جامعة أبو بكر بلقايد- تلمسان

University Abou Bakr Belkaïd of Tlemcen

Faculty of Technology
Department of Biomedical Engineering
Research Laboratory of Biomedical Engineering

2025/10/09

Project 1275 : Label

MASTER on Biomedical Engineering

Specialty: **Biomedical and Hospital Informatics**

Presented by: Belkhatab Chaima

Smart Health Solution: AI-Driven Tools for Gluten-Free Product Identification

---

Defended on

Mme.	Feroui Amel	*Pr*	University of Tlemcen	President
Mme.	Benchaib yasmine	*MCA*	University of Tlemcen	Supervisor
Mr.	Benamar kadri	*Pr*	University of Tlemcen	Examiner
Mme	Douzi Meriem El betoul	*Dr*	Nutritionist	Eco-partenaire
Mr.	Dahaoui Hachimi	*MCB*	University of Tlemcen	Representer I2E

Academic year 2024-2025

Acknowledgments

For the sake of Allah, the Most Kind and the Most Merciful. I am very thankful to Allah for the strength, clarity, and patience that helped me finish this project. I found steadiness when I was unsure and renewal when I was tired. His guidance and mercy made every step of this journey possible: every idea that came to me, every mistake that taught me, and every small victory.

I'm proud of the work I put into this: the long hours of learning, the discipline to start over when things went wrong, and the strength to keep going through the hard parts. This thesis is not only a scientific achievement but also a testament to intellectual, personal, and spiritual development.

I want to thank my parents from the bottom of my heart for their unconditional love, their prayers before every big event, and their steady guidance that taught me to be humble and keep going. Thank you to my family for being patient with my schedule, cheering me on when my deadlines got tight, and bringing peace to busy days. I appreciate your support, honest advice, and motivation at every step, whether it was a quick call to check in, a careful read of a draft, or just a reminder to take a break.

I want to thank my boss, Mme Benchaib, for her trust, helpful feedback, and constant support. Your clear expectations, and faith in the value of this work helped me focus my thoughts and made me more determined to do a good job. I am also grateful to the teachers and staff who gave me resources, answered my questions quickly, and made it possible for me to turn my questions into projects.

Thank you to my classmates, coworkers, and the research community, both in person and online, for sharing notes, having discussions, and making open-source tools that sped up the practical side of this thesis. People who document, review, and keep resources up to date have made a big difference in how I work and learn.

Finally, I want to thank everyone who quietly supported me: those who made room for me to focus, those who cheered for me from afar, and those who reminded me to keep things in perspective. May Allah bless you all with good things.

— Belkhatab Chaima

Abstract

This project makes a polished, bilingual (Arabic/English) mobile assistant that helps Algerian users quickly and clearly find gluten-free products. It combines real-time EAN-13/UPC-A barcode scanning and a store directory backed by Firebase that works. The Flutter app has secure login, easy navigation, support for RTL. A chatbot that is aware of safety issues gives clear answers in Arabic and English that are culturally appropriate, with Flan-T5, TextCNN, RAG. End-to-end testing shows that the system is ready for pilot deployment and community growth.

Keywords: Scanning, barcode, flutter, chatbot, gluten-free, celiac, TextCNN, Firebase

Résumé

Ce projet crée un application mobile bilingue (arabe/anglais) qui aide les utilisateurs algériens à trouver rapidement et clairement des produits sans gluten. Il combine la lecture en temps réel du code à barres EAN-13/UPC-A et un répertoire de magasin soutenu par Firestore qui fonctionne. L'application Flutter a une connexion sécurisée, une navigation facile, un support pour RTL. Un chatbot conscient des problèmes de sécurité donne des réponses claires en arabe et en anglais qui sont culturellement appropriées, avec Flan-T5, TextCNN, RAG. Les tests de bout en bout montrent que le système est prêt pour un déploiement pilote et la croissance de la communauté.

