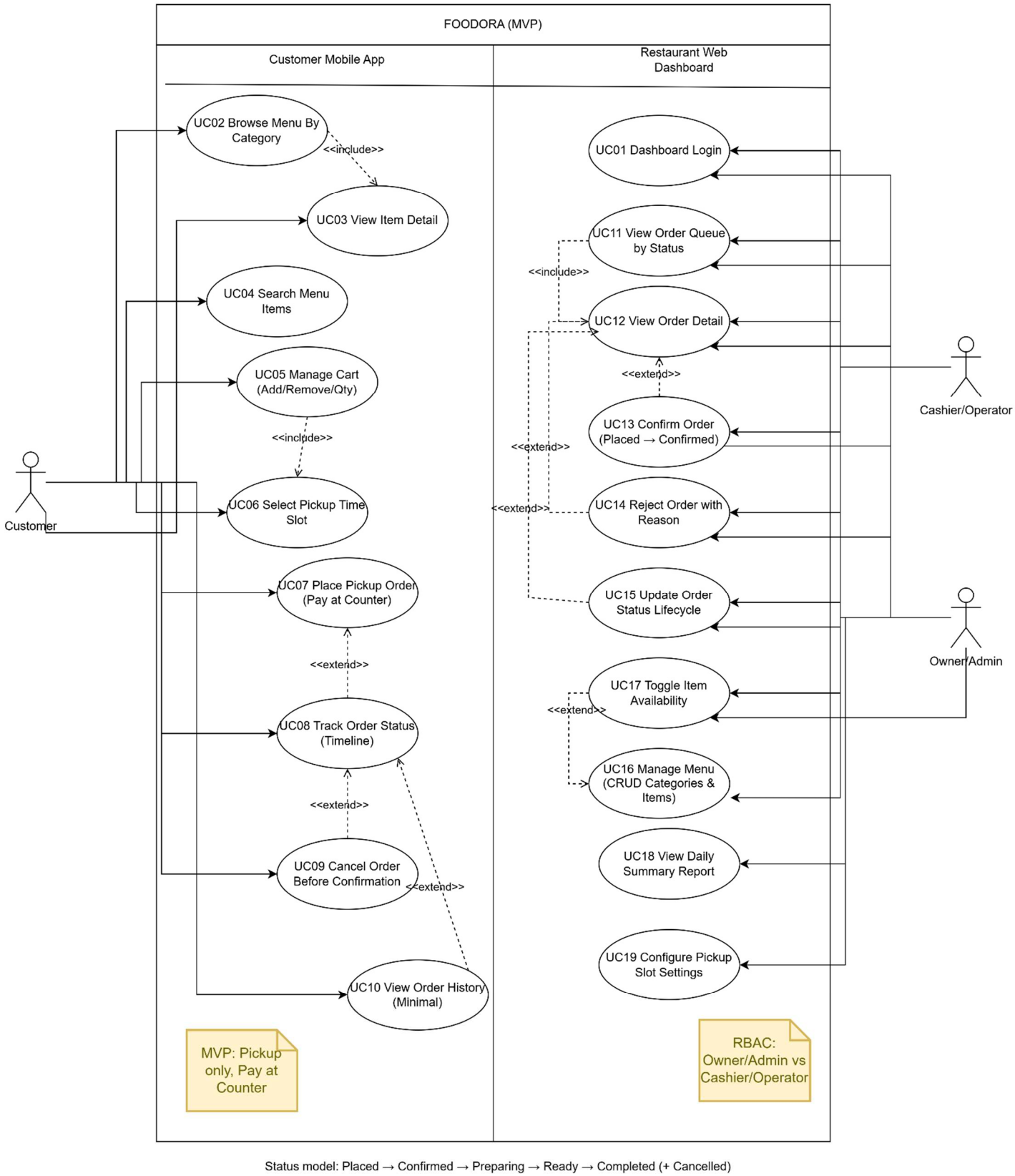
FOCUSED USE CASE

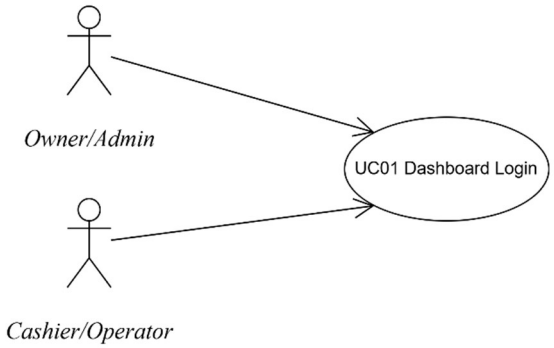
Shared Business Rules (References)

- BR-STAT-01: Order
  statuses: Placed → Confirmed → Preparing → Ready → Completed; plus Cancelled.
- BR-STAT-02: Block invalid transitions (e.g., Ready → Preparing).
- BR-CAN-01: Customer can cancel **only** when status = Placed.
- BR-CAN-02: Restaurant can reject/cancel before Ready; **reason required**. After Ready, cancellation is restricted (optional).
- BR-SLOT-01: Pickup slots generated from opening hours + interval; no past slots.
- BR-SLOT-02: Optional capacity cap per slot; full slots are unavailable.
- BR-AVAIL-01: Re-validate item availability at checkout submit; if any item becomes unavailable → block submit and list impacted items.
- BR-PRICE-01: Total = Σ(priceSnapshot × qty); snapshot captured at order creation; later price changes do not affect historical orders.
- BR-RBAC-01: Roles: Owner/Admin (full) vs Cashier/Operator (orders + limited menu availability toggle).
- BR-AUD-01: All status changes must log fromStatus/toStatus/changedBy/changedAt.
- BR-CONC-01: Concurrency conflict (two operators) → only one succeeds; other sees "status already updated".
- BR-SEC-01: Dashboard requires authentication; restrict customer phone visibility by role as needed.
- BR-PERF-01: Order queue refresh ≤ 10 seconds (MVP acceptable).



Status model: Placed → Confirmed → Preparing → Ready → Completed (+ Cancelled)

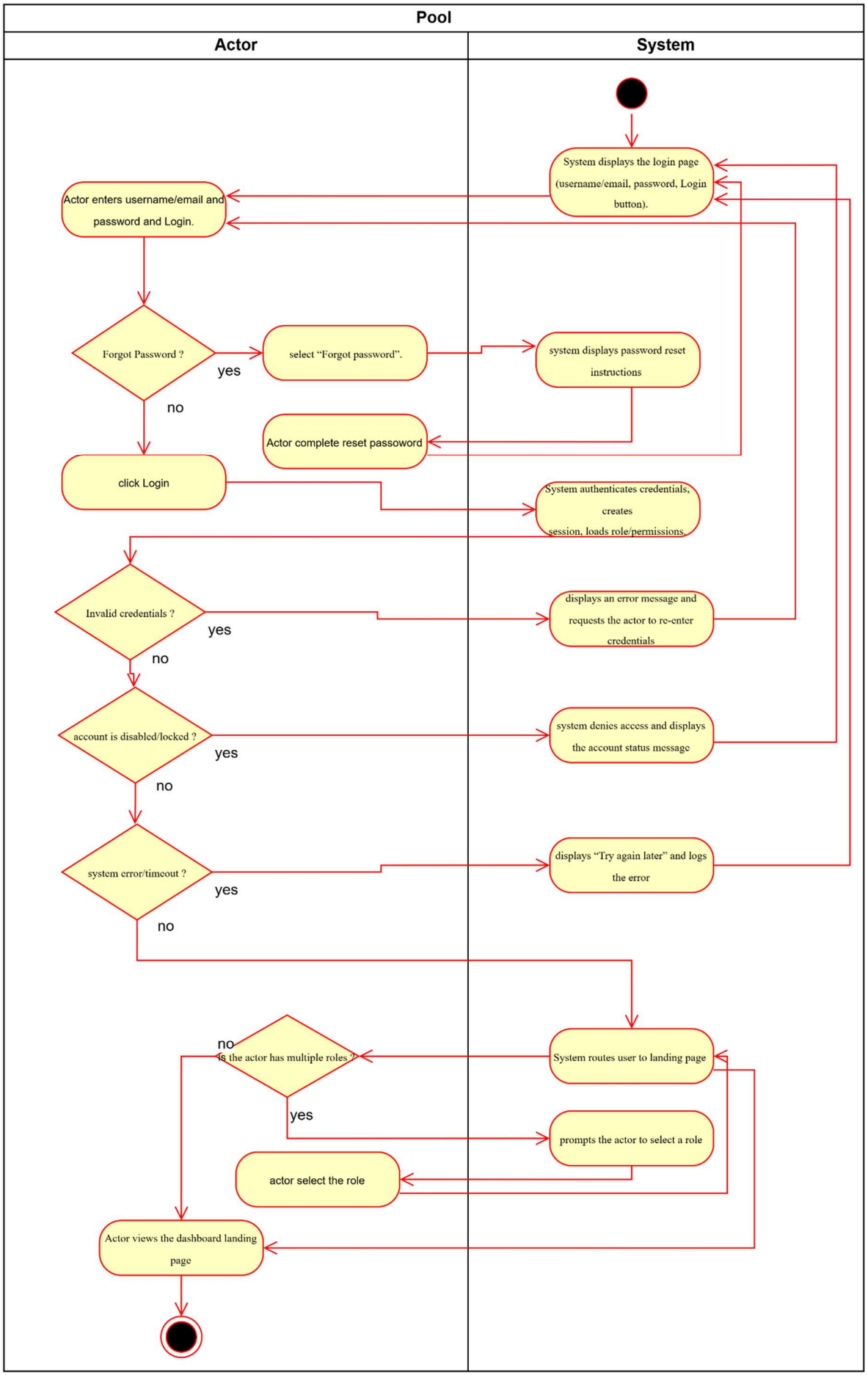
UC01 — Dashboard Login (Auth & Roles)



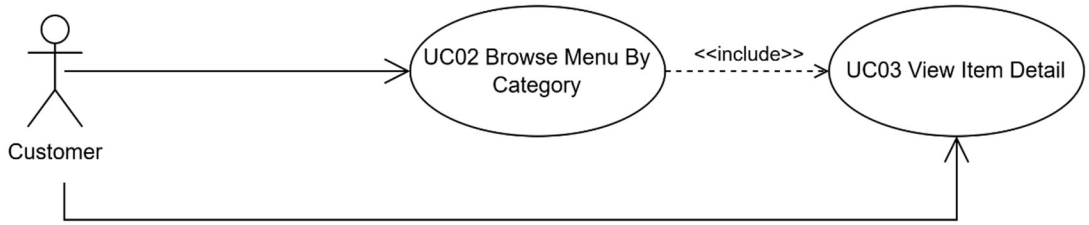
Use Case Number:	UC01	
Use Case Name:	Restaurant Dashboard Login	
Actor (s):	*Owner/Admin, Cashier/Operator*	
Maturity:	Focused	
Summary:	Staff authenticate to access dashboard features based on role (orders/menu/reports).	
Basic Course of Events:	**Actor Action**	**System Response**
	1. Staff authenticate to access dashboard features based on role (orders/menu/reports).	
		2. System displays the login page (username/email, password, Login button).
	3. Actor enters username/email and password.	
	4. Actor clicks “Login”. **A1, E1, E2, E3**	

		5. System authenticates credentials, creates session, loads role/permissions.
		6. System routes user to landing page (Cashier → Order Queue; Admin → Orders/Menu/Reports). **A2**
	7. Actor views the dashboard landing page. The use case ends here.	
Alternative Paths:	**A1. Forgot Password** From **step 4** of the Basic Course of Events, the actor may select "Forgot password". The system displays password reset instructions (portfolio-level). After completion, the actor is returned to the login page. **Control returns to step 2**.	
	A2. Multiple-role user selects role From **step 6**, if the actor has multiple roles, the system prompts the actor to select a role. After the role is selected, the system routes to the corresponding landing page. The use case continues. **Control returns to step 7**.	
Exception Paths:	**E1. Invalid credentials** If the actor enters an incorrect username/password, the system displays an error message and requests the actor to re-enter credentials. **Control returns to step 3**.	
	E2. Account disabled/locked If the actor's account is disabled/locked, the system denies access and displays the account status message. **Control returns to step 2**.	
	E3. System error/timeout If authentication fails due to a system error/timeout, the system displays "Try again later" and logs the error. **Control returns to step 2**.	
Extension Points:	Two-factor authentication (2FA/OTP) (Future)	
Triggers:	Staff needs to access dashboard to process orders or manage menu.	

Assumptions:	User accounts are provisioned by Owner/Admin.
Preconditions:	Dashboard is available; user has valid credentials.
Post Conditions:	User is authenticated; session is created; user lands on role-appropriate page.
Reference: Business Rules	BR-RBAC-01, BR-SEC-01
Author(s):	Business Analyst
Date:	2026-01-07
Activity Diagram:	

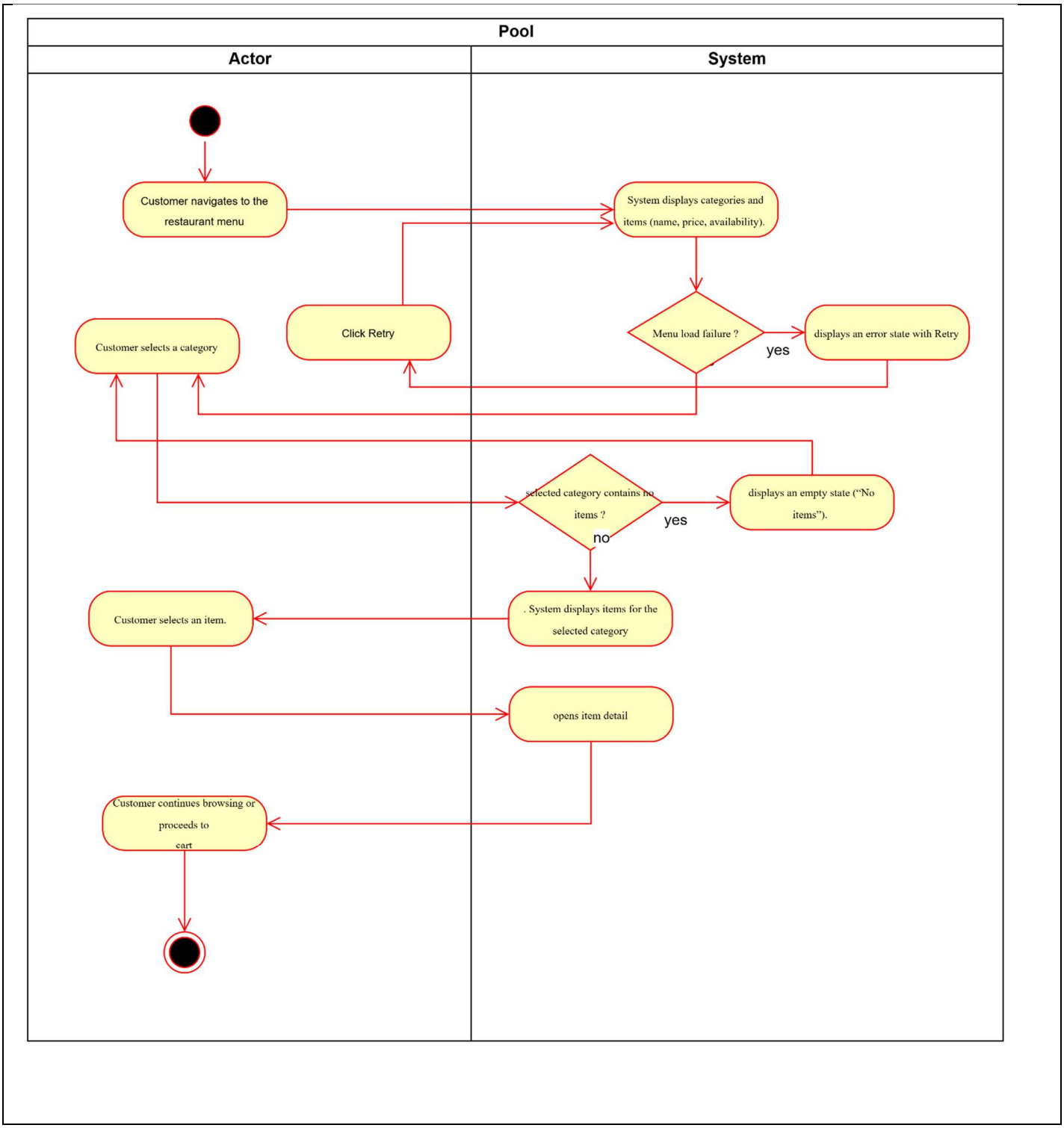

Pool
Actor
System
System displays the login page (username/email, password, Login button).
Actor enters username/email and password and Login.
Forgot Password ?
yes
select "Forgot password".
system displays password reset instructions
no
Actor complete reset passoword
click Login
System authenticates credentials, creates session, loads role/permissions.
Invalid credentials ?
yes
displays an error message and requests the actor to re-enter credentials
no
account is disabled/locked ?
yes
system denies access and displays the account status message
no
system error/timeout ?
yes
displays "Try again later" and logs the error
no
System routes user to landing page
no
Is the actor has multiple roles ?
yes
prompts the actor to select a role
actor select the role
Actor views the dashboard landing page

