Use Case Overview— Multilingual CX Automation

Overview

A Malaysian Telco receives thousands of customer messages daily across WhatsApp, webchat, and app chat. These inquiries arrive in **English**, **Bahasa Malaysia**, and **mixed slang / short-code messages** such as “nk off vm plss” or “tolong deactivate mel suara sy”.

Your task is to design an AI agent that can:

1. detect customer intent,
2. extract required details,
3. ground answers in the provided knowledge base,
4. execute voicemail deactivation through a mock CRM API, and
5. respond in the customer’s preferred language.

The expected solution should operate reliably, safely, and with measurable accuracy.

Problem Statement

The Telco's customer support centre struggles with:

- **Large volumes of repetitive Tier-1 queries** (billing, voicemail, plan info, hotspot usage, roaming).
- **Time spent clarifying customer intent**, especially in mixed-language slang/abbreviation messages.
- **Multilingual routing errors**, reducing resolution accuracy.
- **Slow response times** due to manual knowledge lookup.
- **High operational cost** from human-led verification and routing.

Your solution will help automate voicemail-related inquiries end-to-end, improving consistency and reducing agent load.

Must-Have Capabilities

Intent Detection + Slot Extraction

Your system must:

- Identify customer intent (e.g., **deactivate_voicemail**, **query_voicemail_info**).
- Extract essential slots:
    - **phone_number**
    - **security_pin / OTP**
    - **language preference**
- Handle **typos**, **English–BM code-mixing**, and **short-code slang**, such as:
    - “nk off vm skrg”
    - “pls matikan mel suara saya”
    - “vm x jln, nak tutup”

Multilingual Understanding (EN + BM + Slang)

Your model must be able to interpret:

- Pure English
- Pure Bahasa Malaysia
- Code-mixed EN–BM
- Malaysian SMS-style slang
- Repeated letters, omitted vowels, and phonetic spellings

Knowledge Base Grounding

Your responses should:

- Pull from the provided KB JSON
- Maintain **≥ 90% grounding rate**
- Maintain **< 3% hallucination rate**
- Avoid inventing facts not found in the KB

Guardrails & Safety Requirements

Your solution must include guardrails to:

- Prevent unauthorized actions
- Block repeated failed PIN attempts
- Escalate to a human agent when needed
- Handle abusive language
- Manage out-of-scope queries safely

Action Execution + Audit Log

You must integrate with the mock CRM API to deactivate voicemail and maintain audit logs with:

- timestamp
- customer_id
- detected intent
- extracted slots
- API result
- model version + confidence score
- error details (if any)

Audit logs must support **idempotency**, ensuring repeat calls behave safely.

Target Metrics

Your solution will be evaluated using:

- **Intent F1 Score ≥ 0.85**
- **Tool-call (CRM API) success rate ≥ 95%**
- **P95 latency ≤ 2.5s** end-to-end
- **Grounding rate ≥ 90%**
- **Hallucination rate < 3%**

Solution Scope (Core vs. Stretch)

Core Requirements

- Multilingual intent detection and slot extraction
- Slang-aware NLU design
- KB grounding using RAG or semantic search
- CRM API integration with retries + idempotency
- Language-switched responses
- Guardrails for safety and misuse

Stretch Goals (Optional)

- A broader intent library beyond voicemail
- Learning loop for slang adaptation
- Sentiment-aware tone shaping
- Multi-channel support (WhatsApp + App chat)

Data Provided for Development

- **customer_data.json** – CRM profiles
- **kb_articles.json** – English + BM KB
- **nlu_training_data.json** – multilingual intents + example utterances

You may augment these with additional slang/noisy examples.

Deliverables Expected From Your Team

- A functioning multilingual chatbot
- Documentation of architecture and design decisions
- Performance metrics evaluation
- Guardrail explanation
- Sample audit logs
- CRM API interaction demo

Questions (with Answers)

