Programming II - Assignment 2. proposal

Summary of product idea

My idea is a language learning web application which incorporates AI for conversation simulations / chats and explanations. The end goal would be to make a tool that even I would gladly use for my language learning journey, for this reason I have decided to only concentrate on English to German learning.

(The idea originated from the Creative Thinking and Design group project, where my group's concept is also about language learning, however that is quite a bit different, it is primarily for Slovakians and Hungarians.)

List of all main features

- Conversation simulation using AI
  - General chat, "Just start with any topic"
  - Or maybe even pre-planned situations (E.g.: restaurant, police stop)
- Word bank
  - A word bank, to where the user can easily add / remove new words from.
  - This word bank then can be used for various ways of memorizing. E.g.:
    - Flash cards - German to English / English to German
    - Maybe even a Der / Die / Das Quiz if dictionary scraping works well enough
  - The word bank would have a more detailed description of the word like an actual dictionary with use-cases, etc.
- Games
  - Single player / Multi player - Wordle style game - based on the users word bank
  - Single player / Multi player - Guess the word from emoji game
  - And some other fun ideas I might come up with.
- Grammar
  - A more interactive grammar cheat sheet
- Reading
  - Materials for each levels of proficiency capabilities of easy add to word bank and explanation from Ai or translation services.

Networking and Database

- Networking functions:

    - The multiplayer games would be a main feature of the application, showcasing networking functionalities using sockets where two clients are connected.
- Database: there is a lot of data needed to be stored with an application like this, here is a non-exhaustive list
    - User data, pfp, credentials
    - Chat history for AI conversations
    - Word bank
    - Leaderboards for the games
    - Additional learning materials stored
    - Etc.

