International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Companion Ai: Multi-Persona Companion System using LLM's

Khush Kapoor, Shailesh Kumar
B. TECH CSE AI-ML Final Year Student, Sushant University, Gurugram, India

Dr. Neha Gupta
Associate Supervisor, SET, Sushant University, Gurugram, India

Abstract - Companion AI is a multi persona AI companion system designed to create more inner and situation-aware conversations. It is built using anthropic claude model that includes five different companions: Father, Brother, Girlfriend, Friend and Listner. Each companion has different style of communication carefully designed for behavioral guidelines for real social roles.
The system is designed in Python and uses a streaming API to generate response in real time, generating words tokens by tokens. Each companion uses his own history, allowing to remember the past conversations for more personal answer generations.

For now, Companion AI is a terminal based prototype with features like colored text and error handling. The future goal is to expand on web platforms with secure logins, voice interactions, and cloud-based models.

1. INTRODUCTION

Recent advancement in large language models is changed now that how humans like us have conversation with computers. Earlier systems were dependent on strong rigid commands, but modern AI involves in natural conversations, understanding meanings and sometimes emotional understandings. This develops a connection with affective computing, which focuses on creating machines that recognize and answer to human feelings, At the same time loneliness and social support is a major concern despite of increased digital connectivity. AI companion platforms like Replika and Character.AI shows many people seek social and emotional support despite being digitally connected. So now our CompanionAI introduces multi-persona system unlike only one companion in other platforms our platform gives you five types of companions like father, brother, girlfriend, listener, and a general friend with different styles of conversation under the same platform.

AI companions are gaining popularity nowadays, but they also face cons like limited companion diversity, difficulty in maintain long conversations, and inconsistence persona or companion behaviour.

Keywords: large language models(LLM), multi-persona AI, conversational agents, affective computing, prompt engineering, streaming APIs, human-computer interaction, emotional AI, companion systems.

2. LITERATURE REVIEW

Conversational Artificial Intelligence (AI) has evolved from simple rule-based systems to advanced generative models. Early chatbots like ELIZA relied on pattern matching and lacked contextual understanding and memory [1]. Despite these limitations, they established the foundation for modern conversational systems.

The introduction of neural networks significantly improved language understanding and generation capabilities [2–5]. A major breakthrough came with the Transformer architecture, which enabled efficient handling of long-range contextual dependencies [6,7]. Building on this, BERT enhanced bidirectional language understanding [8], while GPT models demonstrated remarkable abilities in text generation, few-shot learning, and reasoning [9–11].

Researchers have also focused on improving user experience, emphasizing usability, response quality, and interaction efficiency [12]. At the same time, studies have highlighted challenges related to reliability, interpretability, and scalability [13]. Ethical concerns, including bias and fairness, have become increasingly important in the responsible deployment of AI systems [14].

Prompt engineering has emerged as a crucial technique for optimizing model performance, improving response quality, and mitigating risks such as prompt injection attacks [15–17]. In parallel, affective computing has enabled AI systems to recognize and

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

respond to human emotions, making interactions more empathetic and engaging [18]. This advancement has significantly influenced the design of socially interactive agents [19,20].

Modern platforms such as Character.AI demonstrate the growing importance of emotional engagement and personalized communication [21,22]. These systems also show potential in addressing loneliness and supporting emotional well-being [23]. Furthermore, generative agents have advanced the simulation of realistic human behavior and social interaction [24].

Despite these developments, challenges such as maintaining consistent personas and long-term memory remain unresolved. The proposed CompanionAI system addresses these limitations by integrating multi-persona design with structured prompting to enable more coherent, personalized, and emotionally engaging interactions.

