case driven development.
- **James Rumbaugh** led the creation of the Object Modeling Technique (OMT), which offered a rich set of object-oriented modeling concepts and was particularly strong in object and dynamic modeling.



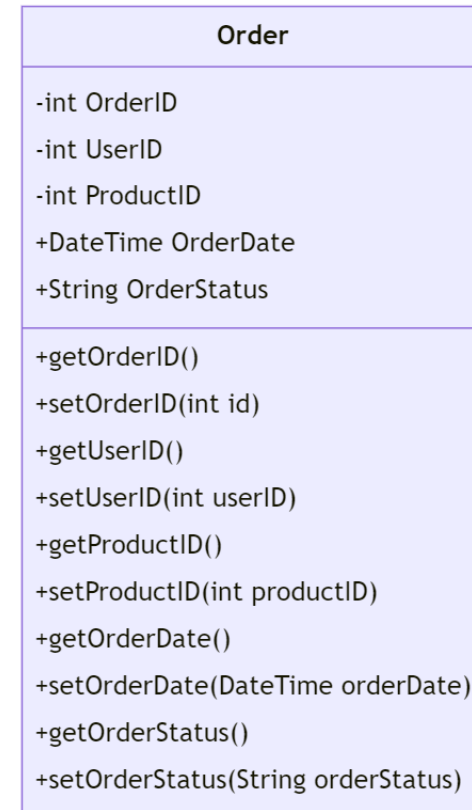
Overview of System Diagrams

- In the upcoming slides, we will delve into several key diagrams that are crucial for understanding our system's architecture and behavior. These include Class Diagrams, which outline the structure and relationships of our system components; Activity Diagrams, which illustrate the processes and workflows; Sequence Diagrams, which detail the interactions over time; and State Diagrams, which show the various states of system components and the transitions between them. Each diagram provides valuable insights into different facets of our system, and we will explore them in detail to provide a comprehensive overview.



