**🧾 Chatbot Design Documentation Structure**

Here’s a scientific structure you can follow (similar to an academic or software engineering design document):

**1. Introduction**

* **Project Title**: UnityToServe Hybrid Smart Chatbot
* **Purpose**: Briefly describe the problem this chatbot solves.
* **Scope**: Define the scope—FAQ answering, user guidance, donation assistance, etc.
* **Audience**: Who will use the chatbot (e.g., donors, volunteers, partners, refugees).

**2. Objectives**

* Offer 24/7 instant support to website visitors
* Automate responses to frequently asked questions
* Provide multilingual, inclusive communication
* Capture leads (email/contact info) for follow-up
* Support volunteers/donors through the onboarding process

**3. System Overview**

* **Type**: Hybrid chatbot using rule-based NLP (spaCy) + generative model (Transformers)
* **Platform**: Website chat widget
* **Architecture Overview**:
  + Frontend: JavaScript embedded chat widget
  + Backend: Python (Flask or FastAPI)
  + NLP: spaCy (intent classification)
  + Response Generation: Transformers (e.g., distilgpt2)
  + Optional: Lead capture + email automation

**🧠 Phase I: Documenting from Scratch (Now)**

Start with the following 3 sections:

**🔹 1. Introduction**

markdown

CopyEdit

## 1. Introduction

### 1.1 Title

UnityToServe Hybrid Smart Chatbot

### 1.2 Purpose

This document outlines the design and implementation plan for a smart chatbot integrated into the UnityToServe.org website. The chatbot will provide 24/7 automated assistance to users seeking to learn more about our programs, donate, volunteer, or request support.

### 1.3 Scope

The chatbot will:

- Answer frequently asked questions about Unity to Serve’s mission and services.

- Direct users to donation and volunteer forms.

- Collect user information (with consent) for follow-up.

- Provide bilingual support (English and Dari in future versions).

- Function as both an information bot (rule-based) and a conversational agent (AI-driven).

### 1.4 Audience

- Potential donors

- Volunteers (local and international)

- Refugees and underprivileged users seeking support

- Unity to Serve staff (internal testing)

**🔹 2. Objectives**

markdown

CopyEdit

## 2. Objectives

- Automate the answering of 80% of routine visitor questions.

- Guide visitors to the correct action pathways (Donate, Volunteer, Contact, Learn More).

- Provide a warm and inclusive conversation experience.

- Use NLP for intent detection and transformers for response generation.

- Enable Unity to Serve to scale outreach without increasing human load.

**🔹 3. System Overview**

markdown

CopyEdit

## 3. System Overview

### 3.1 System Type

Hybrid chatbot combining:

- Rule-based NLP for deterministic responses (via spaCy)

- AI-based response generation using pretrained transformer models (via HuggingFace)

### 3.2 Components

- Frontend Chat Interface (custom or embedded widget)

- Flask/FastAPI-based backend server

- NLP pipeline (spaCy model with defined intents and entities)

- Transformer-based text generation model (DistilGPT2 or similar)

- Lead capture database and webhook integration

### 3.3 Deployment Plan

- Phase 1: Local testing with Flask

- Phase 2: Integration into UnityToServe.org staging environment

- Phase 3: Production deployment with multilingual and lead capture enabled