

Project overview

- The project is a rental management system consisting of separate user and admin panels. Admins are responsible for adding inventories, each identified with a unique barcode, which are displayed to users as artworks. Users can create multiple orders with defined pickup and return dates, and add inventories to these orders similar to a cart functionality, ensuring no duplicate inventories exist within the same order. Once users finalize their selections, they can request a hold for that order. The system allows a maximum of three hold requests for the same inventory across conflicting dates; any further requests mark the inventory as unavailable. Admins then review and confirm orders based on a first-come, first-served approach. When an order is confirmed, all inventories under that order become unavailable to other users until the order's return date.
- After a user requests a hold on an order, the admin reviews and approves the items individually. Only items with hold-request statuses can be approved: on-hold-request items become on-hold, second-hold-request items become second-hold, and third-hold-request items become third-hold. Items with any other status are not eligible for approval.

Default statuses

Order statuses

hold	working	cancelled	confirm	check-out	pack	pick-up
ship	out	returned	check-in	issue	in	rush-order

Order item statuses

on-hold-r equest	2nd-hold- request	3rd-hold- request	unavailable	available	cancelled	loss	out
on-hold	2nd-hold	3rd-hold	unavailable-until	confirmed	clean	damage	in

Task explanation

- We have provided sample JSON data containing 10 inventory records, 5 user records, 2 order records, and 1 order item record for your reference.
- For your reference we have already implemented the Order Creation API and the Add Inventory to Order API, which include the initial status calculation logic for inventories before the request hold stage.
- You need to work on creating the status management logic for the following four API flows:

1. /request-hold – Handles the scenario when a user requests a hold on an order. Define how the statuses of all inventories within that order will be updated based on existing conflicts and hold limits.

2. /remove-order-item – Manages the process when a user removes an inventory item from an order. Determine how this action affects the statuses of other conflicting orders that contain the same inventory item.

3. /update-order – Defines the logic when an admin updates the pickup or return date of an existing order. Based on the new dates, determine how the statuses of all inventories within that order will be recalculated in relation to other conflicting orders.

4. /confirm-order – Covers the flow when an admin confirms an order. Specify the validations that must be performed on the backend before confirmation (e.g., hold conflicts, date overlaps, or availability checks) and how the statuses of conflicting inventories in other orders will be updated accordingly.

- All the status management logic and workflow rules for different stages of the process are explained below.

1. Request Hold Status Calculation Explanation

- If it is the first request for a particular inventory item without any conflicting orders, the status for that order item will be: on-hold-request
- If it is the second request for the same item and there is a conflicting order with the same item having status on-hold or on-hold-request, the status for this order item will be: second-hold-request

- If it is the third request for the same item and conflicting orders exist with the same item having statuses on-hold, on-hold-request, second-hold, or second-hold-request, the status for this order item will be: third-hold-request
- If there are more than three requests for the same item and conflicting orders exist with statuses on-hold, on-hold-request, second-hold, second-hold-request, third-hold, or third-hold-request, the status for this order item will be: unavailable
- If there is any conflicting order with the same item having status confirmed, the status for this order item will be: unavailable-until

The unavailable_until date will be set to the end date of the conflicting confirmed order.

2. Delete Order Item Flow

Removing an item with status unavailable or unavailable-until:

- When an inventory item with these statuses is removed from an order, no changes are required for the status of the same item in conflicting orders, because these statuses indicate that the item is already blocked due to confirmed or over-limit holds.

Removing an item with status on-hold or on-hold-request:

- When such an item is removed, the system needs to recalculate the hold hierarchy for other conflicting orders:
- second-hold-request → on-hold-request: The second-level hold request is promoted to first-level.
- second-hold → on-hold: The second-level hold request is promoted to first-level.
- third-hold-request → second-hold-request: The third-level hold request moves up to second-level.
- third-hold → second-hold: The third-level hold request moves up to second-level.
- Any other lower-level holds (if present beyond the third request) are updated to available.

