

Original Actors and Detailed Use Cases

Subscriber

Create account	2
Modify account	2
Update order	2
Preview selection	2
View delivery history	3
Add delivery note	3
Rate wine	3
View wine detail	4
Change delivery day	4
Return	4
Cancellation	4
Exchange	5

Delivery partner

Get deliveries	5
----------------------	---

Delivery employee

Confirm delivery	5
------------------------	---

Club owner

Get Number of subscribers	6
Get # of new subscribers	6
Get # of cancelled subscribers	6
Get cancellation rate	6
Get amount of money earned	7
Change price of wine	7
Change delivery price	7
Change wine selection	7
Edit wine information	8
Get wine rating	8

Customer Service Employee

Get subscriber's account	8
Get subscriber's history	8
Get pending deliveries	9
Get wine rating	9
Update subscriber's account	9
Modify subscriber's delivery	10

SUBSCRIBER – Create Account

Data: <name>, <street-name>, <city>, <state>, <zip-code>, <email-address>, <phone-number>, [<twitter-name>], [<facebook-name>]

Main Course of Events

- User issues “Create Account” order with the above data
- System validates data
- System creates a record for the subscriber and assigns a subscriber-id to it
- System returns subscriber-id to user

Exception Course

- System delivers error message to user

Implemented as createSubscriber in SubscriberAction.java

SUBSCRIBER – Modify Account

Data: <subscriber-id>, <name>, <street-name>, <city>, <state>, <zip-code>, <emailaddress>, <phone-number>, [<twitter-name>], [<facebook-name>]

Main Course of Events

- User issues “Modify Account” order with the above data
- System validates data
- System retrieves the record corresponding to <subscriber-id>
- System replaces existing customer data with the new data
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implemented as updateSubscriber in SubscriberAction.java

SUBSCRIBER – Update Order

Data: <subscriber-id>, <wine mix>, <size of order>

Main Course of Events

- User issues “Update Order” order with the above data
- System validates data
- System retrieves the record corresponding to <subscriber-id>
- System replaces basic wine mix (RW x1) with the new data
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implemented as updateShipmentDetails in SubscriberAction.java but without <size of order>

SUBSCRIBER – Preview Selection

Data: <RW wine mix>, <AR wine mix>, <AW wine mix>,

Main Course of Events

- User issues “Preview Selection” order

- System validates data
- System retrieves and returns the data

Exception Course

- System delivers error message to user

Implemented as `getShipmentDetailForSub` in `SubscriberAction.java`

SUBSCRIBER – View Delivery History

Data: <subscriber-id>, <delivery history>

Main Course of Events

- User issues “View Delivery History” order
- System validates data
- System retrieves and returns the <delivery history> record corresponding to <subscriber-id>

Exception Course

- System delivers error message to user

Implemented as `getShipmentsForSubDetail` in `SubscriberAction.java`

SUBSCRIBER – Add delivery note

Data: <subscriber-id>, <delivery history>, <delivery date>, <delivery note>

Main Course of Events

- User issues “Add Delivery Note” order
- System validates data
- System retrieves the <delivery history> record corresponding to <subscriber-id>
- System inserts <delivery note> to <delivery history> corresponding to <delivery date>
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implemented as `addShipmentNote` in `SubscriberAction.java`

SUBSCRIBER – Rate wine

Data: <wine name>, <wine score>

Main Course of Events

- User issues “Rate Wine” order with the above data
- System validates data
- System retrieves the record corresponding to <wine name>
- System calculates new <wine average score> using the <wine score> provided
- System returns success or failure to user

Exception Course

- System delivers error message to user

Not implemented

SUBSCRIBER – View wine detail

Data: <wine name>

Main Course of Events

- User issues “View Wine Detail” order with the above data
- System validates data
- System retrieves and returns the record corresponding to <wine name>
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implemented as `getWine` and `getWines` in `SubscriberAction.java`

