# Technical Web Test

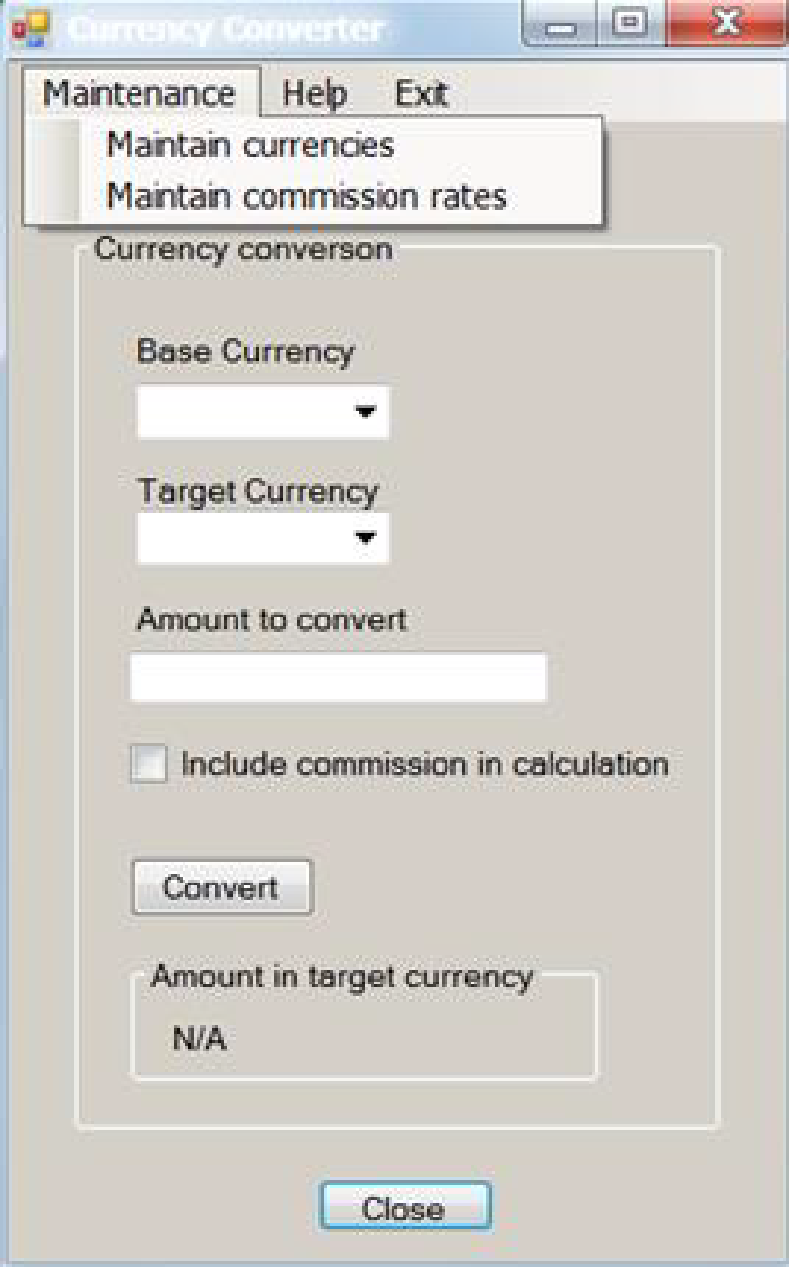
**Question 1**

Introduction:

The following is the main form of an application that converts values from one currency to another. The collection of currencies, exchange rates, and rules for applying commission are held in a shared database, accessed by multiple users of this application on a network. The user of this form selects a base currency, a target currency, specifies an amount (expressed in the base currency), can choose to include or exclude commission, then hits the Convert button to generate the equivalent value in the target currency. That value is shown where “N/A” is displayed in the screenshot below.

The separate “Maintain currencies” form (not shown) provides operations for adding, deleting and editing currencies. A separate process constantly updates the database with exchange rates for any recognised currencies (you do not need to worry about how this works, but you do have control over stopping and starting it).

The separate “Maintain commission rates” form (not shown) provides operations for specifying a minimum commission value and a collection of commission rates for different ranges of values. The minimum commission value and ranges are defined in Sterling. For example, for “amount to convert” values in the range 0.01 to 100.00 pounds Sterling, the commission rate is 5%. An example minimum commission value is £10.



