TriageAssist-SA: AI Clinical Decision Support for ICU Patient Care

Team name: Ubuntu HealthAI Lab

Project name

TriageAssist-SA: AI Clinical Decision Support for ICU Patient Care.

Team

Bernard Swanepoel, MSc Computer Science (2026) led the backend and AI work. He designed the end-to-end workflow and implemented the PhysioNet2012 conversion into long-format data, rolling-window feature generation, weak-supervision labeling rules, optional QLoRA fine-tuning, and the evaluation tooling. He also integrated MedGemma behind a local FastAPI service and added inference safeguards to keep outputs stable under repeated requests and longer generations.

Charlene Cockcroft, BPharm Pharmacy (2026) Led the frontend and authentication work. She implemented the React web app, Flutter mobile app, and Electron desktop app, focusing on consistent rendering of the structured clinical output and predictable behavior across platforms. She also implemented Firebase Authentication across clients and coordinated secure request flows using verified ID tokens. Additionally, with her background in health care she assisted in defining the problem and providing the dataset.

Problem statement

South Africa faces a high clinical workload and unequal access patterns across the public and private health sectors. A minority of people report having medical aid coverage, while most patients rely on a resource-constrained public system [2]. In ICU practice, this translates into time pressure during rounds and handover, where clinicians must interpret many signals that evolve over time and decide whether a patient is stable, deteriorating, or simply under-monitored.

Clinicians often need a concise answer to two practical questions: *Does the patient look stable or worsening from trends?* and *What should I check next to confirm or correct that impression?* When information is missing or contradictory, forcing a confident decision can be unsafe. TriageAssist-SA therefore treats uncertainty as a valid outcome and makes missing data explicit, rather than guessing. The expected impact is reduced cognitive load, more standardized communication of trend-based reasoning, and earlier detection of deterioration signals in high-throughput settings.

This work is framed as a feasibility-oriented prototype aligned with South Africa's direction toward stronger system-wide coverage and governance, including National Health Insurance discussions and implementation planning [3, 4].

Solution

TriageAssist-SA runs MedGemma 1.5 4B-IT behind a local FastAPI service and exposes a simple API contract used by three clients: a React web app, a Flutter mobile app, and an Electron desktop app. The service enforces a strict clinician-facing output format with labeled sections so the response can be scanned quickly and rendered consistently without fragile client-side heuristics.

To keep generation reliable in a demo setting, the server includes safeguards that prevent common failure modes such as empty outputs, special-token-only outputs, and duplicated sections across continuation passes. The system prompt is tuned for ICU triage language and enforces the stability framing: outputs must label **STABLE**, **WORSENING**, or **UNCERTAIN** and then justify that label using only observable facts from the input.

Firebase Authentication is used so the demo can be shared without distributing static keys. Users sign in on the client, the client attaches a Firebase ID token to each request, and the server verifies the token before returning analysis [**?**]. This allows the same service to be accessed from multiple clients while keeping access controlled and auditable.

Technical details

Data baseline: PhysioNet2012 conversion and long-format schema

This project is kept strictly in the context of **PhysioNet 2012**. We use it because it provides ICU time-series measurements that can be transformed into consistent, reproducible trend summaries [1]. Raw records are converted into a long-format table with a stable schema:

$$\langle \texttt{stay\_id}, \texttt{ts}, \texttt{var}, \texttt{value}, \texttt{unit} \rangle$$

A mapping converts PhysioNet2012 variable names into a compact triage set, including core vitals and core labs. In this demo, arterial oxygen saturation (`SaO2`) is mapped to `SpO2` as a practical proxy for a vitals-oriented trend summary, and blood pressure fields are mapped into `SBP`/`DBP`/`MAP` where available. The resulting variable set is intentionally small because the goal is fast, robust prompting rather than exhaustive chart reconstruction.

