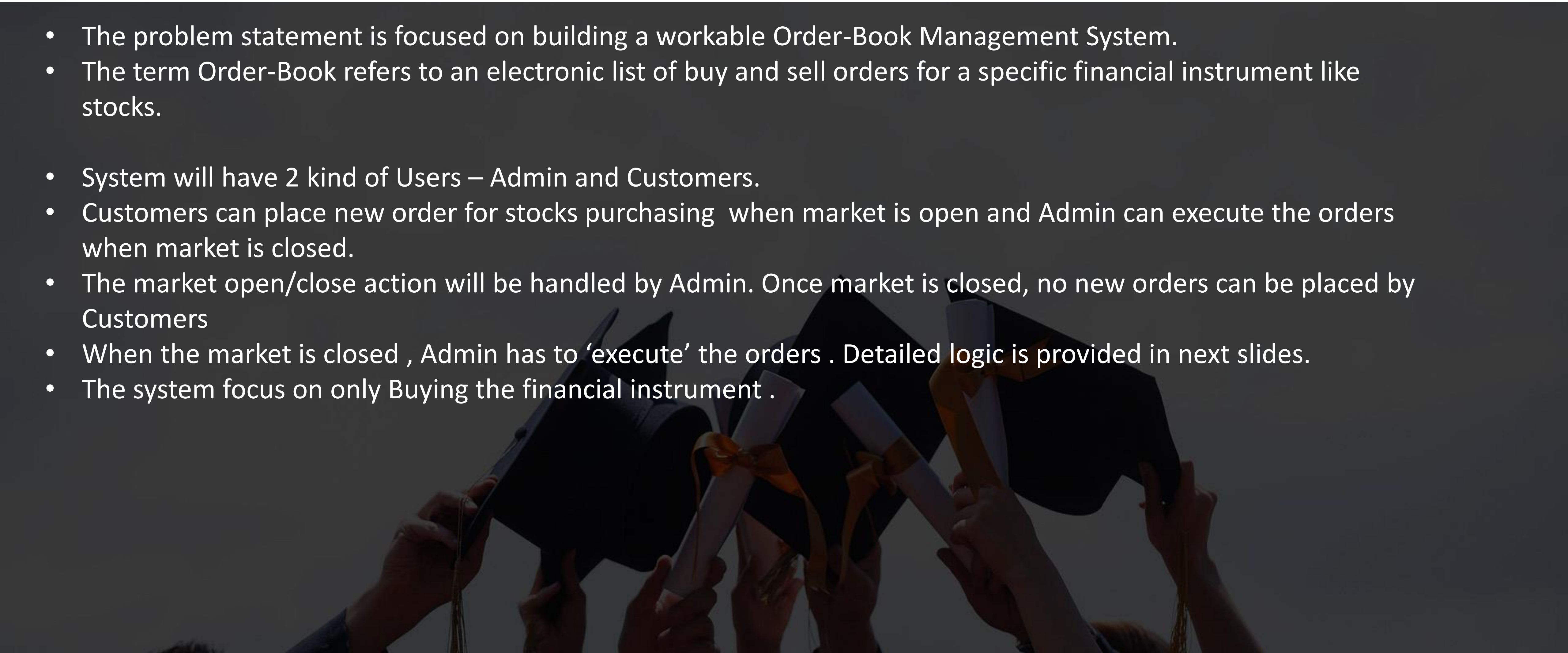
PROBLEM STATEMENT

Order Book System

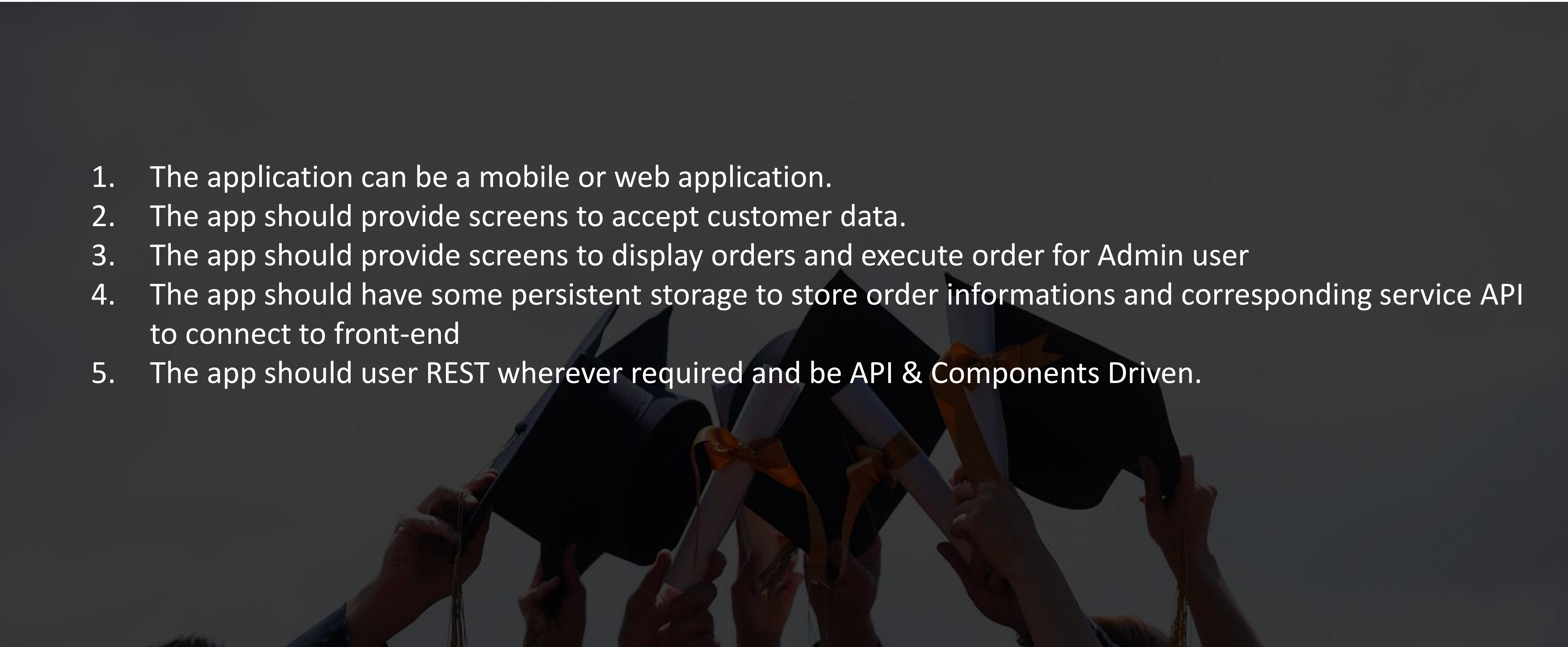
INTRODUCTION



- The problem statement is focused on building a workable Order-Book Management System.
- The term Order-Book refers to an electronic list of buy and sell orders for a specific financial instrument like stocks.

- System will have 2 kind of Users – Admin and Customers.
- Customers can place new order for stocks purchasing when market is open and Admin can execute the orders when market is closed.
- The market open/close action will be handled by Admin. Once market is closed, no new orders can be placed by Customers
- When the market is closed , Admin has to 'execute' the orders . Detailed logic is provided in next slides.
- The system focus on only Buying the financial instrument .

