## The Brief

ING wants to create a trading platform targeting private customers in Indian Market involved in NSE & BSE trading. Customers should be able to invest on equities in multiple sectors.

An equity investment generally refers to the buying and holding of shares of stock on a stock market by individuals and firms in anticipation of income from dividends and capital gains.

You are not required to implement every aspect of the application. For example, assume authentication and authorization is being developed by another team, to be integrated at a later date.

The product owner has listed the criteria by which the application is specified. They are not in any specific order and are not grouped into features but it should give you an idea of what is expected.

### General Requirements

#### User Identification

This platform is a multi-user platform so there must be a way to select a user. This can be a static set of users hard coded but it would be nice if the user list came from the database. For the session you need to nominate which user the session is for and this can be done in any simple way, such as a drop down list, free text box or even just a query string. No password or other credentials should be used to authenticate the user, this is out of scope.

## Use Case 1 – Execute a trade to purchase stock

To purchase stock users must go through a three step process. They must select a stock to purchase, review quote to submit the purchase and view summary of the executed purchase.

### Use Case 1, Part 1 – Select Stock

#### Mandatory

The customer must be able to select one stock to purchase from a list of at least 5 valid pre-defined stocks of your choosing. They must also be able to enter how many of the selected stock they would like to purchase.

Customers can only select one stock name per purchase.

#### Nice to have

Call the API to get a list of names for all available stocks to allow users to select any of these.

### Use Case 1, Part 2 – Confirm Order

#### Mandatory

The customer is to be presented with a quote for the volume of stock requested to purchase. This quote must be generated based on the current market price of the selected stock. The quote must also include the below fees.

A warning must be displayed that stock prices change regularly and the quote is subject to change. The order will be placed with the live price at the time of submission. To help customers with this we will provide a ‘Re-Quote’ option to get the latest price.

The customer must then be able to confirm or cancel the purchase of this stock.

##### Fees & Charges:

* Customer will be charged nominal brokerage fee of their purchases
  + 0.10% for per share if less than 500 shares
  + 0.15% for every 100 shares if great than or equal to 500 shares in purchase

#### Nice to have #1

Include market price of the stock from yesterday, 7 days ago and 14 days ago to give the customer more information when making their purchase. This should be fetched and displayed using Ajax to ensure it doesn’t slow down the critical step of reviewing the quote. This can be done direct from the UI without a Java service.

#### Nice to have #2

It would be nice if the user could edit the volume of stocking being purchased on this page before clicking the ‘Re-Quote’ option.

### Use Case 1, Part 3 – Execute Order

If the customer confirms the order we must fetch the latest price and execute the trade. For the purposes of this exercise inserting a record into your database table with the details of the order is sufficient to be considered an ‘executed order’.

A confirmation screen must now be shown to the users to let them know the order was successful and must also show them the final price for which the order was executed.

## Use Case 2 – Review past orders

In addition to placing orders users must be able to review past orders. You must give them a page they can review to see a table of all completed orders. The table should show a summary of the data available and the customer must then be able to click an order to view the full details of that trade.

Full details must include at least:

1. Time of trade
2. Stock name
3. Stock price
4. Volume purchased
5. Total stock purchase price
6. Fees
7. Total including fees

#### Nice to have #1

Customers can click a ‘Purchase Again’ option which will take them directly to the quote page with stock name, and volume pre-populated.

#### Nice to have #2

As a Trader, I want to view a simple graph showing the split of my investment over different stock