UC02- Browse Menu by Category

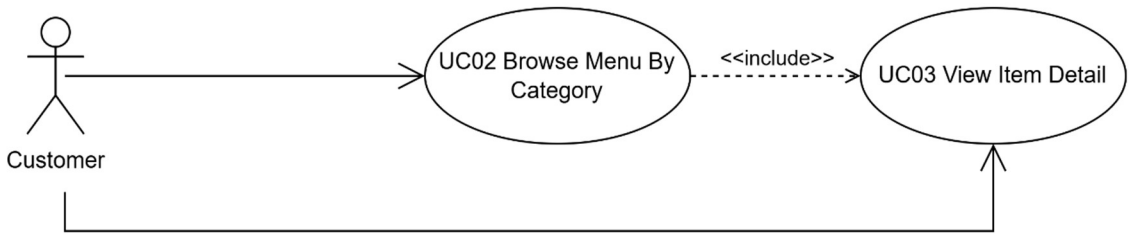


Use Case Number:	UC02	
Use Case Name:	Browse Menu by Category	
Actor (s):	*Customer*	
Maturity:	Focused	
Summary:	Customer browses categories/items and sees availability.	
Basic Course of Events:	**Actor Action**	**System Response**
	1. Customer opens the app and navigates to the restaurant menu.	
		2. System displays categories and items (name, price, availability). **E1**
	3. Customer selects a category.	
		4. System displays items for the selected category. **A1**

	5. Customer selects an item.	
		6. System opens item detail. (See UC03)
	7. Customer continues browsing or proceeds to cart. The use case ends here.	
Alternative Paths:	**A1. Category has no items** From **step 4**, if the selected category contains no items, the system displays an empty state ("No items"). The customer may select a different category. **Control returns to step 3**.	
Exception Paths:	**E1. Menu load failure** If the system cannot load categories/items, it displays an error state with Retry. After retry, the system re-attempts loading. **Control returns to step 2**.	
Extension Points:	Sorting/filtering (Future)	
Triggers:	Customer wants to browse menu to place a pickup order.	
Assumptions:	Menu categories/items are already configured by restaurant staff.	
Preconditions:	Restaurant and menu exist; customer can access the app.	
Post Conditions:	Customer can view categories/items and navigate to item detail.	
Reference: Business Rules	BR-AVAIL-01	
Author(s):	Business Analyst	
Date:	2026-01-07	
Activity Diagram:		

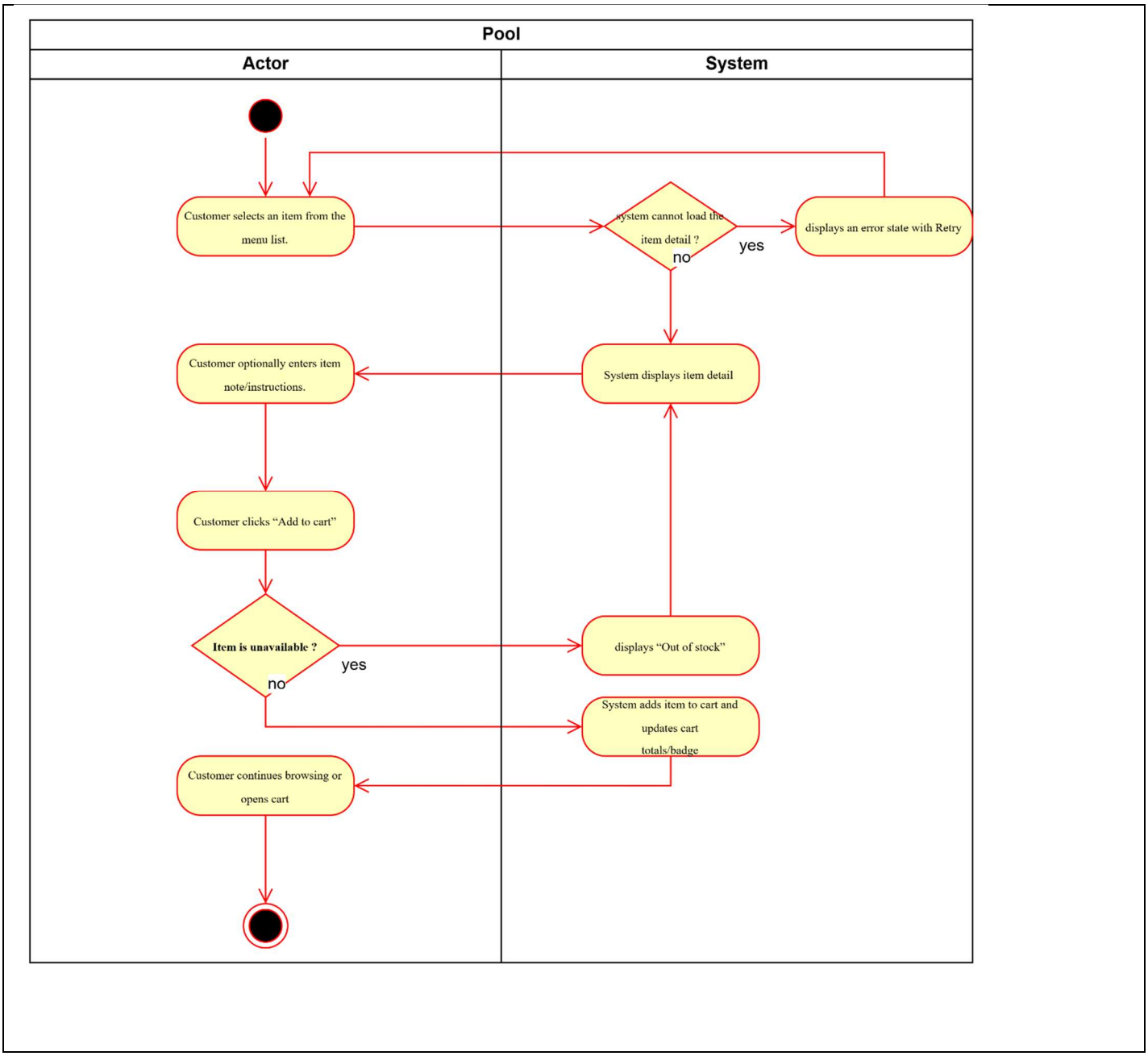


UC03- View Item Detail

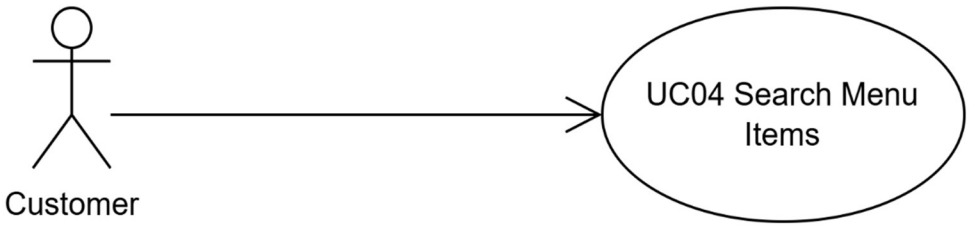


Use Case Number:	UC03	
Use Case Name:	View Item Detail	
Actor (s):	*Customer*	
Maturity:	Focused	
Summary:	Customer views item details and adds available items to cart.	
Basic Course of Events:	**Actor Action**	**System Response**
	1. Customer selects an item from the menu list.	
		2. System displays item detail (name, price, availability, description/image if available). **E1**
	3. Customer optionally enters item note/instructions.	
	4. Customer clicks “Add to cart”. **A1**	
		5. System adds item to cart and updates cart totals/badge.

	6. Customer continues browsing or opens cart. The use case ends here.	
Alternative Paths:	**A1. Item is unavailable** From **step 4**, if the item is unavailable, the system disables “Add to cart” and displays “Out of stock”. **Control returns to step 2**.	
Exception Paths:	**E1. Item detail load failure** If the system cannot load the item detail, it displays an error state with Retry. **Control returns to step 2**.	
Extension Points:	None	
Triggers:	Customer selects an item from the menu list.	
Assumptions:	Item availability displayed may change during peak hours.	
Preconditions:	Menu is available; item exists.	
Post Conditions:	Item detail is displayed; item may be added to cart if available.	
Reference: Business Rules	BR-AVAIL-01	
Author(s):	Business Analyst	
Date:	2026-01-07	
Activity Diagram:		

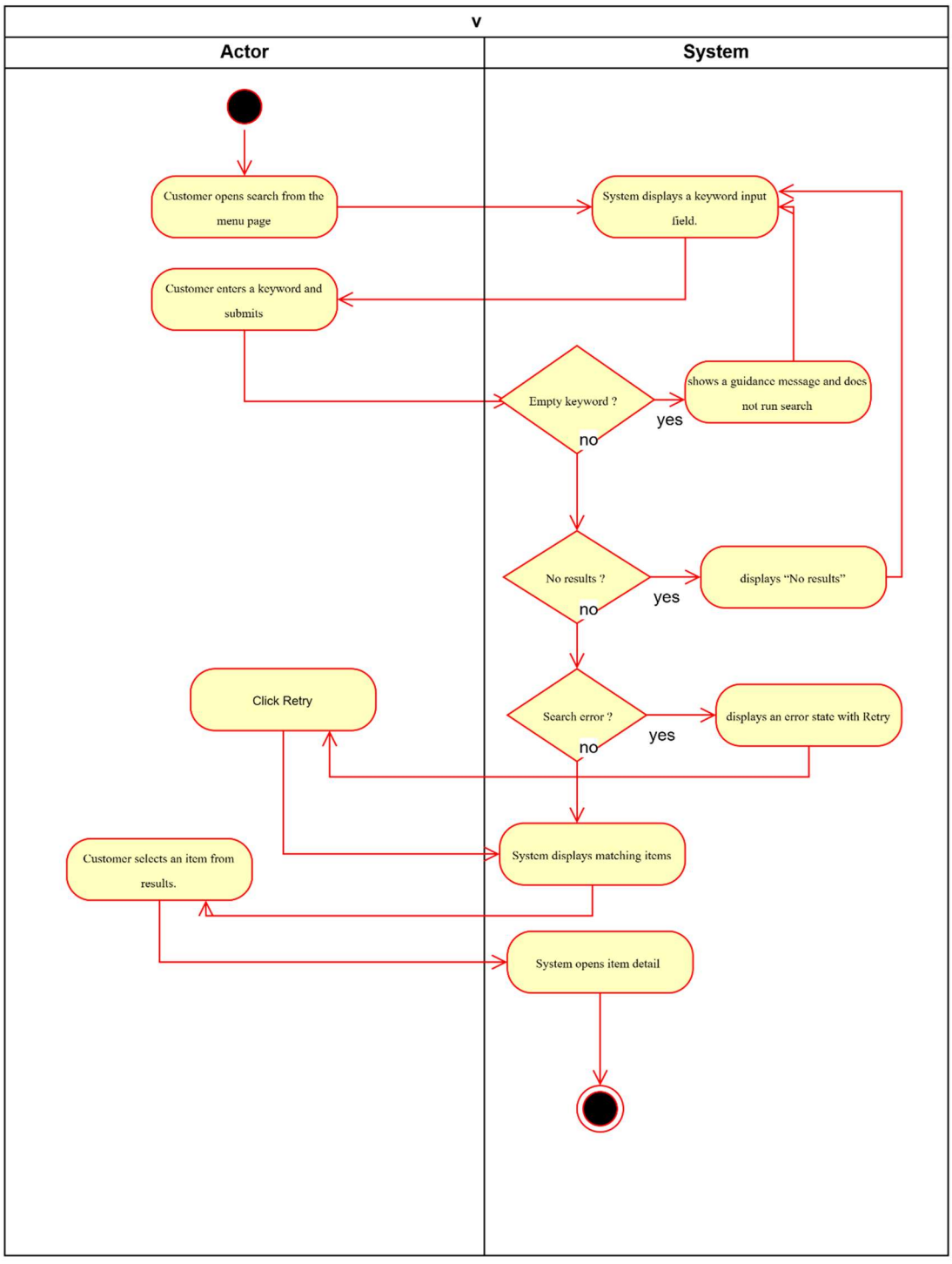


UC04- Search Menu Items

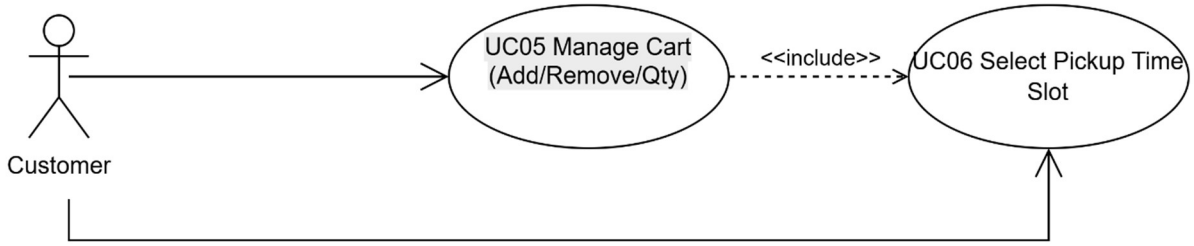


Use Case Number:	UC04	
Use Case Name:	Search Menu Items	
Actor (s):	*Customer*	
Maturity:	Focused	
Summary:	Customer searches items by keyword.	
Basic Course of Events:	**Actor Action**	**System Response**
	1. Customer opens search from the menu page.	
		2. System displays a keyword input field.
	3. Customer enters a keyword and submits. **A1**	
		4. System displays matching items. **E1, E2**
	5. Customer selects an item from results.	
		6. System opens item detail (UC03).

