**Test Plan: Currency Converter App**

### **Introduction**

This test plan outlines the testing approach for the Currency Converter mobile app. The app allows users to convert currencies with two main fields: "Sell" for setting the amount of currency to sell and "Receive" for selecting a currency to receive.

**Purpose**

The purpose of this test plan is to outline the testing approach and activities for the Currency Converter app to ensure that it meets functional, usability, performance, and compatibility requirements.

**Scope**

The scope of this test plan covers testing activities for the Currency Converter mobile app across the following areas:

* Functional Testing
* User Interface (UI) Testing
* Compatibility Testing
* Performance Testing

**Out Of Scope**

* Security Testing
* Localization Testing

**Test Objectives**

* Validate the accuracy of currency conversion.
* Ensure proper handling of user inputs and validations.
* Verify the usability and intuitiveness of the app's UI.
* Assess the app's performance under different network conditions.
* Ensure compatibility with various mobile devices and platforms.

**Test Environment**

The testing environment includes:

* Mobile devices: Android smartphones and tablets.
* Operating systems: Android (latest version).
* Network conditions: Wi-Fi, 3G, 4G, and offline mode.
* Test tools: Testing frameworks, emulators/simulators, network simulators.

Testing the “Currency Converter” mobile app can be done effectively by focusing on these key areas:

**Installation Testing:**

* Verify that the app can be installed, updated, and uninstalled smoothly from app stores (e.g., Google Play Store) and side-loaded onto devices without any installation issues.

**UI/UX Testing:**

* Evaluate the app's user interface (UI) and user experience (UX) design by testing navigation flows, button sizes, touch gestures, and overall responsiveness on mobile touchscreens.

**Functional Testing:**

* Input Validation: Ensure that the app handles various input scenarios gracefully. Test with valid and invalid inputs to check for proper error handling.
* Currency Selection: Verify that the list of available currencies is displayed correctly and that the user can select any currency from the list.
* Conversion Accuracy: Test the conversion accuracy by comparing results with real-time exchange rates or known conversions.
* Conversion Limits: Check if there are any limits on the amount of currency that can be converted.
* Currency Symbols and Formatting: Confirm that currency symbols and formatting are displayed correctly based on the selected currencies.
* Decimal Handling: Test how the app handles decimal inputs and outputs.

**User Interface (UI) Testing:**

* Layout and Design: Ensure that the UI elements are displayed correctly and are visually appealing on different screen sizes and orientations.
* Usability: Verify that the app is intuitive to use and that users can easily understand how to perform conversions.
* Accessibility: Test the app's accessibility features to ensure that it can be used by people with disabilities.

**Integration Testing:**

* API Integration: If the app relies on external APIs for currency conversion rates, test the integration to ensure that it fetches accurate rates and handles errors gracefully.
* Network Conditions: Test the app's behavior under various network conditions such as poor connectivity or no internet connection.

**Performance Testing:**

* Response Time: Measure the time taken by the app to perform currency conversions under normal and peak load conditions.
* Measure the app's performance metrics such as response time, loading time, and resource usage (CPU, memory, battery) on mobile devices under varying network conditions (3G, 4G, Wi-Fi) and usage scenarios.

**Compatibility Testing:**

* Device Compatibility: Test the app on different devices (smartphones and tablets) and operating systems (iOS and Android) to ensure compatibility.

**Localization Testing:**

* Language Support: Test the app with different language settings to ensure that all text and UI elements are localized correctly.

**Regression Testing:**

* Test Updates: Whenever the app is updated with new features or bug fixes, perform regression testing to ensure that existing functionality is not affected.

**Interrupt Testing:**

* Test how the app behaves when interrupted by incoming calls, messages, alarms, or system notifications, ensuring it gracefully handles interruptions without data loss or crashes.

**Offline Functionality Testing:**

* Verify that the app functions correctly in offline mode, allowing users to perform essential tasks such as accessing cached data, creating content, and initiating transactions without an internet connection.

**Test Cases**

1. Functional Testing:

* Verify input validation for the "Sell" and "Receive" fields.
* Test currency selection from the list of available currencies.
* Validate conversion accuracy against real-time exchange rates.
* Check for conversion limits and handling of large amounts.
* Verify the correct display of currency symbols and formatting.
* Test decimal handling for inputs and outputs.

1. User Interface (UI) Testing:

* Evaluate layout and design for consistency and aesthetics.
* Verify ease of navigation and interaction.
* Test responsiveness and touch gestures on mobile devices.
* Assess accessibility features for users with disabilities.

1. Performance Testing:

* Measure response time for currency conversions.
* Test app performance under various network conditions.
* Assess the impact on device battery consumption.

1. Security Testing:

* Validate app behavior under rooted devices.

1. Compatibility Testing:

* Test the app on different devices with varying screen sizes and resolutions.
* Verify compatibility with different versions of Android.
* Conduct cross-browser testing for web versions, if applicable.

1. Localization Testing:

* Verify language support and correct display of localized text, if applicable.
* Validate currency symbols and formatting for different regions.

**Test Execution**

* Execute test cases manually on physical devices and emulators/simulators.
* Use automated testing frameworks for repetitive and regression tests, if applicable.
* Record test results, including pass/fail status and any defects found.

**Defect Management**

* Report any defects found during testing using a standardized format.
* Include detailed information such as steps to reproduce, expected vs. actual results, and severity.
* Track defect status and resolution progress using a defect tracking tool.