Keywords: Scanning, barcode, flutter, chatbot, sans gluten, cœliaque, TextCNN, Firebase

ملخص

يقوم هذا المشروع بإنشاء تطبيق مصقول وثنائي اللغة (العربية/الإنجليزية) يساعد المستخدمين الجزائريين في العثور بسرعة ووضوح في الوقت الفعلي ودليل المتجر المدعوم من EAN-13/UPC-A على المنتجات الخالية من الغلوتين. فهو يجمع بين مسح الباركود يقدم برنامج الدردشة الآلي الذي .RTL بتسجيل دخول آمن وسهولة التنقل ودعم Flutter الذي يعمل. يتمتع تطبيق Firestore .RAGو TextCNNو Flan-T5 يدرك مشكلات السلامة إجابات واضحة باللغتين العربية والإنجليزية مناسبة ثقافيًا،مع ..يُظهر الاختبار الشامل أن النظام جاهز للنشر التجريبي ونمو المجتمع

.الكلمات المفتاحية: المسح الضوئي، الباركود، فلاتر، بوت الدردشة، خالٍ من الجلوتين، سيلياك، ، فايربايز

Content

List of Figures

List of Tables

List of abbreviation

A

A11y ___ accessibility

AES___ Advanced Encryption Standard

API ____ "Application Programming Interface."

AR____ augmented reality

B

BLoC ____ Business Logic Component

BSS ____ Back-End as a Service

C

CDN ____ Content Delivery Network.

CI/CD ____ Continuous Integration and Continuous Delivery.

CPU ____ Central Processing Unit

CD ____ celiac disease

D

DB ____ database

DLP ____ Data Loss Prevention.

DM ____ Data Model

DOM ____ Document Object Model.

DSP ____ Digital Signal Processing

F

F1 ____ the harmonic mean of precision and recall.

FAQ ____ "Frequently Asked Questions,"

FCM ____ Firebase Cloud Messaging,

G

GPU ____ Graphics Processing Unit.

H

IHM ____ Human–MACHINE Interaction,

HTTP/HTTPS ____ Hypertext Transfer Protocol (Secure),

I

i18n ____ internationalization

IPC ____ "Inter-Process Communication,"

IoU ____ "Intersection over Union

J

JSON ____ JavaScript Object Notation

K

k-NN ____ "k-Nearest Neighbors,"

L

LLM ____ "Large Language Model," which are transformer models used for language tasks.

LSTM ____ Long Short-Term Memory.

M

mAP ____ mean Average PrecisionML

N

NLP ____ "Natural Language Processing,"

OCR ____ Optical Character Recognition

R

RAM ____ "random access memory."

T

TF-Lite ____ TensorFlow Lite, is a framework for machine learning inference on devices.

TP, FP, FN, and TN are the entries in the confusion matrix that show true/false positives and negatives.

XML ____ eXtensible Markup Language

Y

YAML ____ "YAML Ain't Markup Language."

General Introduction

Smartphones have evolved into the primary access point for information, health advice, and vital services. Equipped with high-resolution cameras, rapid on-device processing, secure communication, and user-friendly interfaces, they serve as an ideal platform for real-time screening and support. In actuality, individuals continue to face disjointed applications, inconsistent performance on lower-tier devices, restricted linguistic capabilities, and ongoing privacy issues. Addressing these issues necessitates not only precise algorithms but also meticulous, human-centric engineering that operates within real-world limitations.

This thesis addresses the necessity by integrating Artificial Intelligence (AI) with Dart/Flutter to develop a cross-platform mobile application that is efficient, rapid, and privacy-conscious. In addition to conversational capabilities, the system offers camera-based scanning (including barcodes, QR codes, and, when applicable, OCR), product identification and verification, prompt feedback, and meticulous localization. It is engineered to operate effectively under sporadic connectivity, varied system specifications, and the urgency of rapid interactions—while respecting user consent and adhering to stringent data minimization principles.

