

Portfolio Management System

Requirement

As an ABC customer, the customer wants to book trading and track his asset allocations by viewing the portfolio summary , portfolio performance and trading booked on a month-by-month basis.

To enable this, we need to create an application that gives the customer a full view of his / her bookings on trading.

Tech requirement

- Follow TDD
- Consider security aspects and customer data protection.
- Proper logging and tracing needs to be done.
- Proper Exception handling needs to be done
- Ensure 99.9% uptime for the application and the application is performant .

Module 1: Portfolio Summary

The application should provide a Portfolio Summary screen where the customer views his / her portfolio details , asset allocations , holdings and performance bar chart based on search filter. The application should perform a basic input validation on the search filter before retrieving the details from the database.

1. Requirement:-
 - a. Build a user interface, which accepts Order Ref No., Security Name, Transaction Type and From & To Date from the customer as a search filter to retrieve the Portfolio Summary of the customer.
 - b. Validate the input fields and throw error message in case of validation failure.
 - c. Upon successful validation, save the Audit Details and retrieves the Account Summaries from respective tables.
 - d. Add a tab in portfolio summary to view transaction History based on portfolio
 - e. Populate Account Running Balance as \$10,000 by default.
2. Input Fields:-
 - a. transaction Ref No.
 - b. Security Name
 - c. Transaction Type
 - d. From & To Date
3. Input Validation:-

Validation	Expectation	Message
Validate Order Ref No.	Existing Order Ref No.	Invalid Order Ref No.
Validate Security Name	Existing Security Name	Invalid Security Name
Validate Transaction Type	Valid Transaction Type [Buy or Sell]	Invalid Transaction Type
Validate From & To Date	From Date should be before To Date To Date should be after From Date	Invalid From Date or To Date

4. Data Fields:-
 - a. Order Date
 - b. Order Ref No.
 - c. fund Name
 - d. Transaction Type
 - e. Credit

- f. Debit
- g. Running Balance
- 5. Tables involved:-
 - a. ORDER_DETAIL
 - b. PORTFOLIO_DETAILS
 - c. ASSET_DETAILS
 - d. AUDIT_ACTION

Module 3: Order Entry

The platform should provide an Order Entry screen where the customer books the trade and chooses as fund (Note : Funds needs to shown on screen every time an order needs to be created) for purchase or redeem. The platform should perform a basic input validation on the input fields and perform balance check before allowing the customer to book the trade.

For booking the trade they need to integrate with a mock legacy application which can only process one trade at a time , the application that will develop needs to ensure multiple transaction process will have a sla of 1 sec and the performance will not get impacted due to the constraint in legacy application .

1. Requirement:-
 - a. Build a user interface, which accepts fund Name, Transaction Type and Quantity from the customer and computes the Order Value to book for the trade.
 - b. Validate the input fields and throw error message in case of validation failure.
 - c. Upon successful validation, save the Audit Details and Order Entries into respective tables.
 - d. Order Status Workflow upon submission, Submitted -> Cancelled or Submitted -> Executed -> Completed or Submitted -> Failed.
 - e. While developing the application consider proper exception handling if the legacy application goes down .
 - f.
2. Input Fields:-
 - a. fund Name
 - b. Transaction Type
 - c. Quantity
 - d. Order Value
3. Input Validation:-

Validation	Expectation	Message
Validate Security Name	Existing Security Name	Invalid Security Name
Validate Transaction Type	Valid Transaction Type [Buy or Sell]	Invalid Transaction Type
Validate Quantity	Non-negative value and must be more than 0	Invalid Quantity

4. Tables involved:-
 - a. ORDER_DETAIL
 - e. PORTFOLIO_DETAILS
 - f. ASSET_DETAILS
 - b. AUDIT_ACTION

Module 4: Transaction History

The platform should provide an transaction history screen where the customer views his / her Order Histories based on search filter. There is a Reporting system which generates transaction history report every hour from the same tables