In the following questions identifying tests is adequate, details of test data and expected results are not required where they can be inferred. Aim for an answer no longer than 1200 words for this entire section:

**(a) Using the database and the GUI, how would you test the functionality of this form?**

### GUI

First I’ll make sure the happy path i.e. the basic expected functionality works.

The user of this form selects a base currency, a target currency, specifies an amount (expressed in the base currency), can choose to include or exclude commission, then hits the Convert button to generate the equivalent value in the target currency.

The table below lists the two Acceptance Tests for this in Gherkin BDD format.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Scenario ID | [**Scenario1: User can convert base currency to target currency including commission**](https://studio.cucumber.io/projects/259492/test-plan/folders/1854921/scenarios/6090520) | | | | |
| 1 |
| Description |  |  | | | |
| Scenario steps | | | | | |
| Action | Expected Result |  |  |  |  |
| **Given** the currency conversion form is loaded |  |  |  |  |  |
| **When** the user selects a base currency |  |  |  |  |  |
| **And** selects a target currency |  |  |  |  |  |
| **And** specifies an amount (expressed in the base currency) |  |  |  |  |  |
| **And** includes a commission |  |  |  |  |  |
| **And** hits the convert button |  |  |  |  |  |
|  | **Then** the user can see equivalent value in the target currency in the field |  |  |  |  |
|  | And the commission amount is added correctly |  |  |  |  |
|  |  |  |  |  |  |
| Scenario ID | [**Scenario2: User can convert base currency to target currency excluding commission**](https://studio.cucumber.io/projects/259492/test-plan/folders/1854921/scenarios/6090529) | | | | |
| 2 |
| Description |  |  | | | |
| Scenario steps | | | | | |
| Action | Expected Result |  |  |  |  |
| **Given** the currency conversion form is loaded |  |  |  |  |  |
| **When** the user selects a base currency |  |  |  |  |  |
| **And** selects a target currency |  |  |  |  |  |
| **And** specifies an amount (expressed in the base currency) |  |  |  |  |  |
| **And** excludes the commission |  |  |  |  |  |
| **And** hits the convert button |  |  |  |  |  |
|  | **Then** the user can see equivalent value in the target currency in the field |  |  |  |  |
|  | **And** the commission amount is excluded from the calculation |  |  |  |  |

#### Data input considerations for the GUI

In the above two Acceptance tests, I’ll include the following test criteria:

* Select valid input (currencies and commission) in the allowed lowest range to highest range, if allowed select zeros, and confirm the system can handle errors gracefully such as division by zero if commission field allows.
* Test with different currency combinations and make sure they are consistent in converting.
* Invalid input negative, alphanumeric and special characters in the “Amount to convert” field if allowed and make sure the system is throwing warning on these and handles errors gracefully on hitting the convert button
* Make sure currency symbols are displayed correctly (verified on a subset of currencies)
* Change the localisation settings settings and validate the corresponding translated languages and special phonetic characters are displayed correctly for a subset of test scenarios

**Additionally following tests will be considered:**

* The UI/UX improvement items such as accessibility of the screenshot is not obvious, validate against WCAG 2.1 type of standards using automated check with something like Ka11y
* If a Web form is under test, I’ll validate response rate with Google page speed insight or similar tools
* Validate that all the form dropdowns load and are populated with expected values.
* On clicking a dropdown item it gets selected on the form field.
* On clicking convert it converts the input values to expected output values.
* I’ll make sure the links to Help and Exit sub-tabs are functional and take to the expected pages with contents.
* I’ll make sure the include commission checkbox gets applied and removed from the calculation on clicking convert respectively.
* The close button should close the form and land to the expected page. Same functionality on clicking Exit and “X” on to top right of the window.
* The Minimize and Maximize button works as expected. Amount in target currency loads on converting valid values.
* I’ll also test the form with multiple users concurrently logging in while using the form functionalities and observe any functionality or performance issues due to this. This could be done by automated simulated users or in synchronised exploratory testing sessions with a whole team logging in and testing the form functionality in a session
* Make sure the form is responsive in any supported OS, screen resolution, and orientations,
* Validate OWASP Top10 checks passes
* Check if the form persists state on disconnecting and reconnection to a network
* Test the form performance under low bandwidth connections
* For an equivalent web-based form, test on multiple hardware, mobile, OS’s and major supported browsers.

