Elderly Virtual Assistant (EVA) Avatar

Design Process Report

Team Members

#	Name	Specialisation	Role
1	Jackson Herbert Sinamenye	Applied AI	Voice/Text Innovator & lead programmer
2	Aleksander Theo Strand	Applied AI	Project Visionary Leader
3	Vebjørn Berstad	Applied AI	Text Processing Specialist
4	Pratima Kumari	Applied AI	Voice/Text innovator
5	Mubariz Ahmad Rai	Applied AI	Digital Video Creator
6	Alexander Soudaei	Applied AI	Digital Video Creator
7	Majdi Alali	Data Science	Text Processing Specialist

Submitted on: Wednesday 13th December, 2023

Course: ACIT4040 - Applied Artificial Intelligence Project

Instructors
Gustavo Borges Moreno e Mello
Trym Lindell
Tom Glover

Table of Contents

Executive Summary

As technological advancements continue to accelerate, a significant digital divide has emerged, particularly impacting the elderly. The Elderly Virtual Assistant (EVA) project aims to bridge this gap by creating a multimodal avatar specifically designed for elderly users, addressing their unique needs for companionship and accessible information.

In its development, EVA went through several stages, including ideation, design, prototyping, and design justification. The team, through collaborative brainstorming, explored various avatar concepts, eventually converging on EVA due to its potential to significantly aid the elderly demographic.

A key aspect of EVA is its user-friendly interface, designed with the elderly in mind. This includes easy-to-read text and intuitive navigation, ensuring that even those unfamiliar with digital technology can interact with EVA effortlessly. We incorporated advanced Large Language Models (Llama2) for natural language processing, which enabled EVA to generate contextually relevant responses, making the interactions more engaging and meaningful for users.

The core of EVA's functionality lies in its integration of voice-to-text transcription, text processing, text-to-voice conversion, and avatar animation. This integration was achieved through a robust infrastructure that supports seamless interaction across various communication modes. EVA's performance was rigorously evaluated, with a focus on accuracy, robustness, responsiveness, and user satisfaction showcasing EVA's effectiveness in real-world scenarios.

Despite successes, the project faced challenges, including the need to enhance processing speed and improve user experience issues like internet dependency and interface design balance. Future improvement plans include optimizing avatar animation speed, reducing AI model resource intensity, and continuously adapting to the evolving needs of elderly users.

The project team's effective collaboration and decision-making were very important in overcoming challenges, including managing diverse expectations and complex dynamics. Tools like Slack and GitHub were instrumental in ensuring streamlined communication and efficient project management.

EVA showcases the capacity of AI to improve the quality of life for older individuals by efficiently harmonizing technology with user requirements. Although encountering technical and architectural obstacles, the project demonstrates the feasibility of utilizing this technology to meet the distinct requirements of elderly folks. Upcoming improvements are expected to enhance the effectiveness and accessibility of EVA.

EVA stands as an exemplar of the potential of AI in enhancing the quality of life for older individuals. It demonstrates the feasibility and impact of using advanced technology to meet the specific needs of the elderly. Looking forward, continuous improvements and refinements are anticipated, aimed at making EVA even more effective and accessible for its intended users.

The project, initially designed for a specific purpose, has revealed its potential for broader applications, notably in aiding individuals with diverse challenges. This includes those with sight impairments, severe dyslexia, and physical disabilities that hinder keyboard usage. Recognizing this capacity to assist a wide array of people marks a significant finding. It underscores the importance of dedicating additional time and resources to further develop the project. In line with this vision, we have decided to release the project under an MIT license on GitHub. This step is taken with the intent to encourage and facilitate further development and innovation from the broader AI community. By making the project openly accessible, we aim to stimulate collaborative efforts, inviting developers and researchers to contribute, adapt, and enhance the project's capabilities. This open-source approach aligns with our commitment to inclusive technology development, ensuring that the benefits of AI are shared widely and equitably.

1 Introduction

The Elderly Virtual Assistant (EVA) is designed to assist elderly individuals in navigating technology and serving as a source of engagement and information. It is an open-source virtual assistant that utilizes state-of-the-art Large Language Models (LLMs) [1] in a multimodal framework that includes voice-to-text, LLM text generation, text-to-voice, and voice-to-video avatar.

EVA is designed for elderly individuals who may experience challenges adapting to new technologies. By providing a user-friendly interface and real-time interaction, EVA aims to improve the quality of life and social connectivity of the elderly.

Elderly individuals often face challenges in adapting to new technologies, leading to social isolation and reduced empowerment. EVA aims to bridge this gap by providing a digital companion that not only assists with technology but also serves as a source of engagement and information.

The EVA project is anticipated to boost the self-esteem significantly, quality of life, and social engagement of elderly individuals by offering them easy access to advanced technology in a format that is simple to use. This initiative targets the reduction of social isolation and promotes empowerment among the elderly. Moreover, given that EVA's functionality is primarily determined by the customization of the language model, it presents a versatile solution. This flexibility allows us to adapt EVA for various other use cases, particularly for individuals with disabilities or those who are unable to use a keyboard for input or to read responses. Such adaptability ensures that EVA can be a valuable tool not only for the elderly, but also for a broader spectrum of users who face different challenges in interacting with technology.

While the EVA project aims to provide a user-friendly and accessible virtual assistant for the elderly, there are potential risks to consider. These include technical difficulties with the virtual assistant, privacy and security concerns, and limitations in the project's scope. It is also important that the user interface is easy enough so people with little to no technical abilities feel safe using it.

EVA utilizes state-of-the-art Language Models (LLMs) [1] in a multimodal framework, including voice-to-text, tailored LLM text generation, text-to-voice, and voice-to-video avatar. To ensure privacy and security, the project is self-hosted and compliant with GDPR standards [2]. For fast and high-quality voice-to-text conversion, we selected Whisper. For text generation, we opted for Llama-2 13b, but newer models like Mistral can easily integrate into the system. Silero is our choice for text-to-voice due to its fast generation and acceptable quality. By default, our application primarily utilizes GIF images for avatar animation due to their latency and more efficient performance, especially considering our current hardware limitations. However, it is designed with the flexibility to easily switch to the SadTalker model. This adaptability ensures that we can take advantage of the SadTalker model's capabilities as soon as our hardware supports it, offering an optimal balance between performance and innovation. The EVA project is

a worthwhile initiative that addresses a significant and time-appropriate problem faced by elderly individuals in our modern society. The project's expected impact on the quality of life and social connectivity of the elderly is significant, and its use of cutting-edge technology and attention to privacy and security make it a viable and ethical solution. The project is feasible from a technical standpoint. However, we need to consider the cost of running these models on on-premise hardware. The project code and demos are available at GitHub [3]

1.1 Background

In the development of an avatar for elderly companionship, different technologies are employed, namely, voice-to-text, Large Language Models (LLM's), text-to-voice and video generation. Implementations of these technologies often overlook the elderly in user-experience, and only cater to the younger, more technology-inclined generation.

In recent years, LLMs have made significant advancements and rapidly emerged as some of the most revolutionary technologies of this generation. This technology is becoming more prominent throughout a plethora of applications, including digital assistants. This advancement and widespread adaptation of LLMs has not only brought incredible results but has also raised several challenges and risks, where the less technologically inclined are especially vulnerable. There have been many studies discussing the inherent risks of LLMs, highlighting their potential for generating biased or inaccurate information, and the risks of the vast amount of data being stored and processed. In relation to our application, these risks raise concerns, as our user group may be more susceptible to misinformation and less aware of data privacy issues.

In the paper "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?" [4], the authors critically examine the rapid trend in the NLP community towards developing larger language models (LLMs) like BERT and GPT-3. Acknowledging the advancements in large language models (LLMs), the authors raise critical concerns. They point out that datasets, often sourced from the web, carry inherent biases, emphasizing the need for careful curation and documentation to mitigate potential harm. Additionally, they caution against overestimating LLMs' capabilities, noting these models manipulate linguistic forms rather than truly understand language. This misconception could mislead both researchers and the public. Finally, the authors highlight the risks associated with synthetic text generated by LLMs, such as the propagation of harmful ideologies and stereotypes, urging a mindful approach to their development and use.

Moreover, our project explores the use of Llama-2, a group of pre-trained and fine-tuned LLMs introduced by Meta [5]. This advanced model, scaling from 7B up to 70B parameters, has been instrumental in enhancing our avatar's interactive capabilities. Llama 2 and its dialogue-optimized counterpart, Llama 2-Chat, have demonstrated remarkable performance in helpfulness and safety benchmarks. Their efficacy is particularly notable when compared to other open-source models and even some closed-source alternatives. This is a crucial aspect for our application, given the need for reliable and safe interactions for elderly users.

The development process of Llama 2 underscores their commitment to safety and user-centric design. Their development process involved safety-specific data annotation, red-teaming, and iterative evaluations, ensuring that the model is advanced in its capabilities and aligns with ethical considerations and user safety. The fine-tuning methodology and the LLM safety approach adopted for Llama 2 is aligned with the principles of responsible AI development, which is essential for our project.

In December 2023, a new model called Mixtral7bx8 [6] came out, boosting performance of open-source models, placing them right next to closed source models like GPT-4. However, this model comes without safeguards, and would require 34gb of ram to be run efficient, and because it proposes a slightly different setup from traditional LLM's, it will not be considered to implement in this project.

In addition to the concerns raised about LLMs, our project also explores the field of voice-to-text technologies, recognizing their critical role in enhancing the user experience for the elderly. One such technology, the SeamlessM4T model [7] developed by Meta, stands out for its multimodal capabilities, effectively bridging speech-to-text and text-to-speech functionalities. This model addresses the challenges of limited language coverage and the complexities inherent in human-machine communication. By employing a unified multilingual model capable of handling multiple tasks, SeamlessM4T aligns with our goal of creating an intuitive and accessible avatar for the elderly.

The voice-to-text domain is further augmented by WhisperAI [8], a model renowned for its extensive training on a diverse set of data. Whisper's robust generalization across various datasets reduces the need for additional fine-tuning, making it a suitable choice for our project. The model's architecture, a blend of CNNs and RNNs, ensures efficient and accurate speech recognition, essential for real-time user interaction.

In the text-to-voice segment, we explored several models, including Google Text-to-Speech (gTTS) [9] and Pyttsx3 [10]. gTTS, with its deep neural network architecture, excels in generating natural-sounding speech, a key feature for our elderly-focused application. Its extensive training data, covering a wide range of languages and accents, offers the versatility needed to cater to a diverse user base. Conversely, Pyttsx3 provides a cross-platform solution with customizable voice properties, adding flexibility to our application.

Additionally, the Bark model by Suno [11] and Silero model [12] were tested for their text-to-speech capabilities. Bark's ability to generate not only authentic speech, but also non-verbal cues made it a compelling option. Its transformer architecture ensures effective text-to-voice conversion, aligning with our real-time engagement objectives. Silero Models, known for their enterprise-grade speech-to-text and text-to-speech capabilities, offer acceptable and real-time speech output, qualities that are particularly beneficial for elderly users.

In the realm of video generation, we also consider advancements such as Wav2Lip and SadTalker [13]. Wav2Lip specializes in lip-syncing talking face videos with arbitrary

speech, a critical feature for creating realistic and engaging digital companions. Wav2Lip's strength lies in its accuracy for videos in the wild, enabling applications like lip-syncing dubbed movies and public addresses, enhancing accessibility in different languages.