1. Requirement:-

- a. Build a user interface, which accepts Portfolio No or date range or Transaction Type and Order Status from the customer as a search filter to retrieve the Order Histories of the customer.
 - b. Search with at least one search filter and throw error message if not supplied.
 - c. Validate the input fields and throw error message in case of validation failure.
 - d. Upon successful validation, save the Audit Details and retrieves the order histories from respective tables.
2. Input Fields:-
- a. Order Ref No.
 - b. Security Name
 - c. Transaction Type
 - d. Order Status
 - e. From & To Date
3. Input Validation:-

Validation	Expectation	Message
Validate Order Ref No.	Existing Order Ref No.	Invalid Order Ref No.
Validate Security Name	Existing Security Name	Invalid Security Name
Validate Transaction Type	Valid Transaction Type [Buy or Sell]	Invalid Transaction Type
Validate Order Status	Valid Order Status [Submitted / Cancelled / Executed / Completed / Failed]	Invalid Order Status
Validate From & To Date	From Date should be before To Date To Date should be after From Date	Invalid From Date or To Date

4. Data Fields:-
- a. Order Ref No.
 - b. Security Name
 - c. Transaction Type
 - d. Order Status
 - e. Order Date
 - f. Quantity
 - g. Order Value
5. Tables involved:-
- a. ORDER_DETAIL
 - b. ASSET_DETAIL
 - c. AUDIT_ACTION

Database Design

Table Name: USER_DETAIL

Column	Datatype	Remark
ID	INT	PK
FIRST_NAME	VARCHAR	
LAST_NAME	VARCHAR	
EMAIL_ADDRESS	VARCHAR	
CREATED_ON	TIMESTAMP	
CREATED_BY	VARCHAR	
MODIFIED_ON	TIMESTAMP	
MODIFIED_BY	VARCHAR	

Table Name: USER_LOGIN_DETAIL

Column	Datatype	Remark
--------	----------	--------

ID	INT	PK
ID_USER_DETAIL	INT	FK - USER_DETAIL.ID
FIRST_NAME	VARCHAR	
LAST_NAME	VARCHAR	
EMAIL_ADDRESS	VARCHAR	
USER_STATUS	VARCHAR	
CREATED_ON	TIMESTAMP	
CREATED_BY	VARCHAR	
MODIFIED_ON	TIMESTAMP	
MODIFIED_BY	VARCHAR	

Table Name: ORDER_DETAIL

Column	Datatype	Remark
ID	INT	PK
ID_SECURITY_DETAIL	INT	FK - SECURITY_DETAIL.ID
ORDER_REF_NO	VARCHAR	
ORDER_STATUS	VARCHAR	
TRANSACTION_TYPE	VARCHAR	
ORDER_VALUE	VARCHAR	
CREATED_ON	TIMESTAMP	
CREATED_BY	INT	FK - USER_LOGIN_DETAIL.ID

Table Name: ACCOUNT_DETAIL

Column	Datatype	Remark
ID	INT	PK
ID_USER_LOGIN_DETAIL	INT	FK - USER_LOGIN_DETAIL.ID
CREDIT	INT	
DEBIT	INT	
RUNNING_BALANCE	INT	DEFAULT \$10,000
ID_ORDER_DETAIL	INT	FK - ORDER_DETAIL.ID
CREATED_ON	TIMESTAMP	
CREATED_BY	INT	FK - USER_LOGIN_DETAIL.ID

Table Name: SECURITY_DETAIL

Column	Datatype	Remark
ID	INT	PK
SECURITY_NAME	INT	FK - SECURITY_DETAIL.ID
VALUE	INT	

Table Name: AUDIT_USER_LOGIN

Column	Datatype	Remark
ID	INT	PK
ID_USER_LOGIN_DETAIL	INT	FK - USER_LOGIN_DETAIL.ID
SESSION_ID	VARCHAR	
LOGIN_STATUS	VARCHAR	
LOGIN_DATE_TIME	TIMESTAMP	
LOGOUT_DATE_TIME	TIMESTAMP	

Table Name: AUDIT_ACTION

Column	Datatype	Remark
ID	INT	PK
ID_USER_LOGIN_DETAIL	INT	FK - USER_LOGIN_DETAIL.ID
USER_ACTION	VARCHAR	
START_DATE_TIME	TIMESTAMP	
END_DATE_TIME	TIMESTAMP	