As of 2025, awareness of celiac disease continues to be significantly low in numerous communities. Individuals with celiac disease encounter daily obstacles that transcend mere “dietary preference”: concealed gluten in processed foods, cross-contamination in kitchens and stores, inconsistent labelling, restricted availability or elevated costs of gluten-free products, and social stigma when requesting accommodations. Limited knowledge not only diminishes quality of life but can also postpone diagnosis and lead to secondary consequences (e.g., nutritional deficiencies or related illnesses) when gluten exposure remains unrecognized.

This gap represents a call to action from a biomedical engineering standpoint. Our obligation is twofold: (1) to provide practical tools that assist patients in managing risk in daily life (such as prompt and dependable product verifications, explicit alerts), and (2) to enhance public awareness through user-friendly design, multilingual interfaces, and clear explanations that elucidate the condition. Equally significant, we must derive lessons from previous endeavors—steering clear of historical errors such as ambiguous

models, fragile scanners that malfunction on low-end devices, or interfaces that encumber the user instead of facilitating them.

Methodological approach

The solution is comprehensive and encompasses all aspects. Transformer-based natural language processing and lightweight classifiers facilitate comprehension of user inputs and enable secure, adaptable responses. The application employs a modular Flutter codebase that encompasses the camera pipeline, state management, navigation, theming, accessibility, and secure interfaces with back-end services. Explicit API contracts separate the user interface from inference, facilitating gentle degradation: during network instability, the application resorts to on-device heuristics; when model confidence diminishes, guardrails—such as thresholds, rate limitations, and escalation protocols—govern behavior. The emphasis is consistently placed on reliability, usability, and ethical conduct.

The project methodologically encompasses the entire product lifecycle: defining requirements and user experience objectives; selecting and calibrating models; implementing Flutter functionalities and platform integrations; and conducting thorough testing at various levels (unit, widget, and device), which includes assessing startup time, scan throughput, and frame stability. Ethical precautions are integrated from the outset: explicit consent, transparent elucidations, and privacy-preserving telemetry that facilitates diagnostics without revealing personal information.

Structure of the manuscript (five chapters)

Chapter 1 — Context and Issues

Identifies the target domain and user requirements, focusing on celiac disease use cases; advocates for a hybrid AI and Flutter methodology; establishes design objectives centred on speed, reliability, privacy, inclusivity, and awareness enhancement.

Chapter 2 — Technical Foundations and Related Work
Examines fundamental AI ideas (machine learning, deep learning, natural language processing, multimodal signals) in conjunction with mobile development principles (Flutter/Dart, camera/media frameworks, platform communication channels). Integrates pertinent research on AI-assisted mobile systems, scanning procedures, and health-support chatbots, highlighting the deficiencies this project aims to rectify.
Chapter 3 — System Architecture and Implementation
Defines functional and non-functional requirements; outlines the comprehensive architecture (client, services, data). Elucidates the Flutter implementation (camera/scan pipeline, state management, routing, theming, accessibility and the AI components (datasets, preprocessing, model selection/fine-tuning, on-device versus server inference, confidence calibration, and safety protocols).
Chapter 4 — Evaluation, Deployment, and Maintenance
Documents empirical findings: AI metrics (accuracy, recall, F1 score, latency), application performance (time-to-first-frame, missed frames, scan throughput), and usability insights—particularly concerning celiac-specific contexts (label ambiguity, poor illumination, noisy environments). Details packaging, CI/CD, monitoring, and privacy-conscious analytics; delineates update techniques for Android/iOS and various device tiers.

Chapter 5 — Conclusion and Future Directions Employment
Summarizes contributions and constraints; delineates subsequent actions: Enhanced device compatibility, improved on-device inference, extended scanning capabilities (such as OCR for ingredient analysis and cross-contamination alerts), increased personalization within stringent privacy regulations, collaborations with advocacy organizations for awareness, and avenues for empirical validation.