Trend windowing: 12-hour summaries with directional features

Examples are built using rolling windows per stay. A typical configuration uses a 12-hour window with a 6-hour stride. Within each window, each variable is featurized with the last observed value, minimum, maximum, mean, and a simple slope estimate per hour. This representation is compact enough for stable prompting while retaining directional change, which is often the difference between "concerning" and "acceptable" in ICU monitoring. Windows that are too empty are skipped so the model is not trained to hallucinate missing information.

Targets: weak supervision for reproducible training and testing

To keep the pipeline reproducible on PhysioNet2012, supervision targets are generated using transparent heuristic rules rather than private clinical labels. The labeling logic flags potential deterioration signals such as hypotension or elevated lactate, low oxygenation or high respiratory rate, rising creatinine trends, or infection-like patterns such as abnormal white cell count or elevated temperature. These rules are not meant to replace clinical judgment; they provide consistent training signals that encourage sensible triage behavior and reliable formatting.

Each example includes a prompt and a strict JSON target containing keys such as `status`, `drivers`, `what_to_check_next`, `evidence`, a short `narrative`, and a disclaimer. The train/validation split is performed by `stay_id` to reduce leakage, and a configurable per-stay window cap helps prevent a few long stays from dominating training.

Modeling: MedGemma + QLoRA fine-tuning (core approach)

This project centers on fine-tuning **MedGemma 1.5 4B-IT** using **QLoRA** on the generated PhysioNet2012 prompt/target pairs. MedGemma serves as the HAI-DEF foundation model, and the training objective is to produce cautious, triage-centric outputs with stable section structure rather than overclaiming clinical certainty [6].

QLoRA keeps training feasible on limited hardware by loading the base model in **4-bit NF4** while training lightweight adapter parameters. The fine-tuning script uses causal language modeling with **loss masking** so that training occurs only on assistant tokens (the target JSON/structured output) rather than the user prompt. It includes **token_type_ids** handling required by Gemma variants, **pad-safe collation** with label padding set to `-100`, and a conservative strategy to avoid injecting LoRA layers into vision module paths to prevent unintended adaptation of non-target components.

Training was performed locally on an **NVIDIA RTX 4090**, and inference for the multi-client demo was run locally on an **NVIDIA RTX 3070 Ti**. This split reflects a practical deployment pattern: heavier fine-tuning on a high-VRAM workstation GPU, while day-to-day interactive inference remains feasible on a more common developer-grade GPU.

For stability and memory safety, the training pipeline force-disables `use_cache` when gradient checkpointing is enabled, applies a non-reentrant checkpointing configuration when supported, and provides evaluation controls (smaller evaluation sequence length or fully disabling evaluation) to reduce out-of-memory failures during fine-tuning.

System prompt and response contract

A key product decision is that the model output must be easy to scan and stable across clients. TriageAssist-SA enforces a clinician-facing layout that begins with a status label and then follows with short, structured sections. The server prompt instructs the model to use only observable facts from the provided window summary, to avoid hidden reasoning, and to explicitly list missing information when confidence is limited.

The contract is designed for speed and safety:

- **Status** is one of `STABLE`, `WORSENING`, or `UNCERTAIN`.
- **Why** contains a few short bullets that directly reflect the PhysioNet2012-derived input values or trends.
- **Immediate Actions** prioritizes ABCs, monitoring, escalation, and concrete checks.
- **Red Flags** is included only when the input supports it or when risk is clearly high.
- **Missing Data** is used when uncertainty is unavoidable and is written as actionable requests.

This contract stays consistent across presets, which keeps UI behavior predictable and reduces client-side special cases.

Inference server: robust generation and a stable API

The FastAPI service implements a single analysis endpoint and exposes health diagnostics. The `/health` endpoint reports readiness and includes adapter sanity information such as whether an adapter is active and whether LoRA parameters are present. The `/v1/analyze` endpoint accepts a preset plus a case note and returns the structured response.

Presets exist because ICU usage varies. The `quick` preset aims for short triage output, the `normal` preset targets a practical structured summary, and the `detailed` preset supports more comprehensive workup discussion while remaining cautious. Reliability safeguards directly improve product usability: the service retries when decoding yields empty text, optionally bans special-token-only outputs during retry, trims overlap when using continuation passes, and stops early when the quick-format structure is already complete.