SUBSCRIBER – Change Delivery Day

Data: <subscriber-id>, <delivery information>, <new delivery day>

Main Course of Events

- User issues “Change Delivery Day” order with the above data
- System validates data
- System retrieves the <delivery information> record corresponding to <subscriber-id>
- System updates <delivery day> with <new delivery day>
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implemented as `updateDelivery` in `SubscriberAction.java`

SUBSCRIBER – Return Order

Data: <subscriber-id>, <delivery history>, <order number>, <return reason>

Main Course of Events

- User issues “Return Order” order
- System retrieves the <delivery history> record corresponding to <subscriber-id>
- System updates <order number> as a return
- System includes <return reason> in <delivery history>
- System processes return payment for <subscriber-id>
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implementation no longer required

SUBSCRIBER –Order Cancelation

Data: <subscriber-id>, <delivery information >

Main Course of Events

- User issues “Order Cancelation” order
- System retrieves the <delivery information> record corresponding to <subscriber-id>

- System cancels the next order to be delivered
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implementation no longer required

SUBSCRIBER –Order Exchange

Data: <subscriber-id>, <delivery history>, <order number>, <exchange reason>, <wine exchanged>, <wine received>

Main Course of Events

- User issues “Order Exchange” order
- System retrieves the <delivery history> record corresponding to <subscriber-id>
- System updates <order number> as an exchange
- System includes <exchange reason> in <delivery history>
- System processes return of <wine exchanged>
- System processes order of <wine received>
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implementation no longer required

DELIVERY PARTNER – Get Deliveries

Data: <deliveries>, <subscriber info>

Main Course of Events

- User issues “Get Deliveries” order
- System retrieves the <deliveries> and corresponding <subscriber info>
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implemented as getDeliveries in DeliveryAction.java

DELIVERY EMPLOYEE – Confirm Delivery

Data: <deliveries>, <subscriber info>

Main Course of Events

- User issues “Confirm Deliveries” order
- System retrieves the <deliveries> and corresponding <subscriber info>
- User updates delivery status as confirmed
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implemented as addReceipt in DeliveryAction.java

CLUB OWNER – Get Number of Subscribers

Data: <subscriber info>

Main Course of Events

- User issues “Get Number of Subscribers” order
- System retrieves the <subscriber info>
- System returns success or failure to user

Exception Course

- System delivers error message to user

Not implemented

CLUB OWNER – Get Number of New Subscribers

Data: <subscriber info>, <time period>

Main Course of Events

- User issues “Get Number of New Subscribers” order
- System retrieves the <subscriber info>
- System checks how many subscribers joined during <time period>
- System returns success or failure to user

Exception Course

- System delivers error message to user

Not implemented

CLUB OWNER – Get Number of Cancelled Subscribers

Data: <subscriber info>

Main Course of Events

- User issues “Get Number of Cancelled Subscribers” order
- System retrieves the <subscriber info>
- System checks how many subscribers cancelled their membership
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implementation no longer required

CLUB OWNER – Get Cancellation Rate

Data: <subscriber info>, <time period>

Main Course of Events

- User issues “Get Cancellation Rate” order
- System retrieves the <subscriber info>
- System checks how many subscribers canceled during <time period>
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implementation no longer required**CLUB OWNER – Get Amount of Money Earned**

Data: <subscriber info>, <prices>

Main Course of Events

- User issues “Get Amount of Money Earned” order
- System retrieves the <subscriber info>
- System checks how many order have been filled and at what prices
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implemented as `getRevenue` in `AdminAction.java`

CLUB OWNER – Change Price of Wine

Data: <prices>

Main Course of Events

- User issues “Change Price of Wine” order
- System retrieves the <prices>
- User updates them with new values
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implementation no longer required

CLUB OWNER – Change Delivery Price

Data: <prices>

Main Course of Events

- User issues “Change Delivery Price” order
- System retrieves the <prices>
- User updates them with new values
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implementation no longer required

CLUB OWNER – Change Wine Selection

Data: <wine selection>

Main Course of Events

- User issues “Change Wine Selection” order
- System retrieves the <wine selection>

