C++ Workshop Series for UCSC

Flower Exchange – Group Project





Flower Exchange Group Project

- Two persons per group
- Duration Three weeks
- This is the only Assessment for this workshop
- Date of completion 30th March 2023
- Demonstration of the project 31st March 2023 at LSEG Technology Malabe

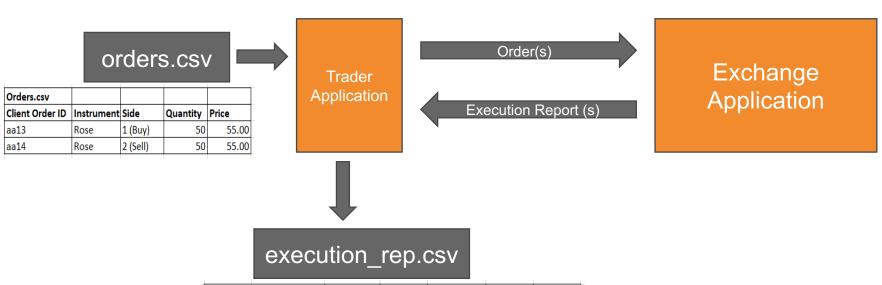


Flower Exchange Story

- The flower exchange is a system which supports basic trading.
- **Trader Application** Traders can submit buy or sell orders for flowers via the Trader Application.
- **Exchange Application** will process the incoming order against existing orders in the order container(commonly known as Order Book) and do a full or partial execution.
- Every order is replied with an Execution Report by the Exchange Application indicating the status of the order.
- Orders sometimes could be rejected due to quantity limitations, invalid flower type, etc.



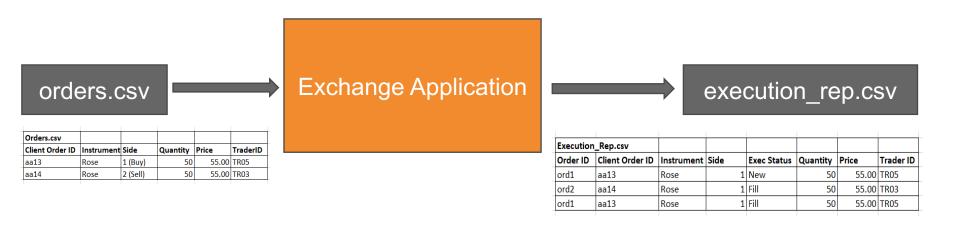
Flower Exchange – High Level Architecture



Execution	_Rep.csv					
Order ID	Client Order ID	Instrument	Side	Exec Status	Quantity	Price
ord1	aa13	Rose	1	New	50	55.00
ord2	aa14	Rose	1	Fill	50	55.00
ord1	aa13	Rose	1	Fill	50	55.00



Flower Exchange – A simpler implementation (2nd option)



Basically, read an orders.csv fila and produce am execution_rep.csv file



Flower Exchange – Input order

Field Name	Туре	Possible Values	Mandatory	Notes
Client Order ID	String	Alpha numeric string (max 7 chars)	Yes	This unique ID identifies the submitted order
Instrument	String	(Rose, Lavender, Lotus, Tulip, Orchid)	Yes	We will limit the instruments for these 5 types only
Side	Int	1 : Buy ; 2 : Sell	Yes	Specifies if the input order is a buy order or a sell order
Price	double	Price > 0.0	Yes	Price of one unit
Quantity	int	{10, 20, 30,, 1000} Order size must be a a multiple of 10. Min 10, Max 1000	Yes	Quantity of the order



Flower Exchange – Output Execution Report

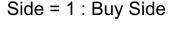
Field Name	Туре	Possible Values	Mandatory	Notes
Client Order ID	String	Alpha numeric string (max 7 chars)	Yes	This is the Client Order ID of the submitted order
Order ID	String	Alpha numeric string	Yes	System generated unique order ID
Instrument	String	(Rose, Lavender, Lotus, Tulip, Orchid)	Yes	We will limit the instruments for these 5 types only
Side	Int	1 : Buy ; 2 : Sell	Yes	Specifies if the order is a buy order or a sell order
Price	double	Price > 0.0	Yes	Price of one unit
Quantity	int	{10, 20, 30,, 1000} Order size must be a a multiple of 10. Min 10, Max 1000	Yes	Quantity of the order
Status	int	0 – New 1 – Rejected 2 – Fill 3 - Pfill	Yes	The status of the execution report
Reason	String	Max 50 chars	No	Contains the reject reason, when an order is not accepted into the system due to validation failure
Transaction Time	String	YYYYMMDD-HHMMSS.sss	Yes	Every execution report should have the transaction time in the given format. We will this data check the speed of your application. This is mandatory.



