# AI Receptionist - Step-by-Step Workflow

## Part 1: Functional Workflow

1. Step 1: Customer sends a message on WhatsApp (text or voice).
2. Step 2: Flask receives the message via /webhook.
3. Step 3: If the message is voice:
4. - Download audio.
5. - Transcribe using Whisper.
6. Step 4: If the message is text:
7. - Use the text as-is.
8. Step 5: Translate message to English.
9. Step 6: Identify intent using Groq LLM.
10. Step 7: Decide which tools to run:
11. - check\_in\_db
12. - check\_in\_document
13. - suggest\_clothing\_combination
14. - calculate\_profit
15. - get\_current\_date
16. Step 8: Run selected tool(s).
17. Step 9: Format the result.
18. Step 10: Send response back to WhatsApp via Twilio.

## Part 2: Technical Workflow

### Flask Setup

* Set up Flask with routes:
* - /webhook
* - /logs
* - /chat
* - /

### Twilio Integration

* Connect Twilio to /webhook.
* Message Handling
* - Read incoming message.
* - Check if text or voice.
* - For voice:
* - Get media URL.
* - Download and convert to WAV.
* - Transcribe to text using Whisper API.
* - For text:
* - Use message body.

### AIReceptionist Class Flow

* Input goes to orchestrator(user\_input).
* Translate input using translate\_to\_english().
* Identify tools using Groq.
* Run matched tools:
* - check\_in\_db()
* - check\_in\_document()
* - suggest\_clothing\_combination()
* - calculate\_profit()
* - get\_current\_date()
* Combine results.
* Return final response.

### Tool Functions

* translate\_to\_english(text)
* extract\_parameters(text)
* check\_in\_db(user\_input)
* check\_in\_document(user\_input)
* query\_llm\_with\_chunks(response, groq\_api, user\_input)
* suggest\_clothing\_combination(user\_input)
* get\_current\_date(user\_input)
* calculate\_profit(user\_input)

### Pinecone

* Initialize if needed.
* Store vectorized chunks.
* Query index with user input.
* Return top results.

### Product DB

* Connect to database.
* Run query with color/material/quality.
* Return rates in table format.

### Voice Transcription

* Download audio.
* Convert to WAV.
* Transcribe using Whisper.

### Logging

* Use logger.info and logger.error in all methods.
* View logs via /logs.