- User updates them with new information
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implemented as addMonthlySelection in AdminAction.java

CLUB OWNER – Edit Wine Information

Data: <wines>

Main Course of Events

- User issues “Edit Wine Information” order
- System retrieves the <wines>
- User updates them with new information
- System returns success or failure to user

Exception Course

- System delivers error message to user

Not implemented

CLUB OWNER – Get Wine Rating

Data: <wines>

Main Course of Events

- User issues “Get Wine Rating” order
- System retrieves the <wines >
- System retrieves scores for each wine
- System returns success or failure to user

Exception Course

- System delivers error message to user

Not implemented

CUSTOMER SERVICE EMPLOYEE – Get Subscriber’s Account

Data: <subscriber-id>,

Main Course of Events

- User issues “Get Subscriber’s Account” order
- System retrieves the subscriber information corresponding to <subscriber-id>
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implemented as findSubById in SubscriberAction.java

CUSTOMER SERVICE EMPLOYEE – Get Subscriber’s History

Data: <subscriber-id>,

Main Course of Events

- User issues “Get Subscriber’s History” order
- System retrieves the subscriber history corresponding to <subscriber-id>
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implemented as getShipmentsForSub in SubscriberAction.java

CUSTOMER SERVICE EMPLOYEE – Get Pending Deliveries

Data: <deliveries>,

Main Course of Events

- User issues “Get Pending Deliveries” order
- System retrieves the <deliveries> information
- System checks the status of each delivery
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implemented as getDeliveries in DeliveryAction.java

CUSTOMER SERVICE EMPLOYEE – Get Wine Rating

Data: <wines>

Main Course of Events

- User issues “Get Wine Rating” order
- System retrieves the <wines >
- System retrieves scores for each wine
- System returns success or failure to user

Exception Course

- System delivers error message to user

Not implemented

CUSTOMER SERVICE EMPLOYEE – Update Subscriber’s Account

Data: <subscriber-id>, <name>, <street-name>, <city>, <state>, <zip-code>, <emailaddress>, <phone-number>, [<twitter-name>], [<facebook-name>]

Main Course of Events

- User issues “Update Subscriber’s Account” order with the above data
- System validates data
- System retrieves the record corresponding to <subscriber-id>
- System replaces existing customer data with the new data
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implemented as updateSubscriber in SubscriberAction.java

CUSTOMER SERVICE EMPLOYEE – Modify Subscriber’s Delivery

Data: <subscriber-id>, <delivery information>, <new delivery day>

Main Course of Events

- User issues “Modify Subscriber’s Delivery” order with the above data
- System validates data
- System retrieves the <delivery information> record corresponding to <subscriber-id>
- System updates <delivery day> with <new delivery day>
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implemented as updateDelivery in SubscriberAction.java

Actual Actors and Detailed Use Cases (those strikethrough are detailed above)

SUBSCRIBER

Create Subscriber	2
Update Subscriber	2
Find Sub by Id	8
Get Shipments for Sub	8
Get Shipment Detail for Sub	3
Update Shipment Details	2
Get Shipment Notes	12
Add Shipment Note	3
Get Shipment Note	12
Update Shipment Note	12
Delete Shipment Note	12
Get Wines	4
Get Wine	4
Get Wine Notes	13
Add Wine Note	13
Get Wine Note	13
Update Wine Note	13
Delete Wine Note	14
Get Delivery	14
Update Delivery	4

ADMIN

Create Admin	14
Get Admins	14
Update Account	15
Get Admin	15
Get Revenue	7
Get Monthly Selections	15
Add Monthly Selection	7
Get Monthly Selection	15

DELIVERY

Get Deliveries	5
Add Receipt	5
Get Receipts	16
Get Receipt	16

SUBSCRIBER – Get Shipment Notes

Data: <subscriber-id>, <shipment-id>

Main Course of Events

- User issues “Get Shipment Notes” order with the above data
- System retrieves the all the notes corresponding to <subscriber-id> and <shipment-id>
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implemented

