

**Question 1:** Is there any restriction/validation/requirement which is not included in the description but could be important for the stakeholders/product team/developers or QA? If so, could you describe them?

Currently, given requirement is not captured accurately and missing validations on user input fields. It also looks like that at database layer attributes are not being validated.

**Requirement 1** specifies how existing trades in the database display to the user on the frontend but the given layout in figure 1 is not populating some attribute values in correct format such as Sell Amount, Buy Amount, and Rate, and as a result value of these attributes are appeared to be calculated incorrectly.

Attribute	Example Value
ID	TR0123456
Sell Currency	USD
Sell Amount	500,00
Buy Currency	EUR
Buy Amount	633,13
Rate	1,2756
Date Booked	2018/05/27 23:13:05

Highlighted values should populate as 500.00, 637.8, and 1.2756. In my view, these attributes values are shown incorrectly because, in the database, table schema is created incorrectly to store these attribute values in correct format. To handle these values being shown incorrectly on trade view, the table schema should look like below:

```
CREATE TABLE `booked_trade_tb` (  
  `id` INT(20) NOT NULL AUTO_INCREMENT,  
  `sell_currency` varchar(3) NOT NULL,  
  `sell_amount` double(12,2) NOT NULL,  
  `buy_currency` varchar(3) NOT NULL,  
  `buy_amount` double(12,2) NOT NULL,  
  `rate` double(4,4) NOT NULL,  
  `date_booked` datetime DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY (`id`)  
);
```

I assume sell currency, sell amount, buy currency, buy amount and rate are mandatory fields in the New Trade form in the frontend.

**Requirement 2** describes a core feature of the application where the user can wire money from one currency to another currency and therefore this feature should be designed and develop efficiently and to do so requirement of this module should be captured and defined accurately. The current requirement is not sufficient and not covering the following:

1. Should the user allow to enter non-numeric values in the Sell Amount field? If not, does the system validate this by showing an appropriate error message to the user?
2. What is the minimum Sell Amount that a user can Sell?
3. What is the maximum limit of the Sell Amount that the user can sell?
4. Can the user sell the amount in decimal value i.e.1.5?
5. On new trade layout should buy amount be calculated with two decimal value or more?
6. Does the system add the currency symbol of Sell Currency as a prefix in the Sell Amount text box?
7. Does the system allow users to enter the sell Amount in negative numbers? If not, does the system validate this by showing a relevant error message?
8. Does the system allow to enter zero in the Sell Amount field? If not, does the system validate this by showing a relevant error message?
9. Does the system allow users to submit blank forms? If not, does the system show relevant error messages for each mandatory field?
10. When the user clicks on Create button to create new trade, does the system show a confirmation pop up to confirm a New Trade? If a confirmation pop-up appears and the user clicks on OK to confirm, does the system show final confirmation to confirm that a New Trade has been created and where the user should be navigating to?. If the user clicks on NO on the confirmation pop-up, then does the New Trade form get empty? and where the user should navigate to?
11. User select Sell Currency, Buy Currency and enter valid Sell Amount and click on Cancel button. Does the system show a confirmation pop-up to the user? If the user selects OK to confirm, does the New Trade form get empty? And where the user should navigate to?
12. Does system have ability to auto correct Sell Amount user input? From 1.000,00 to 1.00.
13. What is the complete flow for the user to navigate to Booked Trades page?

Currently, on the New Trade layout/form, the Sell Amount user input is not appeared to validate correct format, and the buy amount field is showing value in the wrong format.

## New Trade

Sell Currency		Rate		Buy Currency
<div style="border: 1px solid black; padding: 5px; display: inline-block;">             GBP ▼           </div>	▶	1.2561	▶	<div style="border: 1px solid black; padding: 5px; display: inline-block;">             USD ▼           </div>
Sell Amount				Buy Amount
<div style="border: 2px solid red; padding: 10px; display: inline-block;">             1.000,00           </div>				<div style="border: 2px solid red; padding: 10px; display: inline-block;">             1.256,10           </div>
<div style="border: 1px solid gray; padding: 10px; display: inline-block; width: 100px;">             Create           </div>				<div style="border: 1px solid gray; padding: 10px; display: inline-block; width: 100px;">             Cancel           </div>

