图形用户界面, 文本, 应用程序, 信件

AI 生成的内容可能不正确。

图示

AI 生成的内容可能不正确。

A: Yes, it can display both types because:

* Both SpecialEvent and RegularEvent inherit from Event class
* CurrentEventDisplay has a "displays" relationship with Event (0..)

Through polymorphism, CurrentEventDisplay can handle any subclass of Event

-------------------------------------------------------------------------------------------------------

B: Yes, it can display multiple events because:

* The multiplicity between CurrentEventDisplay and Event is "0.."
* This means zero to many events can be displayed

Through the "displays" association with multiplicity 0..

-------------------------------------------------------------------------------------------------------

C: No, the Schedule cannot contain Announcements because:

* There is no direct association between Schedule and Announcement
* Schedule only has a direct association with Event

Announcements are handled separately through AnnouncementDisplay

-------------------------------------------------------------------------------------------------------

D: No, based on the diagram:

* There is no direct or indirect association between Announcement and Date
* Only Events have the association with Date (1..)

-------------------------------------------------------------------------------------------------------

E: Yes, because:

* SpecialEvent inherits from Event
* Event has a "isCurrentOn" association with Date with multiplicity 1..

Through inheritance, SpecialEvent gets the same Date association as its parent class Event

-------------------------------------------------------------------------------------------------------

图示

AI 生成的内容可能不正确。

-------------------------------------------------------------------------------------------------------

A: Activities (rounded rectangles):

* Record Order
* Pick Order
* Create Invoice
* Box Order
* Respond
* Ship

-------------------------------------------------------------------------------------------------------

B: Object/Data nodes (rectangles):

* order: Order
* letter: Rejection\_Letter
* Items
* Invoice
* Full Box
* Labeled Box
* shipment: Shipment

-------------------------------------------------------------------------------------------------------

C: Actions

* Record Order
* Pick Order
* Create Invoice
* Box Order
* Respond
* Ship

-------------------------------------------------------------------------------------------------------

D: Decision nodes (diamonds):

* The diamond after "order: Order" with guards [else] and [existing customer]

-------------------------------------------------------------------------------------------------------

E: Fork nodes (thick horizontal bars):

* The first bar after "Record Order" (splits flow into Pick Order and Create Invoice)
* The second bar before "Labeled Box" (joins the two parallel flows)

-------------------------------------------------------------------------------------------------------

F: Join nodes (thick horizontal bars): Same as fork nodes in this diagram (the second horizontal bar acts as a join)

* The flow from Full Box
* The flow from Invoice”

-------------------------------------------------------------------------------------------------------

G: Main Sequential Flow:

* order: Order → Decision Node
* Decision Node [existing customer] → Record Order

Represented by arrows (→) connecting elements

-------------------------------------------------------------------------------------------------------

H: "Record Order → Fork Node → Pick Order"

WHY this is a control flow:

* It shows the sequence of execution
* It uses arrows (→) to indicate direction
* It connects action nodes and control nodes
* It represents the process logic, not data movement

-------------------------------------------------------------------------------------------------------

I: Yes, because:

* They are connected by a fork node
* Fork nodes indicate parallel processing

After "Record Order", both activities can execute simultaneously

-------------------------------------------------------------------------------------------------------

