AI Voice Agent

Group Temp

Team Members:

- ❖ Sayed Babiker
- ❖ Chukwudera Duru
- ❖ Nathaniel Joseph
- ❖ Madina Maingua
- ❖ Alvin Ododa
- ❖ Yashwanth Pentakota

Project Description

The purpose of this project is to create an efficient backend system that enables real-time voice conversation between users and an AI assistant. The system aims to process the voice input, identify the user's intention, track the conversation context, and provide a quick response to the user while handling multiple users concurrently. The system is designed using Amazon Web Services (AWS) to ensure scalability and reliability by implementing a cloud-based system. Rather than operating a constant server, the system utilizes serverless elements, meaning that the system consumes resources only when requests are made, making it more efficient and economical.

The user's request enters the system via AWS API Gateway, which serves as the interaction interface between the client application and the backend system. The API Gateway receives audio requests, validates them, and forwards them to processing elements. The request is processed by AWS Lambda functions based on an event-driven approach, where the system processes voice input, transmits the text to the AI module, produces a response, and delivers the result to the user. Since Lambda scales automatically, the system can handle multiple conversations concurrently without requiring configuration.

To ensure the continuity of the conversation, Amazon DynamoDB is utilized for storing session information, which enables the assistant to recall past conversations and engage in multi-turn conversations rather than processing each request as a standalone task. The project also tackles the issues of delays and failures in the backend by incorporating logging functionality, timeouts, and retries to ensure that the application remains operational even when errors occur.

The platform showcases how serverless cloud technology and AI processing can be integrated to develop a scalable and responsive voice assistant system.

Project Schedule and Timeline (Gantt Chart)

Team Deliverable 1 – Project Proposal
This phase is dedicated to planning the project and establishing the scope and requirements of the project, as well as developing the initial project plan and establishing the risk management plan.
Jan 20 – Feb 1

- Planning
- Project scope & requirements
- Initial project plan and risk management

Team Deliverable 2 – System Specifications & Requirements
This phase involves defining the system design and selecting the AWS services to be used, as well as establishing the functional and non-functional requirements.
Feb 2 – Feb 15

- System architecture design
- AWS service selection
- Functional & non-functional requirements

Project Phase I – Core Implementation
This phase is dedicated to developing the core backend system and establishing the AWS services and AI voice features.
Feb 16 – Apr 5

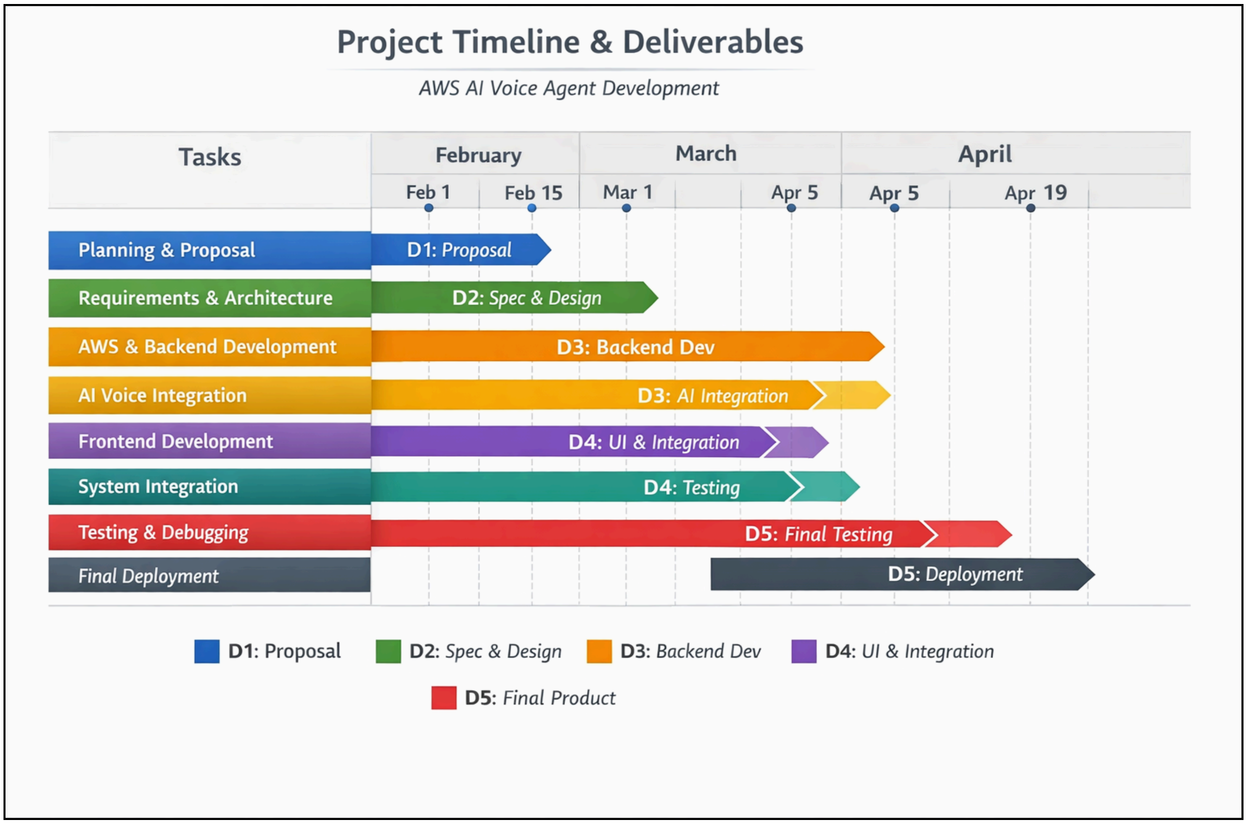
- AWS setup (API Gateway, Lambda, DynamoDB)
- Backend development
- AI voice (STT / TTS) integration
- Core functionality completed