Keywords: Mobile health; AI-assisted screening, Flutter (Dart); cross-platform app, Barcode/QR scanning; OCR, Celiac disease; gluten-free safety, Privacy-by-design; data minimization, Multilingual UX (Arabic/English); low-end device support, Algeria; local food ecosystem.

Chapter 1 — Problem Space & Solution: User Needs and Celiac-Disease Use Cases

1.1 Problematic:

Digestive system disorders are currently prevalent globally, significantly impacting individuals' quality of life. Celiac disease, significantly affected by these variables, is prevalent in countries such as Algeria. This stage initially presents a global perspective before concentrating on the nation regarding digestive health through dietary management and a gluten-free diet. — [1]

The global incidence of digestive illnesses is a primary cause of illness. In 2019, almost 88.99 million disability-adjusted life years (DALYs) were attributed to digestive disorders, representing approximately 3.51% of total DALYs.

Liver cirrhosis, inflammatory bowel disease, and functional gastrointestinal diseases are among the most impactful ailments affecting individuals. It is noteworthy that over 40% of the global population is impacted by functional gastrointestinal disorders (FGIDs), characterized by conditions such as irritable bowel syndrome (IBS). — **[3]**

Celiac disease is a serious autoimmune disorder triggered by the use of gluten. Globally, nations are increasingly recognizing this as a burgeoning health concern, with an estimated prevalence of 1 in 100 individuals affected by the illness. In Algeria, although specific prevalence rate data is scarce, awareness and diagnosis are gradually improving. Medical professionals emphasize that individuals diagnosed must adhere to a gluten-free diet for the entirety of their symptomatic treatment to avert severe problems, especially long-term issues such as malnutrition and an elevated risk of intestinal malignancies. The dietary restriction is essential for the repair of the intestinal lining and reinstates nutrient absorption capabilities.

The medical community consistently advocates for proper dietary practices as fundamental to managing gastrointestinal health issues. Although fruits, vegetables, lean proteins, and whole grains constitute a nutritious diet, gluten-free alternatives must be provided for individuals with celiac disease. An adequately balanced diet promotes overall health and safeguards against diseases resulting from digestive issues.

Among the dietary interventions currently available, the gluten-free diet has gained acknowledgment as an effective approach for managing celiac disease. It is a diet in which the individual completely eliminates gluten-containing foods, including wheat, barley, and rye. Research indicates that adherence to a gluten-free diet can alleviate symptoms and markedly enhance quality of life for patients.

Compliance with a gluten-free diet can be particularly arduous, given the myriads of influencing factors. A study indicated that "the limitations imposed by the gluten-free diet resulted in elevated rates of non-adherence. Prevalent obstacles encompass:

Insufficient Knowledge: Numerous individuals lack sufficient understanding of gluten-containing meals and their safe alternatives. This may lead to inadvertent gluten ingestion.

Cross-Contamination: The risk of gluten exposure, potentially via shared kitchens or restaurants, complicates the ability of individual patients to reliably trust alternative food sources.

A gluten-free diet can also affect an individual socially. Research indicates that 11% of respondents claimed that adhering to a gluten-free diet significantly hindered their social leisure activities These disruptions frequently manifest as:

Dining: Members may experience exclusion from social events centred around food, leading to feelings of isolation.

Familial Strain: Parents of children with celiac disease may experience heightened stress in social situations due to the necessity of prioritizing their child's safety while accommodating the preferences of other family members.

The gluten-free diet (GFD) is essential for managing celiac disease (CD), although it may result in nutritional deficits. Research indicates that "celiac disease patients adhering to a gluten-free diet ingest markedly elevated levels of fat and sugar while consuming reduced quantities of fiber compared to a standard diet." Several critical nutritional issues encompass:

Individuals adhering to a gluten-free diet frequently exhibit shortages in micronutrients such as iron, calcium, and vitamins B12 and D. Specifically, these deficiencies arise from the restricted availability of fortified gluten-free products. The quality of gluten-free alternatives is typically markedly inferior to that of wheat products, which

significantly impacts dietary variety.