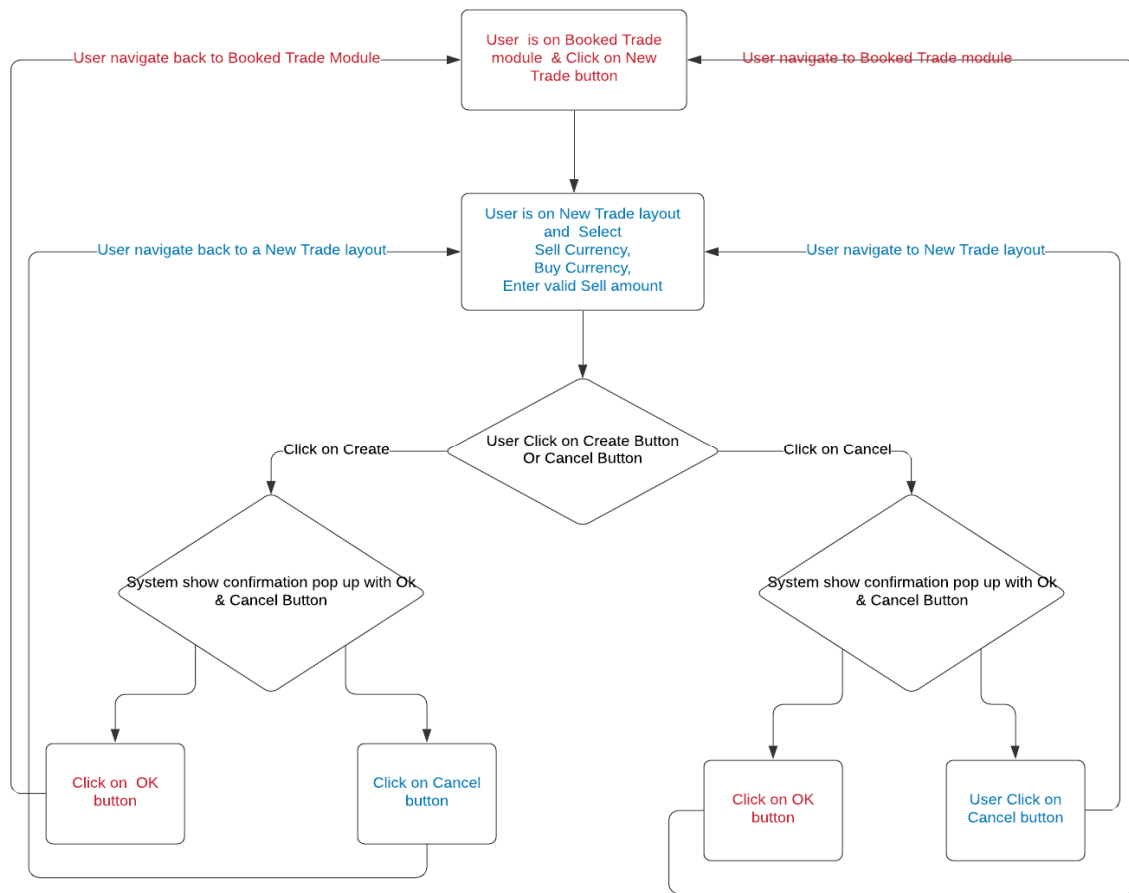
In my opinion, this is a result of an insufficient requirement specification. To design and develop this feature in a correct manner all validations and relevant error messages for mandatory user input must be outlined in requirement specification along with use case flow diagram.

For example:

**Note:** My example is based on assumption that the Sell Currency, Buy Currency and Sell Amounts fields are mandatory.

Element Name	Element Type	User Action	Valid Data	Error State
Sell Currency	Dropbox	Select Value	Mandatory Valid Selection. By default empty.	If Blank – “Please select Sell Currency”
Buy Currency	Dropbox	Select Value	Mandatory Valid Selection. By default empty.	If Blank – “Please select Sell Currency”
Sell Amount	Text Box	Enter Valid Value	Mandatory 1. Minimum 1 currency unit 2. Only numeric value	If enter < 1 – “Please enter an amount greater than 0” If enter non-numeric – “Please enter a valid amount”
Buy Amount	Text Box	None	Not editable & Auto calculate	
Create	Button	Click		If any mandatory field/s are blank – “Show error message for the respective field”
Cancel	Button	Click		

**Task flow diagram**

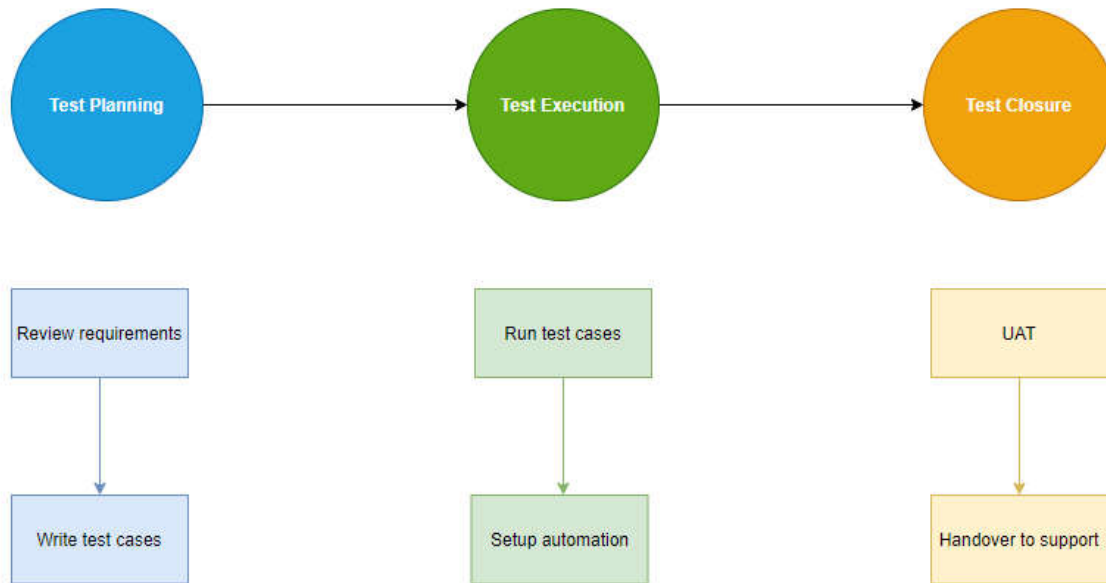


In my opinion above flow diagram will help QA to derive test cases accurately and will also help non-technical stack holders to understand how the New Trade feature flow should work.

