

Tongues (Concept Video Report)

Ishita Prasad, Alex Wu, and Anna Yelizarova

Tongues is an accurate, real-time translation application powered by the crowd. When automatic translators just can't get it right, Tongues enables users to ask the people around them what the *correct* way to say a word or phrase is. This project is being developed by Alex Wu (Team Manager and User Testing), Ishita Prasad (Visual/Interaction Designer and Documentation Coordinator), and Anna Yelizarova (Developer and Visual/Interaction Designer).

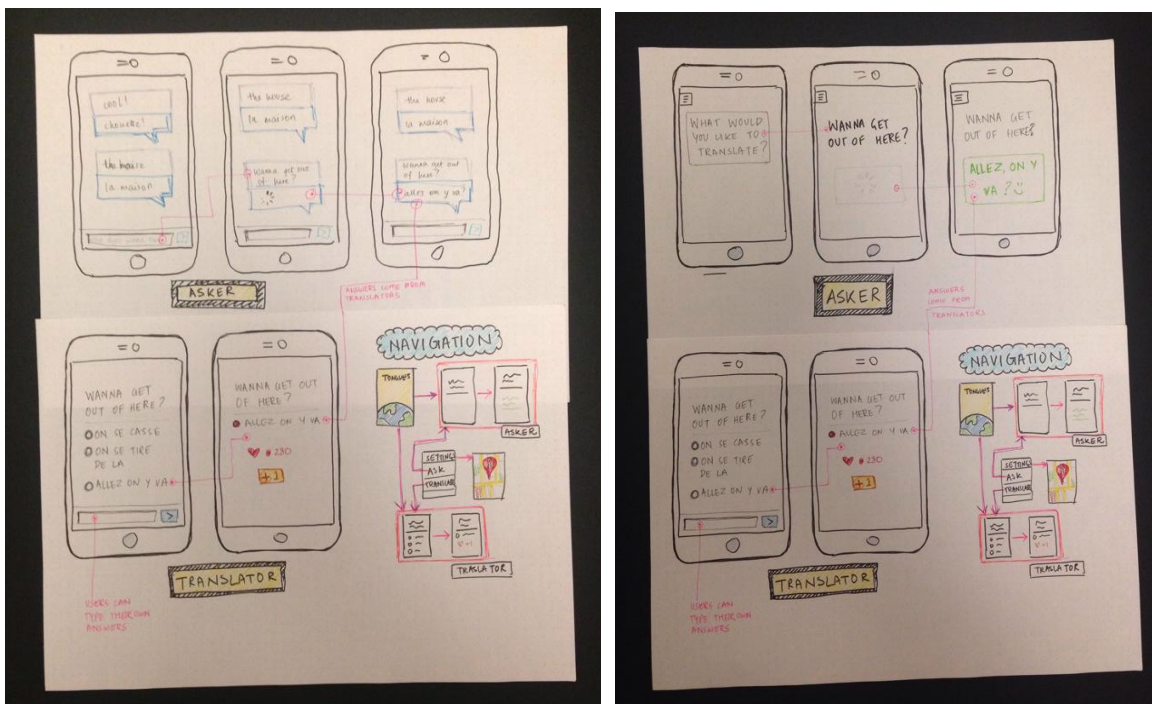
PROBLEM AND SOLUTION OVERVIEW:

When it comes to communicating across languages in day-to-day life, people often need to know how to say the right thing, both fast and accurately. Human-sourced translations are accurate, but aren't ideal for this kind of communication because they cost both money and time. Automatic translators (like Google Translate) were designed to solve part of this problem, but are unfortunately often incoherent and unreliable. How, then, can a person find a way to effectively communicate in a different language while on-the-go?

Our solution is **Tongues**: using the power of crowd technology to enable users to translate snippets of text. By using the knowledge of the masses, users can both find and curate effective translations of useful tidbits of text. Additionally, crowdsourced translations have the power to happen in real-time - a user could request a translation, and have a well-formed phrase sent back right away.

UI SKETCHES:

Here are some UI Sketches we created for a couple of different interface designs:



SELECTED INTERFACE DESIGN:

In the end, we ended up going with a more simplistic, functional design. From our interviews, we discovered that people valued simplicity, ease of use, and speed when it came to translations.

On the asker side, we decided to go with a simple “speech bubble” like interface, where the user types their phrase and immediately gets a response. We went with this design because it is simple, intuitive, and easy to use. It also greatly simplified the search history feature we wanted to include (as opposed to only having one question and translation visible at once), since users can see their previous search queries by swiping up, just like they would for text messages.