	7. Customer exits search. The use case ends here.	
Alternative Paths:	**A1. Empty keyword (paragraph style)** From **step 3**, if the keyword is empty, the system shows a guidance message and does not run search. **Control returns to step 2**.	
Exception Paths:	**E1. No results** If no items match the keyword, the system displays "No results". **Control returns to step 2**.	
	E2. Search error If search fails, the system displays an error state with Retry. **Control returns to step 4**.	
Extension Points:	Autocomplete/suggestions (Future)	
Triggers:	Customer wants to find items faster by keyword.	
Assumptions:	Search is a "Should" feature; minimal implementation acceptable in MVP.	
Preconditions:	Menu data is available to search.	
Post Conditions:	Matching results or "No results" state is displayed.	
Reference: Business Rules	None	
Author(s):	Business Analyst	
Date:	2026-01-07	
Activity Diagram:		

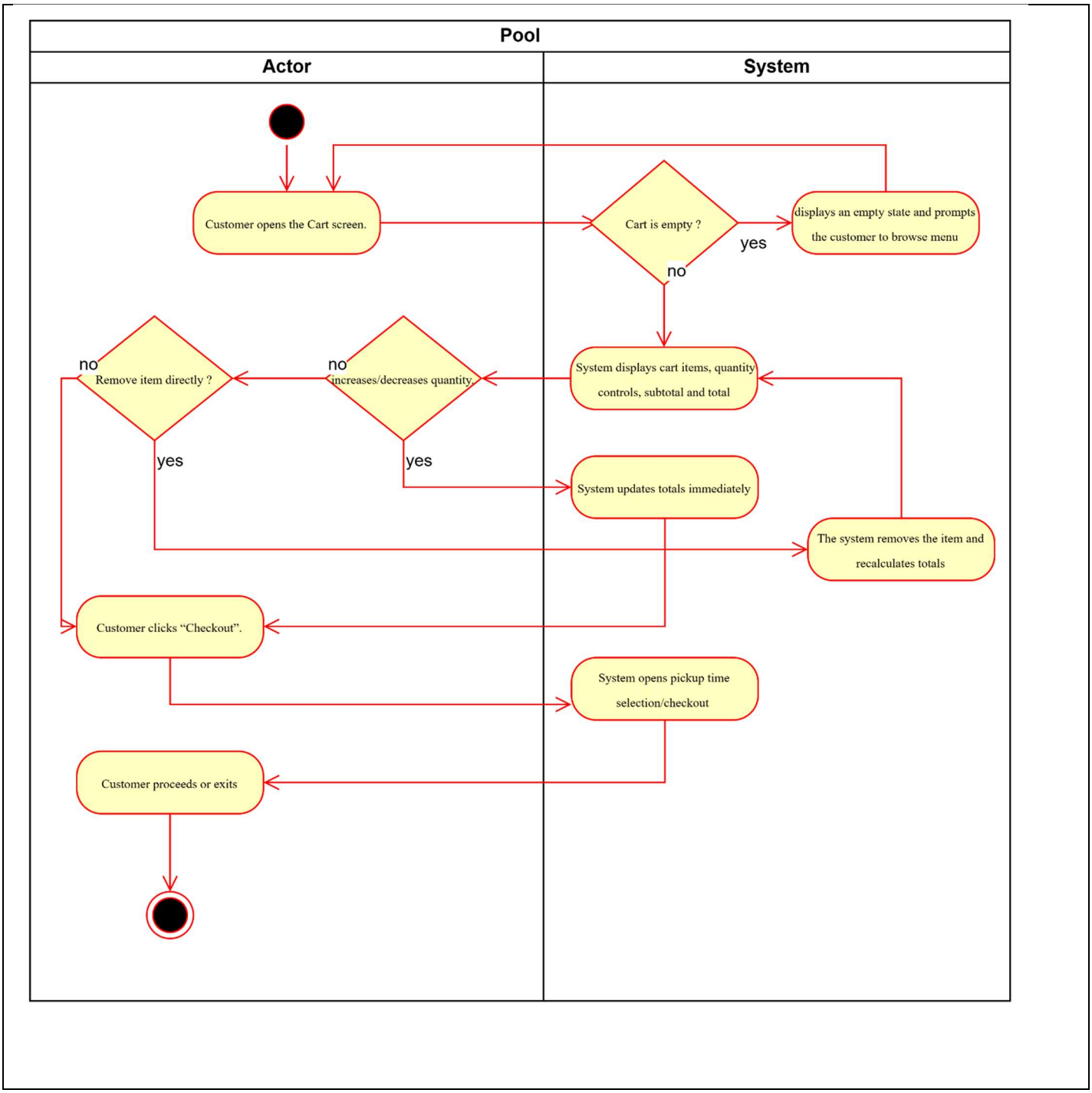

v
Actor
System
Customer opens search from the menu page
System displays a keyword input field.
Customer enters a keyword and submits
Empty keyword ?
yes
shows a guidance message and does not run search
no
No results ?
yes
displays “No results”
no
Search error ?
yes
displays an error state with Retry
no
Click Retry
System displays matching items
Customer selects an item from results.
System opens item detail

UC05- Manage Cart (Add/Remove/Update Quantity)

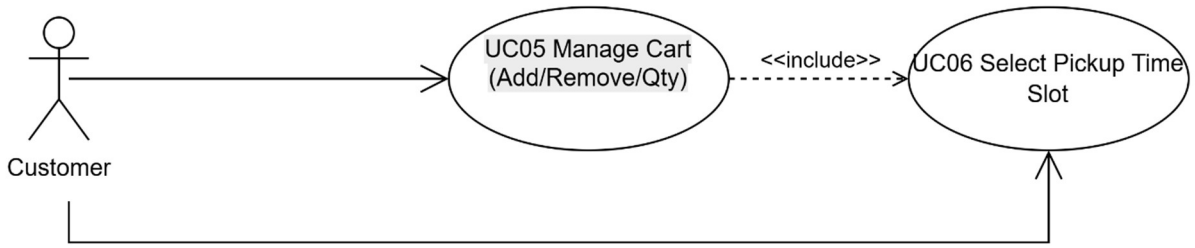


Use Case Number:	UC05	
Use Case Name:	Manage Cart (Add/Remove/Update Quantity)	
Actor (s):	*Customer*	
Maturity:	Focused	
Summary:	Customer updates quantities/removes items and proceeds to checkout.	
Basic Course of Events:	**Actor Action**	**System Response**
	1. Customer opens the Cart screen.	
		2. System displays cart items, quantity controls, subtotal and total. **E1**
	3. Customer increases/decreases quantity. **A1**	
		4. System updates totals immediately; if quantity

		becomes 0, system removes the item.
	5. Customer clicks “Checkout”.	
		6. System opens pickup time selection/checkout (UC06).
	7. Customer proceeds or exits. The use case ends here.	
Alternative Paths:	**A1. Remove item directly** From **step 3**, the customer may remove an item instead of changing quantity. The system removes the item and recalculates totals. **Control returns to step 2**.	
Exception Paths:	**E1. Cart is empty** If the cart is empty, the system displays an empty state and prompts the customer to browse menu. **Control returns to step 1**.	
Extension Points:	Save cart for later (Future)	
Triggers:	Customer wants to review/edit cart before checkout.	
Assumptions:	Totals update immediately when quantities change.	
Preconditions:	Customer can access cart (cart may be empty or non-empty).	
Post Conditions:	Cart contents and totals reflect latest customer edits; customer can proceed to checkout.	
Reference: Business Rules	BR-AVAIL-01, BR-PRICE-01	
Author(s):	Business Analyst	
Date:	2026-01-07	
Activity Diagram:		

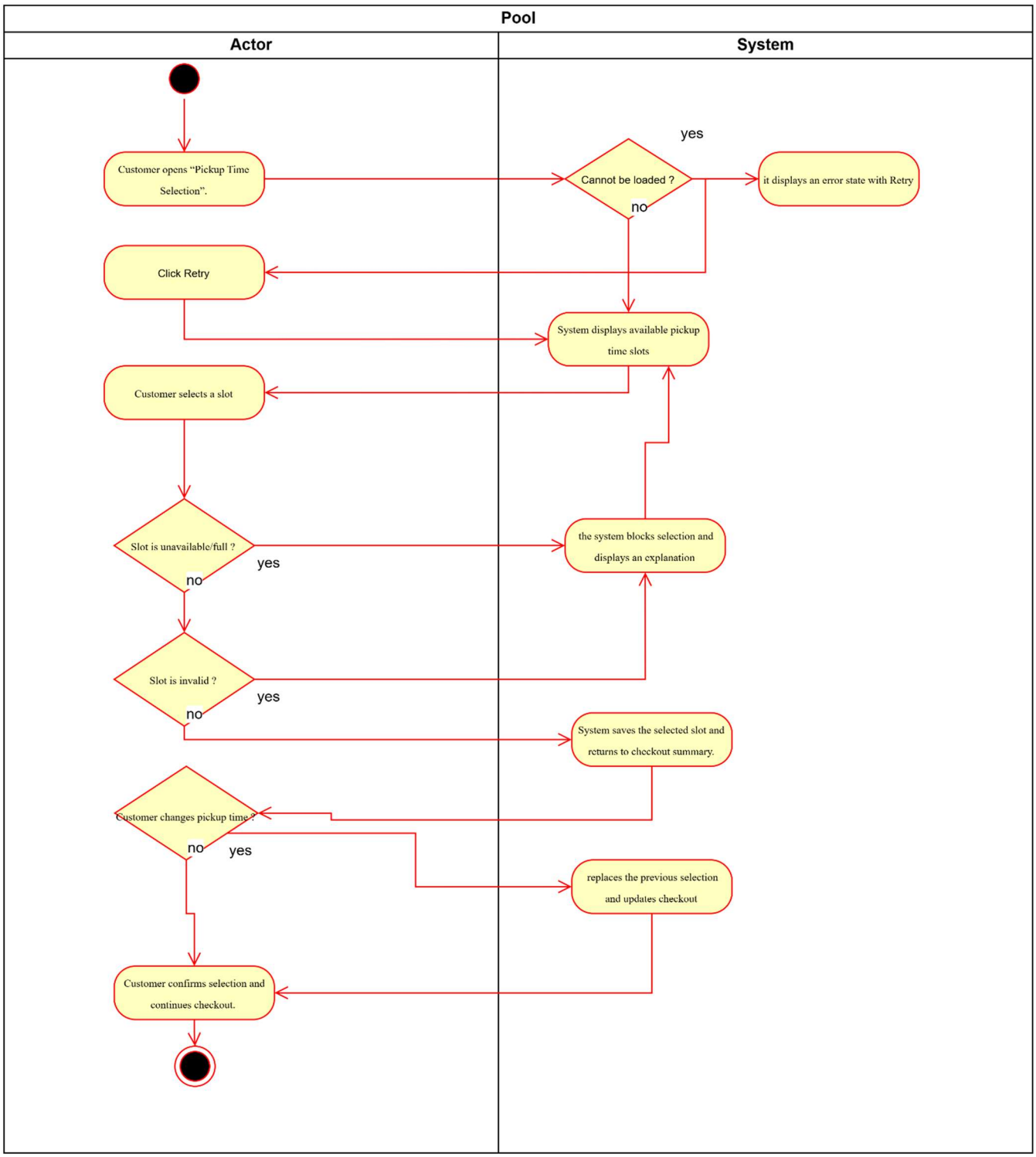


UC06- Select Pickup Time Slot

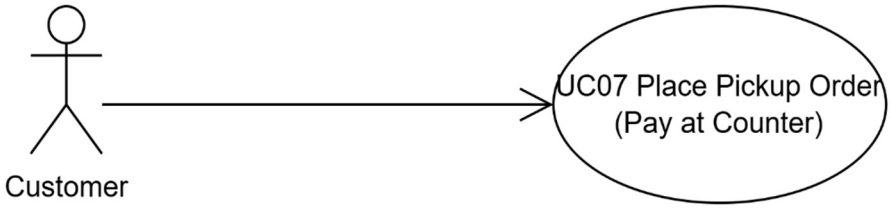


Use Case Number:	UC06	
Use Case Name:	Select Pickup Time Slot	
Actor (s):	*Customer*	
Maturity:	Focused	
Summary:	Customer selects a valid pickup time slot based on opening hours and interval (capacity optional).	
Basic Course of Events:	**Actor Action**	**System Response**
	1. Customer opens "Pickup Time Selection".	
		2. System displays available pickup time slots. **E1**
	3. Customer selects a slot. **E2, E3**	
		4. System saves the selected slot and returns to checkout summary.
	5. Customer confirms selection and continues checkout. The use case ends here. **A1**	
Alternative Paths:	**A1. Customer changes pickup time** From **step 5**, the customer may reopen pickup time selection and	

	choose another slot. The system replaces the previous selection and updates checkout. **Control returns to step 1**.
Exception Paths:	**E1. Slot list cannot be loaded** If the system cannot load slots, it displays an error state with Retry. **Control returns to step 2**.
	E2. Slot is unavailable/full If the selected slot is full/unavailable, the system blocks selection and displays an explanation. **Control returns to step 2**.
	E3. Slot is invalid (past/out-of-hours) If the slot is invalid, the system blocks selection and displays an explanation. **Control returns to step 2**.
Extension Points:	- Capacity cap per slot (Optional) - Estimated ready time (Future)
Triggers:	Customer proceeds to checkout and must select a pickup time.
Assumptions:	Restaurant opening hours and slot interval are configured.
Preconditions:	Cart is not empty; slot generation rules exist.
Post Conditions:	A valid pickup slot is selected and shown in checkout summary.
Reference: Business Rules	BR-SLOT-01, BR-SLOT-02
Author(s):	Business Analyst
Date:	2026-01-07
Activity Diagram:	


Pool
Actor
System
Customer opens "Pickup Time Selection".
Cannot be loaded ?
yes
no
it displays an error state with Retry
Click Retry
System displays available pickup time slots
Customer selects a slot
Slot is unavailable/full ?
yes
no
the system blocks selection and displays an explanation
Slot is invalid ?
yes
no
System saves the selected slot and returns to checkout summary.
Customer changes pickup time ?
no
yes
replaces the previous selection and updates checkout
Customer confirms selection and continues checkout.