SadTalker [13], on the other hand, presents a novel approach to generating talking head videos through a combination of a face image and speech audio. It addresses challenges such as unnatural head movement and distorted expression by generating 3D motion coefficients from audio and using a 3D-aware face renderer. This technology is particularly relevant to our project as it offers a more lifelike and engaging avatar representation, enhancing the overall user experience for the elderly.

With careful selection and integration of these technologies, our avatar aims to overcome the often overlooked needs of the elderly in digital applications. By addressing the challenges posed by LLMs and harnessing the strengths of voice-to-text, text-to-voice and video generation technologies, we strive to provide an avatar that is not only technologically advanced but also empathetic and accessible to its intended user group.

This project focuses on developing a multimodal avatar for elderly people, that is accessible to less-technical inclined individuals for companionship and access to information. However, several aspects are considered for the validity of the project. The project scope is defined by the following considerations;

- **Timeframe Constraints:** The duration of this project is one semester, which means development, testing, and deployment must be completed within the semester. Therefore, this sets a hard deadline for each phase of the project.
- **Hardware Limitations:** Due to limited access to GPUs, the possibility of utilizing large-scale models is limited. Therefore, this project prioritizes efficient models that can run on the available hardware resources.
- **Budgetary Constraints:** As a semester project, the budget for developing EVA is limited. This necessitates a cost-effective approach to development.
- **Testing Constraints:** Limited access to GPUs restricts extensive testing of the application, as testing must work within the constraints of the computational resources available.
- **User Testing Limitations:** The testing of our application is being limited to a smaller group, due to the timeframe constraints.

The main objective of the Elderly Virtual Assistant (EVA) project is to explore the use of state-of-the-art technologies in creating an accessible avatar for elderly individuals. This attempt aims to connect the revolutionizing technology of artificial intelligence with the specific needs of the elderly, a demographic often looked past for technological innovations. Specific objectives are defined;

- **Intuitive User Interface:** Design and develop a user-interface tailored for elderly users, ensuring ease of use and accessibility. The user interface should accommodate to users with limited

technical abilities, allowing them to navigate the application, EVA, comfortably.

- **Optimize Latency:** Minimize system latency to ensure real-time responsiveness and a seamless user experience.
- **Integrate LLMs:** Utilize state-of-the-art LLMs for text generation, ensuring that EVA can handle complex queries, and provide insightful responses.
- **Multimodal Communication:** Integrate voice-to-text, LLM, text-to-voice and video generation technologies to offer a multimodal interaction application.
- **Social Engagement:** Serving as a source of engagement and companionship for the elderly, addressing issues of social isolation and empowerment.
- **Data Privacy:** Follow the GDPR standards, by handling user conversational data responsibly.
- **Modularity in Design:** Develop each component of the EVA system as a modular entity, enabling ease of updates and replacements. This approach allows for the seamless substitution of different models, ensuring the system remains updated to evolving advancements in the fields.

2 Design and Development

2.1 Ideation

The ideation phase of any project is crucial for setting the stage for innovation and creativity. In the case of our project, we embarked on a brainstorming journey to explore various possibilities for an avatar-based solution. Our team, comprised of individuals with diverse backgrounds and expertise, gathered to generate ideas, as illustrated in figure 1, that could leverage technology to address social issues.

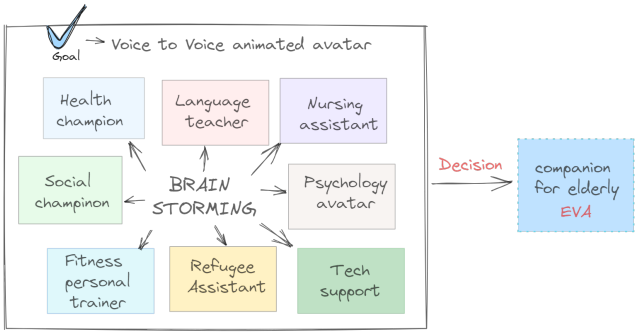


Figure 1: Main ideas from brainstorm

2.1.1 Brainstorming Different Avatar Ideas

Health Companion: Our first idea was to develop an avatar that could function as a health companion. This avatar would assist users in managing their health by reminding them to take medications, schedule doctor's appointments, and provide basic health-related advice. The idea was to create a supportive and interactive tool for individuals, especially those living alone or with chronic health conditions.

Language Teacher: Another concept was an avatar designed as a language teacher. This avatar would help users learn new languages through interactive sessions, utilizing advanced natural language processing capabilities. It would offer personalized learning experiences, adapting to the user's proficiency level and preferred learning style.

Social Companion: Recognizing the increasing issue of loneliness, especially among the elderly, we considered developing a social companion avatar. This avatar would engage

users in meaningful conversations, provide companionship, and help maintain their social skills. It would be especially beneficial for those who are isolated or have limited opportunities for social interaction.

Tech-Support: With the rapid advancement of technology, we identified a need for an avatar that could offer tech support. This would assist users in troubleshooting common technology-related problems, provide guidance on using various devices and software, and offer tips to expand their technology intuition.

Nursing Assistant: The nursing assistant avatar was envisioned as a tool to assist healthcare professionals and caregivers. It would help in monitoring patients, providing reminders for medication, and even aiding in routine check-ups, thereby enhancing the care provided to patients.

Personal Trainer (Fitness): A fitness-focused avatar was also considered. This personal trainer avatar would motivate users to maintain their fitness regime, provide workout guidance, dietary advice, and track their fitness progress.

Psychology Avatar: Addressing mental health was another critical area we explored. A psychology avatar could offer basic counseling, stress-relief exercises, and mindfulness practices. It would be a tool for users to learn coping strategies and maintain their mental well-being.

Refugee Assistant: Lastly, we brainstormed the idea of an avatar designed to assist refugees. This avatar would provide information on legal rights, access to local resources, language assistance, and help in navigating the challenges faced by refugees in a new environment.

Narrowing Down to a Specific Concept: EVA After extensive discussion and analysis of each idea's potential impact and feasibility, we decided to focus on developing a companion for the elderly. This decision was driven by the growing need to address the challenges faced by the aging population, including loneliness, health management, information access and the need for social interaction.

We envisioned this avatar, named EVA (Elderly Virtual Assistant), as a multifaceted tool that would offer companionship and access to information, provide reminders for important tasks, and engage the elderly in stimulating conversations.

Development of EVA: Combining Various Aspects, EVA was conceptualized by combining various features from the brainstormed ideas. It would incorporate aspects of a health companion, social interaction from the social companion concept, and the user-friendly interface of the tech-support avatar. The goal was to create an avatar that was not only technologically advanced, but also empathetic and easy to interact with for the elderly.

At the start of this project, we hosted three workshops where we pitched our ideas to each other. Many of the ideas that came up were in the field of assisting individuals with disabilities, or not non-technical inclined. Due to the diverse culture background within the team, we

discussed elderly care in Norway compared to other countries. We focused on the Norwegian tendency to leave their elderly at homes, which can lead to them feeling isolated.

This discussion started the brainstorming that lead to EVA. We wanted to create a friendly avatar, that could help stimulate conversational need and provide information about the use of technology.

To ensure the design works effectively for our user group, we aimed for an intuitive and ease of use platform. Our solution would allow users to interact with the avatar using their voice to ask questions and receive spoken responses.

Additionally, an important objective is to incorporate anthropomorphism into the design. Anthropomorphism, as defined in the paper "On seeing human" [14], is the "attribution of human characteristics or behavior to a god, animal, or object." This extends beyond merely attributing life to nonliving entities (as in animism), involving the ascription of humanlike properties, characteristics, or mental states to both real and imagined nonhuman agents and objects. It goes beyond simply describing behaviors (like saying 'the dog is affectionate') to representing an agent's mental or physical characteristics using humanlike descriptors (e.g., 'the dog loves me') [14].

To achieve this, we decided to render a robot avatar to lip-sync with the voice output, and tune the language model to behave like an emphatic and understanding human. This approach is designed to foster a more natural and engaging interaction, making users feel more comfortable and connected. The avatar's design incorporates subtle human-like expressions and gestures, enhancing the feeling of interacting with a 'thinking' entity.

Each task is in a field that is ever evolving, especially natural language processing. Therefore, the application is at risk of becoming obsolete before deployment. To combat this, we decided that the application must be modular, meaning each model can be replaced without affecting the rest of the application.

After defining the project's objectives, we divided the group into various subgroups. Each subgroup was tasked with researching and implementing a distinct segment of the pipeline. This collaborative approach allowed us to leverage diverse expertise and perspectives, enhancing the overall design and functionality of the platform. We created an initial design flowchart, as depicted in figure 2, to outline the various components and their interconnections. This flowchart served as a roadmap, guiding the development process and ensuring that all aspects of the system were coherently integrated.

Furthermore, we conducted a series of user experience (UX) tests to refine the interface and interaction. These tests were crucial in identifying usability issues and gathering feedback on the avatar's anthropomorphic qualities. We iteratively improved the design based on this feedback, aiming to create a seamless and intuitive user experience.

By clearly specifying the expected input and output for each segment of the project, we enabled each subgroup to work autonomously while ensuring they delivered the required results.

