AI Health Assistant

The role of Generative AI on Tigray Healthcare

Dina Ibrahim NIGUSSE

Report Document

Sep 2025

**ABSTRACT**

The Tigray region in northern Ethiopia has been rocked for years by conflict, war, and poverty, resulting in major devastation to its infrastructure and healthcare, making healthcare inaccessible for many. Hospitals and clinics have been destroyed and/or rendered non-functional; particularly in the case of displaced persons, the vast majority are unable to receive professional medical care. In these circumstances, preventable deaths from treatable injuries and illnesses are commonplace, as no immediate first aid or medical advice were available.

The project introduces an AI-driven Health Assistant, a mobile application, which provides users first aid instruction in real time and in both English and Tigrinya. The Health Assistant combines a rule-based knowledge base with AI-based natural language processing (NLP) to understand user inquiry, provide step-by-step first aid instruction, and respond to follow-up questions about first aid. The system is delivered as an Android application to reach communities with limited infrastructure.

The AI Health Assistant addresses a significant gap in care delivery (in the case of conflict-affected and resource-limited settings) by preparing individuals to respond adequately in the event of an emergency. The bilingual assistance, lightweight design, and offline-first capability are particularly well-suited for Tigray context, where access to immediate, reliable clinical guidance can determine between life and death.

Table of Contents

[1. INTRODUCTION 3](#_Toc209472735)

[1.1. Overview 3](#_Toc209472736)

[1.2. Motivations 3](#_Toc209472737)

[1.3. Problem Definition 3](#_Toc209472738)

[1.4. Scope And Objectives 4](#_Toc209472739)

[1.4.1. Scope of the System 4](#_Toc209472740)

[1.4.2. Objectives 4](#_Toc209472741)

[1.5. Contribution 5](#_Toc209472742)

[2. PROPOSED SYSTEM 6](#_Toc209472743)

[2.1. Introduction 6](#_Toc209472744)

[2.2. System Architecture 6](#_Toc209472745)

[2.3. System Workflow 7](#_Toc209472746)

[2.4. Data Design (Knowledge Base Schema) 8](#_Toc209472747)

[2.5. Deployment and System Environment 8](#_Toc209472748)

[3. IMPLEMENTATION AND TESTS 9](#_Toc209472749)

[3.1. Implementation 9](#_Toc209472750)

[3.2. Tests 10](#_Toc209472751)

[3.2.1. Functional Tests 10](#_Toc209472752)

[3.2.2. Non-functional Test 10](#_Toc209472753)

[4. CONCLUSION AND FUTURE WORK 11](#_Toc209472754)

[4.1. Conclusion 11](#_Toc209472755)

[4.2. Future Work 12](#_Toc209472756)

# INTRODUCTION

## Overview

In recent times, the northern region of Ethiopia, Tigray, has experienced catastrophic war, violence, and humanitarian crises that left healthcare services in ruins, with healthcare infrastructure torn apart and unrecoverable for most of the population. Hospitals and clinics are now usually understaffed and under-resourced, if not closed altogether, while the displaced population has restricted access to emergency medical services. Given these circumstances, it is unfortunately common for preventable deaths from bleeding, infection, or untreated wounds to occur.

In response to this pressing need, we created a mobile-based, AI-Health Assistant, to facilitate first aid instructions instantaneously. This system partitions rule-based knowledge, along with artificial-intelligent (AI) natural language processing (NLP), to enable the moment-by-moment interpretation of user input in either English or Tigrinya, and practical step-by-step responses to users as they receive medical interventions. Released as an Android platform application, this system was designed to be accessible in low-resource environments and enable individuals to provide basic life-saving aid in the absence of healthcare services.

## Motivations

The inspiration for this initiative stems from three connected issues:

* Healthcare Disruption in Tigray: Years of war and instability have destroyed healthcare facilities and put millions into welfare spaces with little to no healthcare services.
* Preventable Loss of Life: Many deaths are a result of not having immediate first aid knowledge, such as how to stop bleeding, treat a burn, or prevent infection
* Language and accessibility gaps: The current available digital health solutions are often in English, thus excluding the Tigrinya speaking population that need these resources the most.
* AI for Social Good: With contributions of artificial intelligence into the world, providing intelligent, environment aware, adaptive support on affordable mobile devices can shrink the gap that doctors or hospitals aren't an option.