Apart from the above, in order to do end-to-end testing to cover Web Application view from the frontend, backend API, and database layers, following is a list of questions I would like to get discussed or clarified.

1. What is the Data type and length for each attribute?
2. Traded list stored in DB sorted by data asc or desc?
3. Date timestamp in Req1 is yyyy/mm/dd hh:mm:ss which is displayed as dd/mm/yyyy hh:mm under "Book Trades" in Req2, which one is correct?
4. Which fields are mandatory and which fields can be empty in the database? (If I ignore my assumption that Sell Currency, Buy Currency and Sell Amounts fields are mandatory.)
5. How will the "Book Trades" page look like if the trade list is empty?
6. Can the trade list be sorted by the user? if yes then by which column value?
7. What is a supported currency list?
8. Who can view the trades list? Is there any authentication required?
9. What is a success or failure messages for each access level? i.e. View trades, book new trade, create a trade with invalid parameters?
10. Does testing also need to perform on different OS, cross-browser, and devices? If yes, which OS, browsers, and devices?
11. Is there a POST request API to start testing New Trade functionality if frontend is still not ready?

**Question 2: Define a test plan for every requirement.**



## Test Planning

### Review Requirements

- Read business requirements fully and precisely and user flow diagram that helps to understand feature/functionality which needs to be delivered.
- Ensure that the requirements are clear and easy to understand. If they are not understandable or accurate then please do not hesitate to ask the Project Manager or other stakeholders.
- Once the requirement is reviewed thoroughly, look for any gaps in requirements.

Requirement 1 – [Confluence page reference URL](#)

This requirement will turn into a feature that pulls out existing trade transactions from the database and display them on the main view of the application.

Requirement 2 – [Confluence page reference URL](#)

This requirement is relating to developing a New Trade module which will allow the user to sell their currencies and in return to buy their preferred currency at the prevailing rate.

This and any other requirements that out of scope may include later must be tested as a part of integration and system testing.

QAs to ensure that you are fully aware of requirements and what needs to be tested to coverage requirements. Ensure that all available resources are available prior to testing commenced. This includes:

- QA application URL
- Jenkins job access
- QA database access
- JIRA board access
- JIRA Confluence access

- Access to dedicated team channel for this project on Microsoft Team for internal communication and daily stand up
- Backend application access

**Note** - Please refer to the [Confluence page](#) to get all URLs and contact DevOps/System Admin to get access to other required tools.

Once Sprint kick start, provide testing estimation thinking about-

- Cross Browser Testing
- Cross Device testing
- End to end testing on QA and smoke testing on STG and LIVE environments

After having the requirement reviewed, start writing test cases/Test Scenarios at a high level in relevant tickets which may help developers to write code precisely.

## Test Execution

### Testing Approach

**The New Trade** module is a customer-facing module, so it requires thorough testing which includes the following type of manual testing.

#### Functional Testing

- Integration testing
- System Testing
- Acceptance Testing (Smoke testing on stage environment)

#### Non-Functional Testing

- Performance Testing using Jmeter tool
- Compatibility testing on cross device and cross browser
- Usability testing

**View Trade** layout is customer-facing but does not capture value from the frontend. It only shows existing trade from the database therefore backend /database testing need to perform to ensure that each new trade store data in a relevant table accurately and display on view trade page correctly.

Detailed test cases/test scenarios should be log in the **JIRA confluence page** in the below template and the reference URL of this page to be linked in the respective JIRA ticket.

Test case Template:

- Scenario
- Pre-Conditions
- Steps
- Expected Result
- Test Status

### Run Test Cases

- On desktop app (Chrome, FF, IE)
- On Mobile (iOS, Android), Tablet (iOS, Android)

## Defects reporting template

- Environment URL
- Detailed Description
- Preconditions
- Detailed steps to reproduce
- Actual result (With screenshot added and SQL script if needed to reproduce issue)
- Stacktrace from log
- Codebase snippet
- Expected Result
- Actual Result

## Automation Setup

Create test suite in Gherkin style to cover regression suite for automation (I've specified in feature file in the project directory)

Prepare cucumber test automation framework to automate test for Trade view and New Trade module.

Set up knowledge-sharing sessions to get everyone's input on how to set up the framework and what needs to be automated.

## Test Closure

### UAT

- Prepare UAT document using friendly language
- Run through UAT scripts manually to ensure flows work as expected

### Handover

- Add documentation to internal knowledge base
- Demo feature to support relevant stakeholder
- Ensure that JIRA confluence page is up to date with all details