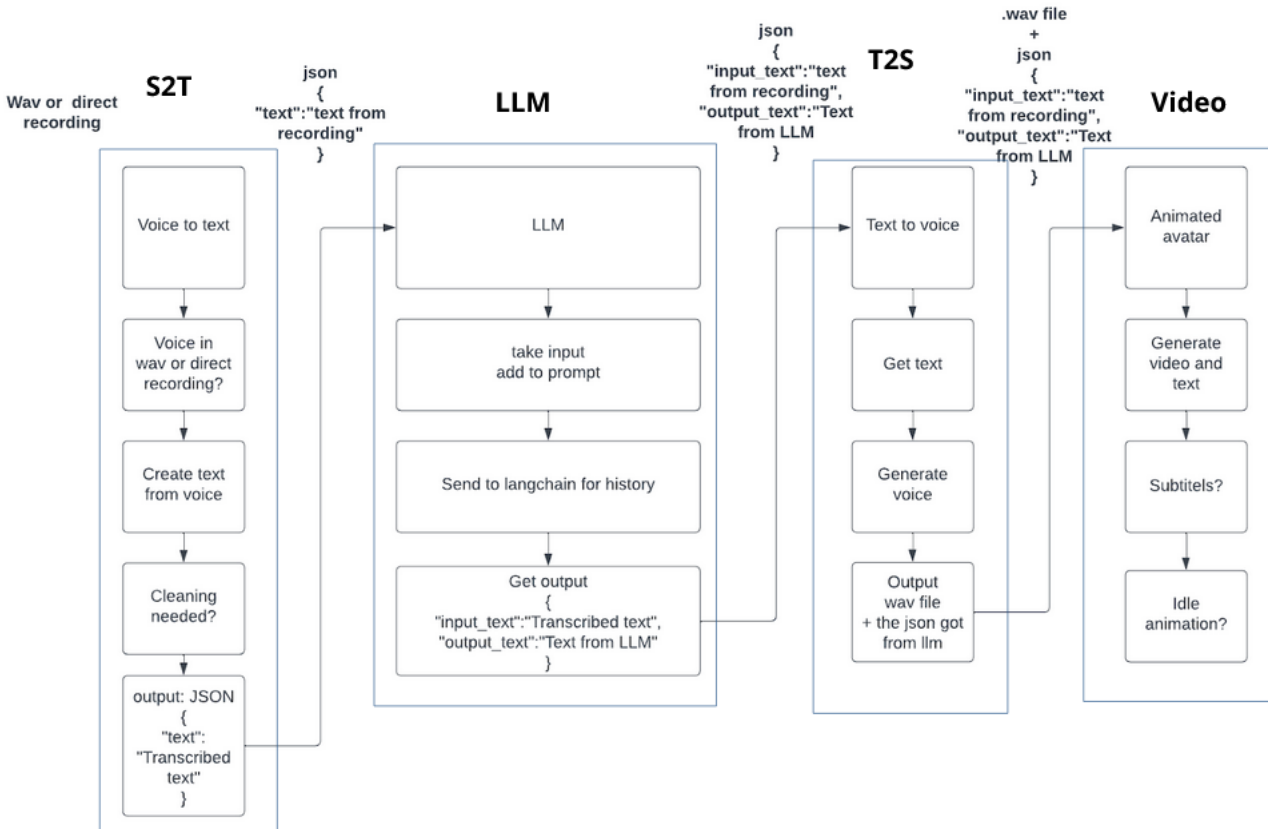


Figure 2: Initial concept development flowchart

2.2 Design Concepts

The design of EVA required careful consideration of various aspects, including its visual representation, user interface, functionalities, and technological integrations. As an AI avatar, EVA needed to possess a visual form that resonated with the intended users, while also being equipped with features that cater to their specific needs. Below, we go into the detailed process of conceptualizing and refining EVA's design.

We explored designs that gave EVA a humanoid appearance, as we believed a relatable and friendly figure could enhance our intended user's interaction. Alternatively, an abstract or symbolic representation was considered, focusing on simplicity, avoiding any potential uncanny valley effect and limiting the computational resources required. Taking inspiration from pet therapy, we contemplated an animal-inspired avatar, which could be perceived as comforting and non-threatening.

We gathered feedback from group members regarding preferences for EVA's visual form. This helped us understand the comfort level with different designs. Balancing the feedback and the goal of creating a warm and approachable avatar, we decided on a humanoid form with soft, friendly features. This design was found to be most effective in establishing a connection with the users. We focused on a voice-activated interface, considering its natural and intuitive interaction style, making it accessible for elderly users. Gesture recognition was another concept we explored, allowing users to interact with EVA through simple hand movements. Given the familiarity of touch

interfaces, especially for users with previous exposure to smartphones or tablets, this was also a considered option.

Choosing the interaction method, a voice-activated interface was selected as the primary mode of interaction, supplemented with touch interaction for users more comfortable with tactile inputs. Context features like medication reminders, health tips, and emergency contact alerts were designed to assist users in managing their health.

For social engagement and companionship, EVA would be designed with advanced language processing to engage users in conversation, storytelling, and even simple games to provide companionship and mental stimulation.

2.3 Prototyping

The prototyping stage of EVA represented a significant phase in our project, where we transformed initial concepts into a series of progressively refined prototypes. Our objective was to develop a multimodal avatar, adept in seamlessly integrating various AI technologies. This involved a comprehensive exploration and integration of diverse AI models, including Voice to Text model for transcribing voice recording, Text processing model for processing transcribed text to generate text response, Text to Voice model to convert text response into voice/audio form, and Voice to Video model for generating an avatar using the audio and a photo. Each step in this process was thoroughly designed and iteratively refined to enhance EVA's usability, functionality, and overall appeal, particularly focusing on the unique needs and preferences of elderly users. This phase was not just about technological development but also about ensuring that EVA could provide empathetic, intuitive, and valuable assistance to its users.

2.3.1 Voice-to-Text Conversion

In developing the voice-to-text component of EVA, our exploration began with SeamlessM4T and WhisperAI. Initially, SeamlessM4T by Meta caught our attention due to its robust speech-to-text capabilities and extensive multilingual support. However, despite its advanced features, SeamlessM4T encountered challenges with maintaining context during transcription, which raised concerns about potential miscommunication. In light of these issues, we shifted our focus to WhisperAI, developed by OpenAI, renowned for its efficient and accurate speech recognition and transcription capabilities. WhisperAI's low latency and high accuracy aligned better with our workflow requirements, ensuring clearer and more consistent transcription for our application.

2.3.2 Text-to-Voice Models

Despite its initial promise, we moved away from SeamlessM4T's text-to-voice translation due to issues similar to its speech-to-text counterpart, particularly regarding context. This led us to explore quite numerous other models since we needed efficiency and improved performance.

Next, we considered gTTS for its wide language and voice support. However, its reliance on an internet connection and lack of emotional expressiveness in speech output were

significant drawbacks. In addition, gTTS being closed sourced would be at compromise with our objective of data privacy. Seeking a faster response rate, we then experimented with pyttsx3, a Python library known for its quick processing. Its cross-platform compatibility and customization options for voice properties made it a strong candidate. However, achieving high speech quality and naturalness across various engines posed a challenge.

Ultimately, we were impressed by the Bark [11] model and its advanced voice cloning capabilities that captured a wide range of emotional and tonal nuances. Despite some speed issues, it provided high-quality audio output, enhancing the real-time interaction experience with EVA.

We also explored the Silero Text-To-Speech Model for its ease of use and fast integration. The model's efficiency was impressive, but the quality of output required further optimization. Silero was then fine-tuned to an acceptable quality of output. While Silero did not quite match the Bark model in output quality, its low latency was a significant advantage.

2.3.3 Text processing

Right from the beginning, we focused on using Llama-2, a large language model, for processing transcribed text to get appropriate text feedback. Our selection of Llama-2 was solely based on its advanced language processing capabilities and open-source availability. The model's GPU implementation and sliding window feature ensured contextual responses but posed memory limit challenges. Fine-tuning was done to allow an extra number of output tokens, enhancing the system message for efficient and improved quality of output. We also integrated LangChain into this model, which played a crucial role in refining the EVA's user experience [15].

One of the important features of the application is the sliding window history, ensuring the EVA maintains a context-aware memory of ongoing conversations. This allowed for more coherent and contextually relevant responses. It was essential to implement this memory as a sliding window, especially considering our primary users of elderly individuals. These users might not be familiar with the use of LLMs, and there was a potential risk of the model becoming overly focused on specific contexts. A static or overly extensive memory might result in EVA delivering responses that, while contextually accurate, could confuse or seem irrelevant to the user. The sliding window approach countered this risk. It ensured the EVA remained context-aware without becoming too focused on past interactions, offering a smoother and more intuitive conversational experience for our elderly users.

In parallel, experiments with CPU utilization were conducted as a backup plan, exploring alternative deployments. This CPU-based approach, optimized for Windows, followed a client-server architecture using a Flask API. The core of this setup involved a main executable file within the llama.cpp folder, supporting various OS and language models. The CPU version showed a response time of 7–9 seconds for simple prompts and up to 17 seconds for complex ones. The project was containerized in a Docker image for independent

operation, including all necessary components and dependencies.

Since we chose to use llama-2-chat we did not need to implement safeguards, as the model itself has those. This was also the main reason for not switching to the newer and more promising models released in late 2023 because none of them has this implemented.

With safeguards, the model will alert the user when it's asking about harmful topics, such as suicide, bomb making, bullying or racist content. For an app like EVA, we feel that ensuring safe responses is more important than switching to the top preforming model at any time. However, since everything is built modular, and llama-2 is implemented using Hugging Face transformers [16], changing to a new model is as easy as changing one line of code.

2.3.4 Voice to Video with image (Avatar animation)

In the avatar animation part of our project, we tried out two different models to create a visual avatar. At first, we used Wav2Lip [17] because it is good at making the avatar's lips move in sync with the audio. But we ran into some problems with getting it set up and making it work with our other tools. Then we switched to SadTalker, which was easier to use because it only needed a picture and audio input. SadTalker worked better and was somewhat efficient for what we needed, compared to Waw2Lip. However, it was slow in processing, so we came up with a backup plan using animated GIFs that simulate a video avatar. This means when we need to respond quickly, we use these GIFs, but if latency is not essential, it is possible to use SadTalker to create the video output. We conducted several experiments to enhance SadTalker's performance, as detailed in Appendix C, by adjusting various settings. While these modifications yielded improvements in certain scenarios, they were less effective in others. Notably, processing new, highly detailed photos proved more time-consuming compared to preprocessed ones, due to the extensive rendering required. Given this challenge and with the goal of optimizing EVA's interaction efficiency, we decided to supplement SadTalker with GIFs. This approach aims to balance detail with quicker response times, thereby enhancing the overall user experience.

The GIFs solution works by listening to events from the user. These events happen when the user clicks the record button, when recording ends, and when the audio file with the answer is created. As one can see in figure 3, there is a dedicated GIF for each of these stats. At the start of a new session, the bot will stay in greeting mode until the user presses the record button. Once the recording ends, it will swap to the processing state, then to the talking state and then the Idle state. In the Idle state, the user can then repeat the process by clicking record again.

2.3.5 Component and Integration Testing

Throughout the prototyping phase, a rigorous and iterative testing process was integral to our development strategy. This approach was important in ensuring that each individual component of EVA functioned as intended before integrating it into the larger system.

Initially, we conducted isolated tests on

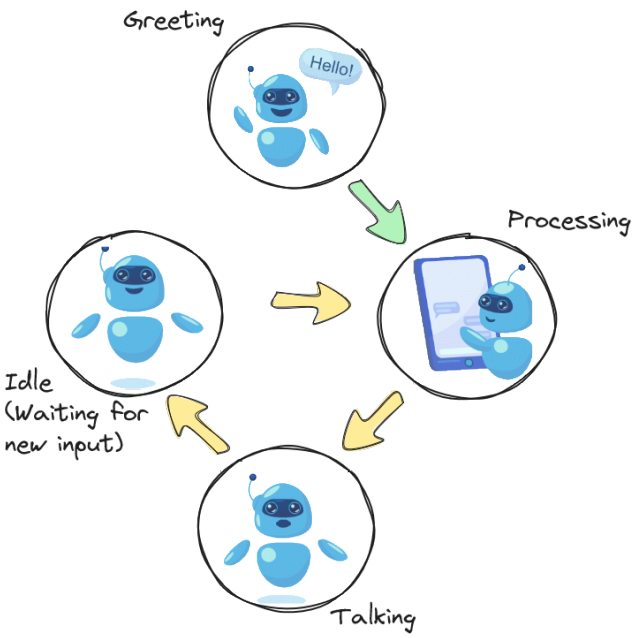


Figure 3: Schema for switching GIF images

each component - Voice-to-Text, Text-to-Voice, Text Processing, and Avatar Animation. These tests were crucial in validating the output of each model, confirming that it met our predefined expectations and requirements. Particularly, we focused on assessing the accuracy, response time, and reliability of each component under various scenarios.

After achieving satisfactory performance in isolated tests, we progressed to testing pairs of components, such as integrating Voice-to-Text with Text Processing, and then Text-to-Voice with Avatar Animation. This step-by-step approach allowed us to methodically identify and address any integration issues, such as data format mismatches or communication bottlenecks.