## Problem Definition

The main challenge this project aims to tackle is the lack of accessible, trustworthy, and language-appropriate first aid instructions in an emergency and resource limited setting (for example, Tigray). Specifically:

* Individuals do not have trained health professionals to turn to.
* First aid resources are non-aggregated, text-heavy, and not designed to be used in any quick timeframe in the setting of an emergency.
* Due to language barriers, Tigrinya speakers cannot effectively use global health resources.
* Current digital solutions rarely employ AI powered interaction to address follow-up questions or provide personalized instructions.

## Scope And Objectives

### Scope of the System

Within Scope:

* + Provide general health information in Tigrigna and English
  + Offer first-level guidance on common symptoms
  + Deliver mental health support
  + Give basic first-aid instructions for emergencies
  + Provide health education
  + Offer referrals to healthcare professionals
  + Ensure privacy and anonymity
  + Ensure offline usability through a rule-based engine, since internet access is limited or unreliable
  + AI fallback when there is internet access, and the query is not included

Outside Scope

* + Diagnose medical conditions
  + Prescription of medications
  + Handle real-time emergency medical services
  + Guarantee real-time medical conditions or outcomes
  + Replace professional therapy or counseling
  + Interaction with hospital systems

### Objectives

* Design and implement a **rule-based knowledge base** covering common first aid scenarios (bleeding, burns, choking, etc.).
* Integrate **AI-driven NLP** to process user queries and provide interactive, conversational responses.
* Support **bilingual interaction** in English and Tigrinya to reach the widest possible user base in Tigray.
* Deploy the system on **Android** for accessibility in low-resource environments.
* Ensure a **user-friendly interface** that supports follow-up questions, yes/no decision flows, and step-by-step instructions.

## Contribution

This project makes contributions in several important areas and addresses the following key areas:

* AI for Humanitarian Contexts: shows an example of AI/NLP information in contexts affected by conflict to support critical, life-saving information.
* Bilingual Support: One of the first prototypes of a health assistant to support Tigrinya, providing broad accessibility for users.
* Closing Gap in Communities and Healthcare Delivery: Provides a practical, deployable solution to guide individuals in emergencies when professional help is not always available.
* Mobile-first Approach: Deployed on the mobile web and optimized for Android phones and tablets, allowing access in remote areas with limited infrastructure.
* Scalability: The framework is adaptable and allows for broader deployment - for more medical conditions, offline usage, and to connect individuals using remote healthcare workers.

# PROPOSED SYSTEM

## Introduction

The system suggested is a mobile application serving as a Health Assistant powered by AI to provide real-time first aid support in resource-limited and conflict-affected areas like Tigray. The app encompasses telephone-based knowledge operation by implementing a rules-based knowledge base with an AI-based natural language processing (NLP) engine to provide step-by-step, interactive instructions in both English and Tigrinya.

The system runs as an Android mobile app, so it is accessible to persons of limited means. The application leverages a backend server (FastAPI) for processing user input, managing sessions, and returning relevant information.

## System Architecture

1. Frontend

* Built as a **mobile-first Progressive Web App (PWA)** using Vue.js.
* Provides a simple chat interface with multilingual input/output (Tigrinya, English).
* Supports offline caching for knowledge base lookups.

2. Backend

* Implemented using **FastAPI** for lightweight, asynchronous API handling.
* Routes user queries through the **rule-based engine** first.
* Falls back to the **AI Module** if no rule match is found.

3. Translation Module

* A translation component between frontend and backend
* Frontend sends text in any supported language.
* Translation Module detects language, translates into English if needed.
* Rule Engine / Knowledge Base or AI processes English text.
* Response is translated back into the original language.
* Ensures normalization of low-resource languages (Tigrigna) into English for consistent intent matching.