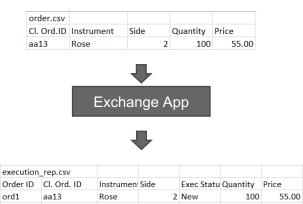
Order book – Example 1

- Exchange app maintains one order book for each Instrument (flower type). Since there are 5 types of Instruments, there will be five separate order books in our system.
- Orderbook has two sides. Buy (blue), Sell (pink)
- Initially the orderbook is empty
- The order book receives an order. (Side = Sell, Price = 55.00, Qty = 100)
- This order will go inside sell side of the order book, and an execution report with status "New" will be disseminated





Side = 2 : Sell Side





order.csv				
Cl. Ord.ID	Instrument	Side	Quantity	Price
aa13	Rose	2	100	55.00
aa14	Rose	2	100	45.00
aa15	Rose	1	100	35.00





State – Final									
Order ID	Qty	Price	Price	Qty	Order ID				
ord3	100	35.00	45.00	100	ord2				
			55.00	100	ord1				

order.csv				
Cl. Ord.ID	Instrument	Side	Quantity	Price
aa13	Rose	2	100	55.00
aa14	Rose	2	100	45.00
aa15	Rose	1	100	35.00





execution	_rep.csv					
Order ID	Client Order	Instrumen	Side	Exec Status	Quantity	Price
ord1	aa13	Rose	2	New	100	55.00
ord2	aa14	Rose	2	New	100	45.00
ord3	aa15	Rose	1	New	100	35.00



Order Book - Example 3 (A FILL execution)



order.csv					
Cl. Ord.ID	Instrument	Side		Quantity	Price
aa13	Rose	2	2	100	55.00
aa14	Rose	2	2	100	45.00
aa15	Rose	1	1	100	45.00





execution	_rep.csv					
Order ID	Client Order	Instrumen	Side	Exec Status	Quantity	Price
ord1	aa13	Rose	2	New	100	55.00
ord2	aa14	Rose	2	New	100	45.00
ord3	aa15	Rose	1	Fill	100	45.00
ord2	aa14	Rose	2	Fill	100	45.00



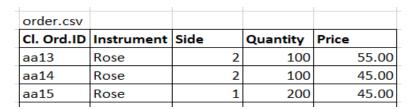
Order book

- Buy side of the orderbook is sorted in the ascending order of the price. (Higher the buy price, more attractive the order)
- Sell side of the orderbook is sorted in the descending order of the price. (Lower the sell price, more attractive the order)
- Orders with the same price are ordered in the time priority. (Priority sequence)

Orderbook	: Rose								
Pr.Seq	Order ID	Trader ID	Qty	Price	Price	Qty	Trader ID	Order ID	Pr.Seq
1	ord4	Tr04	100	4.00	5.00	200	Tr02	ord2	1
1	ord8	Tr08	200	3.00	5.00	100	Tr03	ord3	2
2	ord7	Tr02	200	3.00	6.00	100	Tr01	ord1	1



Order Book - Example 4 (A PFILL execution)







State – Final								
Order ID	Qty	Price	Price	Qty	Order ID			
ord3	100	45.00	55.00	100	ord1			

Instrument	Side	Quantity	Price
Rose	2	100	55.00
Rose	2	100	45.00
Rose	1	200	45.00
	Rose Rose	Rose 2	Rose 2 100 Rose 2 100

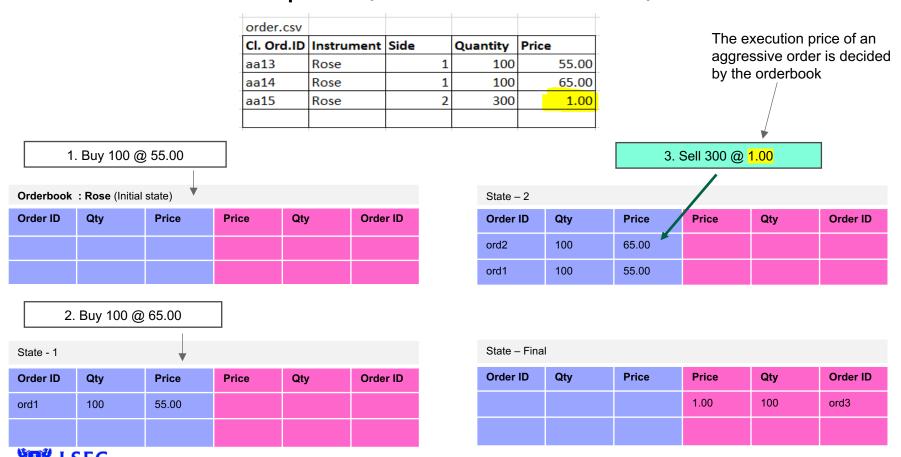




execution_rep.csv						
	Client Order	Instrumen	Side	Exec Status	Quantity	Price
ord1	aa13	Rose	2	New	100	55.00
ord2	aa14	Rose	2	New	100	45.00
ord3	aa15	Rose	1	Pfill	100	45.00
ord2	aa14	Rose	2	Fill	100	45.00