Subsequently, the scope of testing expanded to include three or more components, progressively scaling up to the entire system. Here, our focus shifted to testing the communication between components, ensuring seamless data transfer, and verifying the overall system's responsiveness. The iterative nature of this testing process was crucial in refining the user experience, as it allowed us to make incremental improvements and immediately observe their impact.

Through these tests, we gained valuable insights into the interdependencies and operational dynamics within our service orchestration, which were pivotal in optimizing EVA's performance. This comprehensive testing phase not only ensured the technical robustness of EVA, but also significantly contributed to achieving a cohesive and intuitive user experience.

2.3.6 Infrastructure setup

In designing our project's infrastructure, we strategically chose to integrate docker-compose due to its excellent orchestration capabilities. This key component plays an important role in the efficient management and coordination of our services. To boost the performance of our models, we hosted our project on a powerful GPU server, which serves as the foundational element of our computational resources. This powerful server is instrumental in enhancing the functionality and responsiveness of our models, ensuring optimal performance throughout the system.

Our docker-compose setup has evolved to encompass a variety of specific functions. These include a recording service, a transcription service, a language model service, a text-to-audio conversion service, and a voice-to-video service. Each of these services plays an important role in the seamless operation of the whole system. These services are defined

in the $docker - compose.yml$ file presented in Listing 1, which outlines the dependencies crucial for the service orchestration.

The $record$ service is independent and initiates the user interaction by capturing audio input. The $transcribe$ service depends on $record$, ensuring it starts only after audio capture, to transcribe audio into text. Subsequently, the llm service, which processes the transcribed text, is dependent on both $record$ and $transcribe$, highlighting its role in the sequence after text transcription.

The $texttovoice$ service waits for $transcribe$and llm to complete their tasks before starting. It converts the processed text responses back into speech. Finally, the $voicetovideo$ service, which depends on the aforementioned services, synchronizes the audio with avatar movements to produce the final video output.

This ordered startup sequence enforced by the $depends_on$ directive ensures a smooth data flow and operational integrity of the EVA system.

The use of $volumes$ and $ports$ in the $docker - compose.yml$ is essential for the services to interact with the host system and each other. Volumes are utilized to maintain persistent data and share data between the host and containers, as well as among different services. For instance, the record service maps the $./aClient$ directory on the host to $/app$ inside the container, allowing for real-time access to the recorded audio files. Similarly, shared volumes in $services$ like transcribe and llm ensure that text and processed data are accessible across the necessary components of the system, enabling a seamless transition from one service to the next. The configuration of $ports$ is crucial for external access to the services. Each service exposes specific $ports$ to the host, which allows for communication with the services from both inside and outside the Docker network. For example, the $record$ service exposes $port4999$, making the audio capture service accessible at this port. This is vital for integrating the Dockerized services with external applications or services that may need to interact with EVA. In our pipeline, $volumes$ ensure that data generated or needed by services is available where it is required, while $ports$ make services reachable, facilitating both internal and external communication necessary for the EVA system's operation.

The recording component of our infrastructure, which serves as a client to the other models, operates on the FlaskAPI platform. This platform was selected for its capacity to utilize endpoints effectively, which are essential for the interconnectivity of our services. Additionally, FlaskAPI offers an intuitive user interface, significantly enhancing the user interaction aspect of the system.

The prototyping phase of EVA was an extensive process involving the integration of sophisticated AI models and innovative avatar design techniques. We navigated through various challenges, from computational limitations to ensuring naturalness in speech output. The exploration of different avatar generation methods, coupled with our commitment to adaptability and user-centred design, shaped EVA into a technologically advanced yet accessible assistant for the elderly.

This journey highlighted the dynamic nature of AI development, underscoring the importance of continual learning and adaptation in the field of technology.

2.4 Design Justification

The final design of EVA was shaped through research, development, and user testing. Each design choice, from the user interface to the backend models, is based on testing, and to align the application with our objective of creating an accessible, engaging and supportive companion for elderly individuals. A concern of the model is the evolving fields making the model outdated before the end of the project. We decided to design the application with this in mind, by creating the application modular, meaning that the inner workings of each task would be separated from each other.

For EVA to understand speech quickly and accurately, we needed a top-notch voice-to-text system. Through prototyping the model, we experienced the best results with the WhisperAI model. Therefore, we decided to use WhisperAI for this task. It stands out because of its latency and also accuracy in transcribing speech, using context. WhisperAI is also good at handling sound perturbations, which may be a problem for individuals with voice impediments. This is especially important for elderly users, as it ensures that conversations with EVA are clear, easy to follow, free from frustrating delays and relevant.

When deciding on our text-to-speech model, we carefully considered our objectives. Initially, we decided on the Bark model by Suno. This model excelled in natural sounding speech, addressing the objective “Social Engagement”. However, the model had drawbacks with latency. Recognizing this issue, we decided to move forward with Silero. This model did not quite match the Bark model in terms of quality, and natural sounding speech, but it provided a significant advantage in latency, adhering to our objective of latency. The balance of Silero in quality and latency aligned better with our objectives in creating both a high-quality and responsive assistant for elderly users.

For handling transcribed text after a user speaks to the avatar, Llama-2 stood out to be effective. Llama-2 excels in understanding and responding to languages very well, making sure EVA’s responses are relevant and easy to understand. One of our objectives for this project is to protect the privacy of our users, and adhere to the GDPR standards, and therefore the open-source nature of Llama-2 is essential. Using Llama-2 enhances EVA’s ability to process queries accurately, making the interaction natural and user-friendly, especially for elderly users.

SadTalker was employed for avatar animation, SadTalker is known for its capability to produce high-quality, lifelike animations. SadTalker excels in making the avatar’s movements, especially lip-syncing with spoken words, appear realistic and engaging. This level of detail contributes significantly to the overall user experience, making interactions with EVA more natural and relatable.

However, one of the challenges we encountered with SadTalker was its processing time. Creating these detailed animations requires a certain amount of computational

effort, which can lead to delays in response. To ensure that EVA remains efficient and responsive, particularly in situations where immediate feedback is essential, we incorporated an alternative solution using GIFs.

These GIFs, pre-designed and less complex than full animations, offer a much faster way to provide visual feedback. They are used in scenarios where speed is more critical than the detailed animation provided by SadTalker. By switching to GIFs for quick responses, we strike a balance between maintaining visual engagement and ensuring prompt interaction. This approach allows EVA to adapt dynamically to different user needs, ensuring that the avatar remains both expressive and efficient.

The final design of EVA was all about making sure it was helpful, easy to use, and enjoyable for older people. We wanted EVA to not just be a smart tool, but also a friendly and understanding companion for its users.

3 Result

EVA is an innovative project that combines state-of-the-art artificial intelligence technology with a profound interest to cater to the needs of elderly individuals. Our objective was to create an accessible and engaging companion for the elderly, a group often overlooked in technological advancements. By leveraging state-of-the-art AI, we aimed to craft a solution that embraces modern technology and addresses the unique challenges faced by older users.

3.1 Solution description

We outline a comprehensive solution for the avatar project, EVA, which is made up of several integral components. These components work together, in harmony, to provide an interactive and user-friendly experience for the elderly. Following is a detailed look at each component, as illustrated in Figure 4

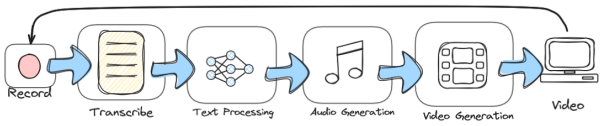


Figure 4: Components integration

3.1.1 Voice to Text component — Transcription

The voice-to-text component captures a voice recording from the user and converts it into text. It is crucial for understanding user commands and queries, transforming verbal interactions into a format that EVA can process. This step is fundamental in initiating a responsive dialogue between the user and EVA.

3.1.2 Text processing

At the heart of EVA's conversational capabilities is the text-processing component. It analyses the transcribed text, interprets the user's intent, and formulates contextually relevant responses. This functionality enables EVA to interact in a meaningful and coherent manner, catering to the specific queries or needs of the user.

3.1.3 Text to Voice Component

After processing the user's input, EVA responds audibly. This component converts EVA's text responses back into speech, providing a natural, conversational experience. It ensures that the responses are not only informative but also engaging, replicating a human interaction.

3.1.4 Avatar animation

The animation of the avatar aspect brings EVA to life visually. It synchronizes the avatar's lip movements with the audio of the responses, enhancing the realism of the interaction. When speed is essential, GIFs provide a faster, albeit less detailed, visual response, maintaining user engagement without delays.

3.1.5 Data flow in the processing pipeline

The complete interaction cycle is described with an outline of data flow within EVA, from the initial user interaction to the final output. Understanding this flow is crucial in comprehending how the various components of EVA interact and process information.

Voice Recording and Conversion The cycle starts with the user's voice input in .wav format, a choice made due to its uncompressed nature, which preserves the original sound quality, crucial for accurate transcription. This format, while larger than compressed alternatives, ensures no loss of data that could impact the understanding of user commands.

Transcription to Text The transcription component converts the .wav file into plain text. The size and format of the input audio can affect the speed and accuracy of this process. Although .wav files are larger, their use simplifies the transcription process as there's no need for decompression, allowing EVA to handle lengthy and context-rich conversations without compromising on response time.

Text Processing and Response Formulation The transcribed text is processed to understand the user's intent. Text data, constrained by the accuracy of the transcription, is analyzed, and a response is formulated. The system is designed to manage varying lengths of text, ensuring flexibility in user interaction.

Conversion to Speech This processed text is then converted back into audio, typically in a .wav format for consistency and to maintain quality. The system's capability to handle the conversion process is subject to the complexity and length of the text, with potential impacts on response times.

Animation Synchronization For the animation synchronization, the audio file dictates the avatar's lip movements. The system is optimized to handle standard lengths of dialogue; however, exceedingly long responses may challenge the synchronization precision and require balancing detail with performance.

Final Output The final output combines the audio with the avatar animation, presented as a video or a GIF. The choice between these two

were a trade-off between detail and speed, with videos providing a richer experience and GIFs ensuring quicker interactions. The system's infrastructure, particularly the GPU resources, plays a crucial role in managing these output formats. Video generation's high latency is affected by model design and the unavailability of other models that are compatible with our infrastructure.

These constraints and data specifications are illustrated in Figure 5, which depicts the flow of data through EVA's processing pipeline, highlighting the transformation of user input into an engaging multimodal output.

