PATIENT FEEDBACK ANALYSIS AND REMINDER SYSTEM

HIGH5 - CODE2CARE

TRACK 1 -GENERALITY

The healthcare system in Cameroon, and more specifically at the Douala General Hospital (DGH), faces major challenges related to patient feedback management and user communication. The lack of a centralized, multilingual platform to collect feedback and send automated reminders results in:

- A poor ability to measure patient satisfaction.
- A high rate of missed appointments and poor treatment adherence.
- Delays in identifying and resolving recurring problems (waiting times, quality of care, etc.).

In a context marked by linguistic diversity **(French, English, local languages)** and technological constraints **(low bandwidth, limited access to digital tools),** it is crucial to develop a suitable, inexpensive solution that can be easily integrated into existing systems.

This track aims to design and prototype an integrated system enabling:

- **Multilingual collection of patient feedback** via an accessible interface (text, voice, emojis).
- **Automated feedback analysis** (sentiment, recurring themes) for informed decision-making.
- **Sending personalized reminders** (SMS, voice call) for appointments and medications.
- **A real-time dashboard** for monitoring hospital performance.

The challenges of this project include:

- **Improving the quality of care**: Better listening to patients allows services to be adjusted according to their needs.
- **Resource optimization**: Reducing missed appointments and waste due to poor scheduling.
- **Inclusion and accessibility**: The solution must be usable by patients with low literacy skills or who do not speak official languages.
- **Sustainability:** The system must be easy to maintain and scalable to adapt to the hospital's future needs.

The actual constraints of this project:

- **Technological:**

- Compatibility with existing infrastructure (poor connectivity, legacy information systems).
- Need for a lightweight and functional offline solution if necessary.

- Linguistic and cultural:

- Support for French, English, and local languages (Douala, Bassa, Ewondo).
- Adaptation to varying literacy levels (use of icons, synthetic voice).

- Regulatory:

- Compliance with health data privacy and security standards.

Challenge

How can we design a digital, multilingual, and accessible system capable of improving patient engagement and operational efficiency at Douala General Hospital, while taking into account local technological and socio-cultural constraints?

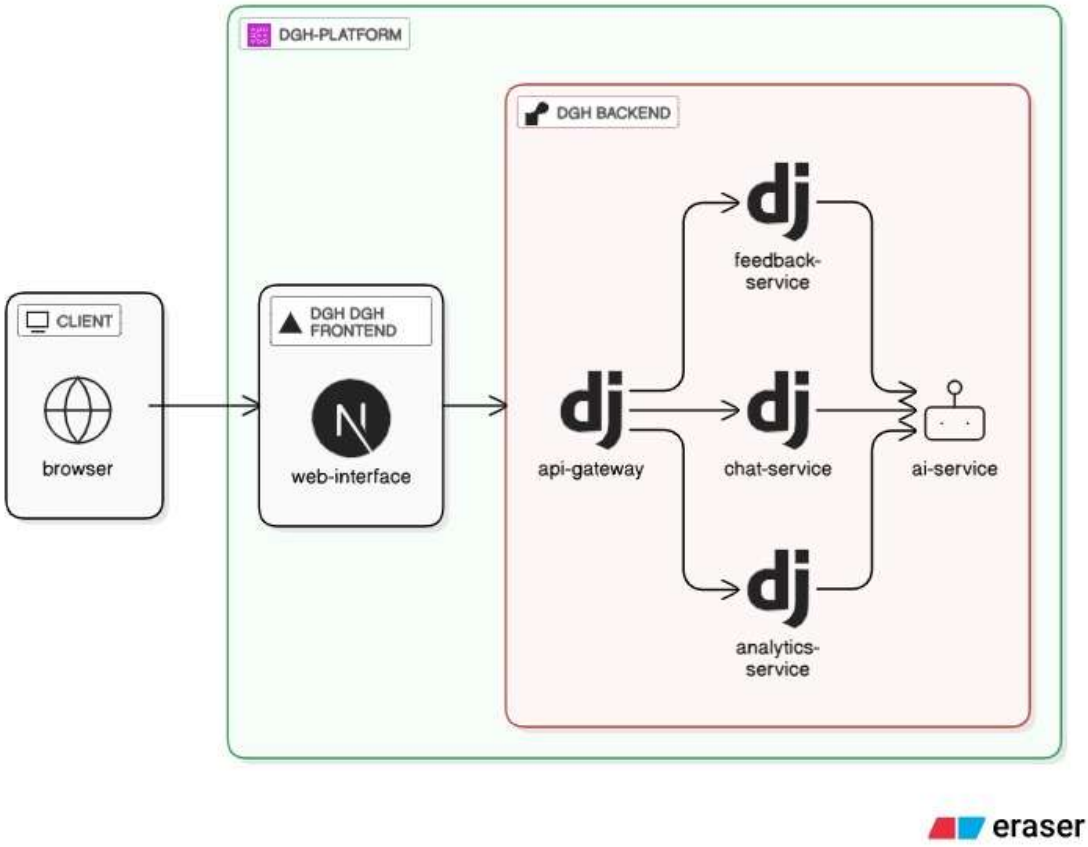
This track aims to address this challenge by developing a modular solution, combining:

- An intuitive user interface (web/mobile).
- Automated analysis tools (NLP, machine learning).
- Automated communication mechanisms (SMS, voice calls).
- A decision-making dashboard for managers.