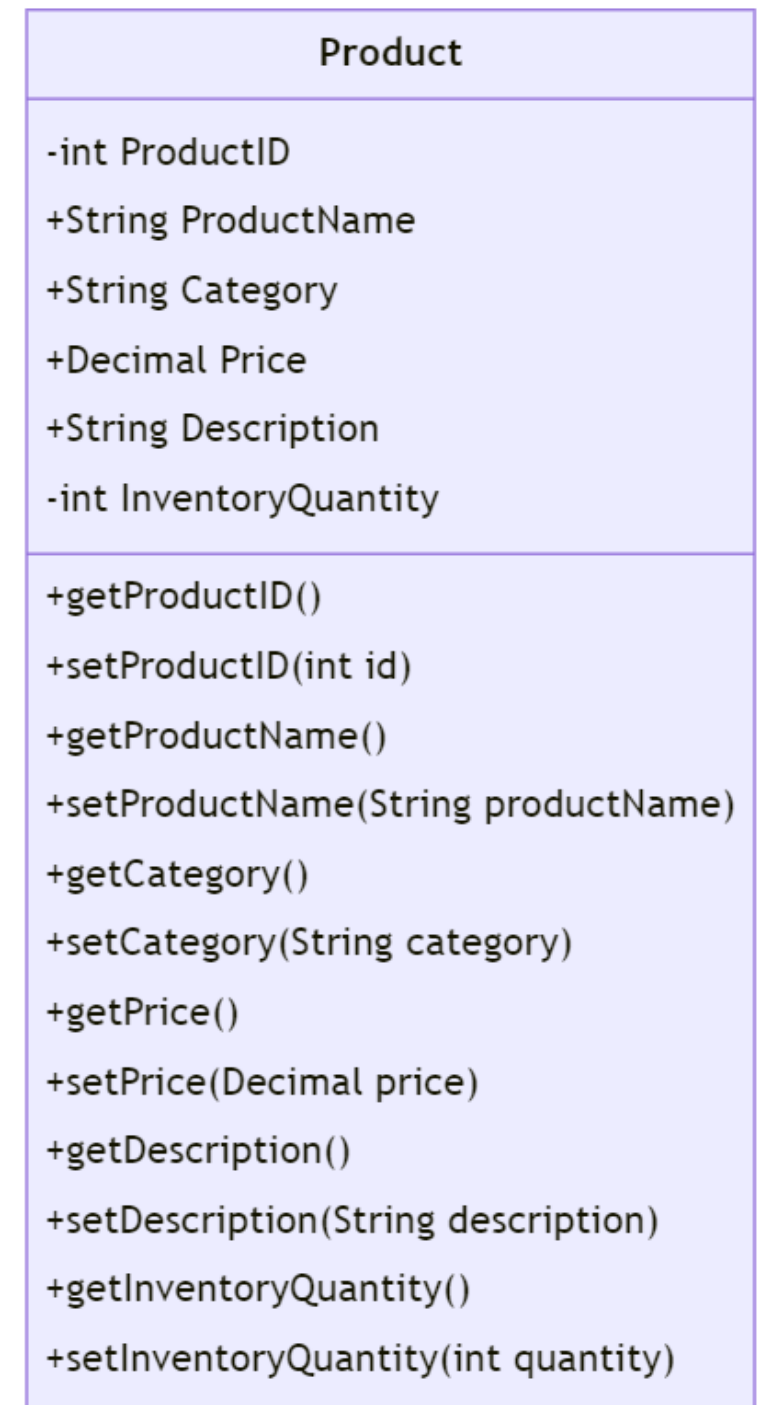
Order Class Diagram

- The provided UML class diagram represents an `Order` class, detailing its attributes and methods. The class has five main attributes: `OrderID`, `UserID`, and `ProductID`, all of which are integers, along with `OrderDate`, a DateTime object, and `OrderStatus`, a string. These attributes capture essential information about an order, including identifiers for the order, user, and product, as well as the date of the order and its current status. The class includes getter and setter methods for each attribute, allowing for the retrieval and modification of order details. This structure ensures encapsulation and provides a clear interface for interacting with `Order` objects in a system.
-



Product Class Diagram

- This UML class diagram depicts a `Product` class, detailing its attributes and methods. The class includes attributes such as `ProductID`, `ProductName`, `Category`, `Price`, `Description`, and `InventoryQuantity`, along with corresponding getter and setter methods for each attribute. This diagram provides a clear and structured representation of the `Product` class, highlighting its key properties and functionalities in an object-oriented design.





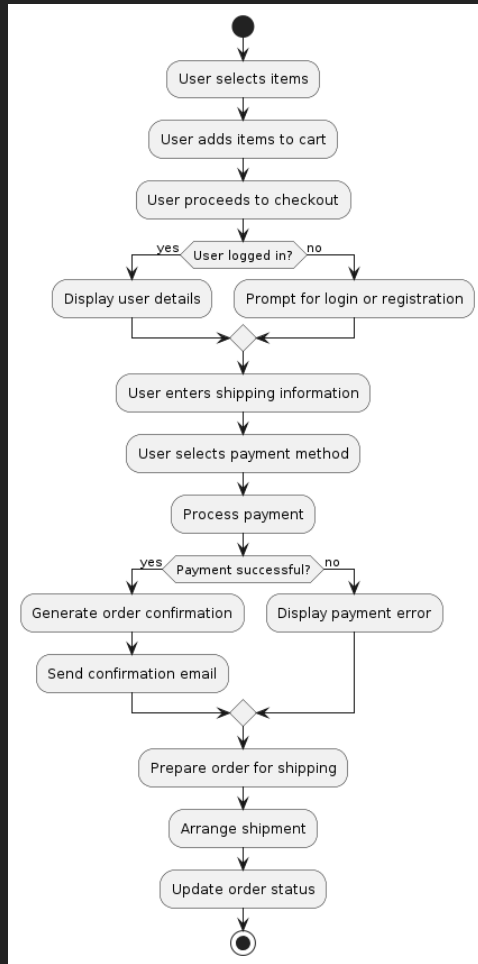
User Class Diagram

- Our User Class diagram outlines a `User` class featuring both public and private attributes. Public attributes include `Username` and `Email`, while private attributes cover `UserID`, `Password`, `Address`, and `PhoneNumber`. The class also provides public getter and setter methods for each attribute, ensuring proper encapsulation and access control.

User
-int UserID
+String Username
+String Email
-String Password
-String Address
-String PhoneNumber
+getUserID()
+setUserID(int id)
+getUsername()
+setUsername(String username)
+getEmail()
+setEmail(String email)
+getPassword()
+setPassword(String password)
+getAddress()
+setAddress(String address)
+getPhoneNumber()
+setPhoneNumber(String phoneNumber)