### **Database**:

* I’ll make sure when I query the DB for input value calculation with and without commissions, it returns the converted values correctly and matches with what's displayed in the GUI
* I’ll query for invalid values and expect the DB to return the graceful error messages as displayed on the GUI

(b) What tests would you perform to test the operations supported by the “Maintain currencies” form?

* I’ll perform the three allowed operations viz. adding, deleting and editing currencies on the Maintain currencies form and then make sure added and edited currencies are displayed correctly on the test under section (a) while deleted currencies do not get populated
* Delete all the currencies (if allowed) and repeat tests in (1) to make sure appropriate warning/error messages are displayed
* Add duplicate currencies (should not be allowed but if allowed see how the system handles these i.e. does it fail gracefully with appropriate error messages or not.
* I’ll stop the service and validate if the add, delete, and edit operations are allowed in this state and if so make sure these operations do not get reflected with the corresponding tests in section (a) while the service is stopped.
* Once the service is started any operations performed should get updated e.g. a new currency added should get displayed on the first form and similar tests for other operations should pass.

**(c) What tests would you perform to test the operations supported by the “Maintain commission rates” form?**

* I’ll specify a minimum commission value and select commission rate for the corresponding ranges and make sure the first form displays the values correctly while testing with Sterling
* Check the first form with other currency combinations than Sterling and check if the commission value calculated in Sterling in this form is reflected correctly on the set currency conversion calculations on the first firm
* Test with specifying the minimum and maximum values as zero and very large numbers and validate the system can handle these without freezing or throwing unknown errors etc.

**(d) What tests would you perform on the form above in order to test accessibility?**

* I’ll do an automated check against any expected standards e.g. WCAG 2.1 using something like MS Inspect
* Validate the form field using a screen reader, check text alternatives, validate contrast
* Validate keyboard navigation works as expected e.g. tab to select, enter to submit etc. are working
* Enable accessibility mode and validate contrast and colour combinations are designed with users with visual impairments

(e) If you were automating the testing of this form for regression test purposes, what would you hope the developer had done that would make the automation easier?

* Developer should add data-testids for all the element selectors and use consistent coding convention
* Included good coverage of functionality tests at the lower level of the test pyramid, so that we need to only automate the happy-path/core user journeys in the e2e regression pack
* Provide mock and stubs e.g. using PACT contract, so that we don’t have to rely on real API responses while validating FE functionalities, especially while running these tests as part of CI/CD
* Map test IDs for a test management tool e.g. TestRail that will make the process of requirement mapping automated and provide fast feedback.
* Integrate CI/CD test reporting tools such as Allure for better comprehension of the test outcome

(f) If this application were used internationally, what further tests would you perform?

* I’ll check the l10n behaviours as to when a specific localisation is applied, the form and the contents are translated and no english content is displayed.
* If available reach out to localisation team/native speakers verify the major languages are translated correctly
* Use automated l10n SaaS solutions for validating the form and content across pages

(g) Looking at the screenshot of the form above, what user interface issues are immediately apparent (you may have mentioned some of these in previous answers)?

* Form title is not visible and will fail accessibility checks.
* The spacing between the characters e.g. “Help” & “Exit” headers are not consistent.
* The usefulness of having a specific “Exit” link on the form is not obvious

**Question 2**

For the following technical test, you can use any JavaScript framework of your choice. We want you to automate the scenarios below using Chrome browser.

1. Open the following URL: <http://www.willistowerswatson.com/ICT>
2. Change the language and region from top left corner to United States English, see example snippet.



1. Search for the word “IFRS 17” using the search box
2. Validate if you have arrived on the result page
3. Check if the result is sorted by “Date”. If not, sort by “Date”
4. Use the “Filter by” functionality and set Content Type to “Article”
5. Validate that each article in the list displays a link that starts with “https://www.willistowerswatson.com/en-US/”

**Evaluation** will be done on the following criteria:

* Clean and tidiness of the solution
* Design and architecture
* Readability
* Stability of the tests

Please provide instructions for setting up the environment and how to run the tests. Share your code either in a public Git repo or compress the files and mail them back.