CONCEPTION & ANALYSIS

ARCHITECTURE

Our solution's architecture adopts a **modular, microservices approach**, perfectly suited to the needs of the Douala General Hospital (DGH).



Here is a detailed analysis of the components and their relevance:

1. Client (Browser)

Purpose: User interface accessible via web browser (computer, mobile).

- **Advantage:**
  - No need to install a native application (reduces technical constraints for patients/staff).
  - Compatible with low-performance devices.

2. DGH Frontend (Web Interface)

Purpose: Dynamic user interface (React.js, Flutter Web?) for:

- Capturing feedback (text/voice/emojis).
- Viewing the administrative dashboard.
- Interacting with the chatbot (if integrated).

- **Advantage:**
  - Unified across all modules (feedback, reminders, chatbot).

3. API Gateway

Roles:

- Routing requests to microservices.
- Authentication (JWT + OAuth2 for medical staff).
- Rate limiting (preventing abuse).

- Critical endpoints:
  - /feedback → Feedback Service
  - /predict → AI Service

4. DGH Backend (Microservices)

Structure: Multiple independent services communicating via an API Gateway:

- **Feedback Service**: Manages the collection and storage of patient feedback.
- **AI Service**: NLP analysis of feedback (sentiment, themes).
- **Analytics Service**: Generates data for the dashboard.

- Advantages:
  - Scalability: Each service can be updated independently.
  - Resilience: An outage on one service does not affect the others.

Architecture Highlights

- **Modularity:** Services are decoupled, facilitating maintenance and adding features (e.g., adding an SMS reminder module).
- **Seamless Integration:** The API Gateway centralizes requests and simplifies access management (authentication, logs).
- **Adaptability to Local Constraints:** Can be deployed on a local cloud (e.g., Google Cloud with servers in Cameroon) to reduce latency.

CHOICE OF TECHNOLOGIES

- **Backend**

Criteria	**Django**	**Fast API**	**Flask**	**Spring Boot**	**Laravel**
Language Type	Python	Python	Python	Java	PHP
	Full-stack	API-first, async	Microframework	Full-stack Java EE	Full-stack
ORM integer	Django ORM	(SQL Alchemy optional)	(optional)	Hibernate	Eloquent
Auto admin UI	Puissant (admin site)	non	non	non	Basque
Security indegree	CSRF, XSS, JWT, RBAC	Basique	non	Advance (Spring Security)	Bonne (mais PHP dependent)
Performance API	Bonne	Très rapide (async)	Moyenne	Très bonne	Moyenne
Courbe d'apprentissage	Moyenne	Faible à moyenne	Faible	Complexe (Java, configs)	Faible
Écosystème santé / NLP	Compatible avec NLP Python, Celery	oui	oui	(Java moins NLP-ready)	(moins adapté à NLP)
Microservices Friendly	(via Django REST + Celery)	Oui nativement	Oui mais nécessite extensions	Excellent (native)	Moins adapté
Communauté Afrique/Global Sud	Large adoption	En croissance	Stable	Moins courant	Large en Afrique (mais PHP-centric)

- **Front-end**

Criteria	**Next.js *[Recommandé]***	**React.js (pur)**	**Vue.js**
Type	Framework complet basé sur React	Librairie UI	Framework complet
Rendu Serveur (SSR)	Intégré nativement (SEO, rapidité)	Besoin de setup (Next, Remix…)	Possible via Nuxt.js
Rendu statique (SSG)	Oui, sans config	Non natif	Avec Nuxt.js
Routing automatique	Basé sur le système de fichiers	Nécessite React Router	Avec Vue Router
SEO / Accessibilité	Excellente pour hôpitaux	Faible (CSR only)	Moyenne (Nuxt recommandé)
Intégration API Django (REST)	Parfait via fetch / SSR API	OK	OK
Multilingue & i18n	next-i18next, facile à intégrer	Moins structuré	Bon via vue-i18n
Performance (vieux téléphones)	SSR/SSG très rapide	CSR = + lent au chargement	Bonne mais dépend des bundles
Déploiement sur Vercel	Automatique, natif	Besoin de setup	Possible mais pas optimisé
Code splitting automatique	Oui, sans config	Non	Oui
Auth & sécurité (RBAC)	Middleware intégrable	Besoin de setup JWT	Possible
Courbe d'apprentissage	Moyenne (React + conventions)	Moyen	Facile pour débutants
Communauté Afrique / Django / Python	Forte adoption	Présente	Moins courante dans écosystèmes IA