On the translator side, we decided to go with a simple upvote system, with the option for a user to enter a custom response. We wanted to keep the user interface simple - in order to avoid overwhelming the translator - so we made the screen as minimalistic as possible, with as few non-essential translator decisions as possible (i.e. they are just given random phrases to translate, and can only upvote an existing answer, create a new answer, or skip the translation).

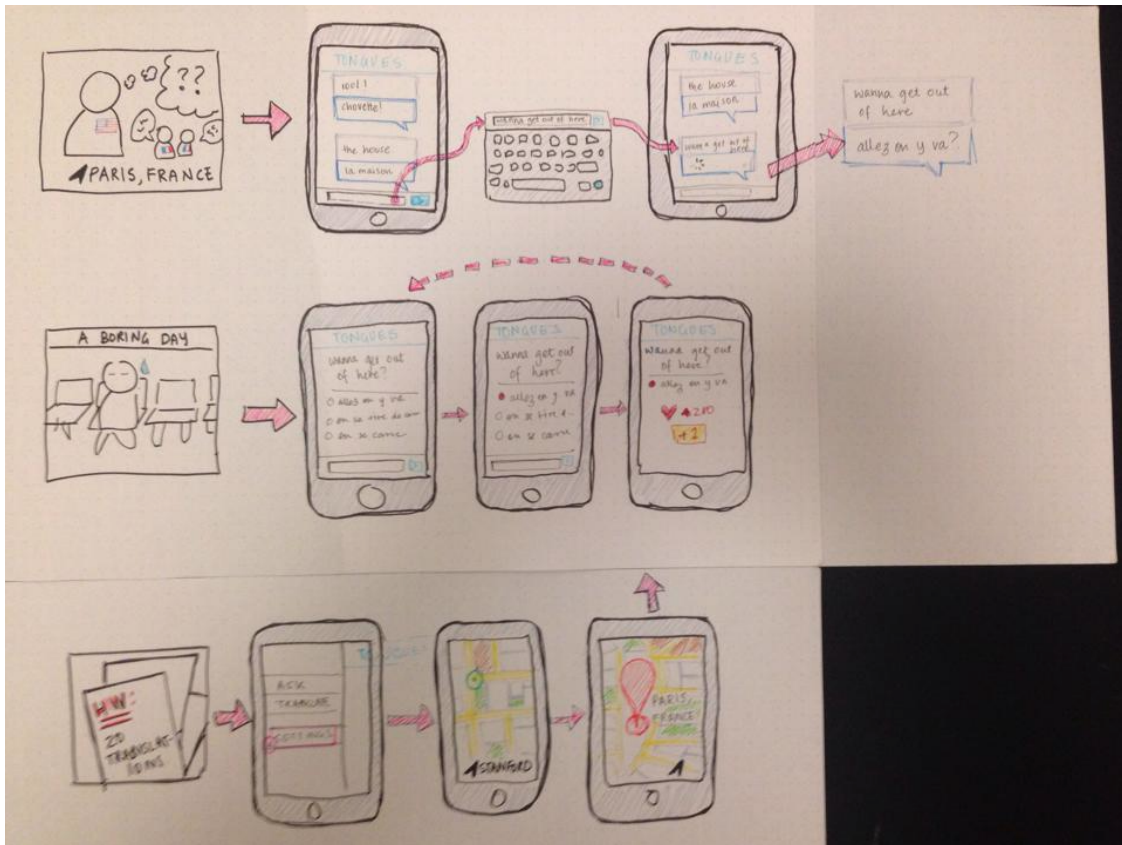
In terms of language selection, we decided to go with a map where the user can choose the region they want (which would default to the region that they are in), as opposed to a drop down list with a giant list of languages. We decided to do it this way because it much better emphasised our colloquial focus on language translation. Region selection makes translations into regional slang much more feasible.

Functionality Summary Table:

	Asker	Translator	Language Selection
Function 1	Allows user to ask questions in a speech bubble interface	Upvote existing answers	Map-based region selection
Function 2	Shows search history	Add new answers	
Function 3	Provides instant translation	Skip translations	

