

Use case

Informal use-case

Translate documents

Main Success scenario:

User uses this user friendly application with simple guidelines to successfully obtain translated outputs from the user's file or typed input. In addition, the user is able to save translated files with just one click. The application produces multiple language translations for user's needs.

Alternative Scenario:

User uses this application with no guidelines so that the user is not able to fully utilize this application.

User tries to get translated data via this application, but this app doesn't produce anything.

User selects import file method either to give a file name or to type input. The application doesn't read user input, so this application doesn't work.

User uses this application but output is not accurate and very poor quality.

Fully-dressed use-case

Scope: Translate documents

Level: user: goal

Primary Actor: UI

Stakeholders and Interests:

UI: wants an accurate fast translation.

User: wants to translate into multiple languages and store the output translation data at the same time.

School: wants this application to be useful for the school and for school administrators to manage their job.

Preconditions: build a fast and high quality translation system.

Success Guarantee: this application gives fast high quality translations to user, and makes user's work efficient.

Main Success Scenario:

1. User uploads a file or types input
2. User is asked for the number of languages they want to translate the file into
3. User inputs the number of languages
4. User is asked which languages they want to translate the input into
5. User inputs the names of languages they want their data translated into
6. User gets the translation through the app
7. User is asked if they would like to save the data as a new file
8. User indicates yes or no
9. User is asked if they would like to continue using the app.

Extensions (or Alternative Flows):

- 1a. User uploads a file type that is not recognized by the system
 1. System tells user to input a valid file type
- 1b. User's typed input includes invalid characters
 1. System tells user to input valid characters
- 1c. System does not recognize the language of user input
 1. System tells user to input other data
- 3a. User inputs an invalid number
 1. System tells user to input a number between 1 and 5
- 5a. User inputs and invalid language name

1. System tells the user to check their spelling or input another language
- 8a. User inputs invalid response
1. System tells the user to enter 'yes' or 'no'

Special Requirements:

- Easily readable system
- Outsourced translation system
- Fast translation response
- Dynamic language python
- Fixable simple structure

Technology and Data Variations List:

- User imports data with its keyboard
- UI shows leading questions
- User answers each question of the UI
- UI gives a new translated file to user
- Multiple options for output

Frequency of Occurrence: Could be nearly continuous.

Open Issue:

- Where should this outsourcing translation be collected?
- If the server is down, where should issues be reported?