- **Sentiment Analysis**

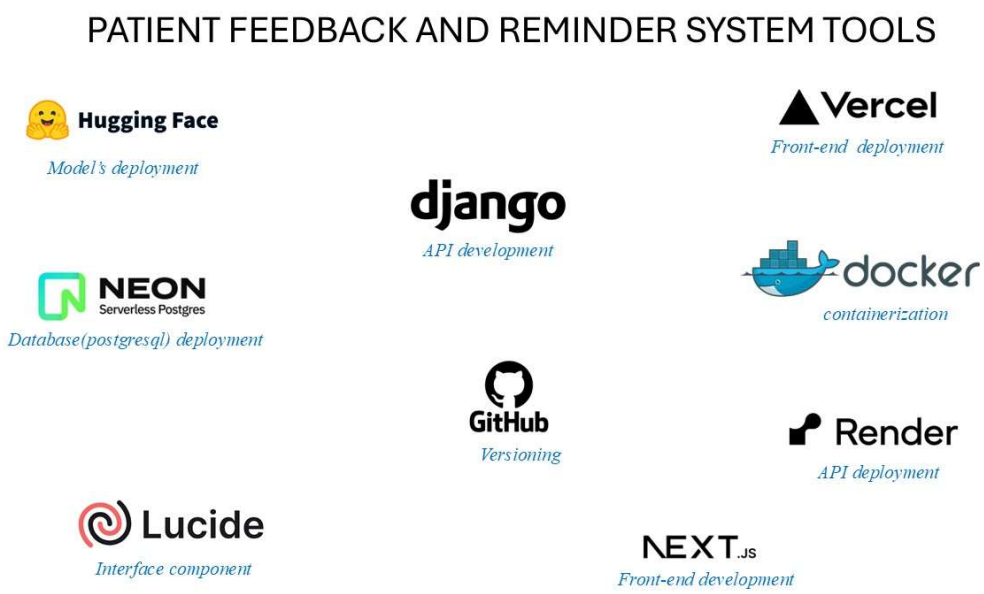
Criteria	RoBERTa Base	BERT	LSTM	Logistic Regression
Performance	9.5/10	9/10	7/10	6/10
	- SOTA on many NLP tasks- Excellent on complex classification- Handles context and nuances	- Exceptional performance- NLP benchmark reference- Good on diverse tasks	- Good performance on sequences- Effective on temporal data- Limited by architecture	- Decent performance- Good for linear data- Reliable baseline
Speed	4/10	3/10	7/10	10/10
	- Training: hours/days- Parallelization possible	-Training: hours/days- Resource intensive	- Training: minutes/hours- Sequential but fast	- Training: seconds/minutes- Very lightweight
Precision	9.5/10	9/10	7.5/10	6.5/10
Adaptability	8/10	8/10	6/10	4/10
	- Efficient fine-tuning- Excellent transfer learning- Multi-domain adaptation	- Standard fine-tuning- Versatile- Fast adaptation	- Architectural adaptation- Hyperparameter tuning- Limited to sequences	- Limited adaptation- New features required- Strict assumptions
Deployment Ease	3/10	2/10	7/10	10/10
	- Resources: GPU (8-16GB)- Complex frameworks- Optimization required	- Resources: GPU (8-16GB)- Heavy infrastructure- Maximum complexity	- Resources: Light CPU/GPU- Standard frameworks- Moderate deployment	- Resources: Minimal CPU- Simple libraries- Immediate deployment

- Theme Extraction

Méthode	Qualité thématique	Besoin de training	Multilingue	Déploiement facile	Temps de calcul	Avantages principaux	Inconvénients
LLM (LLaMA, GPT, etc.)	Très élevée (compréhension profonde)	Aucun (in-context)	Oui (selon modèle)	Moyen à difficile (nécessite API ou Ollama)	Moyen à lent	Compréhension du contexte, personnalisation aisée via prompt	Coût mémoire/temps, difficile à auditer
BERTopic	Élevée (embedding + clustering)	Oui (sur corpus)	Oui	Moyen (besoin d'UMAP/HDBSCAN)	Moyen	Bon équilibre entre qualité thématique et interprétabilité	Moins performant sur petits jeux de données
LDA	Moyenne	Oui	Non (majoritairement anglophone)	Facile	Rapide	Interprétation facile, classique	Moins précis, sensible au prétraitement
NMF	Moyenne à bonne	Oui	Non	Facile	Rapide	Moins bruité que LDA, mieux sur petits corpus	Thèmes parfois moins cohérents
Top2Vec	Bonne	Non (embeddings)	Oui	Moyen (poids modèles)	Moyen	Aucun prétraitement, détecte automatiquement les sujets	Pas toujours stable sur petits corpus
KeyBERT	Bonne (keywords → thèmes)	Non (zero-shot)	Oui	Très facile	Très rapide	Très simple, efficace sur petits textes	Pas de regroupement automatique (juste mots-clés)

Nb: the green cell represents our choice

TOOLS


PATIENT FEEDBACK AND REMINDER SYSTEM TOOLS
Hugging Face
Model's deployment
Vercel
Front-end deployment
django
API development
NEON
Serverless Postgres
Database(postgresql) deployment
docker
containerization
GitHub
Versioning
Render
API deployment
Lucide
Interface component
NEXT.JS
Front-end development