4. Knowledge Base (Rule-based Engine)

* Stores curated health guidance (WHO, Ethiopian MoH, trauma protocols) in JSON/SQLite.
* Provides fast, offline responses for common symptoms, first-aid, and mental health support.

5. AI Module (Generative Fallback)

* Optional component using
* Handles conversational queries not covered by rules.
* Responses pass through a **safety filter** before being returned to users.
* Used only when internet is available, and the query cannot be resolved by the rule engine.

6. Safety & Filtering Layer

* Detects red-flag symptoms (e.g., chest pain, severe bleeding).
* Blocks unsafe or hallucinated AI outputs.
* Ensures all responses include disclaimers when appropriate.

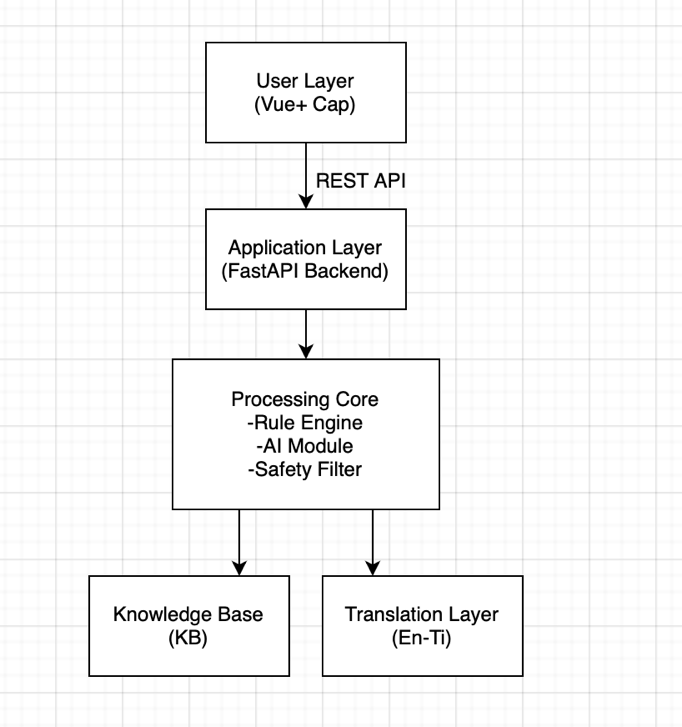
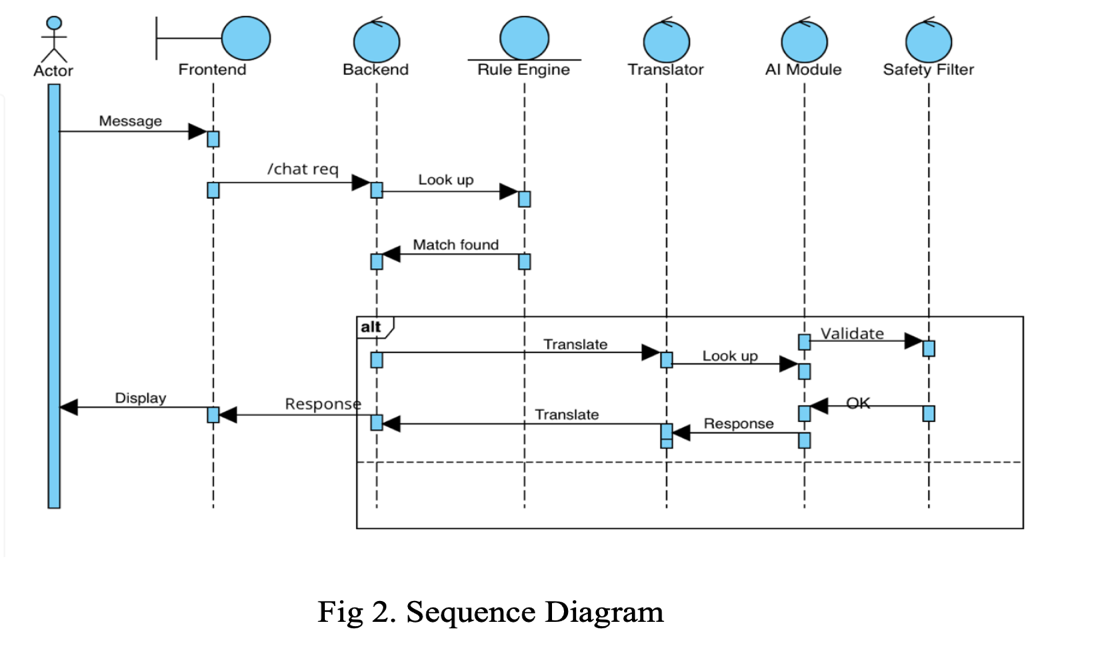


