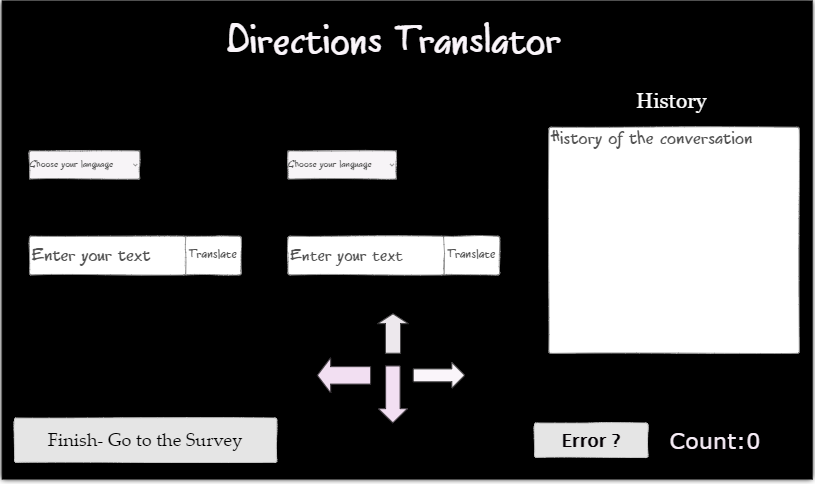
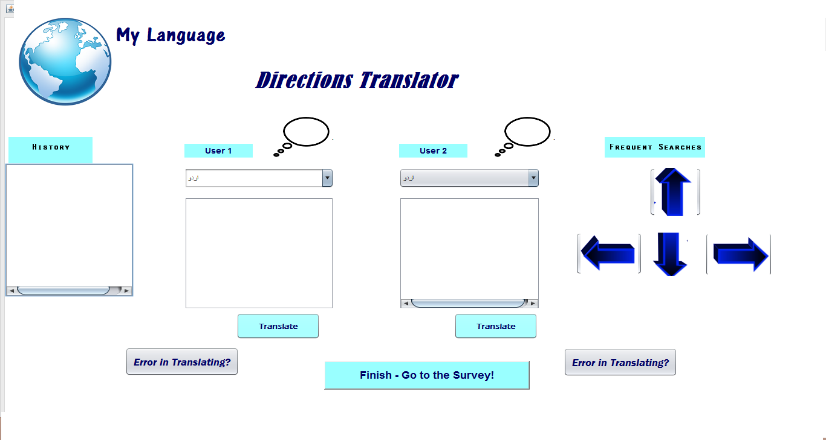
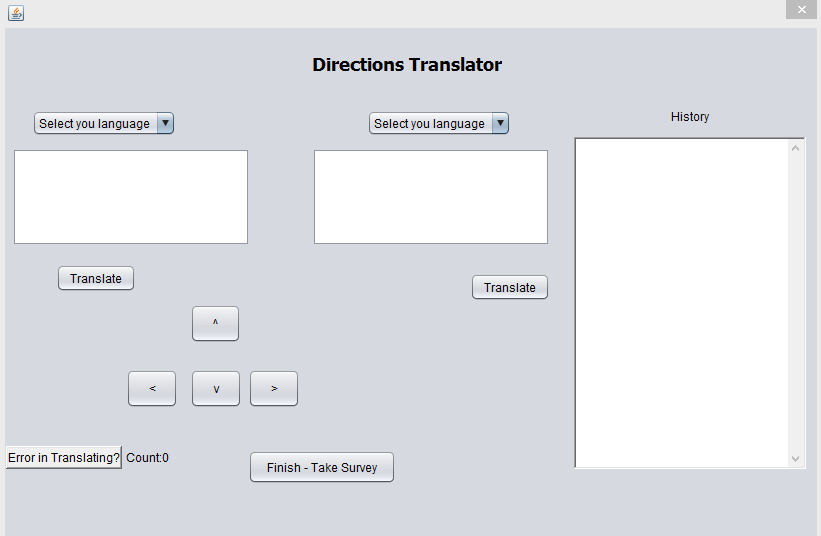
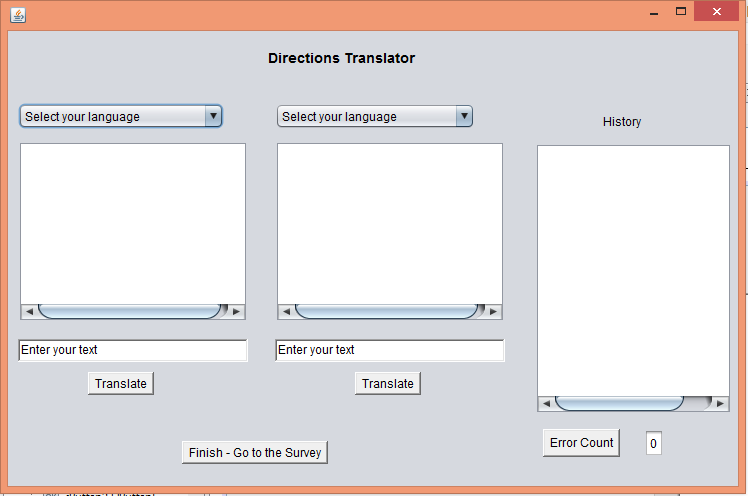
Link to all the **screenshots**: <http://invis.io/AS24NXUHV>