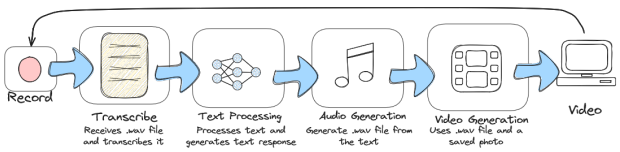
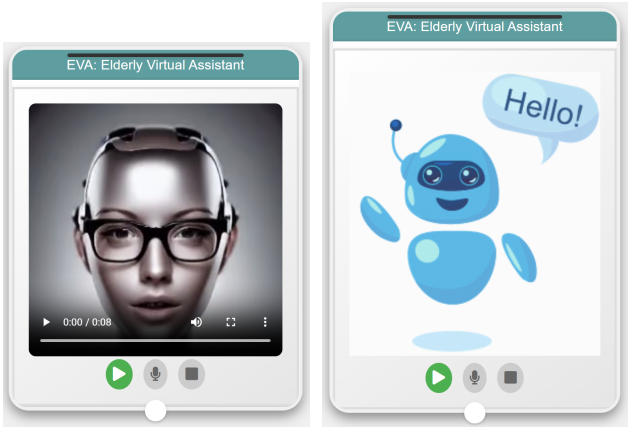


Figure 5: Data flow with in EVA's processing pipeline

3.1.6 User interface

The user interface of EVA is web-based, powered by a Flask application. A user-friendly layout was designed that features an avatar Figure 6a and GIFs frame Figure 6b, a dedicated space for the avatar or GIFs to interact with the user, and a Conversation log frame Figure 7 for real-time display of conversational logs, allowing users to follow the dialogue as they speak with the avatar. It also includes a username entry field for users to enter their usernames as configured in the settings button in Figure 13, personalizing the interaction as well as control buttons including a button for starting the avatar, recording and stopping. These buttons are activated only when they are ready for use, enhancing the user experience.



(a) A Moving Picture (b) Cartoon Face

Figure 6: Illustration of EVA

3.1.7 Eva's Infrastructure

EVA's infrastructure is a well-integrated system combining Docker Compose for service orchestration, a GPU Server for computational power, and client tunneling for seamless local operation of the recording component. We utilized Python, Poetry, and Black for streamlined code management and maintained our workflow through GitHub, ensuring efficient collaboration and code consistency. This robust infrastructure supports EVA's advanced functionalities while maintaining a stable and responsive user experience.

After establishing this foundational infrastructure, we focused particularly on the orchestration capabilities of Docker Compose, a key element in efficiently managing and scaling EVA's services

The topology of our Docker Compose setup is critical in orchestrating the various services that makeup EVA. As illustrated in Figure 8, each service is organized and managed through Docker Compose. This setup streamlines the deployment process and enhances the scalability and maintainability of

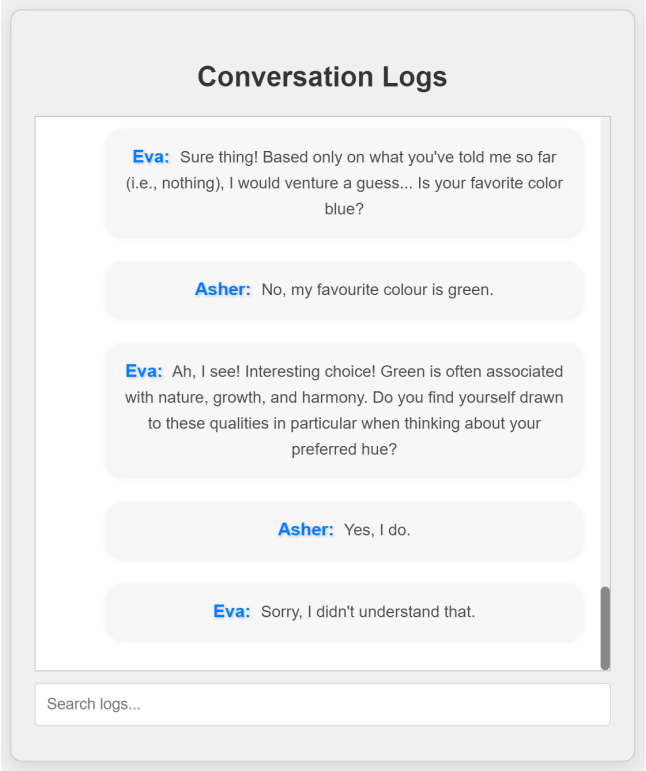


Figure 7: Conversation logs

the entire system.

3.1.8 Service Orchestration

In this orchestration, each service, whether it's for voice-to-text conversion, text/language processing, text-to-voice, or avatar animation, is containerized and managed efficiently. This approach ensures that each component functions optimally and harmonizing with other components, enhancing the overall efficiency and robustness of EVA.

The Record service marks the starting point of user interaction with EVA. Its primary function is to capture audio data (recording) and forward it to the next service via a defined endpoint. It is crucial to note that each service within EVA provides an endpoint for sending or receiving data, ensuring seamless interaction between services. This setup highlights the

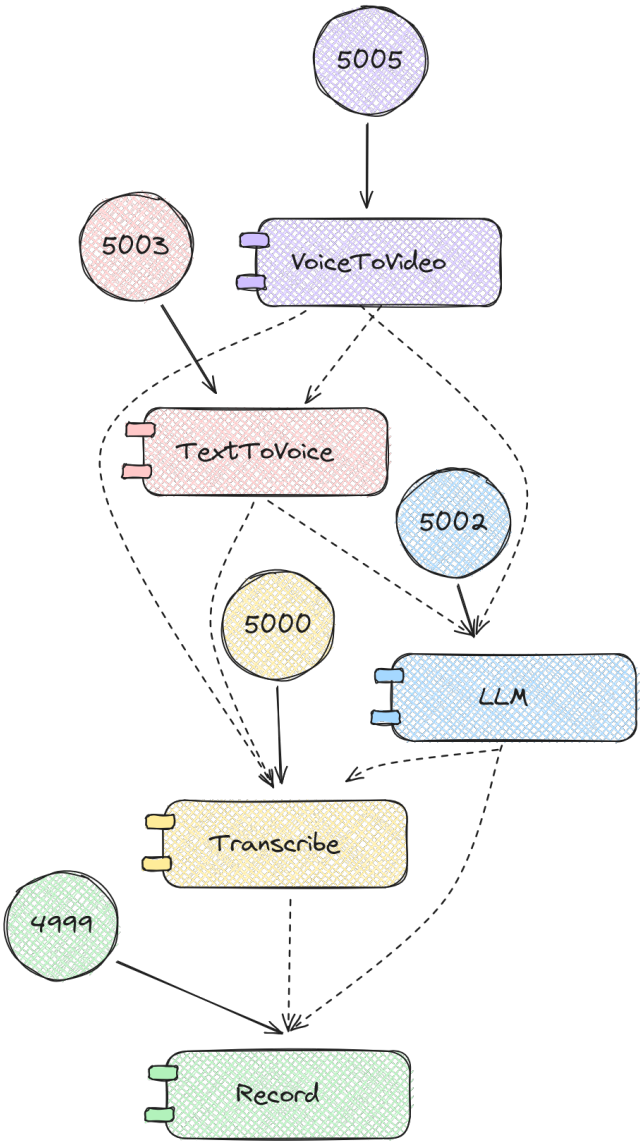


Figure 8: Services Orchestration
Service dependencies
Direct service access

interdependencies that exist within these services, forming the backbone of EVA's operational flow.

The Transcribe service then takes over, transcribing the recorded voice into text, which becomes the input for the LLM (Large Language Model) service. This service processes the text contextually to generate an appropriate response, mirroring the way the user communicates with the avatar.

To maintain the conversational nature of the interaction, the Text-to-Voice service converts this textual response back into audio, effectively bridging the gap between text processing and the final visual output. The culmination of this process is presented to the user as either a video avatar or GIFs animation, created by the Voice-to-Video service. This final step brings the interaction full circle, delivering a dynamic and engaging experience for the users as they interact with EVA.

3.2 Evaluation against project goals and user needs

To assess how effectively the final solution of the EVA meets the project goals and user needs, important aspects of the solution are discussed.

3.2.1 Alignment with project objectives

Accessibility for elderly users. Eva was specifically designed with elderly users in mind. The intuitive web interface, powered by Flask, features large, easy-to-use buttons and a clear layout. This design consideration ensures that EVA is accessible to users who may not be tech-savvy.

Multimodal interaction. By integrating voice-to-text, text-to-voice, text processing and avatar animation, EVA provides a multimodal interaction experience. This approach caters to different user preferences and makes the interaction more engaging.

Effective communication. The use of advanced models like WhisperAI for voice recognition and the Bark model or Silero model for text-to-speech ensures that EVA can communicate effectively, understand user queries accurately and respond in a natural, empathetic manner.

3.2.2 Addressing user needs

The user interface of EVA is straightforward, making it easy for elderly users to navigate and interact with the avatar. Features like real-time conversation logs aid in making the interactions transparent and easier to follow. Integration of high-performing voice-to-text and text-to-voice models ensures that responses are not only quick but also clear and understandable. The option for users to enter their usernames personalized the experience. Additionally, the language processing capabilities of Llama2 ensure that responses are contextually relevant, enhancing the personal touch in interactions. The avatar animation with SadTalker and the use of GIFs for quicker responses keep users visually engaged. This visual aspect is crucial in making the interaction more relatable and lessening the feeling of interacting with a machine.

The final solution of EVA successfully meets the project goals by providing an accessible, multimodal, and effective communication tool tailored for elderly users. Its user-friendly interface, coupled with advanced AI functionalities, ensures that it addresses the specific needs and preferences of its target audience, making it not just a technological solution, but a companion that enhances the daily lives of elderly individuals.

3.3 Performance evaluation

When assessing the performance of EVA, we consider several metrics, including accuracy, robustness, responsiveness, and user satisfaction. Here is a detailed look at each of these aspects.

3.3.1 Accuracy

EVA's voice recognition, powered by WhisperAI, demonstrates high accuracy in transcribing voice recordings into text. This accuracy is needed for clear understanding and effective communication, especially with elderly users who may have unique speech patterns. The Bark model and Silero ensures that the text-to-voice translations are precise and contextually accurate, providing responses that are both relevant and intelligible to users

3.3.2 Robustness

EVA's robustness is evident in its ability to handle a wide range of queries and commands, adapting to different user needs and interaction styles. The use of reliable infrastructure, including Docker compose and a GPU server, contributes to the overall robustness, ensuring that EVA functions smoothly without frequent downtimes or glitches.

3.3.3 Responsiveness

EVA's response time is optimized for quick interaction. While the Bark Model ensures quality in speech output, Silero is used when speed is prioritized, making EVA responsive to user inputs without significant delays. The use of SadTalker for detailed avatar animations and GIFs for quicker responses ensures that EVA remains visually responsive, enhancing user engagement.

With this assessment, EVA stands out as an accurate, robust and responsive solution, that meets the specific needs of elderly users. Its design and functionality reflect a profound understanding of the target user group, leading to a high level of user satisfaction. The performance of EVA, evaluated against these metrics, underscores its effectiveness as a virtual assistant for the elderly.

3.4 Limitations and challenges

While EVA demonstrates significant strengths and capabilities, it is important to acknowledge certain limitations and challenges we encountered. Addressing these transparently allows for a more comprehensive understanding of the project.

3.4.1 Technical limitations

SadTalker, selected for avatar animation in EVA, excels in producing high-quality visuals but faces challenges with slow processing speeds. This is particularly evident when handling new data (input images) containing previously unseen details, leading to prolonged rendering times. Despite efforts to fine-tune the model by adjusting specific settings, as elaborated in Appendix C, these issues persisted. This lag in processing speed critically affects EVA's ability to provide instant visual feedback, a crucial element for real-time interactions. Addressing this limitation is essential for enhancing the overall dynamism and user engagement of

EVA's avatar animations.