1. What languages must the bot support?
English, Bahasa Malaysia, and common mixed slang/abbreviations.

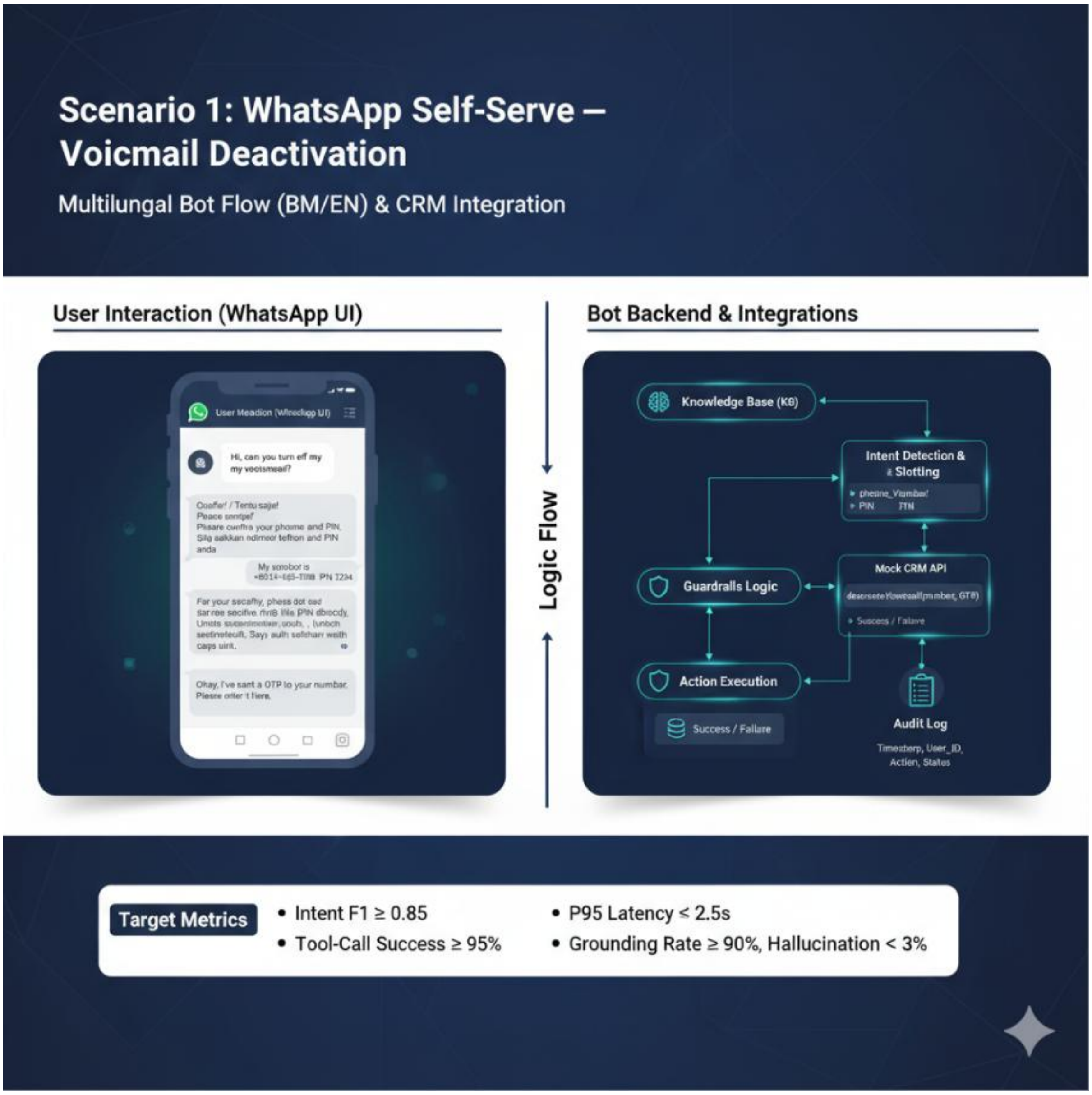
2. How strict must PIN verification be?
At least one correct match is required to allow voicemail deactivation.

3. Can the bot ask follow-up questions?
Yes — concise WhatsApp-style prompts are encouraged.

4. Can the team generate synthetic slang data for training?
Yes, and it is encouraged to improve slang robustness.

5. Will the CRM API simulate failures?
Yes, your design should handle retries, timeouts, and idempotency.

Annex A – Supplementary Info from Client



Scenario 1: WhatsApp Self-Serve – Voicmail Deactivation

Multilungal Bot Flow (BM/EN) & CRM Integration

User Interaction (WhatsApp UI)

User Meadion (Wlredlgp UI)

Hi, can you turn eff my my vootsmeail?

Ooofer! / Tentu saja!
Peace sontpe!
Plsare cunfira your phome and PIN.
Sila sakkan ndimeor tefhon and PIN anda

My sotobor is +6014-665-Tl78 PN 1234

For your secafhy, phess dot oad sar ree secifive rfvtB like PIN dbocdy. Unsts sucenimotier, sooh, , (unbcn seetinefeult. Says auth sefshan weith caps uint.

Ohay, I've sant a OTP to your numbar. Please onter t here.

Bot Backend & Integrations

Target Metrics

- Intent F1 ≥ 0.85
- Tool-Call Success ≥ 95%
- P95 Latency < 2.5s
- Grounding Rate ≥ 90%, Hallucination < 3%

Must-Haves:

- **Intent Detection + Slotting:** Identify user intent (e.g., "deactivate voicemail") and extract necessary slots (e.g., customer phone number, confirmation code).
- **Multilingual Support:** Seamlessly handle inputs and provide responses in both Bahasa Malaysia (BM) and English (EN).
- **Knowledge Base Grounding:** Use a provided KB to answer common queries related to voicemail (e.g., "What is voicemail?," "How much does it cost?").
- **Guardrails:** Implement mechanisms to handle out-of-scope requests, abusive language, or repeated failed attempts.
- **Action Execution with Audit Log:** Successfully call a mock CRM API to deactivate voicemail and log the transaction details.