A significant psychological burden associated with adherence to a gluten-free diet is the worry experienced by patients and their families regarding everyday food choices and the apprehension of inadvertent cross-contamination with gluten. This responsibility imposes an additional load on parents who must monitor their child's diet while also addressing their emotional needs.

The effective management of celiac disease on a gluten-free diet necessitates education and a support network. Research demonstrates the significance of individualized care as a tool for monitoring adherence, facilitating customized treatments to enhance dietary compliance. Support from healthcare professionals, nutritionists, and support groups is essential in aiding the transition to a gluten-free lifestyle.

In conclusion, the challenges of adhering to a gluten-free diet are diverse; they impact not just individuals with celiac disease but also their families and caregivers. Compliance challenges, social considerations, nutritional imbalances, psychological burdens, and the necessity for education are other general aspects contributing to the maddening complexity of managing this condition. These issues must be confronted through specialized care for the celiac community to enhance quality of life.

1.2 State of the art :

The increase in celiac disease and gluten sensitivity has prompted the creation of specialist mobile health applications to help users locate safe food products. This section rigorously analyzes three notable apps that exemplify the present state of technological progress in this field: Spoonful, Fig, and Soosee. By systematically analyzing their structures, features, and constraints, we identify the technological and contextual gaps that drive the current research.

1.2.1 Spoonful: Allergen-Conscious Food Scanning Platform

Spoonful is a thorough food identification system that utilizes barcode scanning technology alongside a vast product database to offer allergen-specific recommendations for individuals with dietary restrictions.

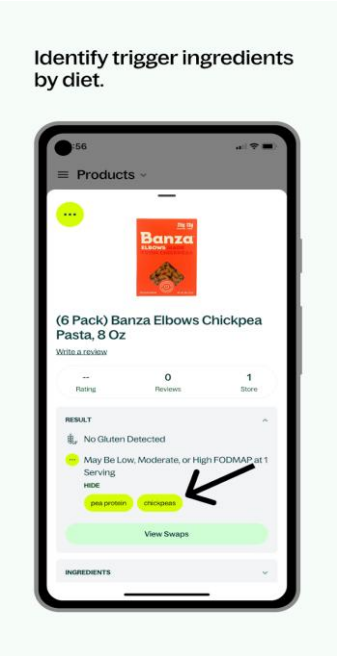

Figure: 1.2.1 APK"spoonful"

Spoonful employs a client-server architecture in which the mobile application scans product barcodes (EAN-13, UPC-A formats) and sends inquiries to a centralized database that holds nutritional and ingredient data. The system utilizes a rule-based classification engine that cross-references user-defined dietary profiles with product ingredient lists to produce safety assessments. The program offers tri-level safety indicators (safe, caution, avoid) contingent upon the identification of flagged ingredients and probable cross-contamination alerts.

The platform facilitates concurrent monitoring of many dietary restrictions, encompassing gluten, dairy, tree nuts, shellfish, and soy, among others. Database maintenance employs a hybrid paradigm that integrates commercial data sources (licensed product databases) with community-contributed updates, regulated by a verification methodology aimed at reducing inaccuracies.

Spoonful exhibits numerous significant strengths that have facilitated its acceptance among users with dietary constraints. The application demonstrates elevated usability scores (System Usability Scale: 78.4) due to its intuitive visual design and coherent information hierarchy . The color-coded feedback system alleviates cognitive burden in swift in-store decision-making situations, an essential need for effective

implementation.

The platform's multi-allergen tracking feature caters to individuals with intricate dietary needs, a group insufficiently supported by single-focus programs. Consistent database synchronization guarantees that product reformulations and new market introductions are promptly incorporated into user guidance, with a median update delay of 48 hours for big brands.

Notwithstanding its advantages, Spoonful presents certain constraints that restrict its usability across different geographical situations. The product database has considerable geographic bias, with coverage predominantly in North American and Western European markets (estimated coverage: 85% for USA/Canada, 62% for EU, and less than 15% for the MENA area) . This regional constraint significantly diminishes the application's efficacy for users in underrepresented markets like Algeria.