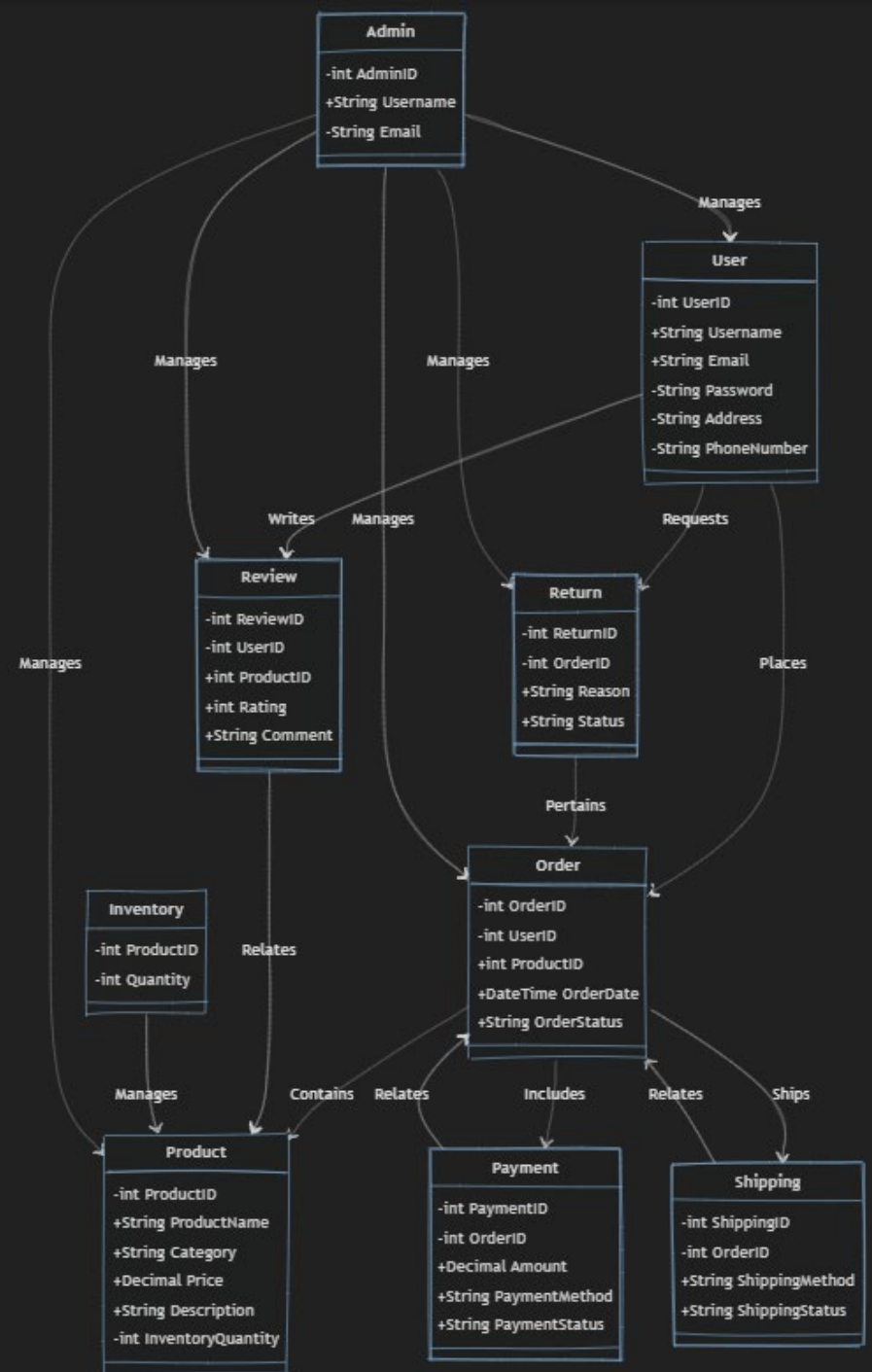
Activity diagram order process



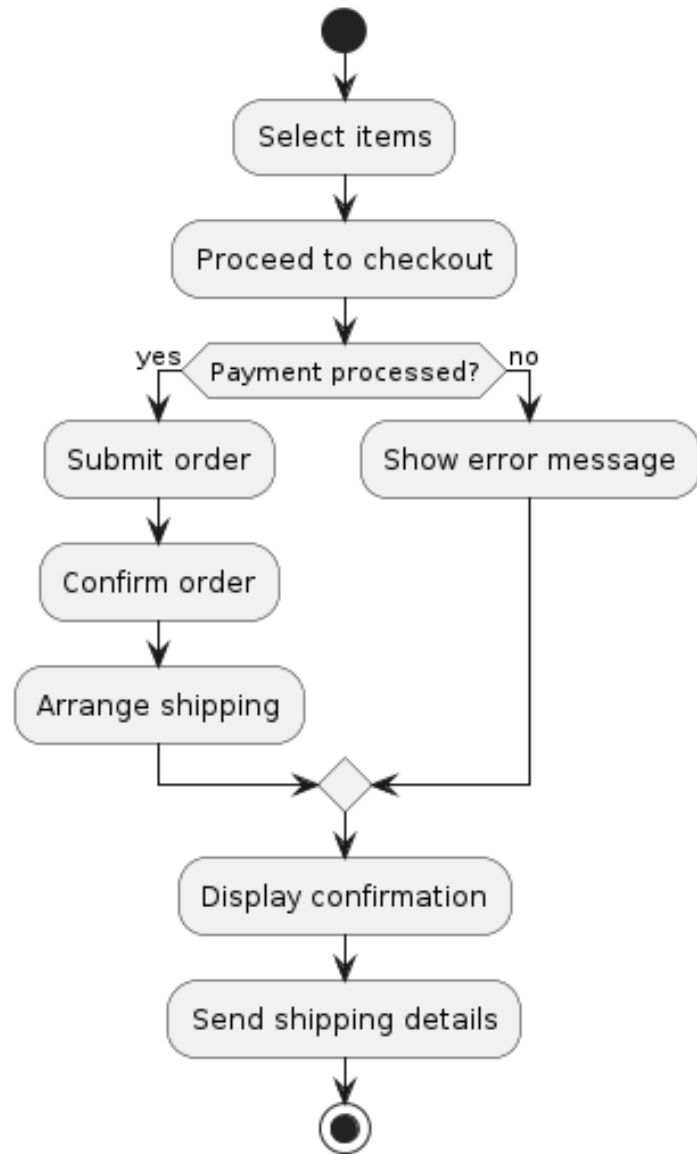
- Our activity diagram illustrates the order process where the user selects and adds items to a cart, proceeds to checkout, and undergoes a login check. If logged in, user details are displayed; otherwise, a login or registration prompt appears. The user then enters shipping information and selects a payment method. The system processes the payment, generating an order confirmation and sending a confirmation email if successful, or displaying a payment error if unsuccessful. Finally, the order is prepared for shipping, shipment is arranged, and the order status is updated.

Class Architect

- The Architect class diagram outlines our e-commerce system's class architecture, showing entities like Admin, User, Product, Inventory, Review, Order, Payment, Shipping, and Return. Admins manage users, products, and reviews, while users place orders, write reviews, and request returns. Orders include products, payments, and shipping details, with Inventory tracking product quantities. It effectively highlights the system's structure and interactions.