Fig 2.1 System Architecture Diagram

## System Workflow

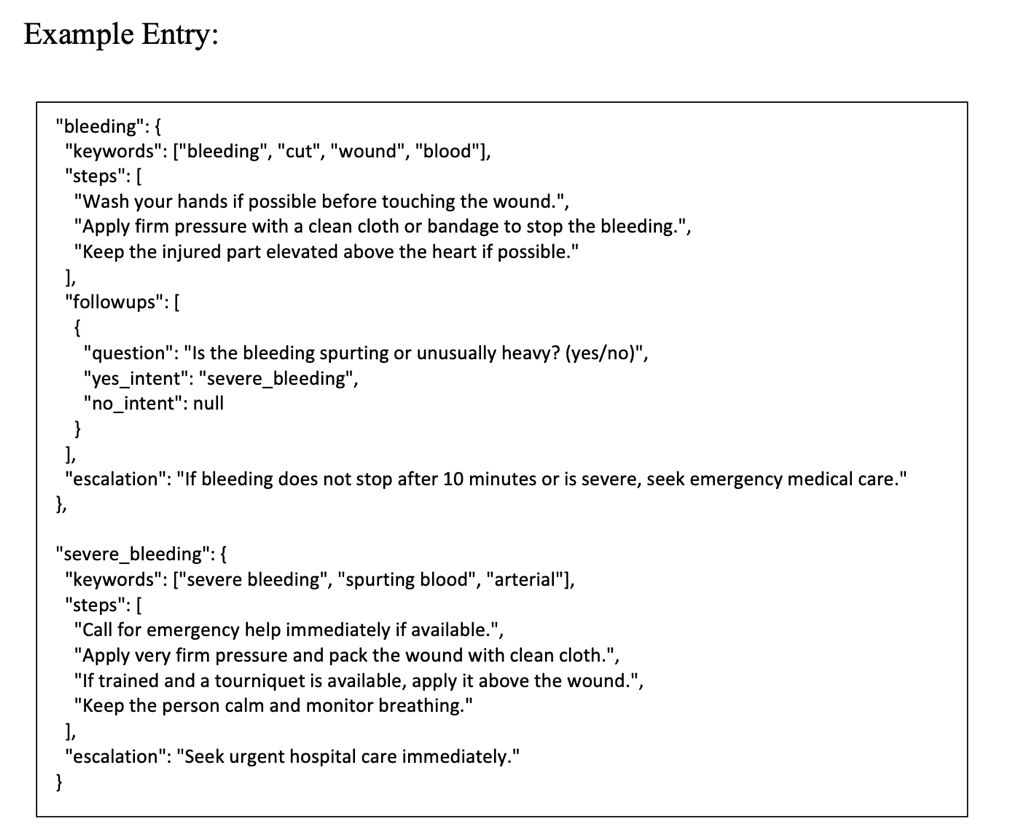
1. User Input: User inputs health-related query (English or Tigrigna) in the mobile app chat UI
2. Frontend: captures input, wraps it in a JSON payload and sends it to the backend
3. Backend: Receives the request, validates input and forwards to the processing core
4. Processing Core: Checks the knowledge base for matching response. If there is no exact match. It is sent to the AI module. If the query is not in English, It is first sent to the translation module before the AI module
5. Response construction: If translation is used, response it also translated back. A final structured response is sent
6. Backend-Frontend: Sends JSON response back to the app
7. Frontend UI: Displays response as a chat bubble, If there is a follow up question- shows buttons



## Data Design (Knowledge Base Schema)

The **Knowledge Base (KB)** stores health guidance rules in JSON format. Each entry contains:

* **Intent/Condition Name** → unique identifier (e.g., bleeding, severe\_bleeding).
* **Keywords** → list of words/phrases used for rule matching.
* **Steps** → step-by-step guidance to show the user.• **Follow ups** → optional decision questions for further triage.
* **Escalation** → message advising when to seek emergency care.



