SRS 28/07/25

{Software Requirement Specifications}.

Topic:
Personal AI Assistant with context retention.

① Introduction
The personal AI Assistant is designed to facilitate intelligent and human-like conversations. It leverages multi-level memory (short-term, long-term and semantic memory) to maintain context across conversations. The assistant learns user preferences and adapts communication style to enhance user experience.

② Purpose & Problems
We have some concerns with the present AI agents and models like not having the personalized experience but our project will serve all the needs of users. The purpose of this system is to develop a conversational AI assistant capable of managing context, learning user habits, and responding intelligently based on historical and semantic understanding. It will use Google Gemini for it's LLM backbone.

③ Technology

Component	Technology/ Tool	Purpose
LLM Backend	Google Gemini Pro	Natural language Processing and context aware responses.
Memory store	Redis (short-term) PostgreSQL (long-term)	Retains session context and long-term user data.
Knowledge Graph	Neo4J / ArangoDB	Represents semantic memory and relationships.
Frontend	ReactJs with Chat UI Libraries	Interactive user interface for web and mobile
Backend	FastAPI with websocket support.	Handles requests user profiles and real-time messaging.
Task Queue	Celery + Redis	Manages background tasks like remainders
Authentication	OAuth 2.0/ JWT tokens	Secures user login and sessions.
Monitoring	ELK stack	Logs analytics and performance monitoring
Security	HTTPS, encryption, RBAC	Ensures secure communication and data handling.

④ Conclusion

The personal AI Assistant aims to transform user interaction through advanced context retention and adaptive Intelligence. By leveraging multi-level memory, natural language understanding and behavioral personalization, the assistant ensures a dynamic and intutive conversational experience.

⑤ Functional Requirements (FR)

- FR1
Maintain short-term and long term conversational context.
- FR2
store and update user preferences and usage patterns;
- FR3
Use Google gemini API to generate contextual responses.
- FR4
Handles multiple requests types
- FR5
Adapt communication style over time.
- FR6
Build and update user profiles based on the interaction history.
- FR7
Provide fallback mechanisms while the LLM is uncertain.

• FR8
Security log and encrypt conversation history for personalization.

⑥ Non-functional Requirements

• NFR1
Response time should be under 2 seconds for standard queries.

• NFR2
Accuracy of contextual memory management must exceed 90%.

• NFR3
Ensure data privacy and security using encryption standards.

• NFR4
Support up to 1,000 concurrent users.

• NFR5
System availability should be upto 99% uptime

⑦ User Identification
our System supports three types of users, each with distinct roles and interactions.

1) Child User
* Primary user interacting with the AI assitant through voice or text.

2) Parent User
Monitor child activity configure permissions and receive alerts.

3) Admin User
Maintain and configure the system backend and AI Logic.

⑧ Workflow for each User:

Child User

- Login or initiate voice/text interaction.
- AI Assistant listens and detect intent
- Emotion & context analyzed.
- System adjusts response time & complexity.
- Response generated and delivered.

Parent User

- Login to Dashboard
- Access child's conversation/emotion history.
- Set permissions or preferences.
- Receive system alerts when needed.

Admin User

- Login to Admin Panel
- Configure assistant behaviour or models.
- Manage security and performance logs.
- Monitor usage analytics.

⑨ Use Case

Use case	Actor	Description
• User interaction	User	User sends a message to AI Assistant.
• Context Retention	System	Assistant maintain short & long term memory of conversation.

Preference learning	System	AI adapts based on user language, tone, or past behaviour.
Profile Management	User	Users can view/ edit preferences
Alert Triggering	System	AI sends alert to parent/ admin based on emotional signals.