The Suno Bark Model, interesting for its advanced speech synthesis capabilities, excels in generating high-quality speech. However, it encounters challenges in ensuring that the voices produced sound natural and human-like, particularly in scenarios involving intricate dialogues or when expressing a range of emotions. This limitation becomes more pronounced in these complex or emotionally charged contexts. Furthermore, a significant drawback of the Bark Model is its slower execution time. This lag in processing speed hinders its ability to deliver responses promptly, which is a critical aspect in real-time applications.

Given these constraints, particularly the need for rapid and reliable voice synthesis, we ultimately chose to adopt the Silero technology. Although Silero may not match the Bark Model in terms of sheer speech quality, it stands out for its higher consistency in output. This consistent performance, especially in terms of speed and reliability across various situations, made Silero a more suitable choice for our requirements.

Models like Llama2, central to EVA's text processing, require significant computational power. This presents a scalability challenge and could limit EVA's deployment on devices with lower processing capabilities. We do, however, believe that this situation will be improved with time, allowing us to run smaller models or model quantization while keeping or improving the quality of the output.

3.4.2 User experiences challenges

The reliance of gTTS on a stable internet connection restricts its functionality in areas with poor connectivity. Furthermore, its API-based operation raises concerns about data privacy during user interactions, leading to our decision not to implement it.

Balancing the simplicity of the user interface with the inclusion of advanced features was a constant challenge. Ensuring that the interface remained intuitive for elderly users without compromising on functionality required careful design considerations

3.5 Future improvement areas

Future versions of EVA should aim to boost the speed of avatar animations while maintaining visual quality. Exploring more sophisticated or efficient animation technologies could be a potential solution.

Reducing the computational demand of our AI models is essential. This would make EVA more versatile and easier to deploy across a broader range of devices.

Regular user feedback and testing are vital to align EVA more closely with the diverse preferences and requirements of elderly users, especially in terms of interface usability and functionality.

In summary, EVA, while effective and innovative, faces challenges related to processing speed, naturalness of text-to-speech output, resource intensity, and balancing user interface design. These limitations provide

valuable insights and directions for future enhancements, ensuring that subsequent versions of EVA are more refined and better equipped to serve the elderly community.

4 Group Dynamic

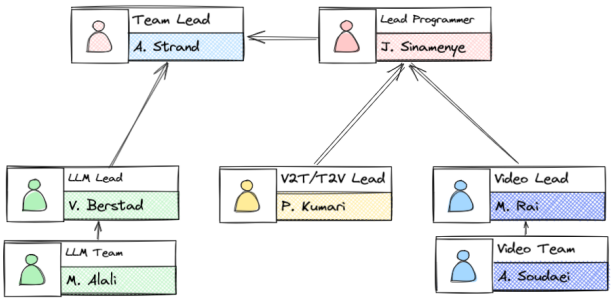


Figure 9: Organization chart

The group formation for the project began with an in-depth consideration of individual strengths, interests, and expertise, even amidst the challenge of two team members joining three weeks late into the project. The initial phase of the project involved aligning the team's thoughts and ideas into one collective plan for our project plan, the first being dividing the team into subgroups. Each member was strategically placed to optimize our workflow. Even if we had a flat structure, we defined an organization chart as shown in figure 9. Vebjørn Berstad and Majdi Omar Alali were designated as responsible for the Large Language Model (LLM). Pratima Kumari, a member of the Voice to Text (V2T) - Text to Voice (T2V) team, was also assigned the role of note-taking during meetings and recording attendance. Jackson Herbert Sinamenye started out as leader of the V2T - T2V team, but due to his skills, he got relocated to Programming lead, and Kumari took the V2T - T2V lead role. Alexander Theo Strand took on the responsibilities of Team Lead. Mubariz Rai and Alexander Soudai were designated as responsible for video avatars.

Each member had a specific responsibility aligned with the project goals, such as LLM development, V2T - T2V team leadership, and avatar creation. This division facilitated a clear understanding of individual contributions. A key advantage of our team, is that we were all able to do more than just our assigned tasks. Team members did not just lead their team; they also helped in other areas when needed. This proved helpful when subgroups got stuck on specific problems, and individuals from other groups stepped in to assist.

4.1 Project tools

When establishing the division of work. We decided to approach the challenge in a flexible, yet structured way. Communication channels were primarily established through Slack, ensuring daily availability and responsiveness. Regular group meetings were scheduled on Mondays and Thursdays to discuss progress and address any issues. Hybrid meetings allowed for both in-person and remote attendance, promoting flexibility.

The collaboration of the group extended beyond communication channels, with GANTT charts used for tracking progress, milestones, and deadlines. In figure 10 we see the GANTT chart for November–December tasks.

Members were expected to dedicate a minimum of 13 hours per week, fostering a commitment to the project. In hindsight, we can conclude that the hourly goal probably was too

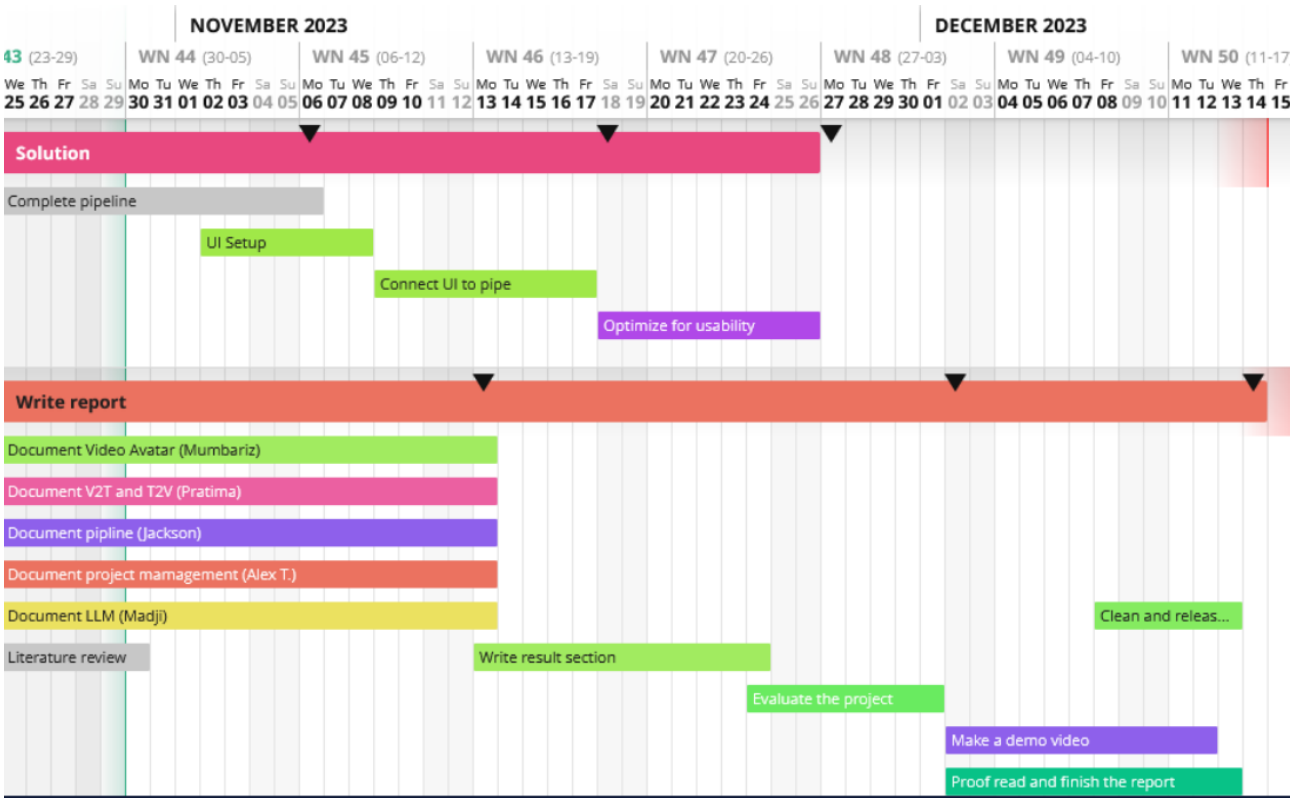


Figure 10: Gantt chart for November–December tasks

high, leaving those who worked as much as expected to do a lot of the work for the other group members as well. The use of Slack for daily check-ins and feedback facilitated seamless communication. Using Slack worked well, but we were forced to host our digital group meetings on Google meet due to technical issues.

For project management, we decided to use Jira. However, Jira usage was inconsistent, leading to some members having to cover for others. This highlighted the importance of good communication within the team. In figure 11 you can see an example of finished tasks.

Source control was done through GitHub, where we also published the finished code and demo videos [3]. On pull requests to main or the develop branch, we preform a formatting test to see that the code follows the Black [18] formatting style, as shown in figure 12. There was a significant difference within the group, in the experience working with Git for source control, which underscored the importance of continuous learning.

Working in a larger group presented unique challenges, as each member had different expectations, with some fine to deliver the less while others pushed themselves for excellence. This has been a challenge throughout the semester, causing some frustration among team members. Another layer of this is the consequences of excluding people from the group, as an exclusion of a group member would lead to that member failing the course and needing to postpone his or her master's thesis.

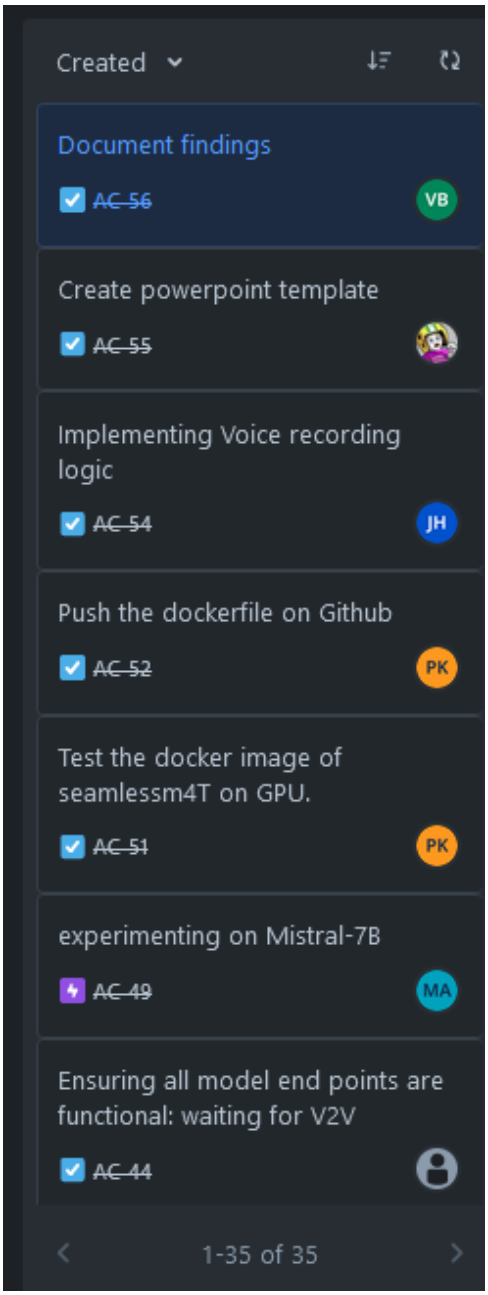


Figure 11: Finished tasks in Jira

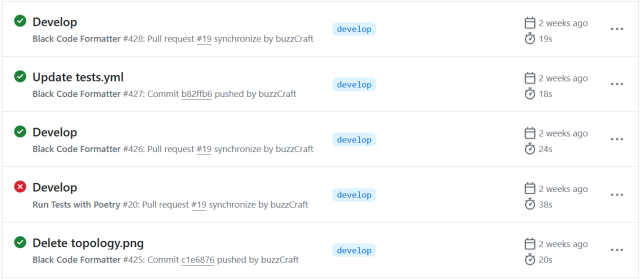


Figure 12: Tests running on pull request on GitHub

4.2 Decision-making

The team made decisions collectively during group meetings, with a democratic vote if necessary, with the Team Lead having an extra vote in case of a tie. Individual tasks were granted autonomy, while crucial decisions were discussed during meetings and voted on if necessary.The team generally agreed on how to proceed, and only once had to conduct a formal vote, regarding whether to exclude a member or not.

