App Development | Report

Lukas Mikulec

My app *Streetschatz* (Schatz means treasure in German) is there to help people automatize learning of everyday words in German. Simply taking a picture of an object and uploading it to the app will generate the German word of that object. Gone are the times when one had to note down the words in English on paper or in the notes app, translate them to German, and then note them down again for later reference. *Streetschatz* takes care of the noting down, translation, noting down, and storage. In addition, it also helps the user remember the words by implementing a quiz function.

When approaching my assignment, I defined the minimum scope of the project, which was the process of generating German words from the uploaded pictures. This would guarantee that no matter the circumstances, the fundamental part of the app would be ready to be worked with and submitted. I gathered feedback from my potential users (learners of German) who liked the idea. They mainly saw the added value of the app in the speed ("It's quick"), no need to note down the objects ("I don't have to note down the name of the object separately."), and the quiz functionality for better memory retention ("It might be helpful in retaining the name of the word in my memory better."). They also proposed other ways of quizzing than just filling in the word to a blank space within a sentence (current implementation), highlighting the appreciation of gamification. One of the concerns was that the objects in the pictures would not always be correctly identified ("say I want to know the word for cafeteria but it only identifies tables, chairs, windows, wall, pillar... but not the object I actually need"). This is a valid concern and the app sometimes displays content that is not precise, but the identification is run by artificial intelligence models, making it hard for me to improve this aspect in my code. For now, the better the picture the user provides, the higher the chance the object gets correctly identified. This is also the reason why the app reminds the user that "Each picture should clearly show only the object without background distractions".

After I finished developing the core part, I had a feeling that there was still time, space, and passion in me to want to not leave it just there with the German words, also because the preliminary user feedback highlighted the quiz feature as valuable. Due to that, I decided to program the extended functionality with a learning part of the app in which the user would be quizzed on the words they uploaded. The last major milestone was the process of getting the app to the cloud. As it seemed that the code did not work in the cloud the way it was intended, I spent some very focused and intense time figuring out where the problem was and the result of that troubleshooting research was that my app only worked once after being started and could not be rerun (both locally and in the cloud, but locally I did not realize it because I would always run the app anew). As the problem was not in the cloud but in my code, I researched a way how to solve it, rewrote it, and managed to make my app functional without the need to restart.

With the app available online, I asked for the second round of feedback from my test users, now based on the actual app experience. The app was labeled as "very intuitive" by one of the users which made

me happy, as the colors, emojis, little text, layout, and animations were supposed to achieve just that. It was also nice to hear that although "Some of the answers were wrong or not exactly what I was expecting", they were "also surpringly accurate in some instances". I expected this kind of result as the app would have to be much more robust in handling various results of the AI models to work fine nearly all the time. The user also shared the images they uploaded and there were multiple objects on some of them, explaining why some results were not precise. The user also pointed out that the example sentences did not always make the most sense, but that seems to be the beauty of the random sentence generation with AI. Another user had trouble uploading the images because they did not choose them from the storage but used the camera function. This makes technical sense as the upload widget is not intended for this purpose and there is another one for taking pictures using the camera from Streamlit. I did not implement it for three reasons. Firstly, it would make the upload process more complex and lengthier for the user (they would first have to choose whether they want to upload photos or take one). Secondly, the app's UX design should nudge the user to upload more images (at least three), so that the quiz function can be used for better memory retention which is not the case if a user takes one picture. Thirdly, this feedback came once the app was fully functional and ready for release, and due to time constraints (as I was about to leave for HK), I decided not to significantly change the upload process at the end with the potential to have to rework too many lines of code. The speech audio files for quiz sentences were generally appreciated, as they help "with the pronunciations". One more question came about the support of more languages which makes sense, but for which the fundamental app design would have to be changed and new AI models found.

After summarizing the developmental process and before listing the limitations and space for growth in the future, let me touch on the app's design. The user comes to the app and can upload images from their device's storage. After they select them and click submit, a process will start in which the images will be sent to an AI model to identify the object on the image, the most probable result of which will then be sent to another AI model for translation to German. The results are then shown in a grid-like manner, with the ability to check each word's dictionary page on Wiktionary, share it on Twitter (X) with friends, and download a txt file with English and German word pairs for later use. If the user has uploaded at least three images, they will be notified through a pop-up (bottom right) and in the left sidebar menu that they can generate a quiz. Once they do, the German word will be sent to an AI model which will write a random text based on that word (only the first sentence of the text will be selected later), while another model will be used to generate text-to-speech audio. The user will then see sentences with blank spaces at the beginning into which they should put the right word. Once they get the answer correct, they can listen to the sentence. When all answers have been answered correctly, they can see a summary and download the sentences as a txt file on the last page of the quiz with the results.

Due to a shortage of space, I will not go more in-depth, but some of the features that are not directly seen and I would like to mention is the system for reconnection with API when connecting to the models so

that even if the connection is not successful at first, the process will not fail immediately. I also had a lot of fun figuring out session states in Streamlit to dynamically change the content on the screen properly, which was at first a challenge as Streamlit has a peculiar architecture with its principle of reloading the whole code after each user interaction that at first made some unwanted changes to happen and some wanted ones to not.

One significant limitation that I see is the process of selecting only the first sentence from the generated text for the quiz. This is currently done by cutting out everything after the first full stop which does not work perfectly if a sentence contains words with full stops (e.g. St. Anton). This should be the first thing to solve in future releases by creating a more robust system for dealing with the outputs of the generative AI texts. Other improvements include the ability to take pictures (although this would have to be considered also from the philosophical point of view of how the app should work) and a major feature would be the support of other language(s). The quiz feature could also be expanded by providing various question types, however, it is also important to not try to mimic some other gamified platforms for language learning.

For now, I consider *Streetschatz* to be a well-working app that is neither too simple nor too complex and that fulfills its purpose by focusing on what is missing on the market and not trying to be something it is not. I enjoyed making it as it was my first real app, because the Streamlit library is well-developed, and because I even had a chance to employ AI in the process. In the end, it is something I myself can use and can share with my friends as a practical tool for their everyday life in German-speaking countries (for now). Enjoy!

Important:

Toward the end of the process, I noticed a **bug in Streamlit**: sometimes, after generating the quiz, after entering the first answer, the whole page will go blank. This does not generate any error and must therefore be a bug. I spent a lot of time figuring out the potential problem but to no avail. Researching other people's problems (here, here, here, and here) and asking on Streamlit Discuss did not yield any results (the bug can be seen on video screenshots in my post there.) What made the process harder to solve was that no error happened and no root cause was described, therefore, I just tried changing different things. Another piece of evidence of this being a bug is that with the same code, the quiz answering sometimes works and sometimes does not. However, there is a workaround that will result in correct behavior: After generating the quiz, do not type in the answers right away (this might result in a blank page). Go back to the main page mode ("Identify German words") and then go back by clicking on mode "Take a quiz". After returning to the already generated quiz, typing in the answers will not result in a blank screen.