Project Phase II – Integration & Testing
This phase involves integrating the entire system and performing tests.
Apr 6 – Apr 19

- System integration
- End-to-end testing
- Bug fixes and performance tuning

Project Phase III – Final Product
This phase is for the final preparation of the system for release, documentation, and presentation.
Apr 20 – Apr 26

- Final deployment
- Documentation
- Final presentation preparation



Project Start: February 3, 2026 (planning week)
Project End: Mid–late April 2026

The Gantt chart illustrates a general project schedule from early February, which starts with planning and architecture design, and then moves to testing and deployment which ends in April.

Risk Management

(i) Technical Performance Risk – High Response Latency

Description:
Since the system uses multiple AWS services (API Gateway, Lambda, AI processing, and database retrieval), the voice agent may experience delays in response to the user's request.

Impact:
If the response is delayed, it will disrupt the normal flow of a conversation and make the voice assistant frustrating or unusable for real-time conversations.

Risk Reduction Plan:

- Have AWS Lambda functions ready so that the responses start quickly
- Store the conversation data in DynamoDB in an efficient way
- Send less data between AWS services
- Send the responses in parts so that they appear fast
- Set time limits and retry the request if there is an error

Backup Plan:
In case of a response taking too long, a message such as ‘Please wait while I process your request’ can be given.

(ii). Input Accuracy Risk - Speech Recognition Errors

Description:
The Speech-to-Text service may fail to recognize the speech of the users due to various reasons such as accents, background noise, and microphone quality.

Impact:
The responses provided by the AI are incorrect, and this affects the reliability and trust of the users.

Risk Reduction Plan:

- Use noise filters and pre-process the input
- Verify speech recognition confidence scores
- Request confirmation of unclear requests from users
- Offer text input option to users

Backup Plan:
The agent will request users to repeat the request if the confidence level falls below a certain threshold.

(iii). System Reliability Risk – Cloud Service Failure

Description:
If the cloud services, like AWS Lambda timeout, API Gateway error, and database connectivity, fail, the ongoing sessions on the server will be terminated.

Impact:
This will cause a loss of conversation states and system downtime.

Risk Reduction Plan:

- Retry the requests if something goes wrong
- Store session data in DynamoDB after every interaction
- Use AWS CloudWatch to monitor system activities
- Use a backup response if something goes wrong

Backup Plan:
If the cloud service fails, the session states will be resumed and the users will be notified.

Team Members Roles and Responsibilities

Madina Maingua — Team Lead & Front-End Developer

Madina serves as the team leader and is responsible for coordinating project activities and guiding the overall development process. She manages the project plan, facilitates team communication, and ensures milestones are met on time. On the technical side, she leads front-end development, focusing on designing and implementing the main user interface for the AI Voice Agent Platform.

Nathaniel Joseph — Front-End Developer

Nathaniel focuses on front-end layout design and user flow implementation. He contributes to building screen layouts, refining user experience, and testing front-end functionality. He also assists with identifying and fixing front-end issues during integration and testing phases.

Sayed Babiker — Front-End Developer

Sayed is responsible for developing user interface components and features for the AI voice agent application. He works closely with the backend team to ensure the front-end integrates smoothly with backend services. His role includes building UI elements that support voice interaction and assisting with connecting the interface to backend APIs.

Yashwanth Pentakota — Risk Management & Front-End Developer

Yashwanth plays a dual role in front-end development and risk management. He helps identify potential project risks, including technical, scheduling, and integration risks, and works with the team to develop mitigation strategies. Additionally, he contributes to front-end development and testing-end development and testing to support system reliability.

Chukwudera Duru — Back-End Developer

Chukwudera is responsible for backend development tasks, including implementing AWS Lambda functions that handle voice input processing and system responses. He works on managing conversation logic and ensuring backend services function reliably. He also contributes to handling data storage and retrieval for conversation sessions.

Alvin Ododa — Back-End Developer

Alvin focuses on backend infrastructure and cloud deployment. His responsibilities include configuring AWS services such as API Gateway and DynamoDB, integrating backend components, and preparing the system for deployment. He also assists in backend testing and system launch preparation.