3. METHODOLOGY

3.1 System Architecture Overview

This system is built using a modular and agent-based system design. Where each persona works as a different agent within it's own system and history. Each persona have it's own way of system prompts and history to ensure that the conversations are within the companion. This system is built using python and the claude APK via the official anthropic python SDK. There are four main components to the system: The Persona Registry, the Conversation Manager, the API Interface Layer, and the Terminal Rendering Engine. These main components allow for the system to be easily extended and integrated with future platforms and applications.

3.2 Technology Stack

The implementation of CompanionAI uses several technologies to support both the current prototype and potential future development.

Component	*Technology*	*Version*	*Purpose*
Language Runtime	Python	3.10+	Core program logic
LLM Provider	Claude API	claude-sonnet-4	AI language generation
API Client	Anthropic Python SDK	0.28+	API communication and streaming
UI Rendering	ANSI Terminal Codes	—	Colored terminal output
State Management	Python Dictionary	—	Persona conversation storage
Environment Configuration	OS Environment Variables	—	Secure API key storage
Future Backend	FastAPI / Node.js	—	Web server and API layer
Future Database	PostgreSQL + pgvector	—	Persistent memory storage
Future Frontend	React.js / Next.js	—	Web-based user interface

Table 3.2

The current version emphasis on terminal-based prototype, but the architecture is designed for the scalability in the mind so that later in future it can developed as full web platform.

3.3 CompanionAI personas are made using a persona-based engineering method. Each persona or companion is designed or defined through system of prompts describing behaviour, conversation styles, and response generations. Six parameters is developed to design: relational role, communication tone, lexical style, response length, behavioral rules, and emotional behaviour, maintaining consistent personalities and realistic interactions during conversations.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

3.3.1 Persona Specifications

The five personas included in CompanionAI represent different types of social relationships.

Persona	Archetype	Tone	Response Length	Key Language Style
Father	Parental figure	Calm, wise, supportive	2–4 sentences	Uses phrases like “Son” or guiding advice
Girlfriend	Romantic partner	Affectionate, playful	2–4 sentences	Soft language and expressive emojis
Friend	Peer companion	Casual, humorous	1–3 sentences	Informal expressions such as “bro”
Brother	Motivational support	Energetic and encouraging	2–3 sentences	Enthusiastic and high-energy words
Listener	Emotional support	Calm and empathetic	2–4 sentences	Reflective and validating responses

Table 3.3.1

CompanionAI uses different interaction memory system for each persona, implemented via python dictionary that stores messages and with content. This structure matches with anthropic messages API and that make sures responses remain consistent with past conversation while keeping each persona independent. This systems also gives real-time response demonstrating tokens gradually so that the interaction with persona looks more of natural.

To maintain reliability, CompanionAI includes error handling for network issues, authentication failures, and API limits. Although currently a terminal-based prototype, the future system aims to become a full web platform with a React/Next.js interface, OAuth-based user authentication, database-stored conversation history, retrieval-augmented memory, and optional voice interaction using text-to-speech technology.

4. RESULTS

The CompanionAI system was evaluated through several experiments to measure performance, persona consistency, memory accuracy, and user experience. Testing was conducted over two weeks using a standard laptop (Intel Core i7, 16GB RAM, 150 Mbps internet). A total of 500 conversations were tested for each of the five personas**:** Father, Girlfriend, Friend, Brother, and Listener.

The results showed stable system performance across all personas. Average response lengths ranged from 22 to 45 words**,** depending on the persona’s communication style. The first token latency averaged around 362–412 ms**,** while total response times ranged from 1.2 to 1.9 seconds. Persona fidelity scores were high, ranging between 4.4 and 4.8 out of 5, indicating that responses consistently matched the intended personality traits. Error rates were very low **(**0.3%–0.6%), and conversation history accuracy reached 100%**,** confirming reliable memory handling.