This ensures that the hold priorities are maintained correctly after removing an item.

Removing an item with status confirmed:

- Identify all conflicting orders that contain the same item with item status as unavailable or unavailable-until.
- Check whether these items are not in conflict with any other orders apart from the current confirmed order being removed.
- If there are no other conflicts, update the status of these items in conflicting orders to available, making them open for new holds or confirmations.

This flow ensures that inventory statuses remain consistent and hold priorities are properly maintained when an item is removed from an order.

3. Update Order Date Flow

Validation against confirmed conflicts:

- When the admin updates the pickup or return date of an order, if any conflicting order with confirmed status exists for the new dates, the update cannot be performed.

Handling conflicts with non-confirmed orders:

- If there are conflicts with other statuses (e.g., holds) but no confirmed conflicts, the system must recalculate the status of all items in the current order based on the new dates. The recalculated current and new statuses are then returned to the admin for review before proceeding.
- Consider below example for understanding

1. Old status before updating dates of Order 1

Order	Dates	Inventory (barcode)	Order item status
Order 1	1-7 to 7-7	10000001	on-hold
Order 2	9-7 to 15-7	10000001	on-hold

2. New status after updating dates of Order 1

Order	Dates	Inventory (barcode)	Order item status
Order 1	11-7 to 16-7	10000001	2nd-hold
Order 2	9-7 to 15-7	10000001	on-hold

Admin confirmation:

- Once the admin confirms the date update, the system updates the order dates and the statuses of all items in the order according to the recalculated values.

Status calculation:

- The new statuses are calculated using the same logic as the request hold flow, taking into account all conflicting orders and hold limits.

4. Confirm Order Flow

Validation before confirmation:

- The order can be confirmed only if all its items have status available or on-hold.

Updating confirmed order items:

- Once the order is confirmed, all items in that order are updated to confirm.

Updating conflicting orders:

- For all conflicting orders containing the same items, update their statuses to unavailable-until. Set oi_unavailable_until to the return date of the confirmed order, blocking these items until they are returned.
- Once you update an order item's status to unavailable-until, you need to find all conflicting items of the same inventory, excluding the current order and the updated item, and then update the statuses of those items based on **FIFO**, using oi_request_hold_at to determine their priority. This should be a recursive function which should go up to the last conflict found.
- In order to check for conflicting order items, an additional mandatory filter must be applied:

1. only consider order items that have been requested for hold.
2. status of order item should not be from below status :-

CANCELLED,
AVAILABLE,
IN,
UNAVAILABLE,
UNAVAILABLE_UNTIL

Example workflows

Example Scenario 1

1. User A adds Inventory X to Order 1
2. User B adds Inventory X to Order 2

order	dates	inventory (barcode)	Order item status
order 1	Jan 1 – Jan 5	10000011	available
order 2	Jan 3 – Jan 7	10000011	available

3. User A requests hold Order 1

order	dates	inventory (barcode)	Order item status
order 1	Jan 1 – Jan 5	10000011	on-hold-request
order 2	Jan 3 – Jan 7	10000011	available

4. User B requests hold Order 2

order	dates	inventory (barcode)	Order item status
order 1	Jan 1 – Jan 5	10000011	on-hold-request
order 2	Jan 3 – Jan 7	10000011	2nd-hold-request

5. Admin approves Order 1

order	dates	inventory (barcode)	Order item status
order 1	Jan 1 – Jan 5	10000011	on-hold
order 2	Jan 3 – Jan 7	10000011	2nd-hold-request

6. Admin confirms Order 1

order	dates	inventory (barcode)	Order item status
order 1	Jan 1 – Jan 5	10000011	confirmed
order 2	Jan 3 – Jan 7	10000011	unavailable-until (Jan 5)

7. User A deletes item from Order 1

order	dates	inventory (barcode)	Order item status
order 1	Jan 1 – Jan 5	10000011	cancelled
order 2	Jan 3 – Jan 7	10000011	available

