How to Use this Template

1. Make a copy [File → Make a copy...]
2. Rename this file: "**Capstone_Stage1**"
3. Replace the text in green

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"

GitHub Username: Idirer

Description

When you want to learn a new language abroad you often don't understand anything at first. Then you know a word or two, and you can start figuring out the way grammar works.
The app jumpstarts your learning by helping you learn basic vocabulary so you can get a better grasp of the language on the field.

Intended User

The app is intended for English speakers who want to learn a new language, more specifically beginners.

Features

The app:

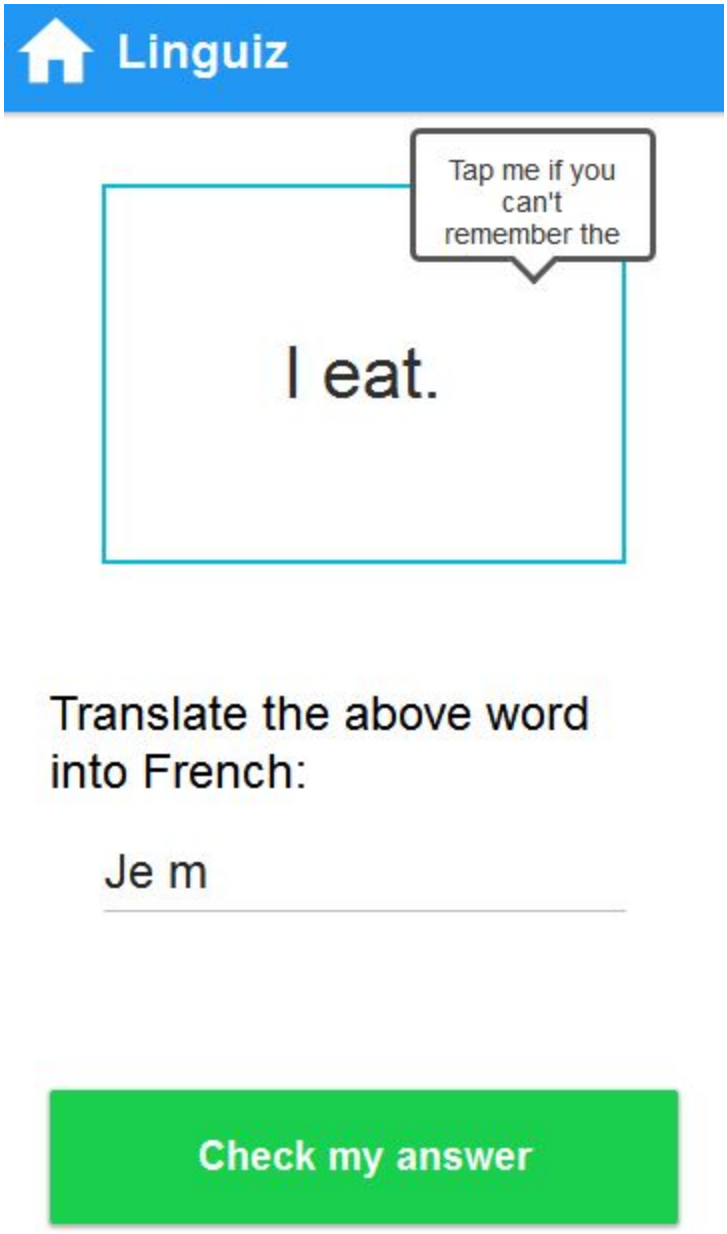
- Allows the user to learn vocabulary using ‘memory cards’ and translation exercises.
- Saves information about the languages the user started learning and the user’s learning progress.
- Adapts the cards based on the user’s progress.

User Interface Mocks

Screen 1

The ‘landing screen’ after Google sign in.