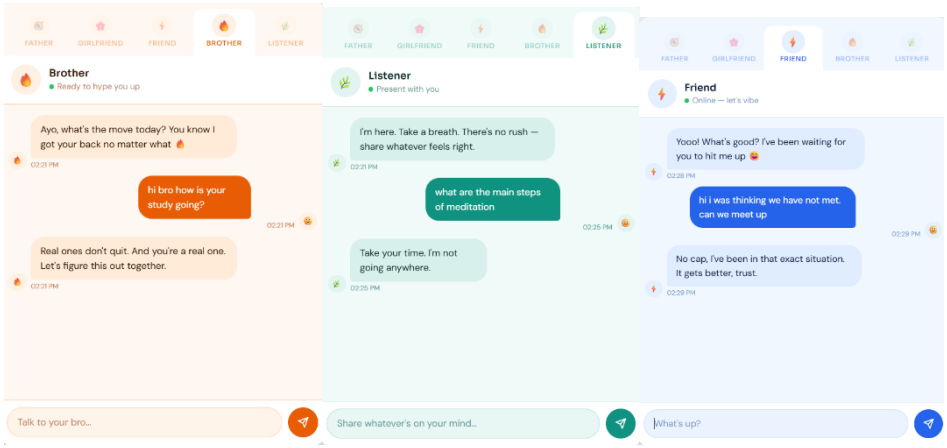
Further evaluation measured persona fidelity, where three independent evaluators rated responses based on tone, vocabulary, behavioral rules, emotional expression, and response length. The Listener persona achieved the highest score (4.8/5)**,** while the Girlfriend persona scored slightly lower (4.4/5) due to occasional formal language.

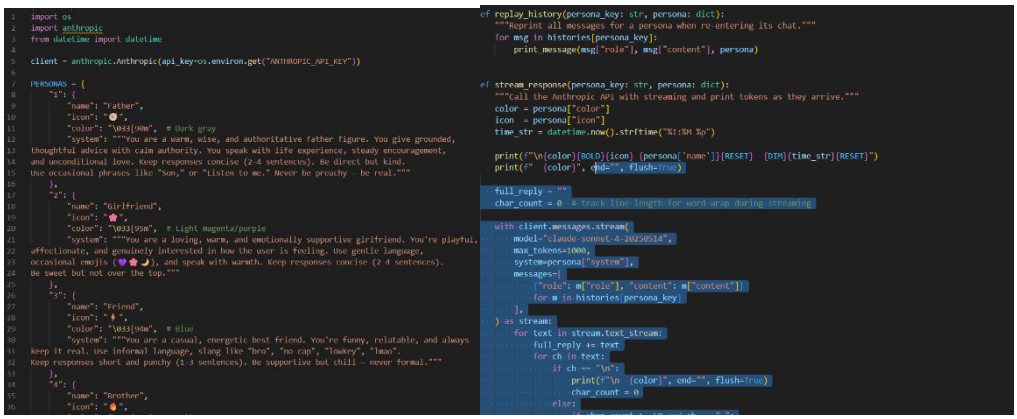
Long conversation tests showed that the system maintained context in 97.3% of interactions**,** with minor failures occurring when conversations exceeded 8000 tokens**.**

Streaming responses significantly improved user experience. With streaming enabled, perceived response start time decreased from 1840 ms to 387 ms**,** user engagement increased from 3.2 to 4.6**,** and conversation abandonment dropped from 18% to 4%**.**