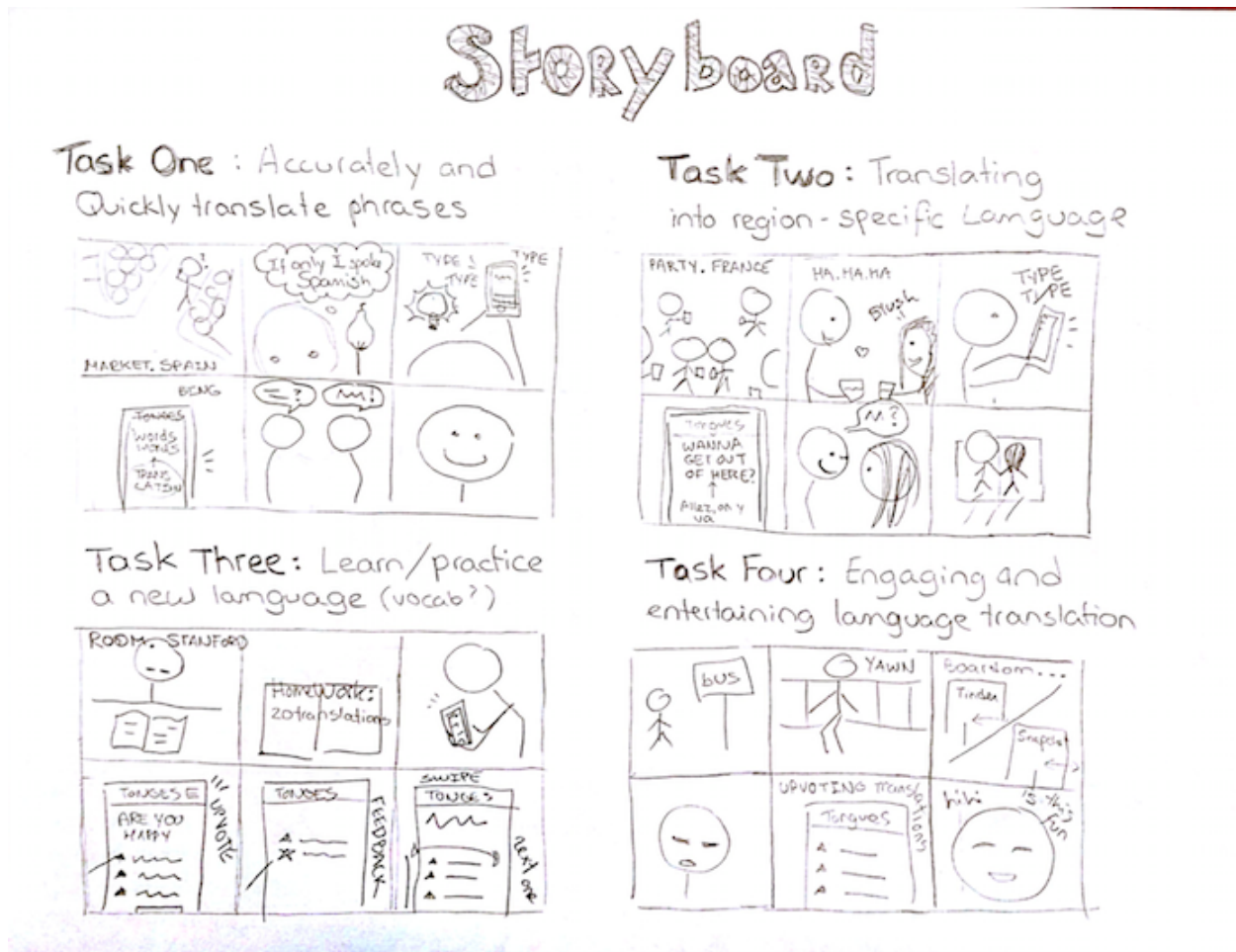
UI STORYBOARDS:

Here are UI Storyboards for three different example tasks carried out with the selected interface design:



VIDEO PLANNING STORYBOARDS:

Here are our Video Planning Storyboards for our three different usage scenarios:



CONCEPT VIDEO DESCRIPTION:

In the end, our concept video ended up being a compilation of four different ways we imagined our app being used. We felt that different types of people would use the app in different ways, prompting us to use different actors for each of the four scenarios (on the bus, at the market, at a party, and while studying).

One of the most difficult parts of filming the video was coordinating our schedules (the the schedules of additional actors) so that we could film all of the necessary scenes. In order to accurately depict the use cases for our product, we needed to film in several different locations, and needed two or more people for every scene (someone to film and one or more people to act). This made finding times to film - especially while coordinating our busy schedules - pretty difficult.

There were a couple of things that ended up working well with this video: filming the video without audio, and filming four separate scenarios with four different actors. Filming the video without audio ended up being a great decision because it dramatically improved our video quality

(versus trying to make our video using the low-quality audio from our camera). Filming four separate scenarios (versus filming it as one person using the app in different ways) ended up working well because - in our opinion - it ended up more clearly showing the variety of scenarios in which the app could be used, as well as the variety of people the app could be used by.

Time wise, design prep took a couple of hours, shooting took 3 - 4 hours, and editing took about 3 hours. A lot of the 3 - 4 hours spent shooting was spent traveling to different locations.