**Test Reporting**

* Create test reports summarizing test activities, including test coverage, pass/fail status, and any identified issues.
* Provide recommendations for improvements and prioritize them based on severity and impact.

**Test Sign-Off**

* Obtain approval from stakeholders based on the test results and recommendations.
* Confirm readiness for app release based on agreed-upon criteria.

**Conclusion**:

This test plan aims to ensure the Currency Converter mobile app's quality, reliability, and user satisfaction by thoroughly testing its functionality, usability, performance, compatibility, and localization aspects. Adhering to this test plan will help identify and address any issues before the app is released to end-users.

**Checklist:**

| **Summary** | **Pass/Fail** |
| --- | --- |
| Verify input validation for the "Sell" field. | Pass |
| Verify input validation for the "Receive" field. | Fail |
| Test currency selection from the list of available currencies. | Fail |
| Validate conversion accuracy against real-time exchange rates. | N/A |
| Check for conversion limits and handling of large amounts. | Fail |
| Verify the correct display of currency symbols and formatting. | Pass |
| Test decimal handling for inputs and outputs. | Fail |
| Evaluate layout and design for consistency and aesthetics. | Fail |
| Verify ease of navigation and interaction. | Pass |
| Test responsiveness and touch gestures on mobile devices. | Fail |
| Assess accessibility features for users with disabilities. | N/A |
| Measure response time for currency conversions. | Pass |
| Test app performance under various network conditions. | Pass |
| Assess the impact on device battery consumption. | Pass |
| Validate app behavior under rooted devices. | N/A |
| Test the app on different devices with varying screen sizes and resolutions. | Pass |
| Verify compatibility with different versions of Android. | Pass |
| Conduct cross-browser testing for web versions. | N/A |
| Verify language support and correct display of localized text, if applicable. | N/A |
| Validate currency symbols and formatting for different regions. | N/A |

**Test Report:**

Exploratory Testing of Currency Converter Mobile App.

**Date:** 23.04.2024

**Tester:** Tester Testerson

**Objective:**

The objective of this exploratory testing session was to evaluate the functionality, usability, and performance of the currency converter mobile app. The testing focused on identifying potential defects, usability issues, and areas for improvement.

**Testing Environment:**

**Device:** OnePlus 10 Pro/Oxygen OS v14.0

**App Version:** v1.0

**Findings Summary:**

During the exploratory testing session, various aspects of the currency converter app were evaluated. The testing revealed several findings related to functionality, usability, compatibility, and interrupt. Overall, the app demonstrated satisfactory performance with some minor issues identified.

**Total defects found (9):**

* Critical defect:

1. The app crashes when BTC currency is selected.

* Minor defects:

1. The horizontal scroll is absent for the "My Balances" field.
2. The Submit button overlaps the 'Sell' field when the keyboard is opened.
3. It is possible to type characters into the "Receive" field.
4. The "Receive" field is editable.
5. It is possible to convert the same currency.
6. The Submit button is active for the empty field.
7. The "Receive" value is truncated when the maximum "Sell" amount is reached.
8. Scroll is unavailable after screen rotation.

### **Suggestions for Improvements:**

1. The "Receive" field should be uneditable.
2. The "Submit" button should be active after the “Sell” field is filled.
3. Should the commission fee be equal to the converting amount?

**Bug Reports:**

**Preconditions:**

1. The app is installed on the device.
2. The app is opened on the device.

| **Summary** | **STR** | **Actual vs Expected** | **Evidence** |
| --- | --- | --- | --- |
| Horizontal scroll is absent for the "My Balances" field. | 1)Convert any 5 currencies. | **Actual:** Only 4 results are visible.  **Expected:** All 5 results should be Visible. |  |
| The Submit button overlaps the 'Receive' field when the keyboard is opened. | 1) Tap on the Sell field to trigger the keyboard. | **Actual:**  The Submit button overlaps the 'Receive' field.  **Expected:**  The Submit button shouldn’t overlap the 'Receive' field. |  |
| The app crashes when BTC currency is selected. | 1) Tap on the Sell currency.  2) Select BTC currency. | **Actual:**  The app crashes.  **Expected:**  The BTC currency is selected and the user is navigated back to the main screen. |  |
| It is possible to type characters into the "Receive" field/The "Receive" field is editable. | 1) Tap on the Receive field.  2) Type letters and special characters | **Actual:**  The text is typed.  **Expected:** the "Receive" field is non-editable. |  |
| It is possible to convert the same currency. | 1) Set the same currency for the “Sell” and “Receive” fields.  2) Tap on the “Submit” button. | **Actual:**  It is possible to convert the same currency.  **Expected:** The Submit button is inactive. |  |
| The Submit button is active for the empty field. | 1)Leave the Sell field empty.  2) Tap on the Submit button. | **Actual:**  The Result modal pop-up is displayed. **Expected:**  The submit button is inactive. |  |
| The "Receive" value is truncated when the maximum "Sell" amount is reached. | 1)Type more than 9 digits into the Sell field. | **Actual:**  The "Receive" value is truncated.  **Expected:**  The "Receive" value is visible. |  |
| Scroll is unavailable after screen rotation. | 1)Rotate the device. | **Actual:**  Unable to scroll the screen to the fields.  **Expected:**  It is possible to scroll the screen in a horizontal rotation. |  |