**Use-Case 1**   **Use-Case 2**

|  |  |  |
| --- | --- | --- |
| Use Case | Translate from one language to another | Log translation errors |
| Primary actors | Users who use the system and know two different languages | Users who use the system and know two different languages |
| Goal in context | User1 enters text in his language into the translator, output is in the target language understandable by User2 | User clicks on “Error in translating?” button, system logs the issue |
| Scope | System | System |
| Level | Summary | Summary |
| Environment | University of Florida Campus | University of Florida Campus |
| Preconditions | We have connection to the internet at all times to make calls to the Google API | The translation result is not understandable for at least one user |
| Minimal Guarantee | Some translation output is visible | The error button is visibly pressed |
| Satisfaction | User1, User2 have understood the translation and replied.User1 reached his destination. | The count of the errors is displayed to the user. |
| Equipment | Computer with access to the internet and Directions Translator | Computer with access to the internet and Directions Translator |

**Design decisions made based on user comments:**

* The interface was made more colorful and attractive.
* The frequent search section having the arrow key buttons was added to the application, which made it user-friendly.
* The dropdown menu was made to display all the languages written in their own scripts, so that it becomes easier for the user to identify their languages.
* The error count was displayed to the user on every click to show the program was keeping count of the errors.
* The conversation was made easier by using two-way translations, where users would never have to swap languages like it is done in a one way translation
* User 1 and User 2 when select their language, all the labels and buttons on their side of the screen would be changed and translated respectively according to their selected languages.
* The chat history feature was added which made it easier to keep a track of the conversation between the users.

**Hypothesis**

Hypothesis (H1): My translator interface is at least as good and user friendly, if not better than my partner’s interface.

Null Hypothesis (H0): My partner’s translator interface is more user friendly than mine.

**Metrics**

* The users were asked to objectively rate their experience on a scale of 1-10.
* The error counts were noted per user.
* Each user could specify the number of errors faced and also state which errors were faced during the translation. The results for the error count will be discussed in Data analysis.
* The users also gave the following feedback in the qualtrics survey:

**Negatives**: The chat history was small, languages not sorted in dropdown, layout was not great

**Positives:** The arrow-image buttons, the dropdown box showing languages in their respective languages, translation of the labels into selected languages, the chat history box

**Procedure**

* The students of the HCI class were paired up randomly such that at least one student knew a language that the other did not.
* They were randomly assigned eight interfaces to test and evaluate.
* The evaluation was standardized by using the same use case once on every interface for a given time interval. This removed any chance of some interfaces being tested more rigorously.
* The random nature of allocation removed any bias in the evaluation.
* On completion of the evaluation, the users were directed to qualtrics survey web-page to record their experience. The survey contained objective as well as subjective questions.
* The users entered feedback individually for the objective and along with their partners for the subjective questions.