Screen 2

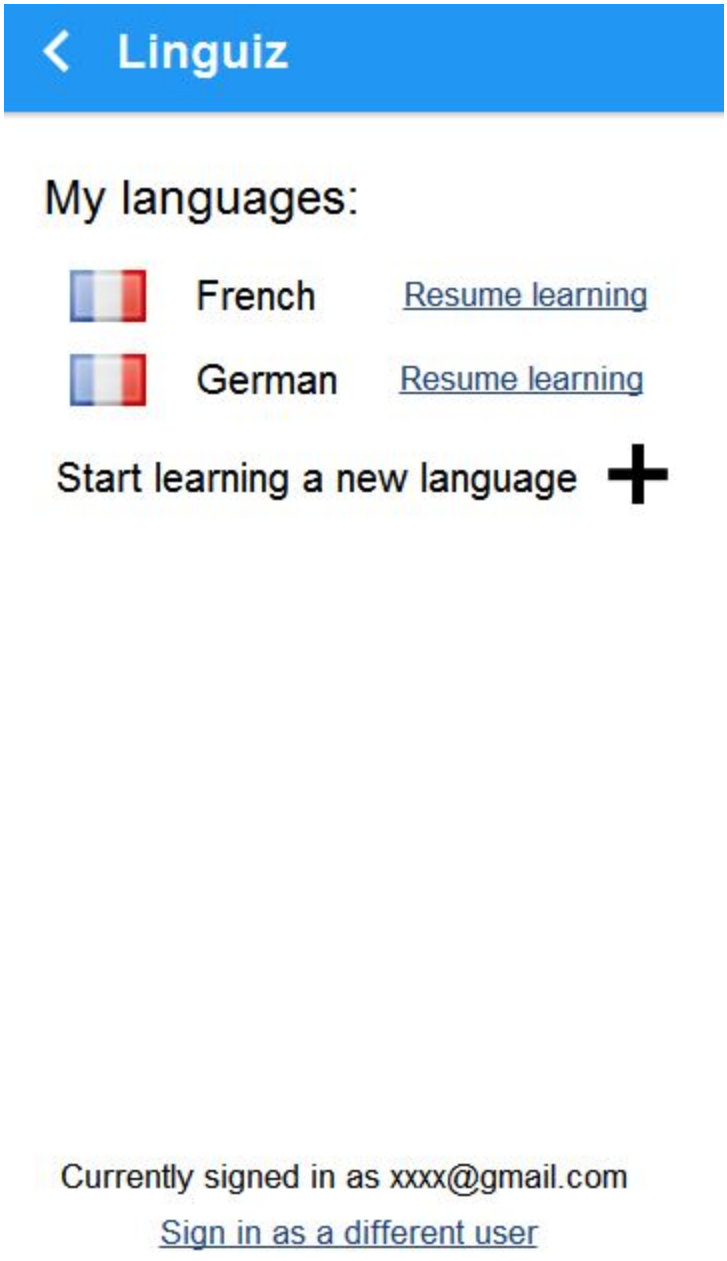


The main playing screen. The card can be flipped by tapping onto it.

Screen 3:

The main playing screen with the card flipped. You can see the translation but you cannot enter new text. You can't check the answer either (The button will be greyed out). The point is to force the user to enter the word from memory (even if it's a super-short-term memory).

Screen 4:



The home page where you can see the languages you are learning and start a new one.

Key Considerations

How will your app handle data persistence?

We want the user to be able to login on another device and keep learning, so we need to store all the progress data on a server.

Additionally when starting a new language the data about this language will be downloaded from the server.
This data will be queried using a rest api.

The app will use an SQLite database to store data locally and provide a fully functional offline experience.

To provide an offline experience we need to store locally the languages that the user can start learning, words with the associated translation for every language the user is learning.

To know which card to show next, we'll store information such as:

* For each word, when the user last saw it (null if the user never saw it).
* The output (success/'failure') for every card: success means the user inputs the correct answer without looking at the back of the card first.
* Number of attempts for a word.

We will have an 'Attempts' table that will store whether the user succeeded for each card. This table will be trimmed every time the data is successfully posted to the server to avoid using too much space on the device.

We will also have to batch requests to save battery life.

I'll build a Content Provider to manage the app data, a SyncAdapter to post data to the server and an AsyncTask that will fetch new data on demand (when the user starts a new language).

Describe any corner cases in the UX.

When playing, the user can touch a card to flip it and see the word's translation. This should be done with a nice animation but degrade gracefully to supported devices.

Describe any libraries you'll be using and share your reasoning for including them.

I'll be using **ButterKnife** since I haven't had a chance to try it yet and it looks good.
I'll also be using **Retrofit** to query my custom backend api.
Finally I might use a library to help with content provider creation. I need to try and see what works before I settle for a specific library though.

Describe how you will implement Google Play Services.

- Google Analytics to record basic statistics.

- Google Sign in. As we need to store and fetch user-specific data from the server, we need some kind of account/authentication.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

- Configure libraries for play services, Retrofit and Butterknife.

Task 2: Implement UI for Each Activity and Fragment

- Build ui for login screen, language selection…
- Build ui for the cards screen.

Task 3: Implement the SQLite database and the content provider

- Write the SQLite scheme, the content provider.
- Link this with the UI.

Task 4: Get some translation data

- I'm not entirely sure how I'll build the words/translation database yet. I'll try to reuse publicly available databases or leverage free translation services (e.g. bing translation or some dictionary).

Task 5: Implement the backend server

- Create db
- Write methods to compute 'probability of appearing next' for a word
- Add rest api

Task 6: Write the SyncAdapter and the AsyncTask

- The AsyncTask should fetch new words on demand for a given language. Used when the user starts learning a new language.
- The SyncAdapter should sync progress data with the server.
- Handle errors: a user needs a network connection to start learning a new language, etc.

Task 7: Implement Google Play Services

- Add Google Signin and coordinate with server.
- Add Google Analytics

Task 8: Implement an App widget

- Add an app widget, for instance with the date the user last played a language potentially with a ‘streak’ count.

Add as many tasks as you need to complete your app.

Submission Instructions

1. After you’ve completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it “**Capstone Project**”
3. Add this document to your repo. Make sure it’s named “**Capstone_Stage1.pdf**”