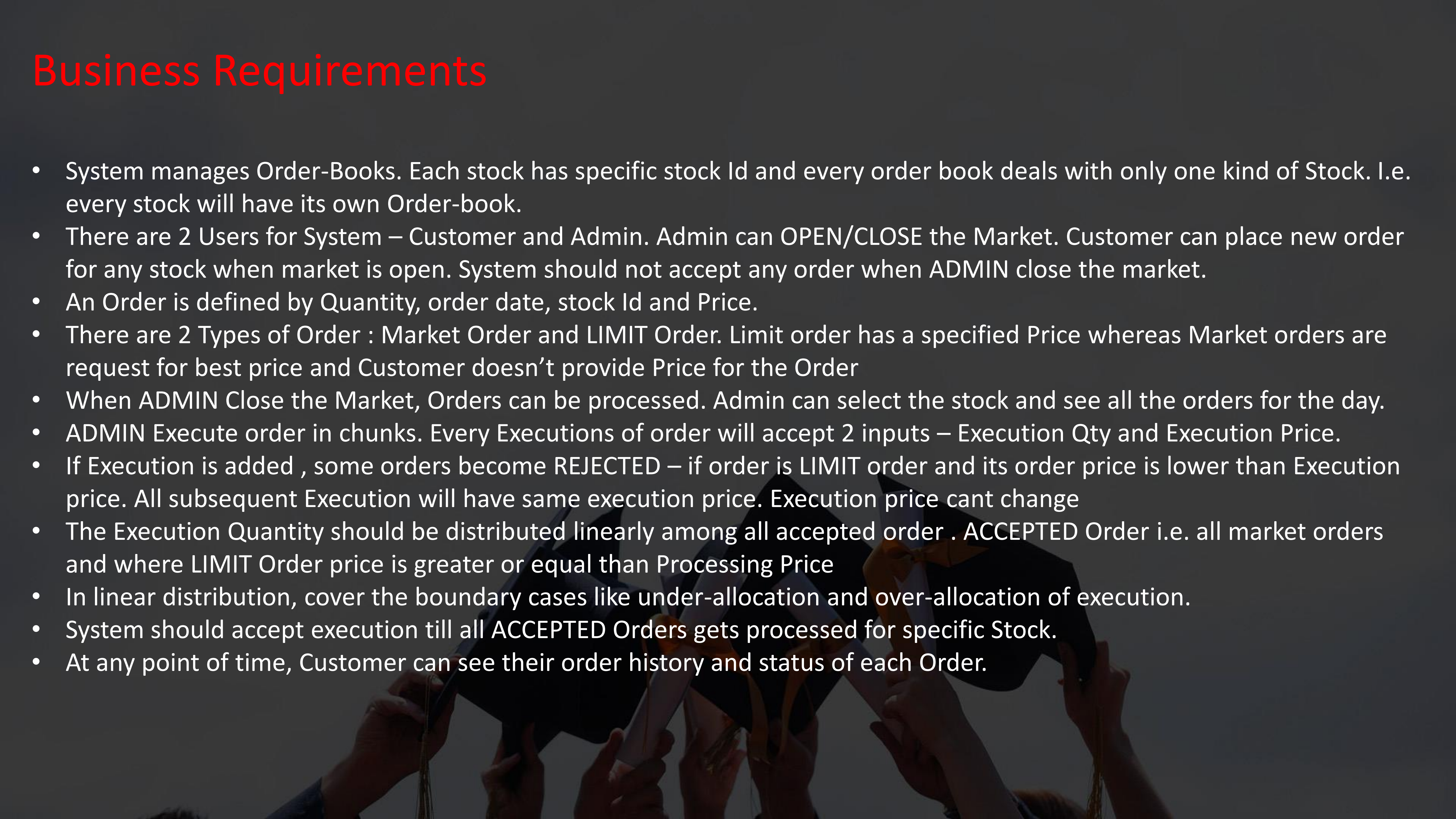
DELIVERABLES



1. The application can be a mobile or web application.
2. The app should provide screens to accept customer data.
3. The app should provide screens to display orders and execute order for Admin user
4. The app should have some persistent storage to store order informations and corresponding service API to connect to front-end
5. The app should user REST wherever required and be API & Components Driven.

Business Requirements

- System manages Order-Books. Each stock has specific stock Id and every order book deals with only one kind of Stock. I.e. every stock will have its own Order-book.
- There are 2 Users for System – Customer and Admin. Admin can OPEN/CLOSE the Market. Customer can place new order for any stock when market is open. System should not accept any order when ADMIN close the market.
- An Order is defined by Quantity, order date, stock Id and Price.
- There are 2 Types of Order : Market Order and LIMIT Order. Limit order has a specified Price whereas Market orders are request for best price and Customer doesn’t provide Price for the Order
- When ADMIN Close the Market, Orders can be processed. Admin can select the stock and see all the orders for the day.
- ADMIN Execute order in chunks. Every Executions of order will accept 2 inputs – Execution Qty and Execution Price.
- If Execution is added , some orders become REJECTED – if order is LIMIT order and its order price is lower than Execution price. All subsequent Execution will have same execution price. Execution price cant change
- The Execution Quantity should be distributed linearly among all accepted order . ACCEPTED Order i.e. all market orders and where LIMIT Order price is greater or equal than Processing Price
- In linear distribution, cover the boundary cases like under-allocation and over-allocation of execution.
- System should accept execution till all ACCEPTED Orders gets processed for specific Stock.
- At any point of time, Customer can see their order history and status of each Order.