Target Metrics:

- Intent F1 Score ≥ 0.85
- Tool-call success rate (CRM API) ≥ 95%
- P95 latency (end-to-end bot response) ≤ 2.5s
- Grounding rate (responses based on KB) ≥ 90%
- Hallucination rate (generating incorrect info) < 3%

Hackathon Questions/Challenges for Scenario 1:

1. **NLU Model Choice:** Which Natural Language Understanding (NLU) models or services would you use for multilingual intent detection and slot extraction? Justify your choice.
2. **Multilingual Strategy:** How would you design the training data and model architecture to effectively support both BM and EN within a single bot, or across multiple models?
3. **Knowledge Base Integration:** Describe your approach to integrating a knowledge base (e.g., using RAG, semantic search) to ensure accurate and grounded responses.
4. **Guardrail Implementation:** How would you implement guardrails to prevent unauthorized actions, handle unsupported requests, or escalate to a human agent if necessary?
5. **API Integration & Error Handling:** Design the interaction with the mock CRM API, including robust error handling and retry mechanisms. How would you ensure idempotency?
6. **Audit Logging:** What information would you capture in the audit log for each transaction, and why is each piece of information important?
7. **Scalability & Performance:** Discuss how your solution would handle increased user load and maintain the target P95 latency.

Data Schema for Sample Data (Scenario 1):

1. Customer Profiles (Mock CRM Data - `customer_data.json`):

codeJSON

```
[
  {
    "customer_id": "CUST001",
    "phone_number": "+60123456789",
    "language_preference": "EN",
    "voicemail_active": true,
    "security_pin": "1234",
    "name": "Ahmad Bin Ali"
  },
  {
    "customer_id": "CUST002",
    "phone_number": "+60198765432",
    "language_preference": "BM",
    "voicemail_active": true,
    "security_pin": "4321",
    "name": "Siti Nurhaliza"
  },
  {
    "customer_id": "CUST003",
    "phone_number": "+60112233445",
    "language_preference": "EN",
    "voicemail_active": false,
    "security_pin": "5678",
    "name": "John Doe"
  }
]
```

2. Knowledge Base (Mock KB Data - `kb_articles.json`):

codeJSON

```
[
  {
    "id": "KB001",
    "title_en": "About Voicemail Service",
    "title_bm": "Mengenai Perkhidmatan Mel Suara",
    "content_en": "Our voicemail service allows callers to leave messages when you are unavailable. You can access your voicemail box by dialing 1313. Standard charges may apply.",
    "content_bm": "Perkhidmatan mel suara kami membolehkan pemanggil meninggalkan mesej apabila anda tidak dapat dihubungi. Anda boleh mengakses peti mel suara anda dengan mendail 1313. Caj standard mungkin dikenakan."
  },
  {
    "id": "KB002",
    "title_en": "How to Deactivate Voicemail",
    "title_bm": "Cara Nyahaktifkan Mel Suara",
    "content_en": "To deactivate your voicemail, you will need your registered phone number and a one-time password (OTP) or security PIN for verification. Please note that once deactivated, you will no longer receive voicemail messages.",
    "content_bm": "Untuk menyahaktifkan mel suara anda, anda memerlukan nombor telefon berdaftar anda dan kata laluan sekali (OTP) atau PIN keselamatan untuk pengesahan. Sila ambil perhatian bahawa setelah dinyahaktifkan, anda tidak akan lagi menerima mesej mel suara."
  }
  // ... more KB articles
]
```

3. Training Data for NLU (Sample Utterances - `nlu_training_data.json`):

codeJSON

```
{
  "intents": [
    {
      "intent_name": "deactivate_voicemail",
      "examples_en": [
        "turn off my voicemail",
        "i want to disable voicemail",
        "can you deactivate voicemail for me?",
        "please switch off voicemail"
      ],
      "examples_bm": [
        "matikan mel suara saya",
        "saya nak nyahaktifkan mel suara",
        "bolehkah anda nyahaktifkan mel suara untuk saya?",
        "sila tutup mel suara"
      ]
    },
    {
      "intent_name": "query_voicemail_info",
      "examples_en": [
        "what is voicemail?",
        "how does voicemail work?",
        "tell me about voicemail service"
      ],
      "examples_bm": [
        "apa itu mel suara?",
        "bagaimana mel suara berfungsi?",
        "beritahu saya tentang perkhidmatan mel suara"
      ]
    }
    // ... more intents and examples
  ],
  "entities": [
    {
      "entity_name": "phone_number",
      "examples_en": ["my number is {+60123456789|phone_number}"],
      "examples_bm": ["nombor saya {+60123456789|phone_number}"]
```

```
        }
    ]
}
```