Members had autonomy, as long as their solutions were relevant within project limits, but during the project work it became apparent that there was a skill gap among the team members. This caused the efficiency of development to be slower than required. This issue was addressed by forming a small group tasked with the coding aspect of the project, which resulted in much faster development but slower report completion.

Challenges in meeting the 13-hour weekly commitment were addressed through open communication with the Team Lead, emphasizing a collaborative problem-solving approach. The structured decision-making process ensured that disagreements were resolved through discussion, fostering a constructive team dynamic.

The switch to a smaller coding team worked exceptionally well, accelerating development and ensuring tasks were completed. This adaptation and restructuring would not have been possible without a thorough assessment of the group's workup to that point. The team learned the importance of adapting to challenges and restructuring when necessary in the context of the course.

4.3 Learning takeaways

The group's experience emphasized the importance of:

- **Effective communication and collaboration in project management:** The team realized that clear and open communication is the foundation of successful collaboration, leading to better problem-solving and project outcomes.
- **Respecting team members' time:** The group recognized the value of respecting each other's time and ensuring that everyone is working towards a unified goal, which helped maintain a positive and productive work environment.
- **Clear task delegation:** The team learned the importance of clear task delegation and the need for regular check-ins to address challenges promptly, ensuring that everyone is on the same page and working towards the common objective.

The group's learning takeaways included;

- **Clear communication:** The team recognized the need for clear communication among members to ensure everyone is on the same page and working towards the common goal.
- **Regular check-ins:** The group learned the importance of maintaining regular check-ins to address challenges promptly, ensuring that the project stays on track and everyone remains engaged and motivated.
- **Code of conduct:** The team emphasized the need for upholding a positive and professional working environment, respecting each other's contributions, and fostering a culture of collaboration and respect.

The group acknowledged that we should have been stricter on members not communicating or meeting the expectations set in group meetings, emphasizing the importance of clear communication and accountability within the team.

As a group, we are pleased with the proactive problem-solving, encouragement for individual and team achievements, and a commitment to learning from challenges and successes to continuously improve collaboration and project outcomes.

5 Conclusion

Large language models have been a major breakthrough in the field of natural language processing, allowing computers to understand and generate human-like text. In this context, integrating LLMs with text-to-voice and voice-to-text technologies has been a significant step forward in making AI-powered communication more seamless and efficient. The ability to translate written text into spoken language, as well as convert speech back into text, has been a game-changer for accessibility and user experience.

However, while these technologies have made significant strides, there are still some limitations. One such area is video generation using SadTalker, which is currently slow and produces suboptimal results. Additionally, the

process requires a significant amount of GPU RAM, which can be a limiting factor for many users.

Since the LLM is center stage in our application, and since the entire pipeline is designed with a modular approach, it will be possible to upgrade it once a better model comes out.

The main objective of our project was to create a virtual assistant avatar that can generate real-time videos of itself while interacting with the user. We started the project with a clear plan and a lot of enthusiasm, but as we started working on the implementation, we realized that we needed to adjust our approach. We decided to prioritize the avatar functionality first, and then work on the video generation feature in the background. This helped us to ensure that the avatar was running smoothly, while also allowing us to work on the video generation feature without compromising the avatar's functionality. It also let us implement a backup solution to the video generation, involving replaying GIF images, making the avatar to operate in near real-time

One of the main challenges we faced was the limited time available to us. We had to prioritize our tasks, making sure that we were working on the most important tasks first. This required careful planning and project management skills, which we had to develop as we went along.

In terms of the avatar's functionality, we are quite pleased with the final result. The avatar can understand natural language and generate responses in a conversationally appropriate manner. It also has a good amount of knowledge about different topics, which allows it to hold a conversation with the user on a wide range of topics.

Moving forward, we would suggest that more focus should be placed on the chat history. In the current solution, the chat history will only stay for 20 messages, not enough for the application over time. More research should also be done concerning video generation, as this is a complex and time-consuming task that requires a lot of knowledge in computer vision and artificial intelligence. The possibility of adding a database to store history, and exploring the possibility of improving the response with RAG [19] techniques, are also areas worth working with in the future.

The team also believes that as soon as a fine-tuned version of the Mixtral model [6], which emphasizes safe content generation, is released, it will significantly enhance the ability to produce reliable and appropriate content. This advancement is expected to address current concerns regarding lower quality generation from open-source models.

The app has been tested briefly with positive feedback, but more testing in cooperation with target users would help to shape the UI to fit their needs. While the current testing has been effective, implementing more rigorous and diverse testing scenarios will ensure a more robust and reliable system. This would include stress testing the system under various conditions and ensuring its performance and stability over extended periods of usage.

It is important to acknowledge the broader

implications of this project as well, especially its potential benefits for various groups beyond its immediate target audience. Particularly, individuals with disabilities that lower their ability to use traditional input methods, such as a keyboard, stand to gain significantly from this technology.

For example, individuals with physical disabilities that limit their manual dexterity or prevent them from using a keyboard can utilize this voice-driven solution to interact with language models. This technology can serve as an essential tool for them, enabling easier access to digital content and services, and facilitating communication.

Moreover, individuals with severe dyslexia, who often face challenges in reading and writing, could find this solution particularly beneficial. The ability to interact with language models through speech rather than text can help bypass some difficulties associated with dyslexia, such as decoding written words. This approach could provide a more accessible and less frustrating way for them to engage with digital content and perform various tasks.

Additionally, this project holds significant promise for people who are blind or visually impaired. Since reading text on a screen or using standard keyboard inputs can be challenging or impossible for them, a speech-based interface offers an invaluable alternative. This technology can empower them to access information, perform tasks, and interact with various digital services independently, using voice commands and audio feedback.

In conclusion, our project was a success in terms of the avatar functionality, but we acknowledge that there is room for improvement in terms of project management and scope. We are happy with the results we achieved, and we believe that with further research and development, we can create a virtual assistant avatar that can truly enhance the user's experience.

References

[1] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, W. Ye, Y. Zhang, Y. Chang, P. S. Yu, Q. Yang, and X. Xie, "A survey on evaluation of large language models," 2023.

[2] B. Wolford, "What is gdpr, the eu's new data protection law?" Sep 2023. [Online]. Available: https://gdpr.eu/what-is-gdpr/

[3] J. Sinamenye, T. Strand, and V. Berstad, "Eva," https://github.com/buzzCraft/Elderly-Virtual-Assistant-EVA, 2023.

[4] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, "On the dangers of stochastic parrots," in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. ACM, Mar. 2021. [Online]. Available: https://doi.org/10.1145/3442188.3445922

[5] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami,

N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom, "Llama 2: Open foundation and fine-tuned chat models," 2023.

[6] M. A. team, "Mixtral of experts," 2023, accessed: 2023-12-12. [Online]. Available: https://mistral.ai/news/mixtral-of-experts/

[7] L. B. Seamless Communication, Y.-A. Chung, M. C. Meglioli, D. Dale, N. Dong, P.-A. Duquenne, H. Elsahar, H. Gong, K. Heffernan, J. Hoffman, C. Klaiber, P. Li, D. Licht, J. Maillard, A. Rakotoarison, K. R. Sadagopan, G. Wenzek, E. Ye, B. Akula, P.-J. Chen, N. E. Hachem, B. Ellis, G. M. Gonzalez, J. Haaheim, P. Hansanti, R. Howes, B. Huang, M.-J. Hwang, H. Inaguma, S. Jain, E. Kalbassi, A. Kallet, I. Kulikov, J. Lam, D. Li, X. Ma, R. Mavlyutov, B. Peloquin, M. Ramadan, A. Ramakrishnan, A. Sun, K. Tran, T. Tran, I. Tufanov, V. Vogeti, C. Wood, Y. Yang, B. Yu, P. Andrews, C. Balioglu, M. R. C. jussà [3], O. . andCelebi andMaha Elbayad andCynthia Gao, F. Guzmán, J. Kao, A. Lee, A. Mourachko, J. Pino, S. Popuri, C. Ropers, S. Saleem, H. Schwenk, P. Tomasello, C. Wang, J. Wang, and S. Wang, "Seamlessm4t—massively multilingual & multimodal machine translation," *ArXiv*, 2023.

[8] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," 2022.

[9] Y. A. Li, C. Han, X. Jiang, and N. Mesgarani, "Phoneme-level bert for enhanced prosody of text-to-speech with grapheme predictions," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, Jun. 2023. [Online]. Available: https://doi.org/10.1109/icassp49357.2023.10097074

[10] P. Karande, S. Borchate, B. Chaudhary, and P. D. Wankhede, "Virtual desktop assistant," *International Journal for Research in Applied Science and Engineering Technology*, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:247823445

[11] D. Schumacher and F. Labounty, "Enhancing bark text-to-speech model: Addressing limitations through meta's encodec and pretrained hubert," 2023. [Online]. Available: https://rgdoi.net/10.13140/RG.2.2.16022.93760

[12] S. Team, "Silero models: pre-trained enterprise-grade stt / tts models and benchmarks," https://github.com/snakers4/silero-models, 2021.

[13] W. Zhang, X. Cun, X. Wang, Y. Zhang, X. Shen, Y. Guo, Y. Shan, and F. Wang, "Sadtalker: Learning realistic 3d motion

coefficients for stylized audio-driven single image talking face animation," 2022. [Online]. Available: https://arxiv.org/abs/2211.12194

[14] N. Epley, A. Waytz, and J. T. Cacioppo, "On seeing human: A three-factor theory of anthropomorphism." *Psychological Review*, vol. 114, no. 4, p. 864–886, 2007. [Online]. Available: http://dx.doi.org/10.1037/0033-295X.114.4.864

[15] "Langchain: A framework for language model-powered applications," https://python.langchain.com/docs/get_started/introduction, 2023.

[16] H. Face, "Transformers documentation," 2023, accessed: 2023-12-12. [Online]. Available: https://huggingface.co/docs/transformers/index

[17] K. R. Prajwal, R. Mukhopadhyay, V. P. Namboodiri, and C. Jawahar, "A lip sync expert is all you need for speech to lip generation in the wild," in *Proceedings of the 28th ACM International Conference on Multimedia*, ser. MM '20. ACM, Oct. 2020. [Online]. Available: http://dx.doi.org/10.1145/3394171.3413532

[18] Łukasz Langa, "Black 23.12.0 documentation — black.readthedocs.io," https://black.readthedocs.io/en/stable/, 2023, [Accessed 12-12-2023].