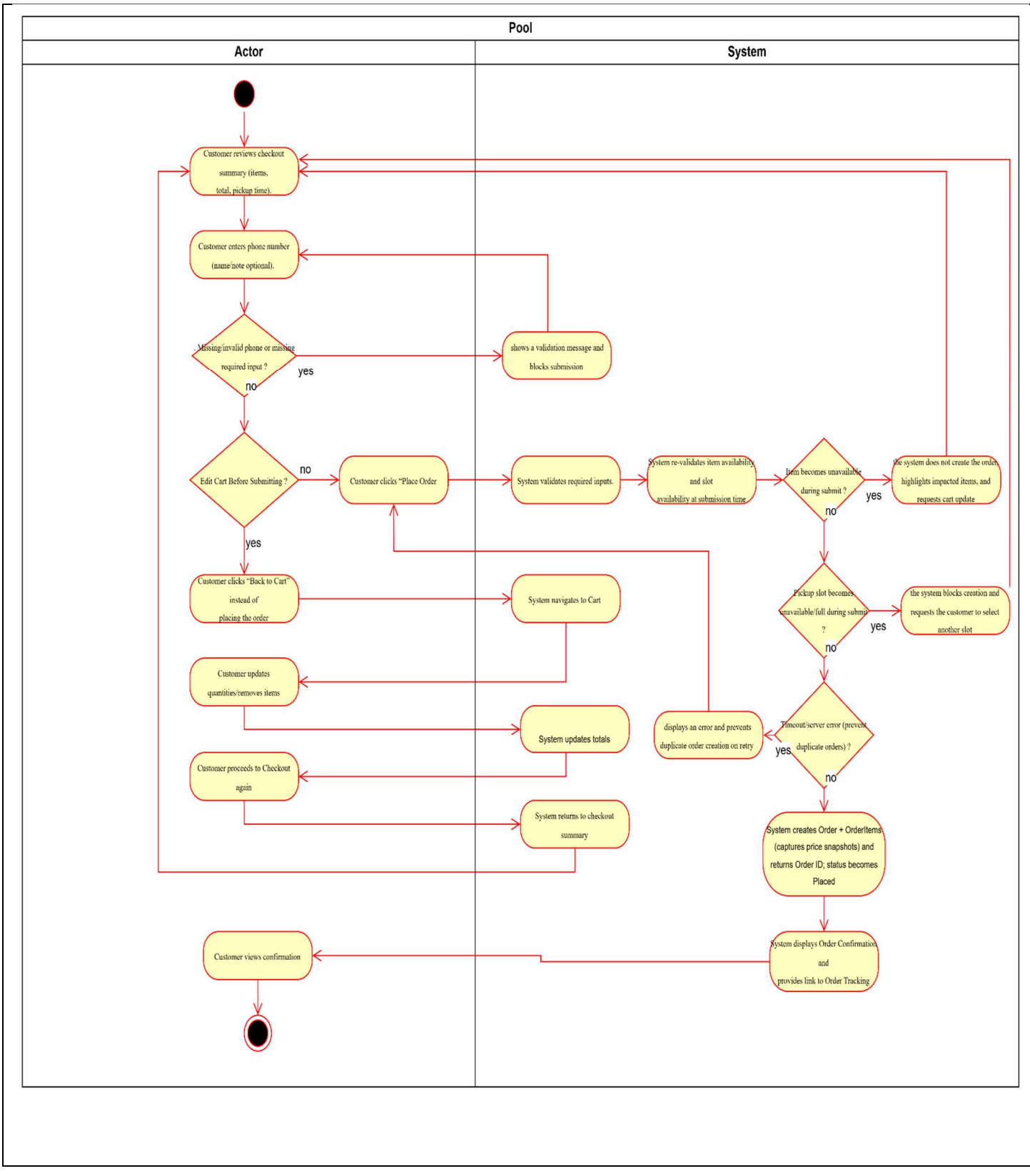
UC07- Place Pickup Order (Pay at Counter)



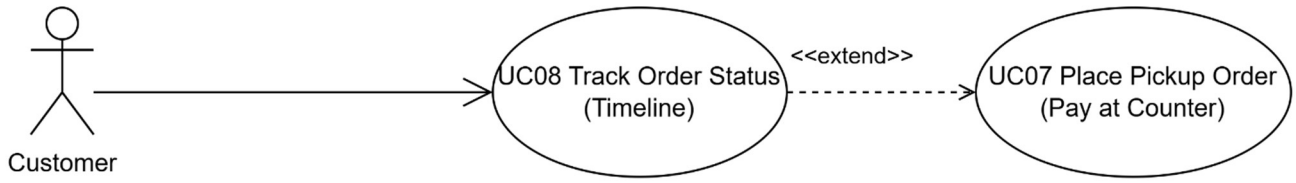
Use Case Number:	UC07	
Use Case Name:	Place Pickup Order (Pay at Counter)	
Actor (s):	*Customer*	
Maturity:	Focused	
Summary:	Customer places a pickup order; system creates an Order ID with status Placed.	
Basic Course of Events:	**Actor Action**	**System Response**
	1. Customer reviews checkout summary (items, total, pickup time).	
	2. Customer enters phone number (name/note optional). **E1**	
	3. Customer clicks “Place Order”. **A1**	

		4. System validates required inputs (cart not empty, pickup time selected, phone present).
		5. System re-validates item availability and slot availability at submission time. **E2, E3**
		6. System creates Order + OrderItems (captures price snapshots) and returns Order ID; status becomes Placed. **E4**
		7. System displays Order Confirmation and provides link to Order Tracking.
	8. Customer views confirmation. The use case ends here.	
Alternative Paths:	**A1. Edit Cart Before Submitting**	
	Actor Action	**System Response**
	3. Customer clicks "Back to Cart" instead of placing the order.	
		4. System navigates to Cart (UC05).
	5. Customer updates quantities/removes items.	
		6. System updates totals.
	7. Customer proceeds to Checkout again.	
		8. System returns to checkout summary. **Control returns to Basic step 1**.
Exception Paths:	**E1. Missing/invalid phone or missing required input** If the customer leaves phone missing or enters an invalid phone format,	

	the system shows a validation message and blocks submission. **Control returns to step 2**.
	E2. Item becomes unavailable during submit If any item becomes unavailable at submission time, the system does not create the order, highlights impacted items, and requests cart update. **Control returns to step 1**.
	E3. Pickup slot becomes unavailable/full during submit If the selected pickup slot becomes unavailable/full at submission time, the system blocks creation and requests the customer to select another slot. **Control returns to step 1** (then customer re-enters pickup selection per UC06).
	E4. Timeout/server error (prevent duplicate orders) If the system fails to create the order due to timeout/server error, it displays an error and prevents duplicate order creation on retry (idempotent behavior). **Control returns to step 3**.
Extension Points:	- Guest checkout vs account-based ordering (Future) - Push notifications (Optional later)
Triggers:	Customer clicks “Place Order” in checkout.
Assumptions:	MVP payment mode is Pay at Counter; system supports price snapshots at order creation.
Preconditions:	Cart not empty; pickup time selected; phone provided; system available.
Post Conditions:	Order is created with Order ID and status Placed; confirmation is displayed; order is visible to restaurant dashboard.
Reference: Business Rules	BR-AVAIL-01, BR-SLOT-01, BR-PRICE-01, BR-STAT-01
Author(s):	Business Analyst
Date:	2026-01-07
Activity Diagram:	

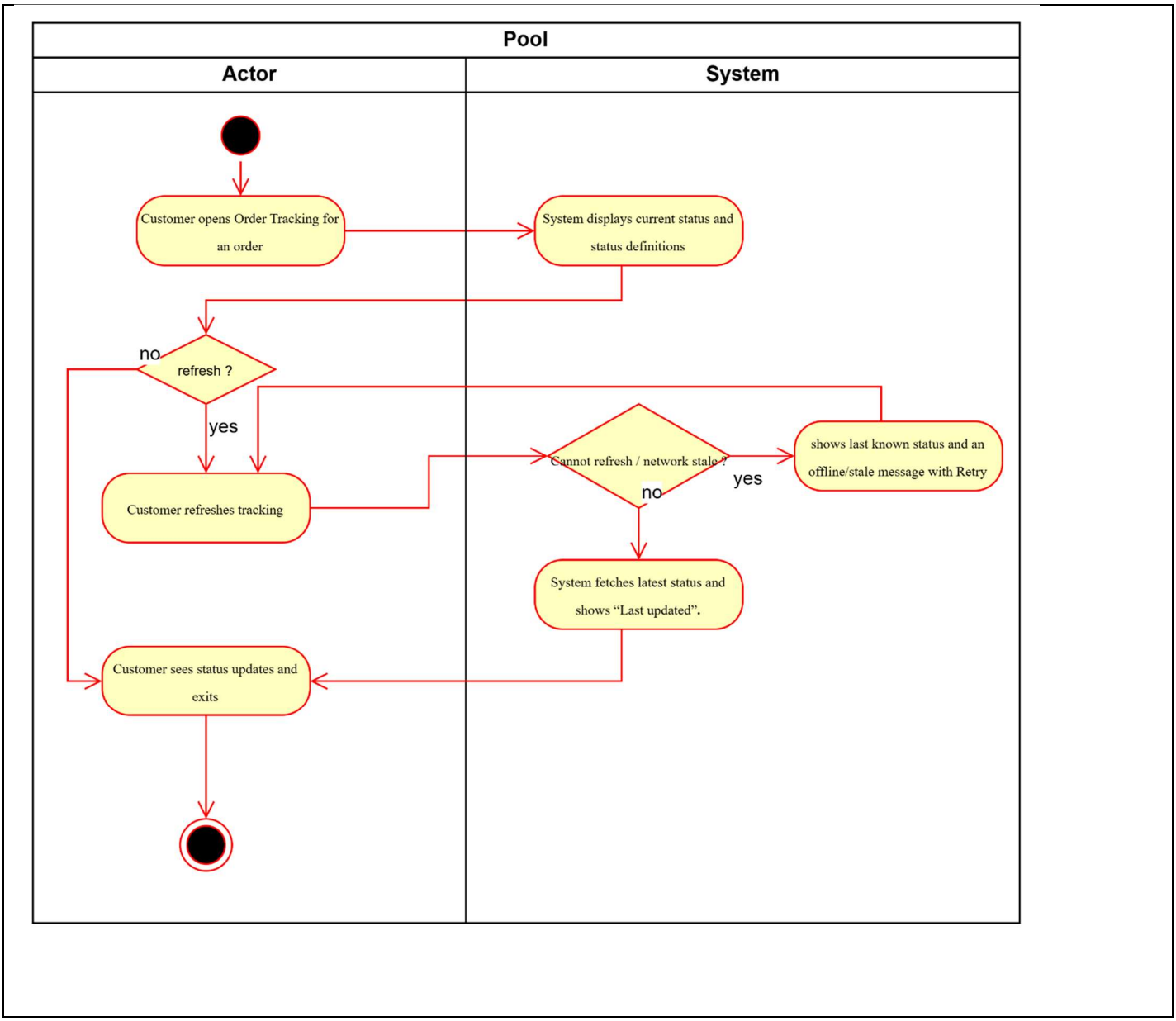

Pool
Actor
System
Customer reviews checkout summary (items, total, pickup time).
Customer enters phone number (name/note optional).
Missing/invalid phone or missing required input ?
yes
shows a validation message and blocks submission
no
Edit Cart Before Submitting ?
no
Customer clicks "Place Order
System validates required inputs.
System re-validates item availability and slot availability at submission time
Item becomes unavailable during submit ?
yes
the system does not create the order, highlights impacted items, and requests cart update
no
yes
Customer clicks "Back to Cart" instead of placing the order
System navigates to Cart
Pickup slot becomes unavailable/full during submit ?
yes
the system blocks creation and requests the customer to select another slot
no
Customer updates quantities/removes items
System updates totals
displays an error and prevents duplicate order creation on retry
Timeout/server error (prevent duplicate orders) ?
yes
Customer proceeds to Checkout again
no
System returns to checkout summary
System creates Order + OrderItems (captures price snapshots) and returns Order ID; status becomes Placed
Customer views confirmation
System displays Order Confirmation and provides link to Order Tracking

UC08- Track Order Status (Timeline)



Use Case Number:	UC08	
Use Case Name:	Track Order Status (Timeline)	
Actor (s):	*Customer*	
Maturity:	Focused	
Summary:	Customer tracks order status timeline and timestamps.	
Basic Course of Events:	**Actor Action**	**System Response**
	1. Customer opens Order Tracking for an order.	
		2. System displays current status and status definitions.
	3. Customer refreshes tracking.**A1**	

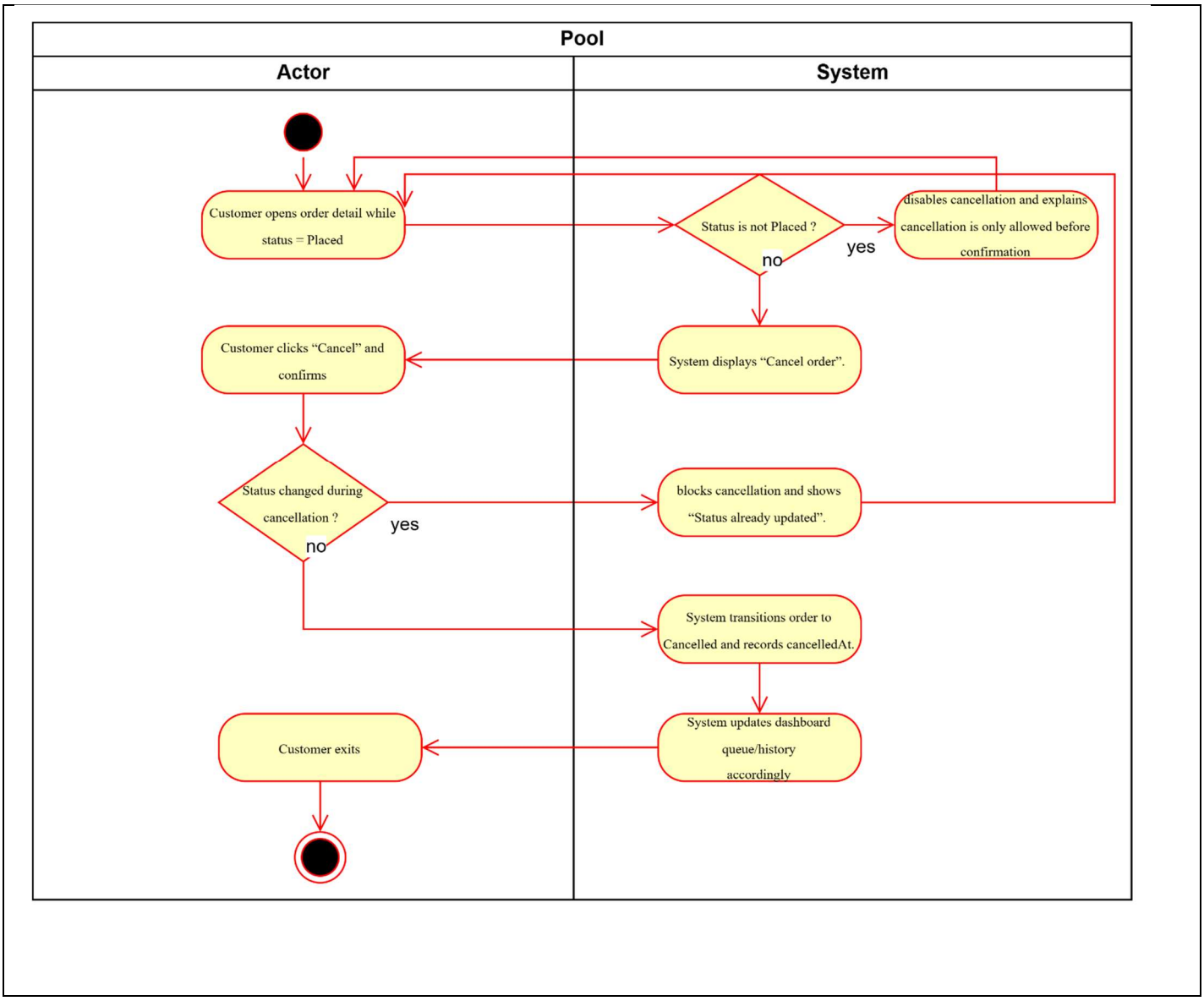
		4. System fetches latest status and shows “Last updated”**.E1**
	5. Customer sees status updates and exits. The use case ends here.	
Alternative Paths:	**A1. View timestamps (paragraph style)** From **step 2**, the customer may view timestamps (confirmedAt/readyAt/completedAt) when available. After viewing, the use case continues. **Control returns to step 4**.	
Exception Paths:	**E1. Cannot refresh / network stale** If the system cannot refresh status, it shows last known status and an offline/stale message with Retry. **Control returns to step 3**.	
Extension Points:	Push notification on Ready (Future/Optional)	
Triggers:	Customer wants to know when to pick up the order.	
Assumptions:	Restaurant staff update statuses via dashboard; customer may need manual refresh in MVP.	
Preconditions:	Order exists and is accessible for tracking.	
Post Conditions:	Customer sees the latest known status timeline and definitions.	
Reference: Business Rules	BR-STAT-01, BR-PERF-01	
Author(s):	Business Analyst	
Date:	2026-01-07	
Activity Diagram:		


Pool
Actor
System
Customer opens Order Tracking for an order
System displays current status and status definitions
no
refresh ?
yes
Customer refreshes tracking
Cannot refresh / network stale ?
yes
shows last known status and an offline/stale message with Retry
no
System fetches latest status and shows “Last updated”.
Customer sees status updates and exits

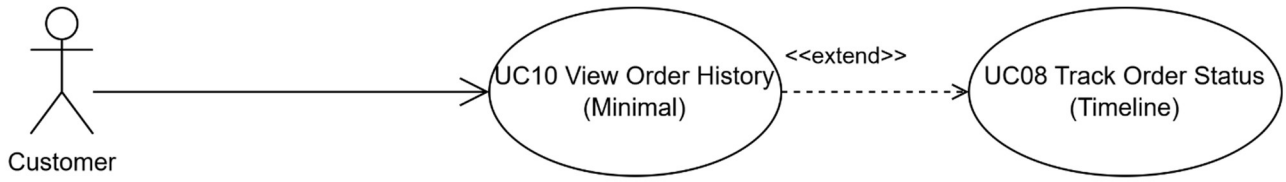
UC09- Cancel Order Before Confirmation

Use Case Number:	UC09	
Use Case Name:	Cancel Order Before Confirmation	
Actor (s):	*Customer*	
Maturity:	Focused	
Summary:	Customer cancels an order only when status is Placed.	
Basic Course of Events:	**Actor Action**	**System Response**
	1. Customer opens order detail while status = Placed.	
		2. System displays “Cancel order”. **E1**
	3. Customer clicks “Cancel” and confirms. **E2**	
		4. System transitions order to Cancelled and records cancelledAt.