External libraries used and technologies

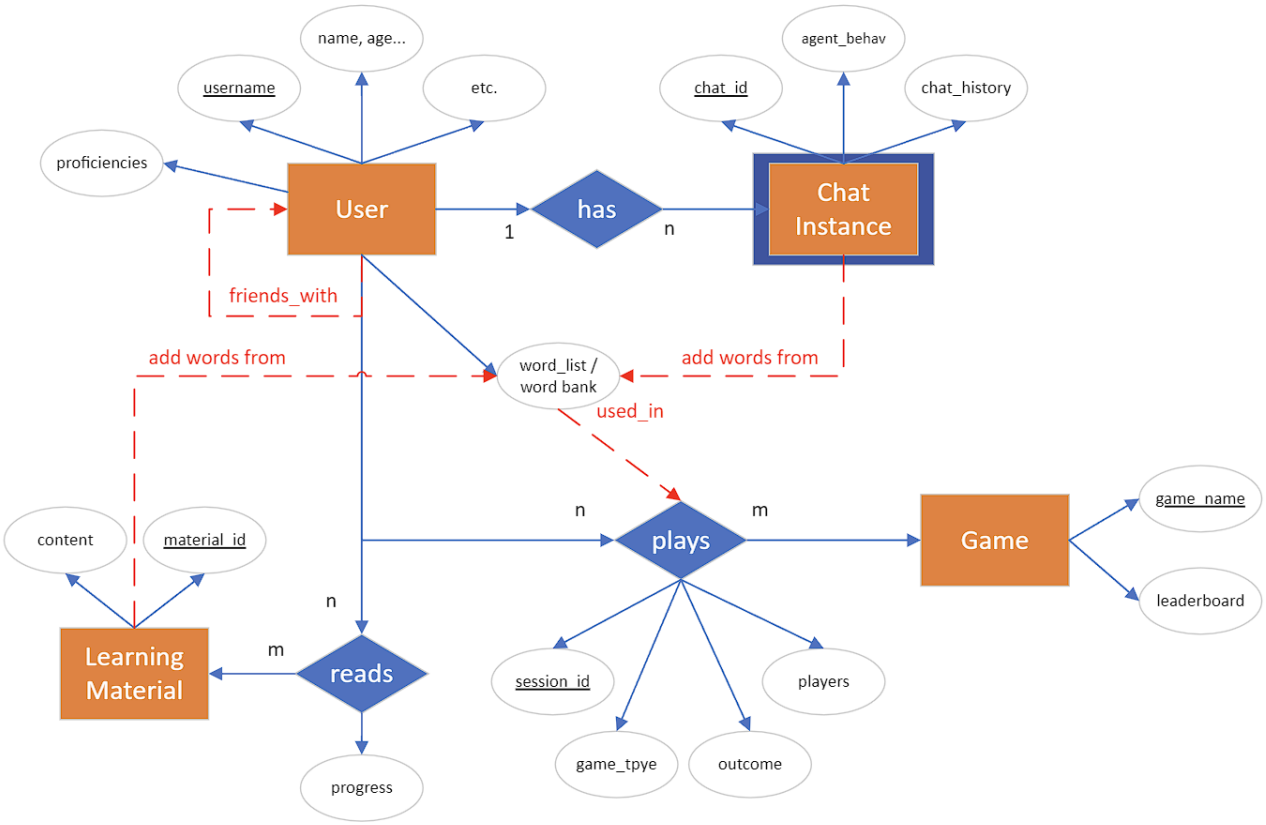
- Backend:
    - Since it has been our main tool for this semester and I have grown quite accustomed to it, I would like to use a Python Flask RESTx backend for my application.
        - I am not sure of all the capabilities Flask has or whether one Flask backend would suffice, but for Websockets I have found that there is a semi-official Socket.IO integration project, which might be worth a look into.
- Frontend:
    - For frontend, obviously an application of this types should be web based, which I hope is not a problem.
    - For frontend frameworks, I have previously worked a bit with React, more specifically NEXT.js, but those are as I have noticed better suited for bigger applications.
        - Thus I have chosen Svelte as my preferred choice. I have seen some short tutorials of it, it seems more lightweight (actually one of the fastest performing frameworks) and has a nice syntax. It was also the second most admired framework in Stack Overflow's 2023 Developer Survey (most loved of 2021) and I would like to try it myself.
        - With this choice I might be giving up some easy to integrate packages by not using React or Angular, but hopefully it turns out alright.
    - This also means that there should be an adequately useable UI for both web and hopefully even mobile.
        - Here I would love to look into Tailwind CSS rather than creating my own CSS from scratch, since we aren't being graded based on looks.
- Hosting:
    - For hosting the backend I have found PythonAnywhere by ANACONDA has a free tier and it is often recommended.

    - And for the frontend I would probably use Vercel since I have already hosted there and it was a breeze to set up.
- Database:
    - For database I decided to use Firebase solutions, not because I think they are the best, but because they offer nice free plans and some additional services I might need, e.g.: offering media storage or easy to use authentication services all in one place.
    - Bare in mind that it is a NoSQL databse solution.
- AI:
    - I am planning on using OpenAI's GPT-3.5 Turbo as that is their cheapest API at 0.5 USD / 1 million tokens, it is not the latest and greatest, but for development purposes it would do it's job nicely. (Also it is very easy to switch from one model to another.)
    - OpenAI also offers nice documentation on how one can use their API.
- Translation services:
    - For a general translator API there are many choices from Google and Amazon to DeepL. I will try and look more into which ones have free tiers / better pricing.
- Dictionary:
    - Unfortunately the solution is not so easy for German as English. For English there is a perfectly adequate free solution like the Free Dictionary API. For German I have found three possible solutions:
        - Using an offline record like dict.cc's downloadable dictionary.
        - Using a dictionary API like Collins.
        - Or even scraping the result of searching the specific word on Wiktionary.

Overview of the planned architecture of the software

I did not want to go into the nitty-gritty of UML and ER diagrams just yet, so I have created an ER-like diagram, showing some main functions of the system and how they interact with each

other.


name, age...
username
etc.
agent_behav
chat_id
chat_history
proficiencies
User
has
Chat Instance
1
n
friends_with
add words from
word_list / word bank
add words from
used_in
content
material_id
n
plays
m
Game
game_name
leaderboard
Learning Material
m
reads
n
session_id
players
game_tpye
outcome
progress