New Order – Customer Screen 1

Search Stock

DBS… ⇩

- This should be Auto complete Dropdown input field to see pre-defined stocks list . Customer can search a stock name here. Each stock is tied to its Stock ID

Order Type

Limit or Market

- Input type Select Dropdown with 2 values ‘LIMIT’ or ’MARKET’. If limit order is selected, then customer has to provide Price in next price field and its mandatory for LIMIT order. For MARKET order type , PRICE field becomes disabled and not required

Price

150.00

- Price for Limit Order.Should be disable for Market Order

Quantity

10

- Quantity of order .Must be a positive integer

Place Order

- Submit button for placing order. This should be enabled only when Market is OPEN by ADMIN User. By default it should be disabled . If market is OPEN , On submitting this form,Customer should be shown some message to inform that Order placed successfully.

Order History – Customer Screen 2

- A Customer can see all the order placed , filter by Date and Stock, all columns should be sortable on click

Select Stock

DBS… ⇩

From Date

To Date

Show Orders

S.No.	Stock Name	Order Qty	Order Type	Executed Qty	Price	Order Status	Order Date
1	DBS	50	Limit	0	150	PLACED	21 Feb 2021
2	DBS	50	Market	0		PLACED	21 Feb 2021
3	DBS	50	Limit	0	140	PLACED	21 Feb 2021

Admin Screen-1

Open Market

CLOSE Market

Buttons to OPEN/CLOSE the Market.

Select Stock

DBS… ⇩

From Date

To Date

Show Orders

Date Picker and Stock select Filters to Show all orders.. Order status can be PLACED, PROCESSED,ACCEPTED and REJECTED

S.No.	Stock Name	Order Qty	Order Type	Executed Qty	Price	Order Status	Order Date
1	DBS	20	Limit	0	150	PLACED	21 Feb 2021
2	DBS	40	Market	0		PLACED	21 Feb 2021
3	DBS	50	Limit	0	140	PLACED	21 Feb 2021

Execution Qty

30

Execution Price

150

Execute Orders

Execution Of Order will be accepted when Market is closed. Once execute orders is clicked.Table should refresh the data .

Admin Screen-2

OPEN Market

CLOSE Market

Buttons to OPEN/CLOSE the Market.

Select Stock

DBS…

From Date

To Date

Show Orders

Date Picker and Order status Filters to Show all orders.. Order status can be PLACED, EXECUTED,ACCEPTED and REJECTED

S.No.	Stocks Name	Order Qty	Order Type	Executed Qty	Price	Order Status	Order Date
1	DBS	20	Limit	10	150	ACCEPTED	21 Feb 2021
2	DBS	40	Market	20		ACCEPTED	21 Feb 2021
3	DBS	50	Limit	0	140	REJECTED	21 Feb 2021

Execution Qty

30

Execution Price

150

Execute Orders

After First Execution, 3rd order became rejected and execution qty distributed ‘linearly’ among all accepted orders i.e. (order Qty / total Accepted order) * execution qty , in this case for first order (20/60) * 30 = 10

Admin Screen-3

Open Market **CLOSE Market**

Buttons to OPEN/CLOSE the Market.

Select Stock

DBS… ⇩

From Date

To Date

Show Orders

S.No.	Stocks Name	Order Qty	Order Type	Executed Qty	Price	Order Status	Order Date
1	DBS	20	Limit	20	150	EXECUTED	21 Feb 2021
2	DBS	40	Market	40		EXECUTED	21 Feb 2021
3	DBS	50	Limit	0	140	REJECTED	21 Feb 2021