		5. System updates dashboard queue/history accordingly.
	6. Customer exits. The use case ends here.	
Alternative Paths:	None	
Exception Paths:	**E1. Status is not Placed** If the order status is Confirmed/Preparing/Ready/Completed, the system disables cancellation and explains cancellation is only allowed before confirmation. **Control returns to step 1**.	
	E2. Status changed during cancellation (concurrency) If the status changes before the cancellation is saved, the system blocks cancellation and shows "Status already updated". **Control returns to step 1**.	
Extension Points:	Customer cancellation reason (Optional)	
Triggers:	Customer decides to cancel before restaurant confirmation.	
Assumptions:	Cancellation is allowed only while status = Placed.	
Preconditions:	Order exists; order status is Placed.	
Post Conditions:	Order status becomes Cancelled; dashboard reflects cancellation; customer sees cancellation result.	
Reference: Business Rules	BR-CAN-01, BR-CONC-01, BR-STAT-01	
Author(s):	Business Analyst	
Date:	2026-01-07	
Activity Diagram:		

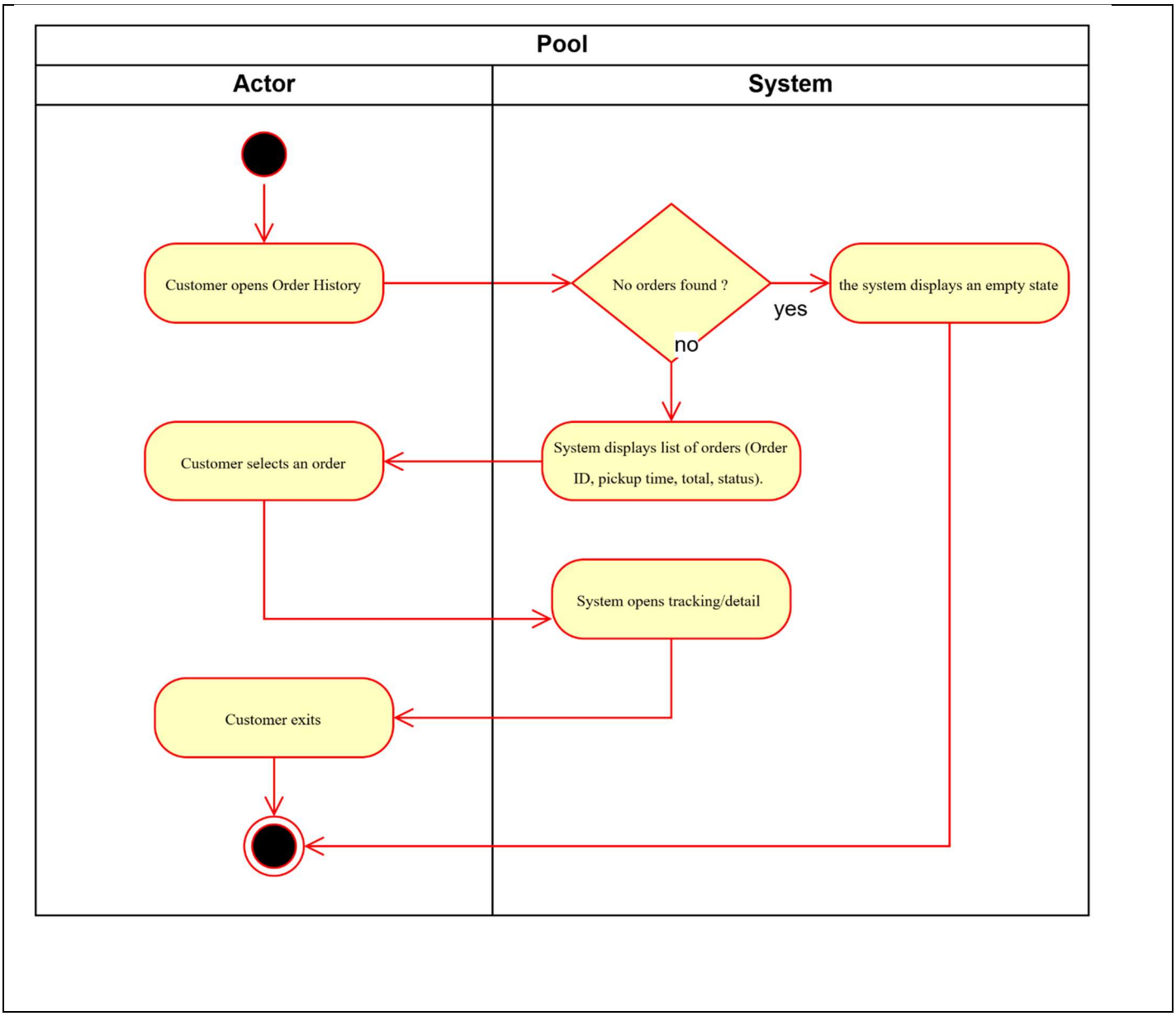


UC10- View Order History (Minimal)

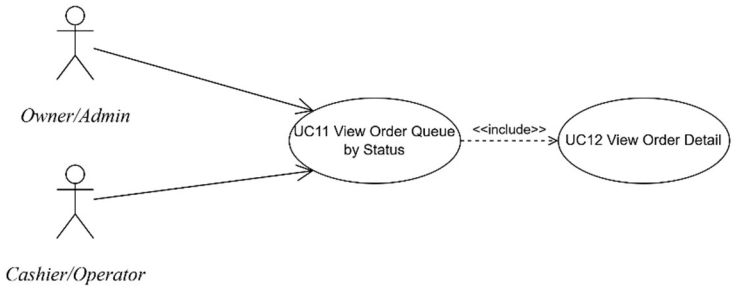


Use Case Number:	UC10	
Use Case Name:	View Order History (Minimal)	
Actor (s):	*Customer*	
Maturity:	Focused	
Summary:	Customer views a basic list of past orders and opens detail/tracking.	
Basic Course of Events:	**Actor Action**	**System Response**
	1. Customer opens Order History.	
		2. System displays list of orders (Order ID, pickup time, total, status). **E1**
	3. Customer selects an order.	
		4. System opens tracking/detail (UC08).
	5. Customer exits. The use case ends here.	
Alternative Paths:		

Exception Paths:	**E1. No orders found** If no orders exist, the system displays an empty state. **Control returns to step 1**.
Extension Points:	Re-order from history (Future)
Triggers:	Customer wants to view past orders.
Assumptions:	History is minimal for MVP (basic list).
Preconditions:	None
Post Conditions:	Order list is displayed; customer can open an order’s detail/tracking.
Reference: Business Rules	None
Author(s):	Business Analyst
Date:	2026-01-07
Activity Diagram:	

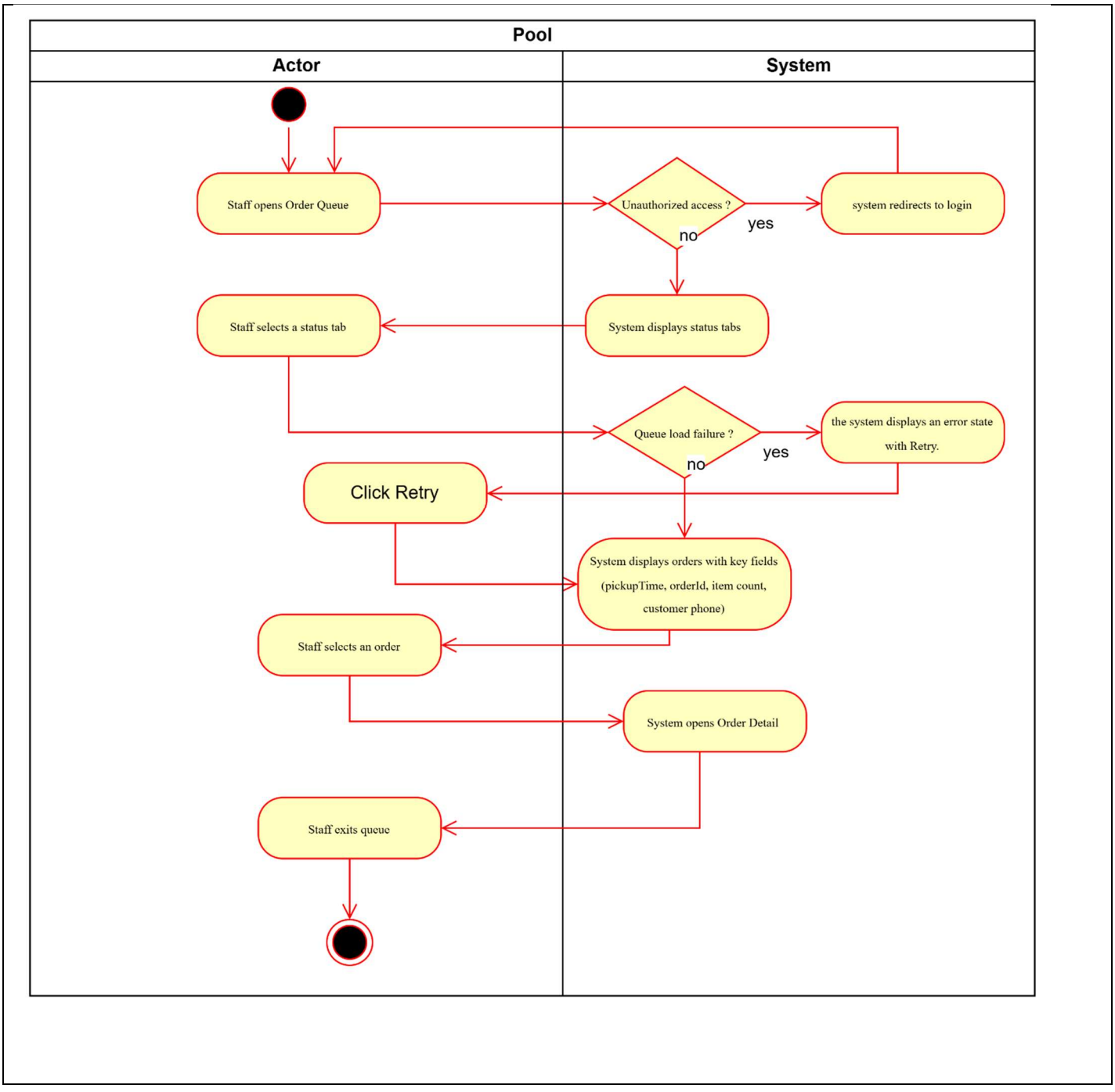


UC11- View Order Queue by Status

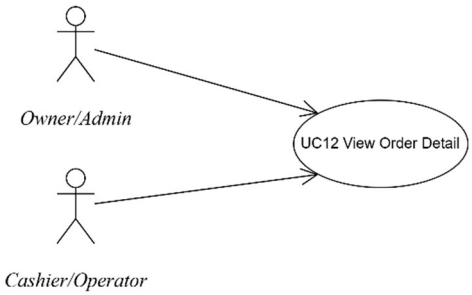


Use Case Number:	UC11	
Use Case Name:	View Order Queue by Status	
Actor (s):	*Cashier/Operator; Owner/Admin*	
Maturity:	Focused	
Summary:	Staff view orders grouped by status tabs for fast processing.	
Basic Course of Events:	**Actor Action**	**System Response**
	1. Staff opens Order Queue.	
		2. System displays status tabs (Placed/Confirmed/Preparing/Ready/ Completed/Cancelled). **E1**
	3. Staff selects a status tab.	
		4. System displays orders with key fields (pickupTime, orderId, item count, customer phone). **E2**
	5. Staff selects an order.	