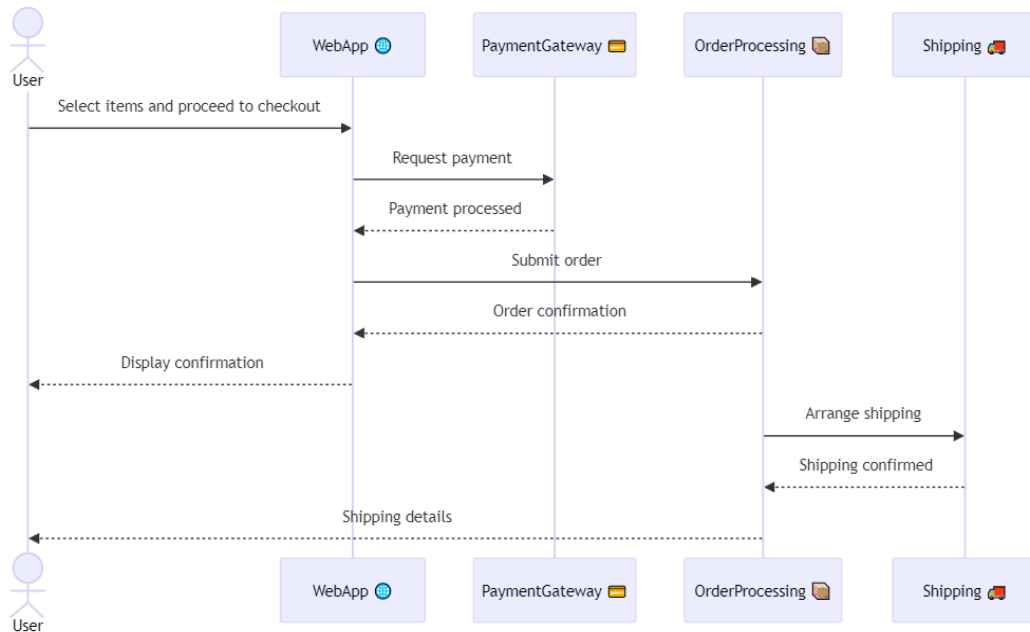


Activity diagram order processing workflow



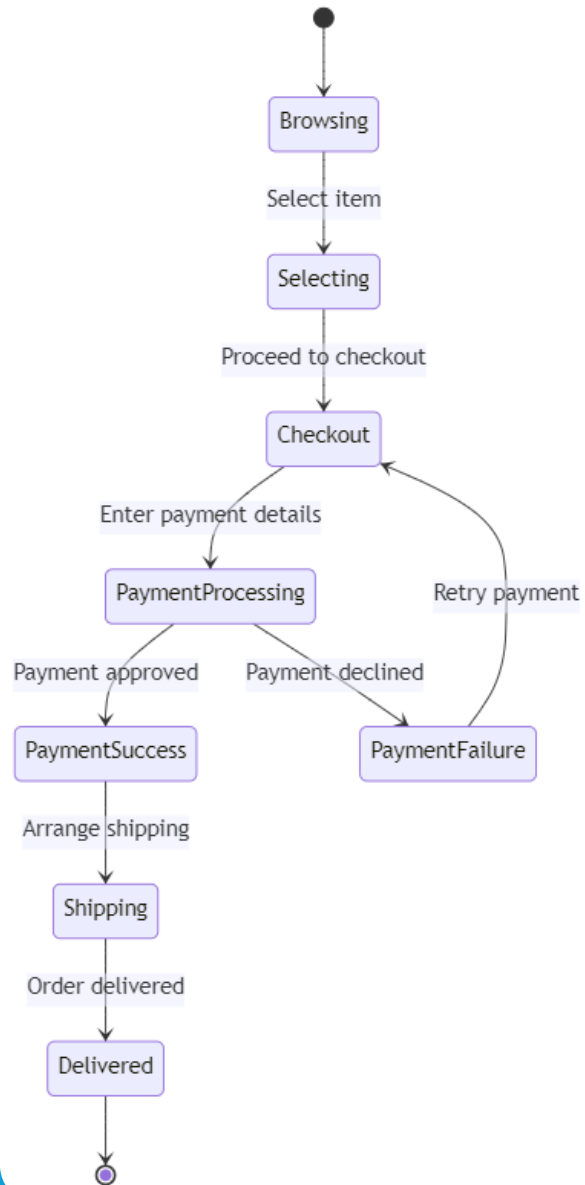
- The activity diagram outlines the order processing workflow. Customers select items, proceed to checkout, and attempt payment. If payment succeeds, the order is submitted, confirmed, shipping is arranged, and shipping details are sent. If payment fails, an error message prompts correction before retrying.

Sequence Diagram



- The sequence diagram depicts the online order process. The user selects items and checks out on the WebApp, which requests and processes payment through the PaymentGateway. The order is then submitted to OrderProcessing, which confirms and arranges shipping via the Shipping service. The WebApp displays the final order and shipping confirmation to the user.

State diagram of order



- . Our state diagram outlines an online shopping order process. It starts with 'Browsing,' then 'Selecting' an item, and proceeding to 'Checkout.' After entering payment details in 'PaymentProcessing,' the payment is either approved, leading to 'PaymentSuccess,' 'Shipping,' and 'Delivered,' or declined, leading to 'PaymentFailure' with an option to retry payment, looping back to 'Checkout.' This diagram clearly maps the order workflow and key decision points

Conclusion

RESULTS

~~EXCUSES~~

- Our diagrams gives a detailed look at our e-commerce system's architecture, using UML diagrams to show how different components like Orders, Products, and Users are structured and interact. It covers various types of diagrams—class, activity, sequence, and state diagrams—which help clarify how our system manages and processes orders. This thorough breakdown not only makes our system's operations clear but also highlights how its design supports scalability and efficiency, key for our future growth and adaptability.

The background of the image is a detailed reproduction of a page from Leonardo da Vinci's sketchbook. It features the central drawing of the Vitruvian Man, a figure inscribed within a circle and a square. Surrounding this central figure are various other sketches, including architectural plans of buildings with arches and stairs, mechanical devices like a crane and a windmill, and a large, ornate architectural dome. The page is filled with Leonardo's characteristic cursive handwriting in Italian, which is mostly illegible but adds to the historical and artistic context of the image.

Art is never finished, only
abandoned.

Grazie per il tuo tempo