South Africa deployment considerations: offline-first and governance-ready

Although training and evaluation are performed on PhysioNet2012, the product design is motivated by South African operating conditions. A local FastAPI deployment supports an "edge" workflow where inference can run inside a hospital network or on a clinician workstation without sending patient data to external services. This approach supports practical privacy-by-design thinking, and it aligns with South African expectations around responsible handling of personal information and health data governance [5].

The multi-client architecture is a feasibility test in itself. Web, mobile, and desktop clients stress the same contract under different network conditions and request timeouts. A stable server-side format reduces UI errors during demos and helps ensure that structured outputs can be audited and versioned as the system evolves.

Clients and authentication: one contract across three apps

All three clients call the same API and render the same labeled sections. The React web client supports fast iteration and clear visualization of the structured output. The Flutter mobile client is designed for longer request times and unstable networks, so it emphasizes timeout control, retries, and clear progress feedback. The Electron desktop client supports a local demo flow where the API runs on the same machine and the user experience remains consistent without cloud dependencies.

Firebase Authentication provides a lightweight identity layer that works across web, mobile, and desktop. Users authenticate in the client, then the client includes `Authorization: Bearer <id_token>` on requests. The server verifies the Firebase ID token before processing. This avoids embedding static keys in applications and supports controlled access when the demo is shared.

Evaluation and reporting on PhysioNet2012 splits

The evaluation script focuses on product-critical properties using the PhysioNet2012-derived validation set. It measures JSON parse rate and required-key coverage because format reliability is essential for a user interface. It also computes status agreement with the weak-supervision target, set-based similarity scores (F1 and Jaccard) for `drivers` and `what_to_check_next`, and ROUGE-L for narrative similarity. Latency and generation length are logged because an ICU-facing tool must remain usable under time constraints.

Limitations and next steps

This system is a demo and research prototype, and its labels are generated by rules rather than clinical outcomes. PhysioNet2012 provides a strong reproducible baseline, but it is not South African data, so domain shift is expected. The trend representation is intentionally compact and can miss rare events between measurements, and it does not incorporate therapy context (ventilator settings, vasopressor dosing, fluid balance), which is often necessary to interpret whether abnormal values represent deterioration or controlled treatment.

Next, we would keep the same pipeline structure but validate it against locally governed datasets once approvals, data agreements, and privacy processes are in place. That would allow calibration of uncertainty behavior and confirmation that the triage categories remain clinically sensible in South African ICU environments. We would also add stronger refusal behavior when essential fields are missing and lightweight monitoring for failure rates and latency, because operational reliability is part of clinical feasibility.

Disclaimer. This project is a demo and research prototype built on PhysioNet2012 for reproducibility. It is not medical advice, and clinician review is required.

References

[1] PhysioNet. *Computing in Cardiology Challenge 2012: Predicting Mortality of ICU Patients.* https://physionet.org/content/challenge-2012/1.0.0/ (accessed 2026-02-15).

[2] Statistics South Africa. *General Household Survey 2021 (Statistical Release P0318).* https://www.statssa.gov.za/publications/P0318/P03182021.pdf (accessed 2026-02-15).

[3] South African National Department of Health. *National Health Insurance: Key Messages*. https://www.health.gov.za/wp-content/uploads/2024/06/NHI_key_messages.pdf (accessed 2026-02-15).

[4] Republic of South Africa. *National Health Insurance Act, 2023 (Act No. 20 of 2023)*. https://lawlibrary.org.za/akn/za-act/2023-20/eng@2024-06-03 (accessed 2026-02-15).

[5] Information Regulator (South Africa). *POPIA Regulations (Final), 21 Jan 2025*. https://inforegulator.org.za/wp-content/uploads/2025/04/POPIA-2021-Regulations-FINAL-21-Jan-2025.pdf (accessed 2026-02-15).

[6] Google (Hugging Face). *MedGemma 1.5 4B-IT model card*. https://huggingface.co/google/medgemma-1.5-4b-it (accessed 2026-02-15).