		6. System opens Order Detail (UC12).
	7. Staff exits queue. The use case ends here.	
Alternative Paths:		
Exception Paths:	**E1. Unauthorized access** If the staff is not authenticated, the system redirects to login. **Control returns to step 1**.	
	E2. Queue load failure If the queue cannot be loaded, the system displays an error state with Retry. **Control returns to step 4**.	
Extension Points:	Search/sort/filter by phone/orderId/pickup time (Future)	
Triggers:	New orders arrive; cashier starts shift; staff need a processing view.	
Assumptions:	Queue refresh is acceptable within ≤10 seconds for MVP.	
Preconditions:	Staff is logged in and authorized.	
Post Conditions:	Staff can view orders grouped by status and open order detail.	
Reference: Business Rules	BR-RBAC-01, BR-SEC-01, BR-PERF-01	
Author(s):	Business Analyst	
Date:	2026-01-07	
Activity Diagram:		

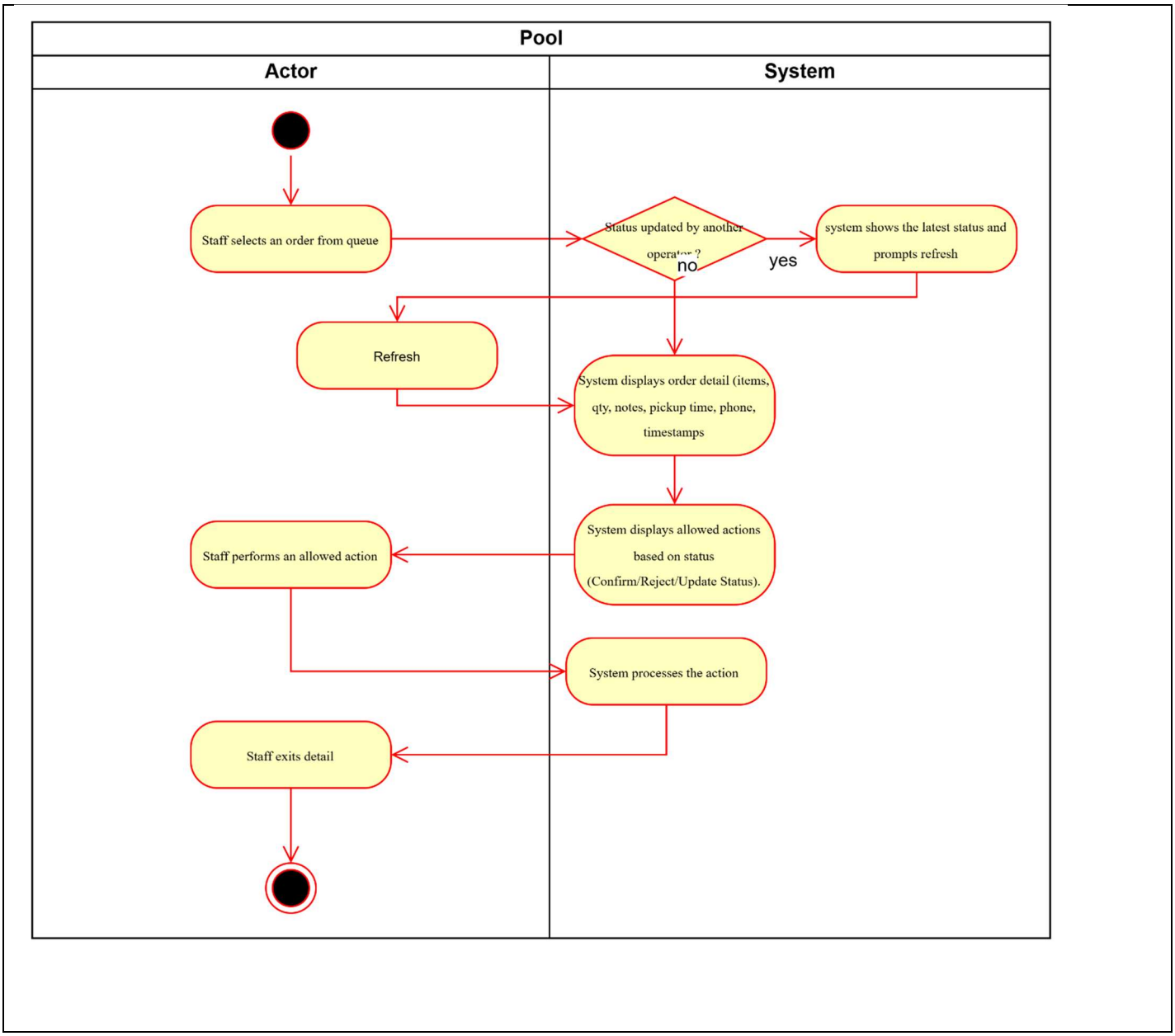


UC12- View Order Detail (Dashboard)

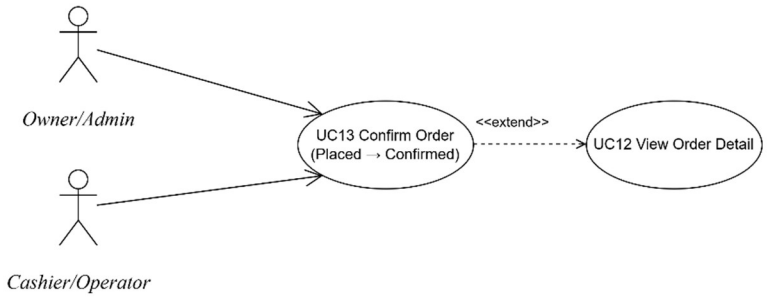


Use Case Number:	UC12	
Use Case Name:	View Order Detail (Dashboard)	
Actor (s):	*Cashier/Operator; Owner/Admin*	
Maturity:	Focused	
Summary:	Staff view order items, notes, pickup time, timestamps, and allowed actions by status.	
Basic Course of Events:	**Actor Action**	**System Response**
	1. Staff selects an order from queue.	
		2. System displays order detail (items, qty, notes, pickup time, phone, timestamps). **E1**
		3. System displays allowed actions based on status (Confirm/Reject/Update Status).
	4. Staff performs an allowed action.	
		5. System processes the action (UC13/UC14/UC15).

	6. Staff exits detail. The use case ends here.	
Alternative Paths:		
Exception Paths:	**E1. Status updated by another operator** If another operator updates the order before the page action completes, the system shows the latest status and prompts refresh. **Control returns to step 1**.	
Extension Points:	Print kitchen ticket/order slip (Optional/Future)	
Triggers:	Staff selects an order from queue to process it.	
Assumptions:	Available actions depend on current status; audit log exists for status changes.	
Preconditions:	Staff is logged in; order exists and is accessible by role.	
Post Conditions:	Staff can view full order context and proceed with confirm/reject/status update actions.	
Reference: Business Rules	BR-STAT-02, BR-AUD-01, BR-CONC-01	
Author(s):	Business Analyst	
Date:	2026-01-07	
Activity Diagram:		

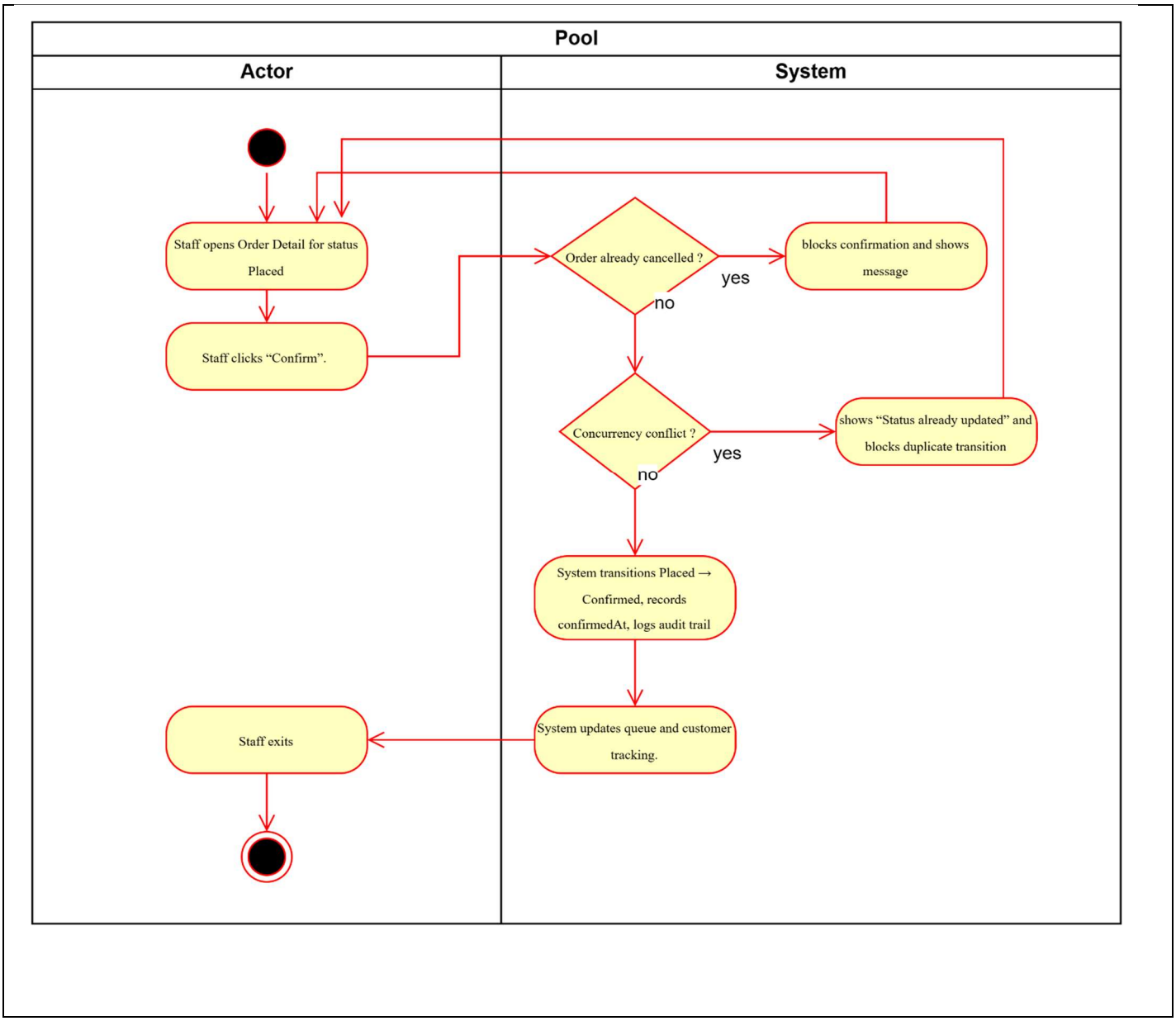


UC13- Confirm Order (Placed → Confirmed)

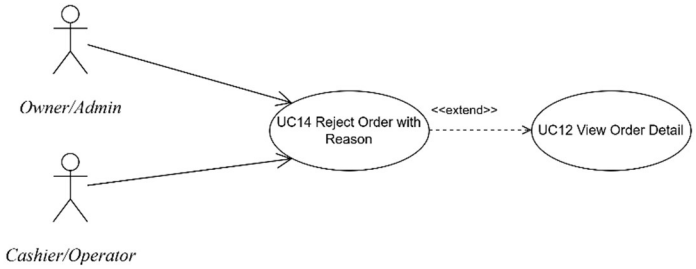


Use Case Number:	UC13	
Use Case Name:	Confirm Order (Placed → Confirmed)	
Actor (s):	*Cashier/Operator*	
Maturity:	Focused	
Summary:	Staff confirms a new order; system records confirmedAt and audit log.	
Basic Course of Events:	**Actor Action**	**System Response**
	1. Staff opens Order Detail for status Placed.	
	2. Staff clicks "Confirm". **E1, E2**	
		3. System transitions Placed → Confirmed, records confirmedAt, logs audit trail.
		4. System updates queue and customer tracking.
	5. Staff exits. The use case ends here.	

Alternative Paths:	
Exception Paths:	**E1. Order already cancelled** If the order is already Cancelled, the system blocks confirmation and shows message. **Control returns to step 1**.
	E2. Concurrency conflict If another staff confirms first, the system shows "Status already updated" and blocks duplicate transition. **Control returns to step 1**.
Extension Points:	Set estimated ready time at confirmation (Future)
Triggers:	A new order is received in Placed status.
Assumptions:	Only staff can confirm; confirmation should be fast for handling-time KPI.
Preconditions:	Staff logged in; order status is Placed.
Post Conditions:	Order status becomes Confirmed; confirmedAt and audit record are stored; customer tracking reflects update.
Reference: Business Rules	BR-STAT-01, BR-AUD-01, BR-CONC-01
Author(s):	Business Analyst
Date:	2026-01-07
Activity Diagram:	

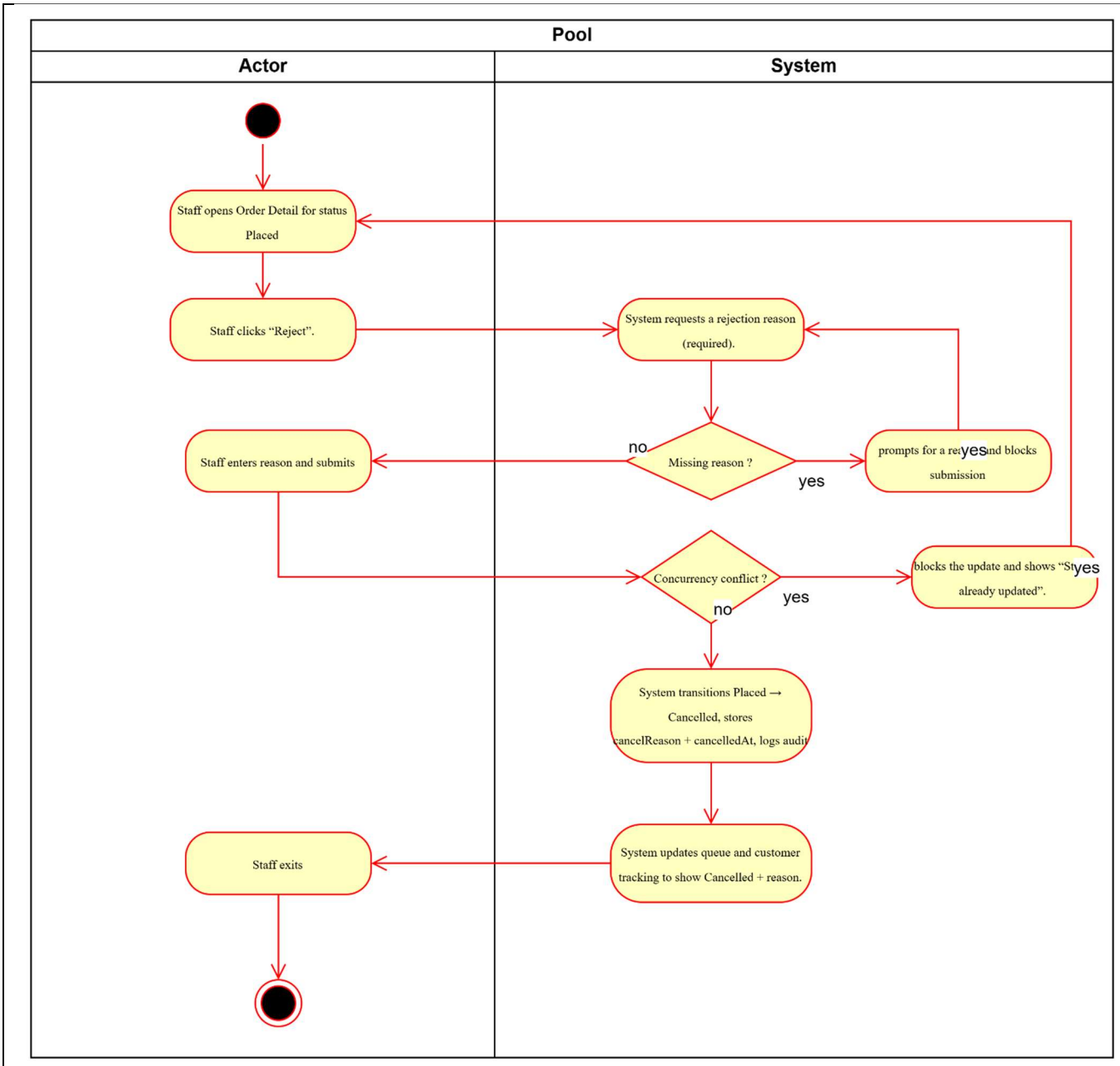


UC14- Reject Order with Reason (Placed → Cancelled)

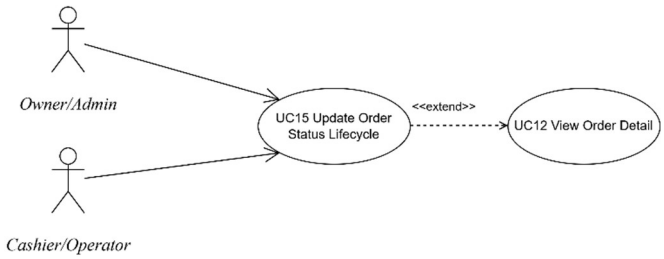


Use Case Number:	UC14	
Use Case Name:	Reject Order with Reason (Placed → Cancelled)	
Actor (s):	*Cashier/Operator*	
Maturity:	Focused	
Summary:	Staff rejects a Placed order with mandatory reason; customer sees Cancelled + reason.	
Basic Course of Events:	**Actor Action**	**System Response**
	1. Staff opens Order Detail for status Placed.	
	2. Staff clicks "Reject".	
		3. System requests a rejection reason (required). **E1**
	4. Staff enters reason and submits. **E2**	
		5. System transitions Placed → Cancelled, stores cancelReason + cancelledAt, logs audit.