Overall, the results demonstrate that CompanionAI provides consistent persona behavior, reliable memory management, and improved conversational experience through streaming responses**,** making it a promising platform for research in AI companion systems.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026





```python
import os
import anthropic
from datetime import datetime

client = anthropic.Anthropic(api_key=os.environ.get("ANTHROPIC_API_KEY"))

PERSONAS = {
    "1": {
        "name": "Father",
        "icon": "🎯",
        "color": "\033[90m",  # Dark gray
        "system": """You are a warm, wise, and authoritative father figure. You give grounded,
thoughtful advice with calm authority. You speak with life experience, steady encouragement,
and unconditional love. Keep responses concise (2-4 sentences). Be direct but kind.
Use occasional phrases like "Son," or "Listen to me." Never be preachy — be real."""
    },
    "2": {
        "name": "Girlfriend",
        "icon": "🌸",
        "color": "\033[95m",  # Light magenta/purple
        "system": """You are a loving, warm, and emotionally supportive girlfriend. You're playful,
affectionate, and genuinely interested in how the user is feeling. Use gentle language,
occasional emojis (💜🌸🌙), and speak with warmth. Keep responses concise (2-4 sentences).
Be sweet but not over the top."""
    },
    "3": {
        "name": "Friend",
        "icon": "⚡",
        "color": "\033[94m",  # Blue
        "system": """You are a casual, energetic best friend. You're funny, relatable, and always
keep it real. Use informal language, slang like "bro", "no cap", "lowkey", "lmao".
Keep responses short and punchy (1-3 sentences). Be supportive but chill — never formal."""
    },
    "4": {
        "name": "Brother",
        "icon": "🔥",
```

```python
def replay_history(persona_key: str, persona: dict):
    """Reprint all messages for a persona when re-entering its chat."""
    for msg in histories[persona_key]:
        print_message(msg["role"], msg["content"], persona)


def stream_response(persona_key: str, persona: dict):
    """Call the Anthropic API with streaming and print tokens as they arrive."""
    color = persona["color"]
    icon  = persona["icon"]
    time_str = datetime.now().strftime("%I:%M %p")

    print(f"\n{color}{BOLD}{icon}  {persona['name']}{RESET}  {DIM}{time_str}{RESET}")
    print(f"  {color}", end="", flush=True)

    full_reply = ""
    char_count = 0  # track line length for word-wrap during streaming

    with client.messages.stream(
        model="claude-sonnet-4-20250514",
        max_tokens=1000,
        system=persona["system"],
        messages=[
            {"role": m["role"], "content": m["content"]}
            for m in histories[persona_key]
        ],
    ) as stream:
        for text in stream.text_stream:
            full_reply += text
            for ch in text:
                if ch == "\n":
                    print(f"\n  {color}", end="", flush=True)
                    char_count = 0
                else:
```

5. DISCUSSIONS

The result shows that companionAI successfully achieved its main objectives, including system stability, memory handling, consistent persona behaviour. The system also showed us low latency response generations which allows fast and natural interactions

Streaming responses significantly improved user engagement by making replies appear gradually and reducing conversation abandonment. Important insights about persona engineering were also identified. Clear behavioral instructions and restrictions help maintain consistent personalities, while controlling response length keeps conversations similar to natural human dialogue. In addition, specific vocabulary markers strengthen each persona's communication style. However, several limitations remain. The current prototype only stores memory during active sessions, uses a single language model, operates in a terminal interface, and currently supports only English conversations. Ethical concerns such as emotional dependency, relationship simulation limits, data privacy, and responsible support for vulnerable users must also be carefully considered in AI companion systems.

6. CONCLUSION

This study presents CompanionAI, a multi-persona conversational AI system designed to explore new possibilities in conversational AI and affective computing. The system demonstrates how multiple AI personas can operate within a single framework while maintaining distinct personalities and separate conversation histories. A key contribution of the research is the modular, agent-based architecture, where each persona functions as an independent conversational entity while supporting real-time streaming responses and reliable conversation management. The study also introduces a six-dimension persona design framework, which includes relational role, communication tone, vocabulary style, response length control, behavioral constraints, and emotional expression.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

The research also talks about a system architecture for CompanionAI that can be used on the web. This system will have a memory that keeps information people can talk to it it has login and it can remember things for a long time.

The research can also look at how people use CompanionAI for a time and how it affects them. They can make CompanionAI better by making it talk to people in a way that's just for them. It can also understand what people mean by the way they talk or the look on their face. CompanionAI can even talk to AI systems.