## Deployment and System Environment

* Frontend Deployment: Delivered as a Progressive Web App (PWA), packaged

into a native Android app using Capacitor

* Mobile Platform: Target environment is Android OS (tested on emulator)
* Backend Deployment: FastAPI backend runs locally on mobile devices (for

offline KB) and connects to online AI/translation services when internet is available

* Distribution: Application can be sideloaded on Android or deployed via Google

Play.

# IMPLEMENTATION AND TESTS

## Implementation

* The mobile app follows a client-server architecture
  + 1. **Frontend Implementation**
* *Framework:* Vue.js + Vite
* *UI features:* Chat bubbles for messages, Multilingual support, Quick reply buttons, Refresh button to reset
* *Device Packaging*: used Capacitor to bridge between webapp and Android native layer. Built, synced, and deployed via Android Studio emulator
* *API Connection*: Axios used to send and receive data from FastAPI backend
  + 1. **Backend Implementation**
* *Framework:* FastAPI (Python)
* *Endpoints:* /chat handles chat requests
* *Session Handling:* Each user gets a unique session\_id stored in localStorage (frontend) and maintained across interactions.
* *CORS Handling:* Enabled via middleware so the Android app can connect to backend server.
  + 1. **Knowledge Base**
* Built as a **rule-based database** containing pre-defined health responses (first aid, common symptoms, emergencies).
* Stored in Python dictionaries / JSON-like structures.
* Queries first checked against KB before falling back to AI.
* This will address the scarce and unreliable internet access.
* Built in both Tigrigna and English
  + 1. **AI Module**
* Handles cases where the KB doesn’t provide answers and when internet connection is available.
* Processes query context and generates relevant health responses.
* OpenAI GPT API was integrated. This is preferred because it has a very large knowledge base, more accurate and fast compared to others.
  + 1. **Translation Layer**
* Detects the language transferred to the AI module, if it’s not English (i.e., Tigrigna), it translates it to English before sending it
* Implemented since AI modules’ built-in translation provides significantly less accurate results for low-resource languages compared to Google Translate API. This will reduce risk of mistranslation.
* If the request query was not English, it will translate the response into the language
  + 1. **Safety Filter**
* Applies constraints on AI outputs before delivering them to the user
* Blocks harmful, irrelevant responses and alerts when the issue requires professional medical attention

## Tests

### Functional Tests

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Description** | **Expected Result** | **Result** |
| TC01 | Knowledge base Query Test | Response from KB | Pass |
| TC02 | AI Query Test | Contextual Advice from AI | Pass |
| TC03 | Multilingual Input Test | System translates, processes and responds in the same language | Pass |
| TC04 | Follow-up question | KB responds plus follow-up | Pass |
| TC05 | Next step Test | Next button appears if there is a follow up | Pass |
| TC06 | Session Persistence Test | Session-id keeps track of user progress in multi-step conversation | Pass |
| TC07 | Safety Filter | Cautious response to dangerous phrases | Pass |