SUBSCRIBER – Get Shipment Note

Data: <subscriber-id>, <shipment-id>, <note-id>

Main Course of Events

- User issues “Get Shipment Note” order with the above data
- System retrieves the specific note corresponding to <note-id>
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implemented

SUBSCRIBER – Update Shipment Note

Data: <subscriber-id>, <shipment-id>, <note-id>, <new note>

Main Course of Events

- User issues “Update Shipment Note” order with the above data
- System retrieves the specific note corresponding to <note-id>
- System updates the note with <new note> data
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implemented

SUBSCRIBER – Delete Shipment Note

Data: <subscriber-id>, <shipment-id>, <note-id>, <new note>

Main Course of Events

- User issues “Delete Shipment Note” order with the above data
- System deletes the specific note corresponding to <note-id>
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implemented

SUBSCRIBER – Get Wine Notes

Data: <subscriber-id>, <wine-id>

Main Course of Events

- User issues “Get Wine Notes” order with the above data
- System retrieves all the notes corresponding to <subscriber-id> and <wine-id>
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implemented

SUBSCRIBER – Add Wine Note

Data: <subscriber-id>, <wine-id>, <new note>

Main Course of Events

- User issues “Add Wine Note” order with the above data
- System retrieves the wine data corresponding to <subscriber-id> and <wine-id>
- System adds note <new note> to <wine-id> and generates <note-id>
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implemented

SUBSCRIBER – Get Wine Note

Data: <subscriber-id>, <wine-id>, <note-id>

Main Course of Events

- User issues “Get Wine Note” order with the above data
- System retrieves the notes corresponding to <note-id>
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implemented

SUBSCRIBER – Update Wine Note

Data: <subscriber-id>, <wine-id>, <note-id>, <new note>

Main Course of Events

- User issues “Update Wine Note” order with the above data
- System retrieves the note corresponding to <note-id>
- System updates note to <new note>
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implemented

SUBSCRIBER – Delete Wine Note

Data: <subscriber-id>, <wine-id>, <note-id>

Main Course of Events

- User issues “Delete Wine Note” order with the above data
- System removes the note corresponding to <note-id>
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implemented

SUBSCRIBER – Get Delivery

Data: <subscriber-id>

Main Course of Events

- User issues “Get Delivery” order with the above data
- System retrieves the delivery corresponding to <subscriber-id>
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implemented

ADMIN – Create Admin

Data: <admin-name>

Main Course of Events

- User issues “Create Admin” order with the above data
- System creates new Admin record and generates <admin-id>
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implemented

ADMIN – Get Admins

Data:

Main Course of Events

- User issues “Get Admins” order with the above data
- System returns names and IDs of all admins
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implemented

ADMIN – Update Account

Data: <admin-id>, <admin-name>

Main Course of Events

- User issues “Update Admin” order with the above data
- System retrieves admin data and updates name with new <admin-name>
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implemented

ADMIN – Get Admin

Data: <admin-id>

Main Course of Events

- User issues “Get Admin” order with the above data
- System retrieves admin information corresponding to <admin-id>
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implemented

ADMIN – Get Monthly Selections

Data:

Main Course of Events

- User issues “Get Monthly Selections” order with the above data
- System retrieves all monthly selections
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implemented

ADMIN – Get Monthly Selection

Data: <selection-id>

Main Course of Events

- User issues “Get Monthly Selection” order with the above data
- System retrieves monthly selection that corresponds to <selection-id>
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implemented

DELIVERY – Get Receipts

Data:

Main Course of Events

- User issues “Get Receipts” order with the above data
- System retrieves all receipts
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implemented

DELIVERY – Get Receipt

Data: <receipt-id>

Main Course of Events

- User issues “Get Receipt” order with the above data
- System retrieves receipt that corresponds to <receipt-id>
- System returns success or failure to user

Exception Course

- System delivers error message to user

Implemented