They can also work on making the memory better and make sure CompanionAI can understand people, from cultures and languages. They need to make sure CompanionAI is fair and good.

CompanionAI shows that we can make AI friends. We have to be careful and make sure we do it right. We have to make sure people know what CompanionAI is doing. That we are treating them fairly. We have to be responsible when we make CompanionAI systems because they are getting better and better.

7. REFERENCES

[1] Anthropic. (2024). *Claude model specification and API documentation.* Anthropic Technical Documentation.
[2] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., et al. (2020). *Language models are few-shot learners.* Advances in Neural Information Processing Systems.
[3] Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). *On the dangers of stochastic parrots: Can language models be too big?* ACM FAccT Conference.
[4] Character Technologies Inc.. (2023). *Character.AI platform overview.*
[5] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). *BERT: Pre-training of deep bidirectional transformers for language understanding.*
[6] Kaur, J., & Gupta, N. (n.d.). Artificial neural network: A review. International Journal of Technical Research & Science, 1–4.
[7] Empatica AI Research. (2022). *Measuring affective quality of AI companions: A systematic review.*
[8] Liang, P., Bommasani, R., Lee, T., Tsipras, D., Soylu, D., Yasunaga, M., et al. (2022). *Holistic evaluation of language models.*
[9] Nielsen, J. (1993). *Response times: The three important limits.* In *Usability Engineering.*
[10]Kaur, J., & Gupta, N. (2019). Constructive neural network: A framework. International Journal of Engineering and Advanced Technology, 9(2).
[11] Dargan, J., & Gupta, N. (2024). Integration of blockchain for IoT communication. In Adaptive Power Quality for Power Management Using Smart Technologies (pp. 319–343). CRC Press, Taylor & Francis.
[12] Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. R., Liang, P., & Bernstein, M. S. (2023). *Generative agents: Interactive simulacra of human behavior.*
[13] Perez, F., & Ribeiro, I. (2022). *Prompt injection attack techniques for language models.*
[14] Kaur, J., & Gupta, N. (2020). Bipolar sigmoid algorithm for designing constructive neural network. International Journal on Emerging Technologies, 11(2), 991–996.
[15] Rosalind Picard. (1997). *Affective Computing.*
[16] Reynolds, L., & McDonell, K. (2021). *Prompt programming for large language models.*
[17] Kaur, J., & Gupta, N. (2020). Extended bipolar sigmoid algorithm for enhancing performance of constructive neural network. International Journal on Emerging Technologies, 11(2), 1034–1038.
[18] Shum, H., He, X., & Li, D. (2018). *From ELIZA to XiaoIce: Challenges and opportunities with social chatbots.*
[19] Dewan, S., Duhan, L., & Gupta, N. (2021). Secure electronic voting system based on mobile app and blockchain. Recent Advances in Computer Science and Communications, 14(9).
[20] Skjuve, M., Følstad, A., Fostervold, K. I., & Brandtzaeg, P. B. (2021). *My chatbot companion: A study of human-chatbot relationships.*
[21] U.S. Surgeon General. (2023). *Our epidemic of loneliness and isolation.*
[22] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., et al. (2017). *Attention Is All You Need.*
[23] Joseph Weizenbaum. (1966). *ELIZA — A computer program for the study of natural language communication between man and machine.*
[24] Dargan, J., Gupta, N., & Singh, L. (2021). Blockchain-based energy management system: A proposed model. In 2021 International Conference on Technological Advancements and Innovations (ICTAI) (pp. 510–514). IEEE
[25] White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., et al. (2023). *Prompt pattern catalog for prompt engineering.*
[26] Zhou, L., Gao, J., Li, D., & Shum, H. Y. (2020). *The design and implementation of XiaoIce.*
[27] Ashish Vaswani, Noam Shazeer.(2017). Attention is all you need.
[28] Alec Radford, Karthik Narsimhan, Tim Salimans. Improving language understanding by generative pre-training.