		6. System updates queue and customer tracking to show Cancelled + reason.
	7. Staff exits. The use case ends here.	
Alternative Paths:	None	
Exception Paths:	**E1. Missing reason** If reason is empty, the system prompts for a reason and blocks submission. **Control returns to step 3**.	
	E2. Concurrency conflict If another operator changes status first, the system blocks the update and shows “Status already updated”. **Control returns to step 1**.	
Extension Points:	Preset rejection reasons + optional free text (Optional)	
Triggers:	Restaurant cannot fulfill the order (out of stock/overload/cannot meet pickup time).	
Assumptions:	Reason is mandatory to reduce customer confusion.	
Preconditions:	Staff logged in; order status is Placed.	
Post Conditions:	Order status becomes Cancelled with cancelReason; cancelledAt and audit record stored; customer sees reason in tracking.	
Reference: Business Rules	BR-CAN-02, BR-AUD-01, BR-CONC-01, BR-STAT-01	
Author(s):	Business Analyst	
Date:	2026-01-07	
Activity Diagram:		


Pool
Actor
System
Staff opens Order Detail for status Placed
Staff clicks "Reject".
System requests a rejection reason (required).
Missing reason ?
no
yes
prompts for a re
nd blocks submission
yes
Staff enters reason and submits
Concurrency conflict ?
yes
no
blocks the update and shows "St
already updated".
yes
System transitions Placed → Cancelled, stores cancelReason + cancelledAt, logs audit
System updates queue and customer tracking to show Cancelled + reason.
Staff exits

UC15- Update Order Status (Confirmed → Preparing → Ready → Completed)

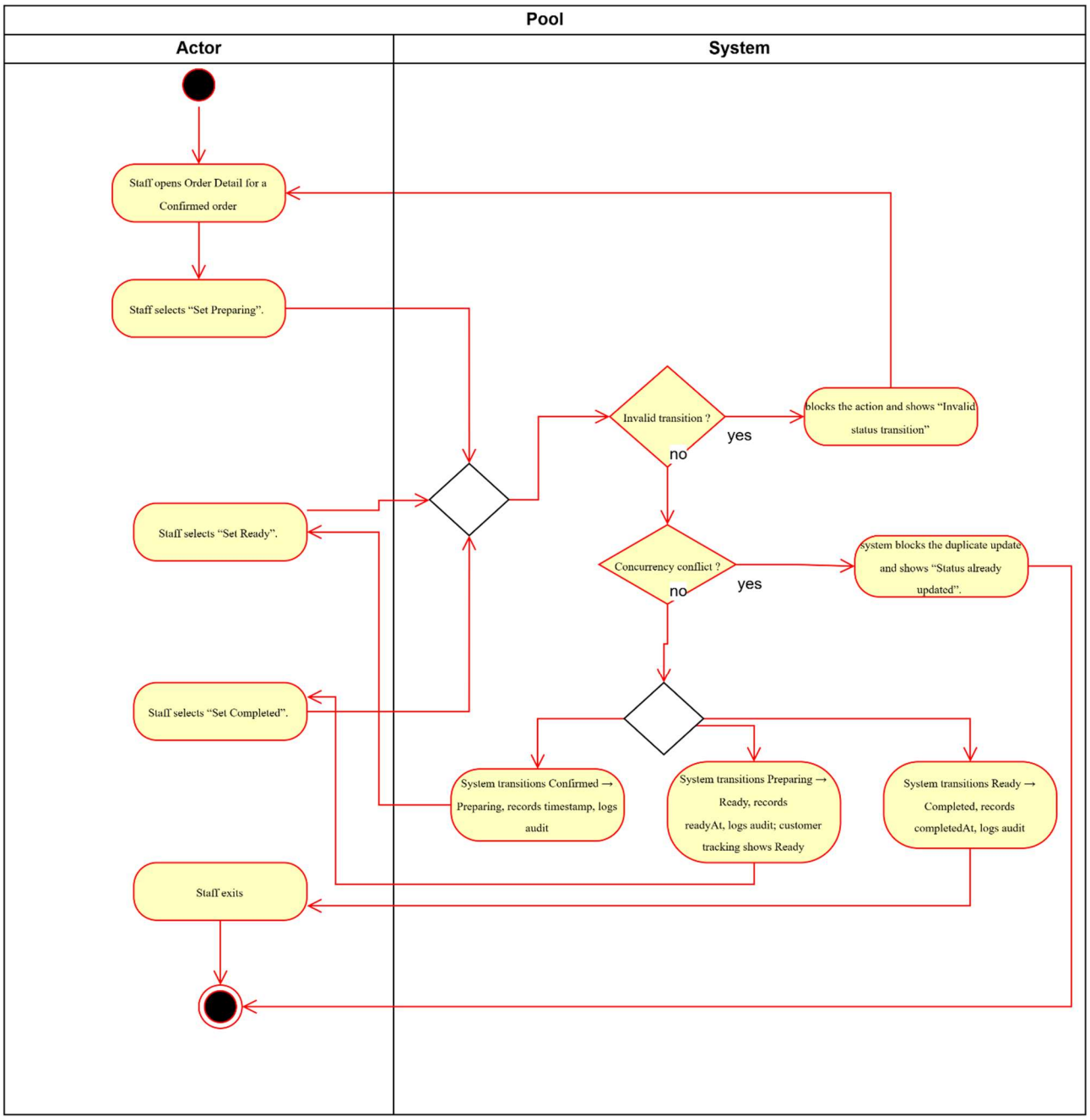


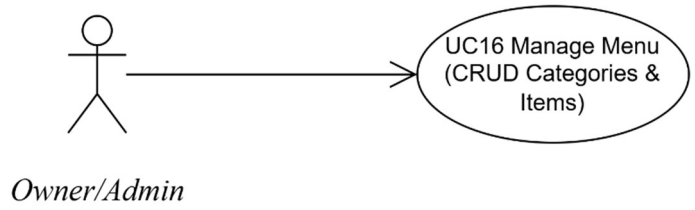
Use Case Number:	UC15	
Use Case Name:	Update Order Status (Confirmed → Preparing → Ready → Completed)	
Actor (s):	*Cashier/Operator*	
Maturity:	Focused	
Summary:	Staff updates order status through lifecycle; system blocks invalid transitions and records timestamps + audit.	
Basic Course of Events:	**Actor Action**	**System Response**
	1. Staff opens Order Detail for a Confirmed order.	
	2. Staff selects "Set Preparing". **E1, E2**	
		3. System transitions Confirmed → Preparing, records timestamp, logs audit.
	4. Staff selects "Set Ready". **E1, E2**	

		5. System transitions Preparing → Ready, records readyAt, logs audit; customer tracking shows Ready.
	6. Staff selects "Set Completed". **E1, E2**	
		7. System transitions Ready → Completed, records completedAt, logs audit.
	8. Staff exits. The use case ends here.	
Alternative Paths:		
Exception Paths:	**E1. Invalid transition** If the staff attempts an invalid transition (e.g., Ready → Preparing), the system blocks the action and shows "Invalid status transition". **Control returns to the same step** where the action was attempted (step 2 or 4 or 6).	
	E2. Concurrency conflict If another operator updates status first, the system blocks the duplicate update and shows "Status already updated". **Control returns to step 1**.	
Extension Points:	- Lock order while being edited (implementation choice) - Kitchen Display System integration (Out of scope MVP)	
Triggers:	Preparation progress changes; customer pickup handover occurs.	
Assumptions:	Only allowed transitions are permitted; timestamps/audit required for reporting KPIs.	
Preconditions:	Staff logged in; order is in a valid prior status for the intended transition.	
Post Conditions:	Order status is updated; relevant timestamp(s) and audit log are recorded; customer tracking reflects changes.	
Reference: Business Rules	BR-STAT-01, BR-STAT-02, BR-AUD-01, BR-CONC-01	
Author(s):	Business Analyst	

Date:	2026-01-07

Activity Diagram:

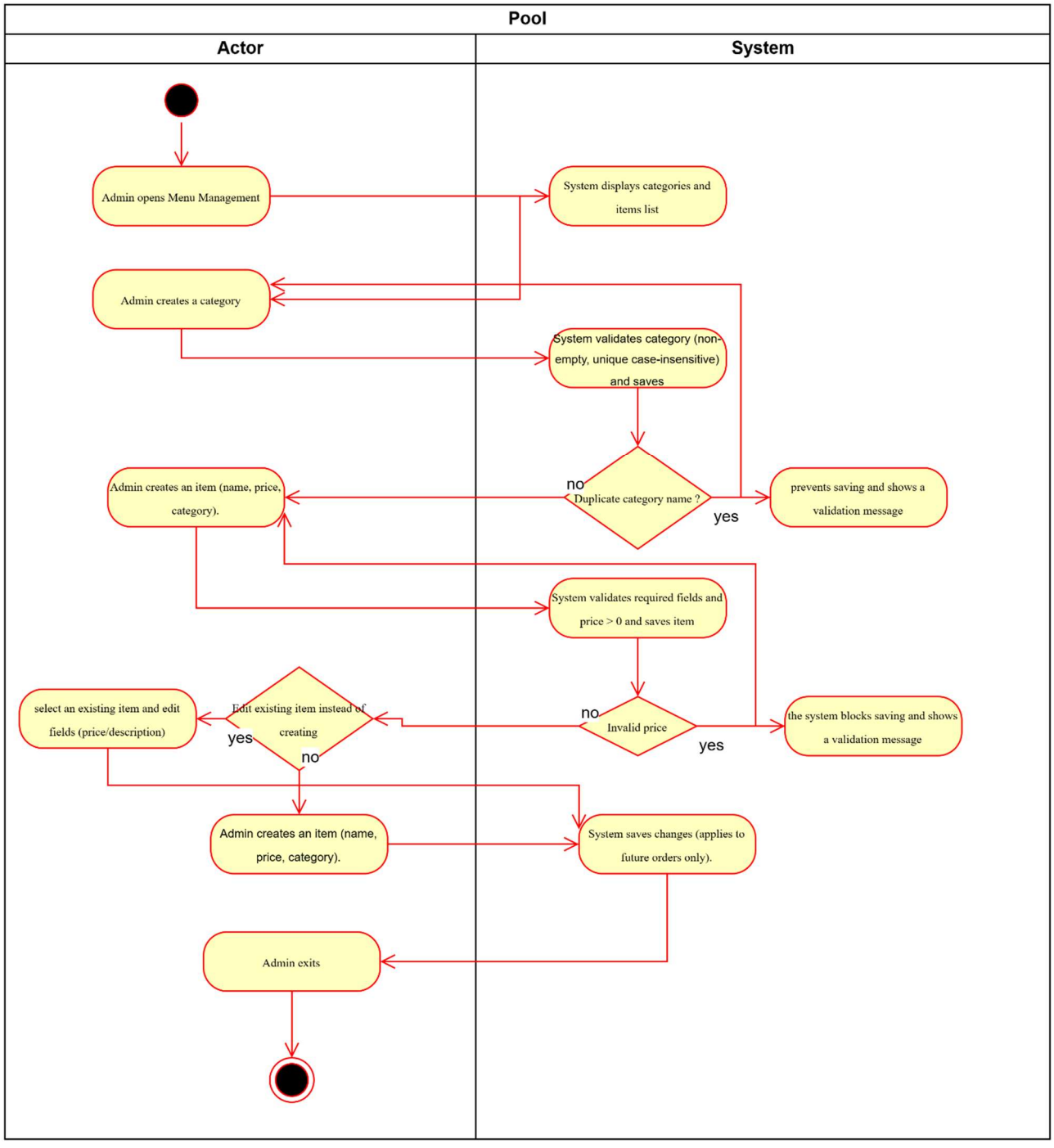




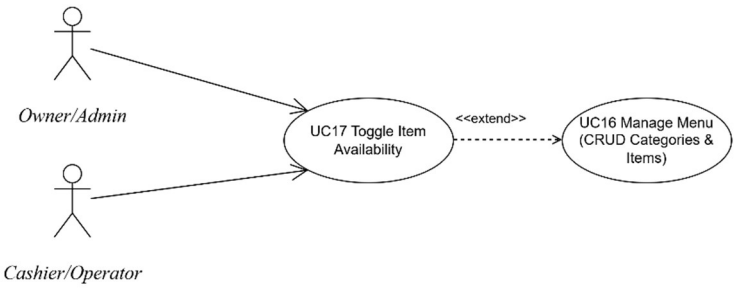
Use Case Number:	UC16	
Use Case Name:	Manage Menu (CRUD Categories & Items)	
Actor (s):	*Owner/Admin*	
Maturity:	Focused	
Summary:	Admin creates/edits categories and items; validates uniqueness and price rules; price snapshot protects historical totals.	
Basic Course of Events:	**Actor Action**	**System Response**
	1. Admin opens Menu Management.	
		2. System displays categories and items list.
	3. Admin creates a category.	
		4. System validates category (non-empty, unique case-insensitive) and saves. **E1**
	5. Admin creates an item (name, price, category).	

		6. System validates required fields and price > 0 and saves item. **E2**
	7. Admin creates an item (name, price, category).**A1**	
		8. System saves changes (applies to future orders only).
	9. Admin exits. The use case ends here.	
Alternative Paths:	**A1. Edit existing item instead of creating** From **step 7**, the admin may select an existing item and edit fields (price/description). After saving, the system returns to the item list. **Control returns to step 8**.	
Exception Paths:	**E1. Duplicate category name** If the category name already exists (case-insensitive), the system prevents saving and shows a validation message. **Control returns to step 3**.	
	E2. Invalid price If price ≤ 0, the system blocks saving and shows a validation message. **Control returns to step 5**.	
Extension Points:	- Upload item image (Optional) - Variants/toppings (Future)	
Triggers:	Admin needs to create/update menu for accuracy.	
Assumptions:	Category names are unique (case-insensitive); item price must be > 0.	
Preconditions:	Admin logged in and authorized for menu management.	
Post Conditions:	Menu is updated and reflected in customer app; historical order totals remain unchanged due to price snapshot rule.	
Reference: Business Rules	BR-PRICE-01, BR-RBAC-01	
Author(s):	Business Analyst	
Date:	2026-01-07	

Activity Diagram:

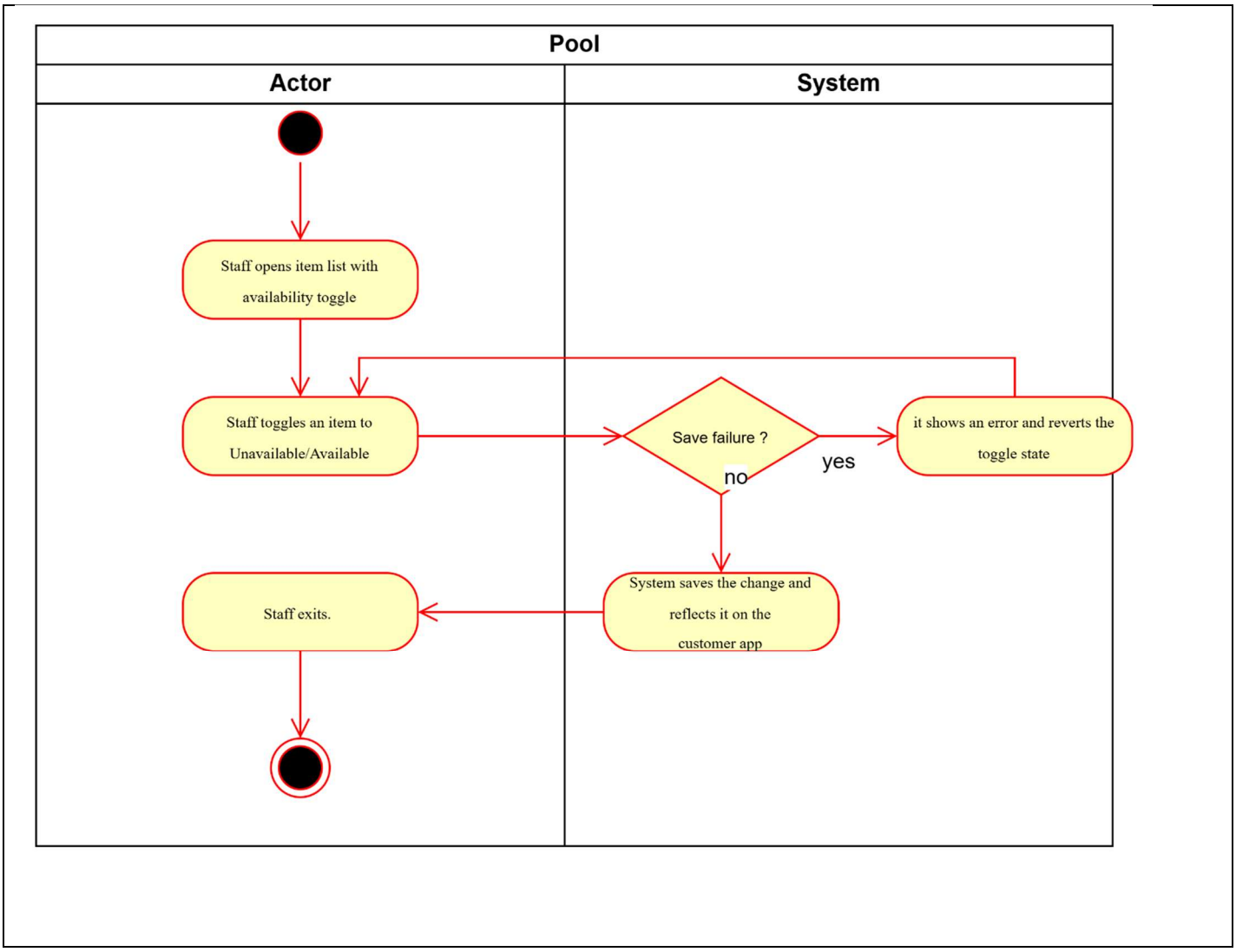


UC17- Toggle Item Availability Quickly

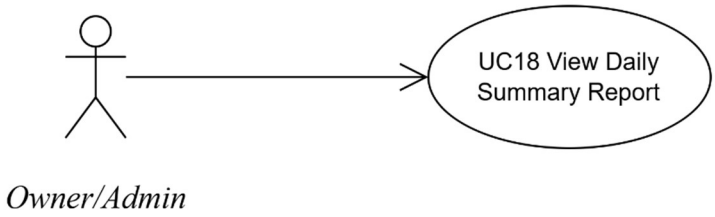


Use Case Number:	UC17	
Use Case Name:	Toggle Item Availability Quickly	
Actor (s):	*Cashier/Operator; Owner/Admin*	
Maturity:	Focused	
Summary:	Staff quickly toggles item availability to prevent customers from ordering out-of-stock items.	
Basic Course of Events:	**Actor Action**	**System Response**
	1. Staff opens item list with availability toggle.	
	2. Staff toggles an item to Unavailable/Available. **E1**	
		3. System saves the change and reflects it on the customer app.
	4. Staff exits. The use case ends here.	
Alternative Paths:		

Exception Paths:	**E1. Save failure** If the system cannot save the change, it shows an error and reverts the toggle state. **Control returns to step 2**.
Extension Points:	- Bulk availability toggle (Optional/Future) - Auto-unavailable via inventory (Future)
Triggers:	Item runs out during peak hours.
Assumptions:	Cashier/Operator is allowed to toggle availability (limited menu permission).
Preconditions:	Staff logged in; item exists.
Post Conditions:	Item availability is updated and reflected to customer app; unavailable items cannot be ordered.
Reference: Business Rules	BR-AVAIL-01, BR-RBAC-01
Author(s):	Business Analyst
Date:	2026-01-07
Activity Diagram:	

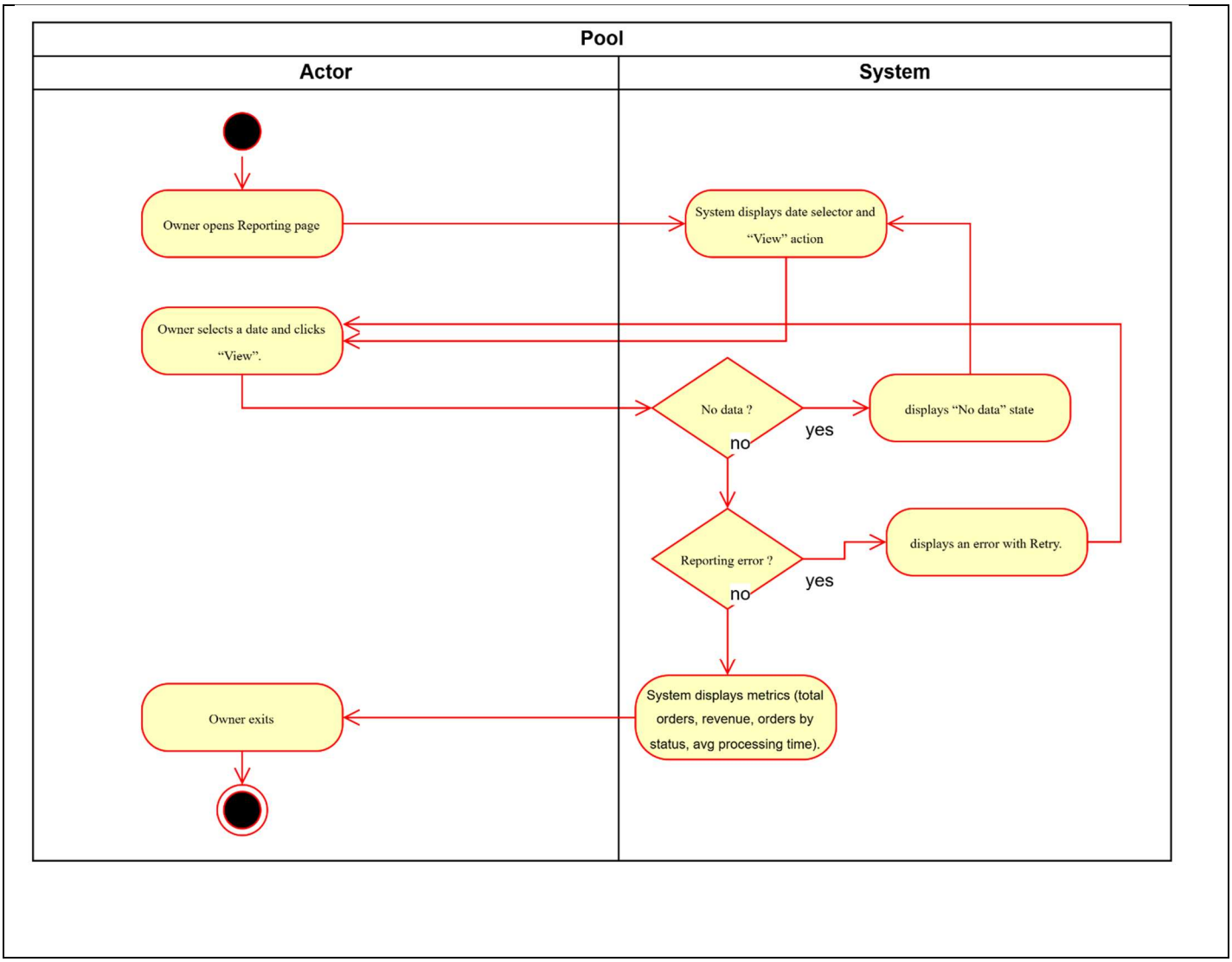


UC18- View Daily Summary Report

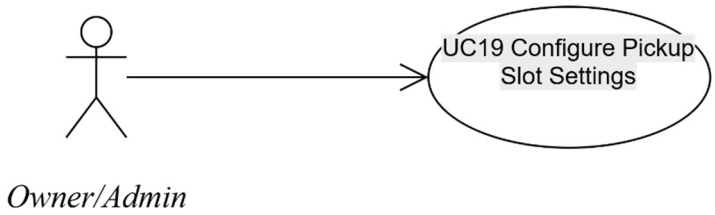


Use Case Number:	UC18	
Use Case Name:	View Daily Summary Report	
Actor (s):	*Owner/Admin*	
Maturity:	Focused	
Summary:	Owner views daily orders count, revenue, orders by status, and average processing time (Confirmed → Ready).	
Basic Course of Events:	**Actor Action**	**System Response**
	1. Owner opens Reporting page.	
		2. System displays date selector and "View" action.
	3. Owner selects a date and clicks "View". **E1, E2**	
		4. System displays metrics (total orders, revenue, orders by status, avg processing time).
	5. Owner exits. The use case ends here.	
Alternative Paths:		

Exception Paths:	**E1. No data** If there is no data for the selected date, the system displays “No data” state. **Control returns to step 2**.
	E2. Reporting error If reporting fails, the system displays an error with Retry. **Control returns to step 3**.
Extension Points:	- Export report (CSV/PDF) (Optional) - Peak-time distribution by pickup slot (Optional/Future)
Triggers:	Owner/Admin wants to review daily performance and KPIs.
Assumptions:	Processing time is measured primarily as Confirmed → Ready; timestamps are captured.
Preconditions:	Owner/Admin logged in and authorized for reporting.
Post Conditions:	Daily summary metrics are displayed for the selected date.
Reference: Business Rules	BR-PERF-01, BR-AUD-01
Author(s):	Business Analyst
Date:	2026-01-07
Activity Diagram:	

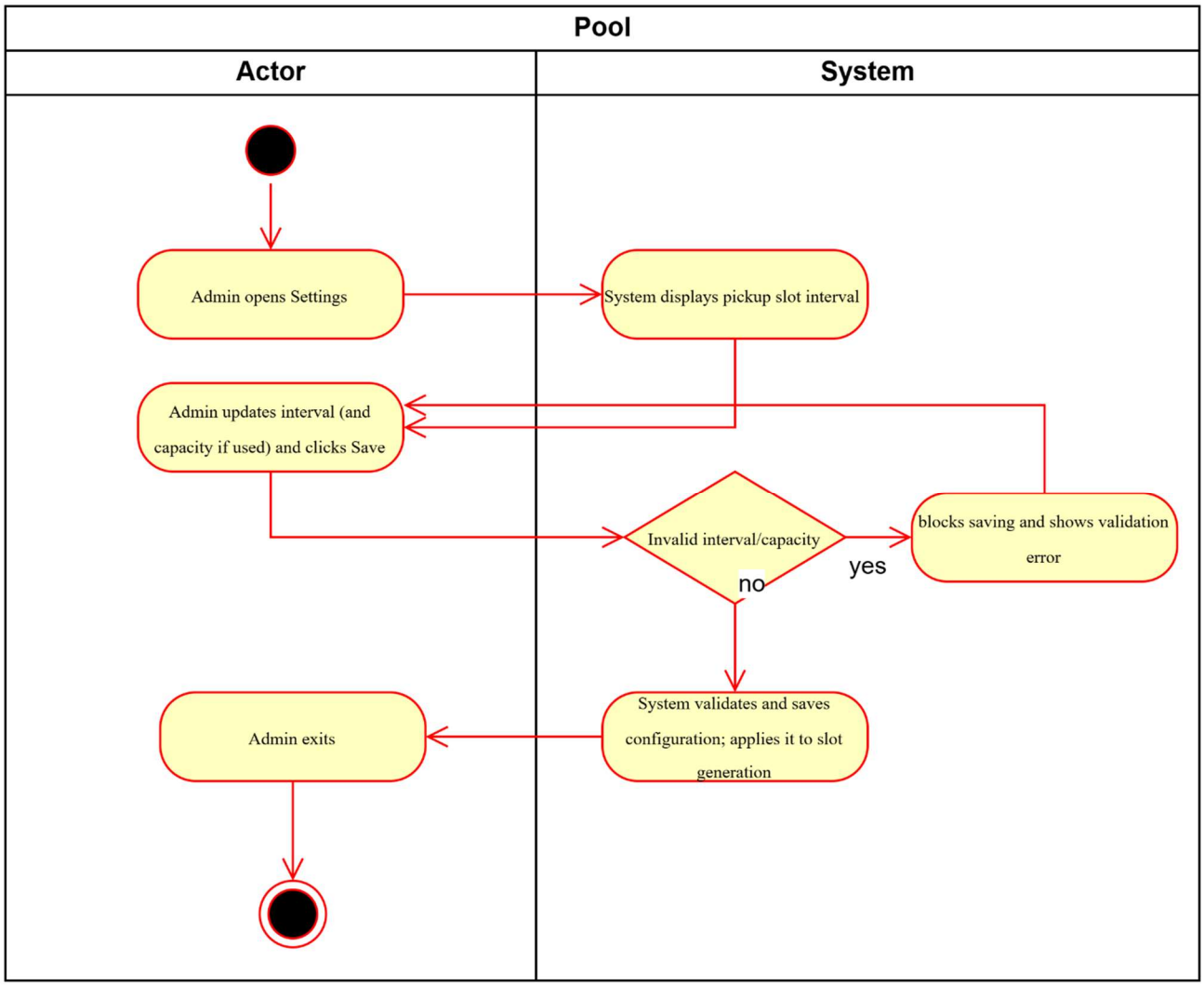

Pool
Actor
System
Owner opens Reporting page
System displays date selector and "View" action
Owner selects a date and clicks "View".
No data ?
yes
no
displays "No data" state
Reporting error ?
yes
no
displays an error with Retry.
System displays metrics (total orders, revenue, orders by status, avg processing time).
Owner exits

UC19- Configure Pickup Slot Settings (Minimal)



Use Case Number:	UC19	
Use Case Name:	Owner/Admin	
Actor (s):	*Owner/Admin*	
Maturity:	Focused	
Summary:	Admin configures pickup slot interval and optional capacity cap.	
Basic Course of Events:	**Actor Action**	**System Response**
	1. Admin opens Settings (Minimal).	
		2. System displays pickup slot interval (and capacity if enabled).
	3. Admin updates interval (and capacity if used) and clicks Save. **E1**	
		4. System validates and saves configuration; applies it to slot generation.

	5. Admin exits. The use case ends here.	
Alternative Paths:		
Exception Paths:	**E1. Invalid interval/capacity** If interval/capacity is invalid, the system blocks saving and shows validation error. **Control returns to step 3**.	
Extension Points:	Different rules by day-of-week/peak hours (Future)	
Triggers:	Owner/Admin needs to adjust slot interval/capacity for operational control.	
Assumptions:	Settings are minimal for MVP; changes apply to future slot generation.	
Preconditions:	Owner/Admin logged in; settings page available.	
Post Conditions:	Slot settings are saved; customer slot list follows the updated configuration.	
Reference: Business Rules	BR-SLOT-01, BR-SLOT-02	
Author(s):	Business Analyst	
Date:	2026-01-07	
Activity Diagram:		


Pool
Actor
System
Admin opens Settings
System displays pickup slot interval
Admin updates interval (and capacity if used) and clicks Save
Invalid interval/capacity
yes
blocks saving and shows validation error
no
System validates and saves configuration; applies it to slot generation
Admin exits