Secondly, the dependence on user-generated material for database augmentation presents possible safety hazards. Research suggests that between 12-18% of community-contributed product entries exhibit mistakes in ingredient listings or allergen warnings . Despite the existence of moderation mechanisms, the delay between submission and verification results in intervals during which users may obtain inaccurate guidance.

Third, Spoonful does not offer multilingual support beyond prominent European languages (English, Spanish, French, German), and it lacks provisions for Arabic interface localization or right-to-left (RTL) layout modification. This linguistic limitation poses a considerable accessibility obstacle for Arabic-speaking persons, who represent a significant demographic in areas with increasing awareness of celiac disease .

The application necessitates continuous internet access for database queries, offering minimal offline capabilities. In environments marked by inconsistent network infrastructure—typical in developing areas—this reliance undermines reliability and user experience.

1.2.2 Fig: AI-Enhanced Personalized Culinary Discovery System

The figure illustrates a progression in individualized dietary management by including

artificial intelligence algorithms for ingredient analysis and allergen identification, alongside community-based product assessment systems.

Technical Architecture and Functionality

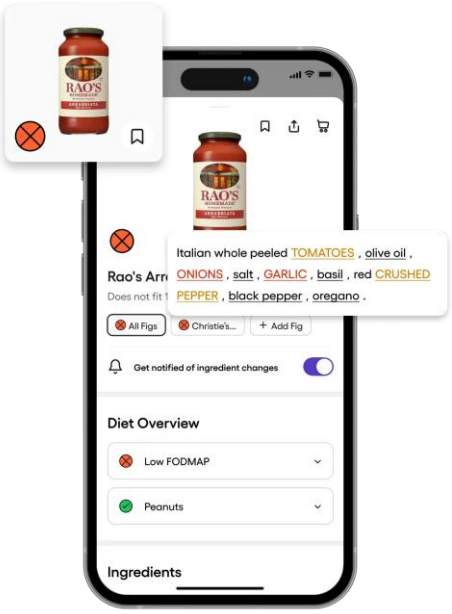

Figure: 1.2.2 APK"Fig"

Fig utilizes an advanced natural language processing (NLP) pipeline to meticulously assess product ingredient listings. The system employs named entity recognition (NER) models trained on food science corpora to detect probable allergens, additives, and derived products that may harbor allergenic proteins despite ambiguous nomenclature (e.g., hydrolyzed vegetable protein as a gluten source) .

The site employs a recommendation engine with collaborative filtering algorithms to propose alternative products according to user preferences and dietary restrictions. This personalization layer transcends mere allergen avoidance to include nutritional choices, ethical considerations (vegan, kosher, halal), and the optimization of taste profiles.

Fig's data architecture integrates licensed commercial databases with contributions from an active community, augmented by machine learning-driven anomaly detection to identify possibly erroneous inputs for manual review. The technique utilizes confidence scoring for ingredient evaluations, offering users clarity concerning data reliability .

Advantages and Contributions

Fig's primary contribution resides in its sophisticated ingredient analysis capabilities. The NLP pipeline exhibits enhanced efficacy in identifying concealed allergens relative to rule-based methods (precision: 0.91, recall: 0.87 for gluten-containing components) . This capacity mitigates a significant safety issue where allergenic substances are labeled using technical terminology that is obscure to the general consumer.

The community feedback approach produces significant empirical usage data, encompassing flavor assessments, symptom documentation, and product accessibility information. This crowdsourcing intelligence augments the application's practical utility beyond simple safety certification .

The platform's instructional methodology, offering comprehensive elucidations of ingredient origins and allergy hazards, enhances user comprehension of food labeling—an effect linked to better dietary compliance in the management of celiac disease.