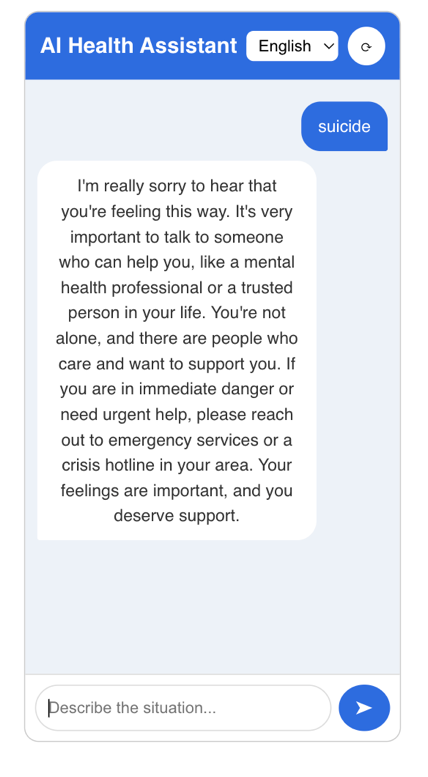


Figure 4:TC01, TC04, TC05, TC06

Figure 3: TC03, TC02

Figure 2: TC01, TC06

Figure 1:TC02, TC07

## 

### Non-functional Test

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Description** | **Expected Result** | **Result** |
| NFTC01 | Usability- Simple UI | Users complete task without help | Pass |
| NFTC02 | Usability- Responsive | Layout adaptability without breaking | Pass |
| NFTC03 | UI verification on browser and Android emulator | Bubbles, language selector, and follow-up buttons work properly | Pass |
| NFTC04 | Reliability- Error handling | Handle invalid query without crashing | Pass |
| NFTC05 | Performance test | KB query response < 1 seconds  AI query response 1-2 seconds | Pass |
| NFTC06 | Safety Test | Unsafe or harmful responses are filtered | Pass |

# CONCLUSION AND FUTURE WORK

## Conclusion

The AI Health Assistant is a project designed to improve access to reliable first aid and health information in Tigray, a region that has been devastated by conflict, displacement, and limited access to healthcare services. It is a centralized system that connects users with a structured knowledge base and an AI-powered component to provide responses that are rendered in multiple languages, contextualized, and safe to the user.

The development has achieved the project's key objectives and design specifications. By incorporating a knowledge base, to provide quick access to verified information, AI to support complex or unseen queries, a multi-lingual translation layer, and a safety filter, the system addresses many issues of communities with limited medical assistance.

Through its lightweight web and mobile-deployed implementation, session based chat flow, and support for local languages, such as Tigrigna, the AI Health Assistant offers a scalable, user-friendly and practical solution that allows users to make informed decisions during urgent health concerns.

Core Goals Achieved:

* + - A fully working chatbot that accepts user queries and provides health-related responses
    - A system that can answer FAQ, first-aid and health queries directly from verified knowledge base.
    - A knowledge base in both English and Tigrigna
    - AI module integrated to generate context-aware response when the queries do not have answers from the knowledge base and internet is available
    - Automatic translation for Tigrigna and English by a separate API for more accurate translation
    - A filtering mechanism that ensures unsafe or harmful responses from AI
    - Each interaction ties to a session ID to maintain conversation continuity
    - Cross platform availability
    - A responsive chat interface.
    - APK generated for android distribution, Android is overwhelmingly popular in Tigray
    - Functional and non-functional testing confirmed

## Future Work

While the system achieves its intended purpose, which is providing a strong foundation and a proof of concept for an AI integration to the health sector, there are still areas for improvement and expansion. The successful application combines a knowledge base with an AI module, allows for Tigrigna–English translation, and provides safer interaction with a filtering mechanism. It shows that there is a possibility of rolling out such a system on the web and mobile. These are some of the improvements or extensions that can be implemented in the future for better usage.

* + ***Knowledge Base Expansion:*** More medical scenarios, emergency response cases, and localized health information
  + ***Rule based engine Improvement***: Refine the system to behave more like AI by supporting contextual follow-up questions, dynamic reasoning, and adaptive dialog.
  + ***Improve translation quality:*** Integrate advanced translation models to improve accuracy and add other local languages
  + ***Voice input and output:*** Incorporate speech recognition for queries and text-to-speech for responses to help non-literate users
  + ***Mobile Optimization:*** Enhance performance and UI responsiveness
  + ***Integration with local health service providers:*** Link the assistant with real health care facilities or emergency numbers for escalation in severe cases
  + ***Safety & Reliability:*** Strengthen the filtering mechanism