

lANGUAGE tRANSLATOR

Detailed Report



User testing provides vital insight in the nuances that have been overlooked in the application development cycle. Four subjects were analyzed, and many findings came to light. This report will discuss the testing methodologies and the experiences associated with usability testing.

# Testing Methodology:

Usability testing was the prime focus. Many of the subjects have come to appreciate the subtlety and that actions speak louder than words. Observing their reactionary behavior and conduct provided invaluable points of interest missing in my application that needed further investigation. This methodology strictly discouraged any sort of communication between the subject and tester. During this process, the idea was to simulate an environment where a program is released and the end user is rarely supervised and must rely on the how the applications performs conceptually. Since the language translator was a minimal viable product, this had reduced the testing efficacy and as a result, lead to future errors in the application. This was reflected in the overall impressions of many of the experimenters. The prohibition of interference with the tester further exacerbated the difficulties faced with the system. Their operation of the program was studied by noting task completion times, program breaking error and missteps. Some could quickly pick up the program navigation while mostly lagged. The programs user interface made the process of completing the tasks longer than the expected.

# Findings:

The overall feedback received from the usability testing can be categorized into the universal principals of visibility, affordance, constraints, mappings and feedback.

## Affordance:

Affordance was a key issue observed in the task analysis. The users were confused about the properties of the buttons. Specifically, many were not sure about the buttons function due to their special arrangement in the page. This was a huge factor, as the button contained in its location did not afford the act of translating the text. Also, many were concerned about the existence of one drop down list. Some were not able to fathom this concept as much time was spent in limbo before any user interaction was made. The characteristics of a translator apps assume the user to select a source and destination language. The text hint was believed to be enough to just type and let the app automatically detect the texts origin. However, this backfired when simply that was not they had in mind.

## Mappings:

Considering the inherent nature of the minimum viable product, there was a degree of disregard for natural mapping. Primarily the special mapping was lacking. After some time of scouring the four edges of the screen, the experimenters had realized the translation button was obscured by the soft keyboard. In addition, the application lacked the logical positioning of the translation icon in relation to the edit text.

## Constraints:

A reoccurring problem was the lack of proper constraints contained in the system. If the users decided to activate text to speech on an empty translation the application would break. So was the perpetual spelling mistake that would give an error. A reactionary process is to disable specific screen controls to guide the user through the translation process

## Feedback:

The limitation of the application was evident in the lack of feedback provided. The users would enter a non-sufficient amount of characters, however as a symptom of agitation they stared down the screen hoping for the text to appear. Also, sometimes the emulator would refuse to show the keyboard upon starting the activity that lead the experimenters to much disarray.

## Visibility:

Lastly the application posed many visibility issues. Firstly, it’s icons misrepresented their purpose aesthetically and thus did not convey its correct purpose. This was exemplified by the sheer amount of issues users has had while moving their cursor back and forth confused as to what they wanted for lunch.

Ultimately, the user testing showcased many areas that my application needed improvement. The power of deduction from mere observation provided valuable feedback that is imperative to building robust applications.