Order Book - Example 5 (A FILL execution with a twist)



order.csv				
Cl. Ord.ID	Instrument	Side	Quantity	Price
aa13	Rose	1	100	55.00
aa14	Rose	1	100	65.00
aa15	Rose	2	300	1.00





execution_rep.csv						
Order ID	Client Order	Instrumen	Side	Exec Status	Quantity	Price
ord1	aa13	Rose	1	New	100	55.00
ord2	aa14	Rose	1	New	100	65.00
ord3	aa15	Rose	2	PFill	100	65.00
ord2	aa14	Rose	1	Fill	100	65.00
ord3	aa15	Rose	2	PFill	100	55.00
ord1	aa13	Rose	1	Fill	100	55.00



_				
order.csv				
Cl. Ord.ID	Instrument	Side	Quantity	Price
aa13	Rose	1	100	55.00
aa14	Rose	1	100	65.00
aa15	Rose	2	300	1.00
aa16	Rose	1	100	2.00







Final state					
Order ID	Qty	Price	Price	Qty	Order ID



order.csv				
Cl. Ord.ID	Instrument	Side	Quantity	Price
aa13	Rose	1	100	55.00
aa14	Rose	1	100	65.00
aa15	Rose	2	300	1.00
aa16	Rose	1	100	2.00





execution_rep.csv						
Order ID	Cl. Ord. ID	Instrumen	Side	Exec Statu	Quantity	Price
ord1	aa13	Rose	1	New	100	55.00
ord2	aa14	Rose	1	New	100	65.00
ord3	aa15	Rose	2	PFill	100	65.00
ord2	aa14	Rose	1	Fill	100	65.00
ord3	aa15	Rose	2	PFill	100	55.00
ord1	aa13	Rose	1	Fill	100	55.00
ord4	aa16	Rose	1	Fill	100	1.00
ord3	aa15	Rose	2	Fill	100	1.00



Flower Exchange – Input validations

An order will be rejected if

It does not contain a required field

It is for an invalid Instrument

It contains an invalid side

Its price is not greater than 0

Its quantity is not a multiple of 10

Its quantity is outside the range (min = 10 max = 1000)

A Rejected execution report is generated when validations are failed



Order Book - Example 7 (Input Validations)

order.csv				
Cl. Ord.ID	Instrument	Side	Quantity	Price
aa13		1	100	55.00
aa14	Rose	3	100	65.00
aa15	Lavender	2	101	1.00
aa16	Tulip	1	100	-1.00
aa17	Orchid	1	1000	-1.00



Exchange App



execution_rep.csv		_rep.csv						
	Order ID	Cl. Ord. ID	Instrumen	Side	Exec Status	Quantity	Price	Reason
	ord1	aa13		1	Reject 🗲	100	55.00	Invalid instrument
	ord2	aa14	Rose	3	Reject	100	65.00	Invalid side
	ord3	aa15	Lavender	2	Reject	101	1.00	Invalid size
	ord4	aa16	Tulip	1	Reject	100	-1.00	Invalid price
	ord3	aa17	Orchid	1	Reject	1000	-1.00	Invalid size

A validation failure results in a Rejected Execution Report



Flower Exchange – Verify your system

- We have provided you 6 sample input files (order.csv) and corresponding sample output files (exec_rep.csv) files in the .
- When developing your system, test your system with the sample input files we provided and verify your system produces the same output as in the sample exec_rep.csv files

SampleOrder1.csv (give this as an input to your app)



execReports1.csv (produced by your app)

Compare:

- exeReports.csv generated by your system matches the sample_exereports.csv Pass
- exeReports.csv generated by your system produces a different output to sample_exereports.csv Fix your bugs, and re-test
- Note that the order of the execution reports in your file does not have to be the same as ours. But the content should be the same.



Flower Exchange – Evaluation

- Demonstration and code review You will demonstrate the system and we will do a code review with you.
- Project will be evaluated for the design, coding practices, efficiency of the program and the speed of the program.
- During the demo, we will test your application against our test data (orders.csv) and verify the functional accuracy.
- To test the speed of the program we will input a large orders.csv and measure the time taken to produce the resulting exec_reports.csv (Note that **Transaction Time** is a mandatory field in the Execution Report)