#### Nice to have #3

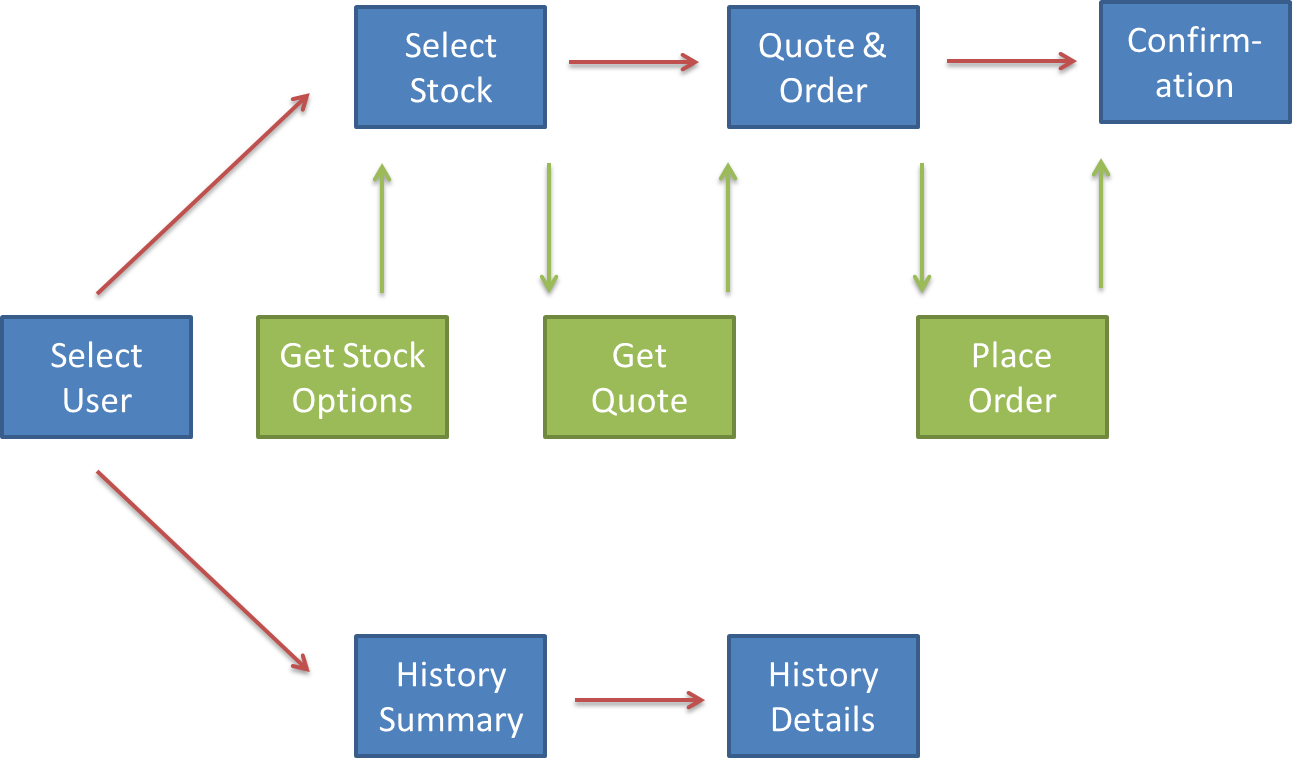
For each order show the current price and calculate profit/loss since the time of the original order

## Acceptance Criteria As a Trader, I want to select the equity and place the order based on the market price.

As a Trader, I want the price to be changed based on the market live pricing during confirm order.

Example: Equity price when customer selecting the order is INR100 but while confirming the order, market price can be increased or decreased. Hence the final price after order placed is INR110 or INR99 depending upon market price.

## Solution Overview



## Expectations

**Technology:** Google Polymer for UI && Java for server side

**Backend:**

* Restful API for backend services
* Consider about the security & validations while inserting into DB

**Testing:**

A balanced testing strategy is critical in every project. A significant amount of effort will be invested in this so it’s important that you have a clear idea on the kind of testing approach that is best for you to follow. We are not expecting you to have fully implemented full-fledged testing solution covering each layer of testing. We think that you have to have a clear idea about the model that you would use in a project like this platform and do accordingly to that, the scope and the time for this exercise.

**Other:**

* It’s better to demonstrate one thing working well, than it is to demonstrate everything half-complete
* Use your product owners
* Be open and honest with your stakeholders and yourselves; if you’re not going to make it, change direction or negotiate scope with your product owners
* Protect your team from themselves in case they overcommit

## End of Day Review & Demo

Don’t underestimate the importance of the review. Each team will have 15 minutes to demonstrate what they have done. Consider finishing 30 minutes before the reviews are scheduled to do a practice run.

## API References

To get live Equities, you can construct API from the below link. You can claim your free API key.

<https://www.alphavantage.co/documentation/#time-series-data>

ING TESTING CHALLENGE 2018

Welcome to ING Testing Challenge 2018, in this challenge we have listed two simple UI &API Testing tasks to be coded and tested.

Please note that this is time boxed challenge, hence you are requested to complete this no later than 1pm. Also you need to demo your work for 5 to 10 minutes after 1pm. You are requested to create a very simple framework which supports UI and API testing.

**Scenario 1:**

You are requested to navigate to the following URL mentioned below and do a UI Automation testing covering maximum test scenarios as possible.

<https://www.distancecalculator.net/>

**Scenario 2:**

You are requested to validate the below API request using restassured + java code.

<https://openweathermap.org/current>

Task:

* You can use any browser that you are comfortable with
* Think as much as test scenarios as possible, think from a regression point of view to project future code changes from breaking the existing functionality
* What can go wrong in development?
* Please make sure to use proper assertion techniques
* Java + Selenium? C# + Selenium? JavaScript? Anything you are comfortable with.