Execution Qty

30

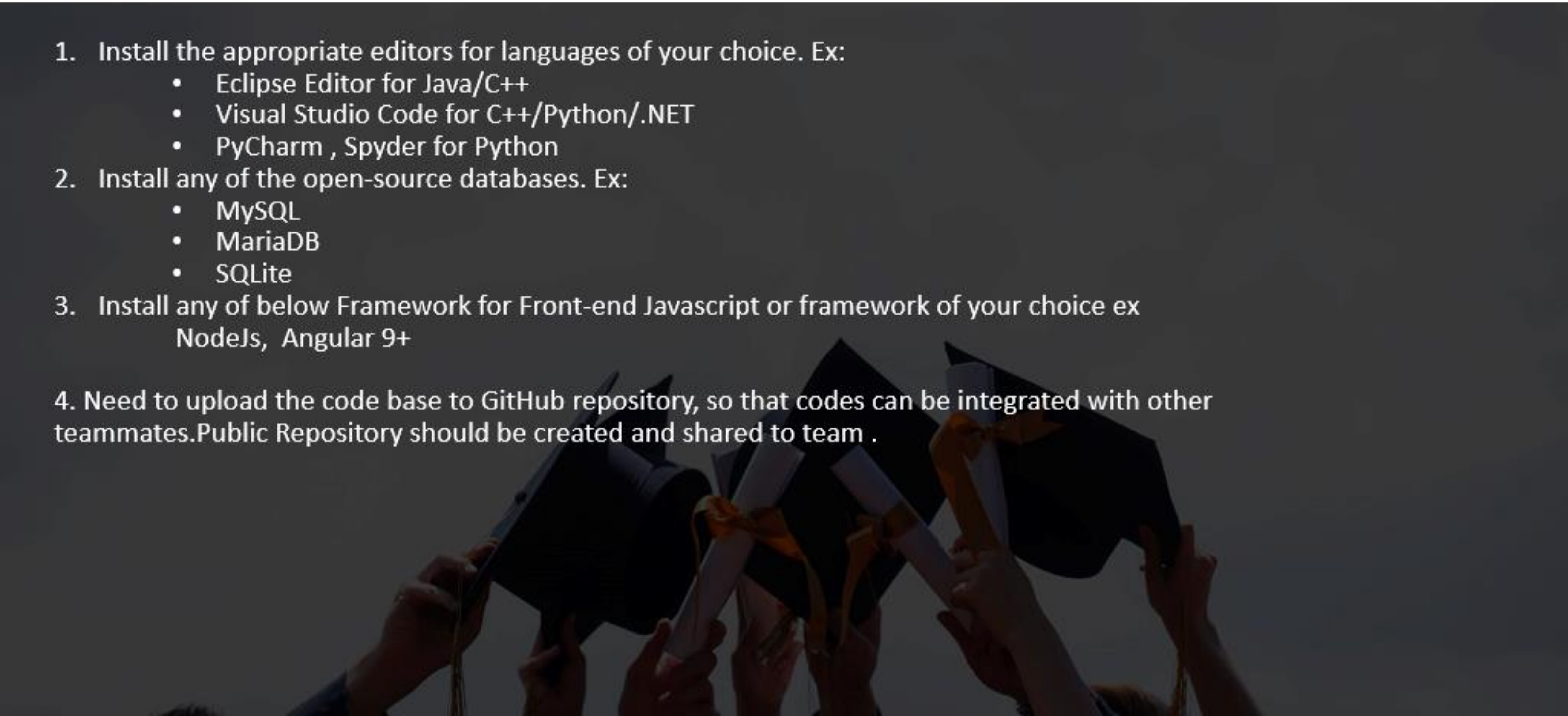
Execution Price

150

Execute Orders

After second execution , both the ACCEPTED orders quantity has been completed and status changes to EXECUTED. After this no new execution can be submitted .

Software Prerequisites:



1. Install the appropriate editors for languages of your choice. Ex:
   - Eclipse Editor for Java/C++
   - Visual Studio Code for C++/Python/.NET
   - PyCharm , Spyder for Python
2. Install any of the open-source databases. Ex:
   - MySQL
   - MariaDB
   - SQLite
3. Install any of below Framework for Front-end Javascript or framework of your choice ex

   NodeJs, Angular 9+

4. Need to upload the code base to GitHub repository, so that codes can be integrated with other teammates.Public Repository should be created and shared to team .