[19] P. S. H. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," *CoRR*, vol. abs/2005.11401, 2020. [Online]. Available: https://arxiv.org/abs/2005.11401

Appendices

A Docker Compose Configuration for EVA System

Listing 1: .docker-compose.yaml with services configurations

```
version: '3.8'
services:
  record:
    build:
      context: ./aClient
      dockerfile: Dockerfile
    ports:
      - "4999:4999"
    volumes:
      - ./aClient:/app

  transcribe:
    depends_on:
      - record
    build:
      context: ./app/VoiceToText
      dockerfile: Dockerfile
    volumes:
      - ./app/TextToVoice/app:/text-to-voice-app
      - ./app/Llm/app:/llm-app
      - ./app/VoiceToText/app/audio_asset:/app/audio_asset
    ports:
      - "5000:5000"
  llm:
    depends_on:
      - transcribe
      - record
    build:
      context: ./app/Llm
      dockerfile: Dockerfile
    volumes:
      - ./app/Llm/app:/llm-app
      - ./app/TextToVoice/app:/text-to-voice-app
    ports:
      - "5002:5002"
    deploy:
      resources:
        reservations:
          devices:
            - driver: nvidia
              count: 1
              capabilities: [ gpu ]
  texttovoice:
    depends_on:
      - transcribe
      - llm
    build:
      context: ./app/TextToVoice
      dockerfile: Dockerfile
    command:
      - python3
      - /text-to-voice-app/voiceGen.py
```

```
    volumes:
      - ./app/Llm/app:/llm-app
      - ./app/TextToVoice/app:/text-to-voice-app
    ports:
        - "5003:5003"
  voicetovideo:
    build:
      context: ./app/VoiceToVideo
      dockerfile: Dockerfile
    volumes:
      - ./app/VoiceToVideo/app:/SadTalker/results
    ports:
      - "5005:5005"
    depends_on:
      - transcribe
      - llm
      - texttovoice
```

The above listing presents the complete $docker - compose.yml$ file used in the EVA project. The configuration details the service definitions, including the build context, dependencies, exposed ports, and volume mappings. It serves as a key reference for understanding the infrastructure setup and service orchestration integral to the operation of the EVA system. Readers can refer to this for a deeper technical insight into how various services like $record$, $transcribe$, llm, $texttovoice$, and $voicetovideo$ are integrated and managed.

B Eva Web GUIs

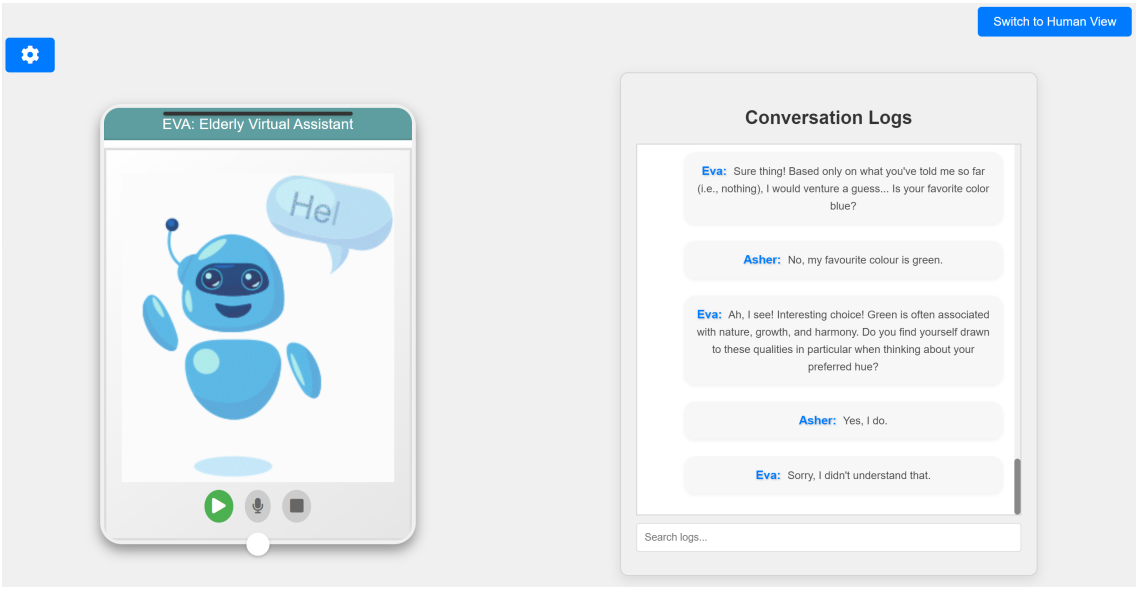


Figure 13: Web Interface Design for GIF Output

This appendix provides a snapshot and a detailed description of the web user interface designed for displaying GIFs in the EVA system. It highlights the layout, user interaction points, and how GIFs are integrated and presented in response to user queries. This part of the UI is crucial for understanding how EVA maintains user engagement through quick visual feedback when speed is prioritized over detailed video responses.

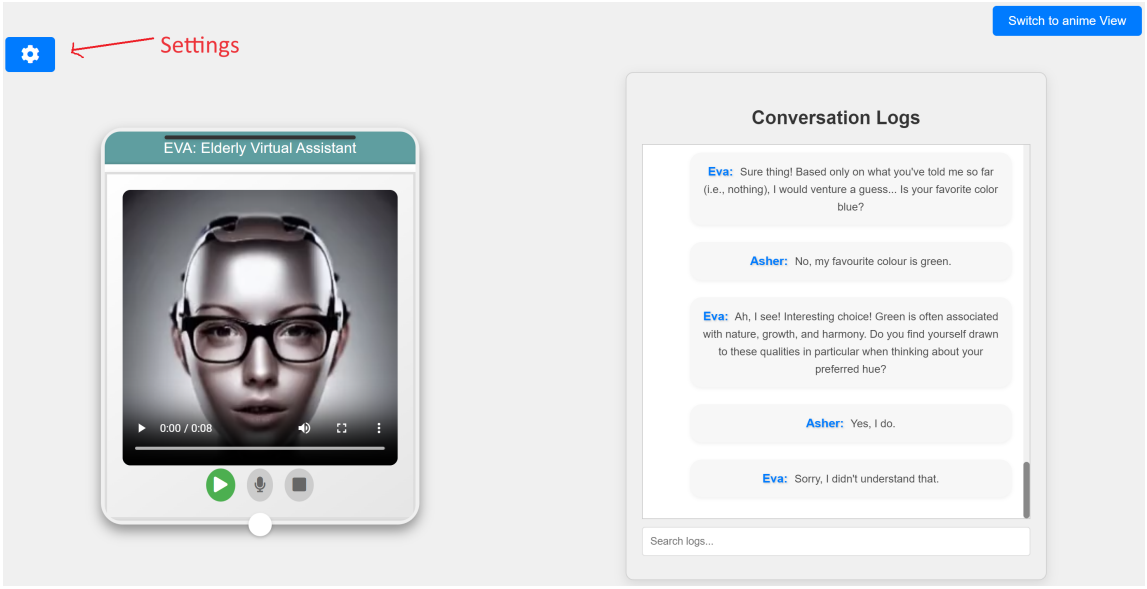


Figure 14: EVA web GUI for Avatar design - cartoon face

Included here, above, are the visuals of the cartoon face used as the avatar in the EVA system. This appendix sheds light on the aesthetic choices, animation features, and how the avatar's design contributes to user interaction and engagement. It provides a visual reference for the avatar that users interact with, emphasizing the importance of user-friendly and expressive design in virtual assistant interfaces.

Figure 13 and Figure 14 come from the same display. Frame for GIFs and cartoon faces are as a result performed by clicking a button “Switch to Human view” or “Switch to anime view”. Both GUIs display conversation logs dynamic frame. The conversation logs are displayed in real-time, aiding users in tracking their interaction history with EVA. This part of the UI is instrumental in ensuring transparency and ease of use, particularly for elderly users, by allowing them to review and follow the conversation as it progresses.

The name “Eva” in the logs represents a fixed name for the application, whereas the name “Asher” is set by the user through the settings button close to the avatar frames. The user can also set their hobbies with the same action of clicking this settings button as shown in Figure 15 to give the avatar context to personalise the response, this is however not functional in our results but a plan for future improvement.

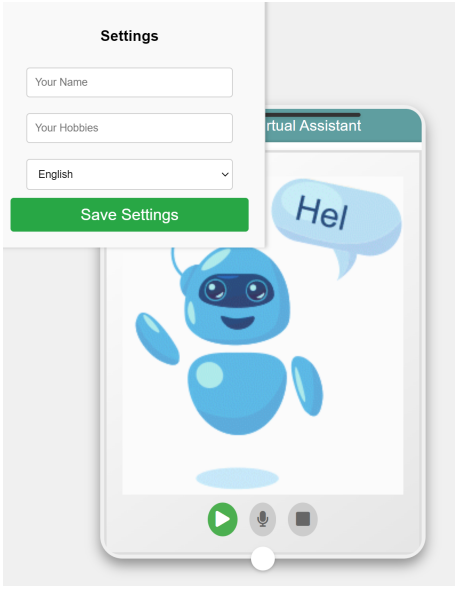


Figure 15: Initial interaction form: Username setting into EVA app

C SadTalker experiments

Initially, when the SadTalked module was tested, one of the main issues was the high execution time of generating video from voice and image. An image of size 410 KB containing 406 x 506 pixels, combined with three seconds long audio, was used to generate the very first sample. To generate this sample, the model ran for over four minutes to generate a three seconds long video.

However, by turning certain settings on or off, as well as selecting an input image with good enough quality with the image size and dimensions being as low as possible, the process was able to run more efficiently. To identify what combination of settings to use, as well as how dependent the execution time was on other factors, an experiment was conducted.

For this experiment, five images with different sizes, shown in Table 1 were combined with six audio files with different sizes and lengths, shown in Table 2. In addition, three different settings shown in Table 3, were tested with different combinations.

These settings have different functionality. By turning *Still* on, the model generates more static images with less eye and lip movement. The *Preprocess Full* keeps the image in full size by turning it on, on the other hand by turning this setting off, the generated images get cropped. The *Enhancer GFPGAP* is a third-party module that enhances the generated images to a higher quality.

By combining every image in Table 1 with every audio in Table 2, as well as with every combination of settings in 3, were we able to track execution time of 150 generated samples. Table xx shows some of the

Table 1: Name and size of images used in the experiment

Image Name	**Image Size**
art_10	0.555 mb
art_13	0.617 mb
art_8	2.970 mb
art_4	3.450 mb
happy1	107.000 mb

Table 2: Name and size of audio files used in the experiment

Audio Name	**Audio Size**	**Audio Length**
chinese_poem1	0.256 mb	5 sec
chinese_poem2	0.450 mb	9 sec
RD_Radio40_000	0.500 mb	8 sec
bus_chinese	0.636 mb	3 sec
japanese	2.500 mb	9 sec
deyu	2.560 mb	9 sec

Table 3: Settings Configuration

Still	**Preprocess Full**	**Enhancer GFPGAN**
ON	ON	ON
ON	ON	OFF
ON	OFF	ON
OFF	ON	ON
ON	OFF	OFF