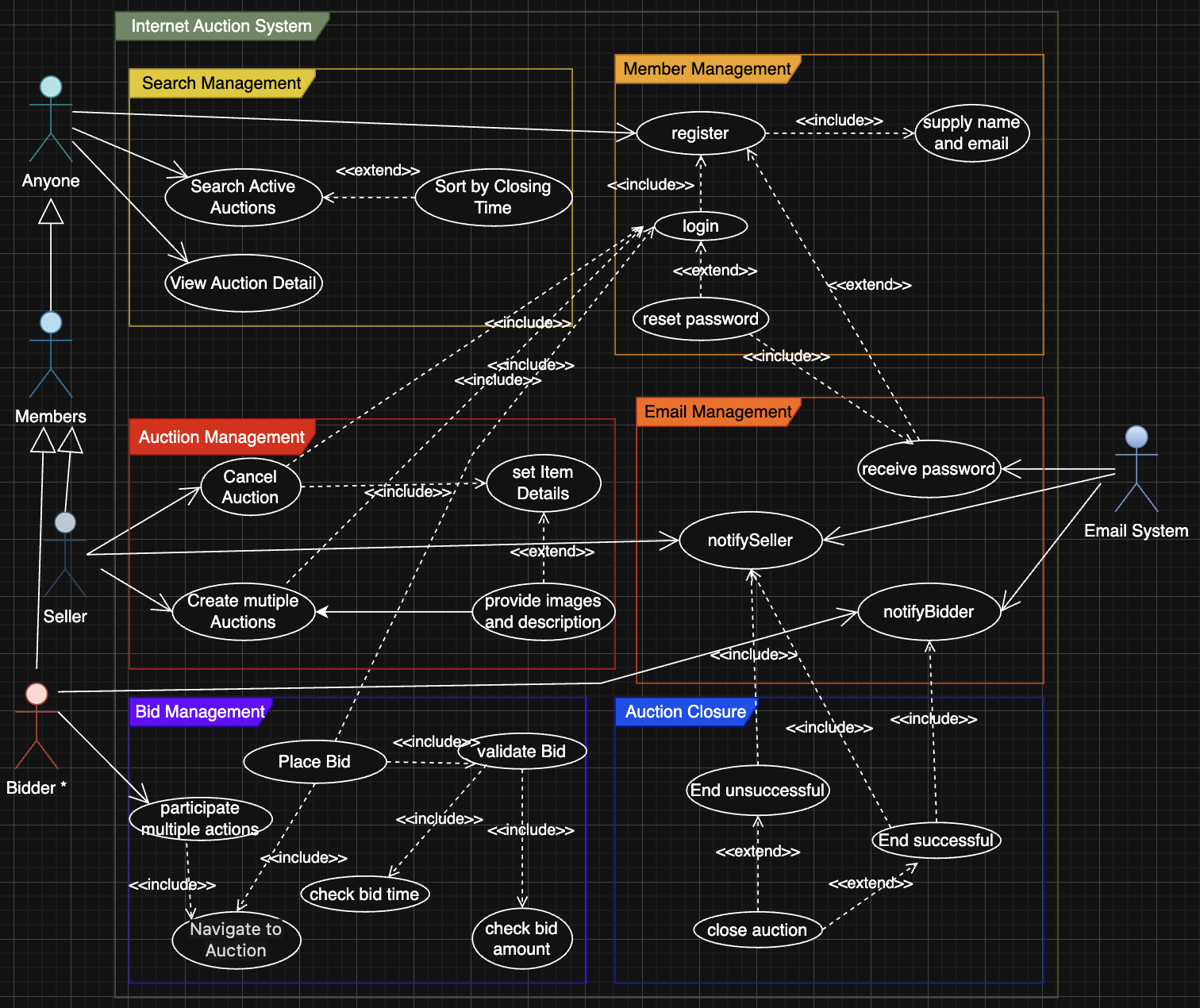
J: No, because:

* They are in a sequential flow
* Ship can only occur after both parallel paths are complete

Ship must wait for the join node to synchronize both parallel paths

-------------------------------------------------------------------------------------------------------

Question 3



### 3.1 Stakeholders (Actors)

### Anyone

* Can search for active auctions and view auction details.

Members

* Must register with name and email.
* Receive password via email.
* Must log in to bid or sell.
* Can reset password (receive via email).
* Can place bids and create auctions.

Seller (a type of Member)

* Can create multiple auctions.
* Can set item details, provide images/description.
* Can cancel auctions.

Bidder (a type of Member)

* Can participate in multiple auctions.
* Can place bids (must be valid and within time/amount constraints).

Email System

* Sends passwords, notifications to sellers and bidders.

### 3-2.Use Cases

1. Search Management

* Search Active Auctions: Anyone can search auctions by keyword.
* Sort by Closing Time: Results are sorted chronologically.
* View Auction Detail: Anyone can view details of any auction.

2. Member Management

* Register: Members must register with name and email.
* Login: Members must log in to access member features.
* Reset Password: Members can request password reset.

Supply Name and Email: Required for registration.

3. Auction Management

* Create Multiple Auctions: Sellers can create multiple auctions.
* Set Item Details: Each auction must have name, closing time, minimum bid.
* Provide Images and Description: Optional for each item.
* Cancel Auction: Sellers can cancel auctions (with notification to bidders).

4. Bid Management

* Place Bid: Members can place bids on auctions.
* Validate Bid: Bids must be above minimum and previous bids.
* Check Bid Time: Bids must be before closing time.
* Check Bid Amount: Bids must be valid.
* Navigate to Auction: Members can access auction pages.

5. Auction Closure

* Close Auction: Auctions are closed after end time or cancellation.
* End Successful: If there are valid bids, notify seller and winning bidder.
* End Unsuccessful: If no bids, notify seller.

6. Email Management

* Receive Password: Members receive password via email.
* Notify Seller: Seller receives notifications (e.g., auction end, cancellation).
* Notify Bidder: Bidders receive notifications (e.g., win, cancellation).

### 3-3.Assumptions

* Sellers and bidders are both members.
* Email notifications are handled by an external system.
* Only registered and logged-in members can bid or create auctions.
* Auctions can be cancelled only up to 24 hours before closing.
* All actions are logged and validated by the system.

### 3-4.Diagram-Scenario Mapping

* **Anyone**: Can search and view auctions.
* **Members**: Must register, log in, and can reset password.
* **Sellers**: Can create, manage, and cancel auctions.
* **Bidders**: Can participate and place bids.
* **Email System**: Handles all email notifications.