**Data Analysis**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Mean | Median | Mode | Standard deviation | Variance |
| My interface | 7.125 | 7.5 | 8 | 1.8850919 | 3.55 |
| Elizabeth’s interface | 7.875 | 8 | 9 | 1.13 | 1.267 |

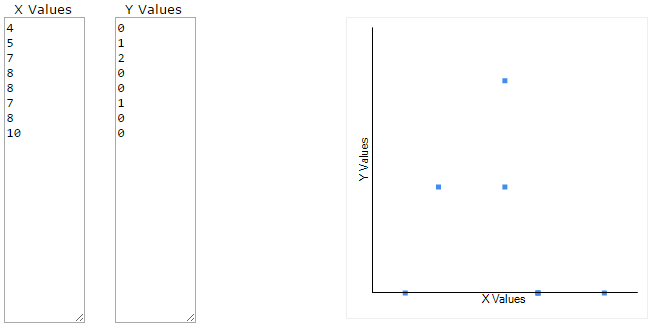
|  |  |  |  |
| --- | --- | --- | --- |
| Meana-Meanb | T | df | p(one tailed) |
| -0.75 | -0.97 | 14 | 0.1742535 |

If t value was larger than 1.96, we could have rejected the null hypothesis.  
Since it is lesser than 1.96, we should check the p value.

|  |  |  |  |
| --- | --- | --- | --- |
| df1 | df2 | F | P |
|  |  |  |  |

If the p value is less than 0.05 we can reject the null hypothesis.  
Since the p value is also greater than 0.05, we cannot reject the null hypothesis based on this collection of data. However this is not equivalent to accepting the null hypothesis.

According to the small sample of data collected it would seem that the interface of my partner is more user friendly than mine but it would be incorrect to assume that this is the case as we do not have evaluations from the same set of students. It would be incorrect to assume that our evaluators were exactly alike. It is possible that as the number of evaluators increase more students may prefer my interface. It is also a possibility that given the same set of evaluators that my interface may have been rated as good as the other one.



The value of R is -0.2506. Although technically a negative correlation, the relationship between your variables is only weak (the nearer the value is to zero, the weaker the relationship).

The value of R2, the coefficient of determination, is 0.0628.

**Conclusion**

**Bias**:

A systemic bias may have been introduced by using the google translate API, in case it could not translate certain words of a language well.

There might have been a learning bias amongst some of the participants and that might have changed their initial feedback on user friendliness of the interface.

It is possible that some of the best features of the apps overshadowed some of the less usable ones or impressed the users so much that they did not concentrate on the less usable features.

It is also possible that I have suffered from experimenter’s bias while analyzing the results.

**Confounds**:

Design confound: Certain users may have not tried all the possible features of all their assigned apps due to impatience, lack of time and therefore this may have resulted in a different rating than what they would have normally rated the interface.

Certain users may have also been faster/slower learners than their peers and this affects the outcome of the experiment.

Population confound: The users were all of a certain demographic (between 18- 40 years of age). This may have resulted in certain trends in the result.

But the level of evaluation done to all the apps was not the same because no two people are equivalent therefore no two groups of people can be equivalent evaluators.

**Design lessons learnt**

During the initial interview phase, I was surprised how my interviewee could think of so many functions like showing history, implementing dropdown which shows languages written in their respective scripts, showing a map and arrow buttons.

Though I was expecting very good evaluations in the class activity, it was surprising that I did around average. Some people didn’t like the frequent search section - arrow button keys and found them strange as they did not think of clicking it initially as no instruction was given.

Some did not like the layout though I felt I was trying something innovative and attractive.

Thus, you will always be surprised by the User’s evaluations!

* I learnt that users love standardization and simplicity. The layout if kept as simple as it was in the beginning of designing phase could have been better for some users as adapting changes is not so easy for them.
* Similarly while in the case of the arrow keys, some users did not try clicking it to observe that they are meant to help with the directions and make their work much easier. This way, I learnt that users love familiarity with the interface.
* The users, when facing lack of time prefer easy and standard interfaces, instead of attractive interface which has more features because they do not have the time to understand or use the additional functions, if they find these features complicated.
* I learnt that one needs to involve more and more users at the initial design phase and testing prototype phase of your development cycle. Without having users guide you during the initial phases, you can never predict how the users will react to different features and changes.