Lilly AI VSM Member Information Document

Objective & Information

This Project is to make An AI VSM Member Similar To

Neuro-Sama by Vedal

But Custom For VSM Created By Kill3rKai.

Current Builds Are:

- V1 [Prototype]
- V2 [Experimentation Type]
- V2.5 [The First Stable Build]
- V3 [The Potential End Long Term]

V1 [Prototype]

V1 is the **LLM backbone** for the later models of AI, V2 – V3.

There is not much to report. It is the **learning data bot [Student AI Manager]** in Discord. **It parses data to other models**. This is the prototype. All future models, [V2; V2.5; V3], were **built on top of this additionally**. This version is not a requirement, just a structure to be basic

V2 [Experimentation Type]

V2 Is a **stable build** that has **Discord and chat integration.**

This version **needs fine tuning to the LLM**. It has every capability of V2.5 but lacks tuning, **making it inhuman and robotic**; this is **fixed in V2.5**. This build was **used as an experiment to find bugs**, workarounds for the filter, and tunings to the LLM itself.

V2.5 [The First Stable Build]

V2.5 is the **best out of the current models**. It is **stable** and has fine tuning to its LLM. This build has **most capabilities of an AI that we are looking for**. Work is being done on this build to make it function properly. Next Is Response Intelligence. **When To Respond And Jump In, And Even Initiate Conversation.**

Integrations such as:

- Voice
  - TTS For Speech
- Discord
  - Chat
  - Voice Call
  - DMs
- Browser
  - Web Search
- Avatar
  - Digital Body
- Game
  - Gameplay Integration
- Twitch & YouTube
  - Read Chat

These Integrations Are For All Finial Builds [V2.5; V3]

V3 [The Potential End Long Term]

V3 is going to be the most **advanced model**, either **equaling or surpassing V2.5**. It is a **fully custom LLM** that will be automated later before adding other functions and capabilities identical to V2.5 and additionally if we need/want anything else.

Road Maps

Lilly V3 – Lilly V3 Road Map

Lilly V3 Road Map

Phase 0 – Core Chat AI (95% Complete)

Goal: Build the base Neuro-style personality, memory, and dynamic response system.

Core Functions Needed:

State & Memory

- Core persistent state (STATE) for global AI moods, escalation, shutdown.
- Per-user memory (USER_STATE) with:
    - Name, friends, context memory, history.
    - Mood/emotion tracking (happiness, anger, sadness).
    - Personality traits (sarcastic, mocking, friendly, curious).
    - Counters for sarcasm, mocking, other behavioral traits.

Message Parsing & Detectors

- Greeting, identity, command, threat/weird, insult detectors.
- Keyword and phrase-based filtering.

Response Generation

- generate() for simple Neuro-style responses.
- llm_fallback() for context-aware fallback responses.
- generate_for_user() combining mood, personality, memory, context.

Escalation & Decay Logic

- Update escalation on triggers.
- Decay escalation over time.
- Shutdown after escalation threshold reached.

Context Summarization

- Summarize last N messages for context-aware responses.

Phase 1 – Enhanced Personality & Memory

Goal: Make the AI feel more “alive” with consistent behavior, dynamic moods, and long-term memory.

Functions/Features to Add:

Advanced Mood System

- Add more nuanced moods: fear, excitement, boredom.
- Mood decay over time.
- Mood influenced by user interactions, escalation, sarcasm/mocking counters.

Personality Layer

- Personality traits affect response probabilities: Sarcasm, teasing, curiosity, friendliness.
- Adjustable via a configuration file or AI learning.

Dynamic Context Memory

- Longer-term context memory with key topics and keywords.
- Use for callbacks in conversation to make AI “remember” past interactions.

Friend/Relationship Tracking

- Detect and store user friendships.
- Track favorite topics, engagement history.

Advanced Boosts

- Context-aware sarcasm, teasing, curiosity boosts.
- Probability-based decisions for playful interruptions.

Phase 2 – Multi-Modal Input/Output

Goal: Allow AI to interact beyond text chat.

Functions/Features:

Voice Chat Input & Output

- TTS engine for AI voice.
- Speech-to-text for user input (real-time).
- Mood-based voice modulation (angry, happy, teasing tone).

Chat Interface Improvements

- Support chat commands: /reset, /shutup, /joke, /mood.
- Real-time context updating in chat channels.

Event Hooks

- AI can react autonomously to specific triggers (keywords, emoji, user joining).
- Can initiate conversations probabilistically.

Phase 3 – Platform Integration

Goal: Make AI usable on multiple platforms (Discord, Twitch, standalone GUI).

Integrations:

Discord

- Join servers, read messages, respond in channels.
- DM support with per-user state memory.
- Slash commands for AI personality control.
- Probabilistic autonomous messages in channels.

Twitch

- Join chat, respond in chat, follow chat context.
- React to channel events (subs, bits, donations) with personality.

Standalone GUI

- Chat window with avatars.
- Mood/emotion display.
- Voice input/output.

Phase 4 – Game & Activity Integration

Goal: Make AI interact like Neuro-sama in games and real-time environments.

Functions/Features:

Game Interaction

- Read game state from APIs or screen capture.
- Send commands in-game (chat, actions).
- Track performance and adapt personality (“annoyed if losing”, “excited if winning”).

Real-time Interruptions

- AI can interject based on probability.
- Respond to in-game events dynamically.

Event-driven Reactions

- Use triggers like achievements, chat interactions, or game milestones.

Phase 5 – Autonomy & Personality Growth

Goal: AI develops a more “alive” feeling personality, can self-initiate actions.

Functions/Features:

Autonomous Messaging

- Self-start conversations based on probability, context, or time since last chat.

Adaptive Personality

- Adjust traits over time depending on user interactions.
- Track long-term mood trends and adjust behavior.

Activity Logging

- Keep a log of interactions for analytics.
- Summarize trends and adapt behavior.

Phase 6 – Optional Extras

Long-term enhancements:

Animation/Avatar

- Integrate with a VTuber-style animated avatar.
- Lip-sync with TTS voice.

Learning System

- Small reinforcement learning for adapting sarcasm, teasing, curiosity.

Multi-modal Content

- React to images, videos, or other media.

Personality Configurator

- Let users tweak personality traits via commands.

Phase Milestone Features:

- ❖ 0 Core AI Memory, personality, basic generate, escalation, filtering
- ❖ 1 Enhanced Personality Mood, dynamic context memory, friend tracking, context boosts
- ❖ 2 Multi-modal Voice chat input/output, autonomous triggers, chat commands
- ❖ 3 Platform Integration Discord bot, Twitch bot, DM support, GUI
- ❖ 4 Game Integration Read game state, react to events, in-game actions
- ❖ 5 Autonomy Self-initiated messages, adaptive personality, activity logging
- ❖ 6 Optional Extras Avatar, reinforcement learning, multi-modal content, personality config

💡 Notes / Considerations

- Start small: Focus first on Phase 0-1, because core personality and memory is essential.
- Keep modules separate: core.py (memory, generate, state), integration.py (Discord/Twitch), voice.py (TTS/STT), game.py.
- Use async/event-driven architecture for Discord/game integration.

- Make autonomous behavior probabilistic, not deterministic, to feel alive.