Example Scenario 2

order	dates	inventory (barcode)	Order item status
order 1	1-7 to 10-7	10000012	on-hold-request
order 2	7-7 to 15-7	10000012	2nd-hold-request
order 3	11-7 to 18-7	10000012	2nd-hold-request
order 4	15-7 to 21-7	10000012	3rd-hold-request
order 5	19-7 to 21-7	10000012	2nd-hold-request

Additional scenarios for the understanding of the flow

Example of hold statuses when an item is removed from a set list that has a hold on it:

1. Order 1 has rental dates **6/1-6/10** has an individual item **ON HOLD**.
2. Order 2 has rental dates **6/5-6/15** has the same individual item and requests hold. The individual item status that would be approved would be **2nd HOLD**.
3. Order 3 has rental dates **6/9-6/19** has the same individual item and requests hold. The individual item status that would be approved would be **3rd HOLD**.
4. Order 1 releases the item therefore releasing their hold, Order 2 now moves up to **ON HOLD** and Order 3 now moves up to **2nd HOLD**.

Example 2 of hold statuses when an item is removed from a set list that has a hold on it:

1. Order 1 has rental dates **6/1-6/10** has an individual item **ON HOLD**.
2. Order 2 has rental dates **6/5-6/15** has the same individual item and requests hold. The individual item status that would be approved would be **2nd HOLD**.
3. Order 3 has rental dates **6/9-6/19** has the same individual item and requests hold. The individual item status that would be approved would be **3rd HOLD**.
4. Order 1 AND Order 2 both release the item therefore releasing their holds, Order 3 now moves up to **ON HOLD**.

Example 3 of hold statuses when an item is removed from a set list that has a hold on it:

1. Order 1 has rental dates **6/1-6/10** has an individual item **ON HOLD**.
2. Order 2 has rental dates **6/5-6/15** has the same individual item and requests hold. The individual item status that would be approved would be **2nd HOLD**.
3. Order 3 has rental dates **6/9-6/19** has the same individual item and requests hold. The individual item status that would be approved would be **3rd HOLD**.
4. Order 2 releases the item therefore releasing their hold. Order 1 maintains their **ON HOLD** status while Order 3 now moves up to **2nd HOLD**.

Example of hold statuses when an item is confirmed for a 1st hold:

1. Order 1 has rental dates **6/1-6/10** has an individual item **ON HOLD**.
2. Order 2 has rental dates **6/5-6/15** has the same individual item and requests hold. The individual item status that would be approved would be **2nd HOLD**.
3. Order 3 has rental dates **6/9-6/19** has the same individual item and requests hold. The individual item status that would be approved would be **3rd HOLD**.
4. Order 1 confirms the item in their set list. This item is now **UNAVAILABLE UNTIL 6/11** and the item now shows with this status in Order 2 and Order 3's set lists on both the user end and in the admin panel since they both have overlapping dates, automatically cancelling out their hold positions.

Example 2 of hold statuses when an item is confirmed for a 1st hold:

1. Order 1 has rental dates **6/1-6/10** has an individual item **ON HOLD**.
2. Order 2 has rental dates **6/5-6/15** has the same individual item and requests hold. The individual item status that would be approved would be **2nd HOLD**.
3. Order 3 has rental dates **6/11-6/21** has the same individual item and requests hold. The individual item status that would be approved would be **2nd HOLD** because while Order 3 does not overlap with Order 1, it does overlap with Order 2.
4. Order 4 has rental dates **6/15-6/21** has the same individual item and requests hold. The individual item status that would be approved would be **3rd HOLD** because Order 4 overlaps with Order 2 and Order 3.
5. Order 1 confirms the item in their set list. This item is now **UNAVAILABLE UNTIL 6/11** and the item now shows with this status for Order 2 since it has overlapping dates with Order 1, automatically cancelling out its hold position. Order 3 gets moved up to **ON HOLD** since Order 2's hold status has been removed and Order 4 gets moved up to **2nd HOLD**.

