System Design Document (SDD)

Project Title: AI-Powered Chatbot to Reduce Customer Support Call Costs
Prepared By: [Your Name]
Date: 23 June 2025

---

1. Objective

The goal of this system is to develop a scalable, intelligent chatbot that can handle customer support queries and significantly reduce the number of phone calls to the helpdesk. Each call to the helpdesk currently costs the company approximately **AUD 20 for a 10-minute call**, with an average of **500 calls per weekday and 100 per weekend day**. This translates into a substantial operational cost, which this chatbot system aims to minimize. The chatbot will leverage **Azure OpenAI for conversational AI**, **Hugging Face for retrieval-augmented generation (RAG)**, and **Azure Synapse Analytics with Power BI** for tracking performance and impact.

---

2. High-Level Architecture Overview

This system integrates various cloud services and machine learning components, each serving a specific purpose:

- **Azure OpenAI**: Handles natural conversation and generative answers.
- **Hugging Face RAG**: Answers based on company-specific documents using Retrieval-Augmented Generation.
- **Chroma Vector Database**: Stores embeddings for similarity-based document retrieval.
- **Gemini Embeddings**: Converts company documents into numerical vector format.
- **Azure Data Lake Storage (ADLS)**: Stores documents, logs, and unstructured data.
- **Azure Synapse Analytics**: Processes interaction data for reporting.
- **Power BI**: Visualizes analytics on chatbot performance.
- **Azure Active Directory (AAD)**: Manages user roles and access control.
- **Azure Cloud Infrastructure**: Hosts databases, services, and integrations.

---

3. System Components and Their Roles

3.1 Chatbot Interface

The chatbot will be integrated into the company's existing **website and mobile app interfaces**. This integration layer is **not in scope** for this project, as the company already has reusable components that support chatbot deployment. The chatbot will appear as an

interactive widget, allowing customers to type queries and receive intelligent responses in real time.

3.2 API Gateway

An **Azure API Management Gateway** will serve as the secure entry point for all chatbot interactions. It will manage routing, API versioning, request authentication, and throttling. This ensures high availability and security for user requests.

3.3 Authentication and Role-Based Access Control

Access to the chatbot and the underlying data will be managed through **Azure Active Directory (AAD)**. Users (customers, support agents, and admins) will be assigned specific roles, and permissions will be enforced accordingly:

- **Customers**: Can interact with the chatbot and receive answers to their queries.
- **Support Agents**: Can access unresolved queries and assist when escalation is needed.
- **Administrators**: Can view analytics dashboards, logs, and configure chatbot settings.

3.4 Chatbot Orchestrator Layer

This is the decision-making engine of the chatbot, responsible for routing queries to either Azure OpenAI for generative responses or to Hugging Face RAG for document-based responses. This orchestration logic can be hosted using **Azure Functions** or **Azure Kubernetes Services (AKS)** for flexibility and scalability.

---

4. AI and Retrieval Components

4.1 Azure OpenAI (GPT-4)

Azure OpenAI will be used to power the natural conversation experience. It can handle general queries, provide polite responses, maintain context, and manage follow-up questions. It will be particularly useful for casual, generic support queries that don’t require specific document lookups.

4.2 Hugging Face RAG

For questions that depend on internal documents (such as policy details, terms, or process guides), a **Retrieval-Augmented Generation (RAG)** model will be used. This model will retrieve relevant documents from a vector database and then generate a response based on both the retrieved content and the customer’s question.

4.3 Gemini Embeddings

Before documents can be used in a RAG pipeline, they need to be converted into a numerical form (embeddings). This will be done using **Gemini embedding models**, which are capable

of creating rich semantic representations of text. These embeddings are then stored in a vector database for fast retrieval.

4.4 Chroma Vector Database

The embeddings generated from documents using Gemini will be stored in **Chroma**, a high-performance vector database. This enables similarity-based retrieval when a customer query is matched with relevant documents from the company's knowledge base.

---

5. Data Storage and Analytics

5.1 Azure Data Lake Storage Gen2 (ADLS)

All raw documents used for RAG (such as FAQs, internal guides, policies) will be stored in **Azure Data Lake Storage (ADLS)**. Additionally, all chatbot interaction logs, escalation records, and user feedback will be stored here for auditing and training purposes.

5.2 Azure Synapse Analytics

Data collected from chatbot interactions and ADLS will be processed using **Azure Synapse Analytics**. This includes:

- Number of conversations
- Query resolution rates
- Escalation metrics
- Chat duration and satisfaction ratings

The output of this processing will be used to feed the dashboard.

5.3 Power BI Dashboard

Power BI will be used to visualize chatbot performance and business KPIs. The dashboard will provide insights such as:

- Number of users served
- Percentage of queries resolved without escalation
- Reduction in call center load
- Average response times
- Cost savings estimation

These dashboards will be shared with operations and leadership teams for continuous monitoring.

---

6. Security and Compliance

- **Data Encryption**: All data will be encrypted at rest and in transit.
- **Access Control**: Enforced via Azure AD with role-based permissions.
- **Logging**: Chat logs and system events will be monitored via **Azure Monitor** and **Log Analytics**.
- **GDPR Compliance**: Personally identifiable data will be anonymized where required.

---

7. Deployment Strategy

The system will be deployed using **Azure-native DevOps pipelines** or **GitHub Actions**. All infrastructure components (databases, vector DB, APIs) will be provisioned using **Infrastructure as Code (IaC)** tools such as **Terraform** or **Bicep**.

The initial deployment will follow a **staging → production** flow to ensure proper testing and validation. Future updates to the chatbot logic or knowledge base will be handled using CI/CD pipelines.

---

8. Scalability and Availability

Component	Scalability Strategy
Azure Functions	Auto-scaled based on traffic
Vector DB (Chroma)	Read replicas and sharding as needed
RAG Inference (AKS)	Horizontally scalable pods
Synapse Analytics	Scalable compute tiers
Power BI	Optimized using imported or live query models

The architecture will aim for **99.9% uptime**, with failover mechanisms and backup policies in place for critical components.

---

9. Monitoring and Alerting

- **Azure Monitor** will track system health, usage, and error rates.
- **Application Insights** will trace user interactions and system performance.
- **Alerts** will be configured for downtime, low response rates, and unusual activity patterns.
- Feedback data from users will also help improve the model's accuracy and usability.

---

10. Future Enhancements

- Support for **voice-based interactions** using Azure Speech Services.
- Integration with **multilingual support** for global users.
- Feedback loop to allow retraining of the model with real customer interactions.
- Proactive notifications via WhatsApp or Email using Azure Communication Services.