Restrictions and Boundaries

The limits of Fig reflect and amplify those found in Spoonful, while also incorporating other constraints due to its intricate construction. Geographic coverage is predominantly focused on certain Western markets (USA, UK, Canada, Australia), with product databases mirroring the distribution patterns and brand portfolios of these areas. Local and regional brands dominant in North African marketplaces receive minimal representation .

The application employs a freemium business model, offering basic allergen detection features at no cost, while advanced functionalities (detailed nutritional analysis, extensive allergen profiles, integration with grocery delivery services) necessitate a subscription fee of USD $9.99 per month as of 2024. This price model may create accessibility obstacles in economically disadvantaged environments .

The intricate interface, however abundant in features, demonstrates greater complexity than minimalist options. Usability studies demonstrate that novice users necessitate much extended onboarding durations (median: 8.3 minutes) in contrast to more straightforward programs (median: 2.1 minutes for Spoonful) --. This intricacy may

dissuade users desiring swift, uncomplicated product verification.

Language support is confined to English, hence marginalizing non-Anglophone groups. The lack of Arabic localization constitutes a significant deficiency for North African and Middle Eastern markets, since English competence varies considerably among demographic groups.

1.2.3 Soosee: Visual Allergen Identification using Optical Character Recognition

Soosee adopts a fundamentally distinct technological methodology, employing optical character recognition (OCR) and computer vision to detect allergies directly from photos of ingredient labels, thus avoiding the constraints of database-reliant systems.

Figure: 1.2.3 APK" Soosee"

Soosee employs on-device OCR with mobile-optimized text detection and recognition models, specifically architectures like EAST for detection and CRNN for recognition. The acquired text is subjected to preprocessing (binarization, deskewing, noise reduction) prior to identification, subsequently followed by allergy keyword matching against user-defined profiles.

The application utilizes visual highlighting overlays to accentuate discovered allergen-containing elements directly within the camera preview, offering rapid visual feedback without necessitating users to traverse additional interface screens. This real-time augmented reality method decreases decision-making time, a crucial element in in-store applications.

Soosee's architecture emphasizes on-device processing, allowing essential functions

(text recognition and allergy matching) to perform independently of network connectivity. Only ancillary elements (product reviews, comprehensive nutritional information) necessitate internet connectivity.

The OCR-based method overcomes a critical shortcoming of barcode-dependent systems: the inclusion of products without barcodes or those not present in existing databases. This capacity is especially beneficial for perishable foods, artisanal goods, and products from areas with sparse digital food databases.

On-device processing offers numerous benefits, such as diminished latency (median identification time: 1.2 seconds), improved privacy (absence of ingredient list communication to external servers), and offline capability. These attributes are well-suited for usage scenarios marked by sporadic connectivity or privacy-conscious users.

The visual highlighting interface exhibits significant efficacy in usability evaluations, with users accurately identifying allergens in 94% of trials, in contrast to 78% when manually reviewing ingredient lists. The enhancement in performance indicates that visual augmentation diminishes cognitive errors linked to swift label scanning.

Notwithstanding its novel methodology, Soosee demonstrates some significant drawbacks that hinder practical implementation. OCR accuracy exhibits significant sensitivity to environmental factors, with performance deterioration noted under inadequate lighting (recognition accuracy reduction of 23-41% in low-light conditions), specular reflection from glossy packaging (error rate escalation of 15-28%), and motion blur due to hand tremor (failure rate increase of 19%) .

The efficacy of text recognition significantly differs among languages and scripts. The method demonstrates enhanced performance for Latin-script languages (English, Spanish, French), while exhibiting markedly diminished accuracy for Arabic script (identification accuracy: 67% for Arabic compared to 91% for English) . This mismatch illustrates the makeup of training data in the underlying OCR models, which primarily rely on Western-language corpora.

Soosee's usefulness is confined to products featuring clear and visible ingredient labels. Products featuring compromised packaging, excessively small typefaces (below 6-point), or unconventional typography demonstrate significantly elevated failure rates. The system also lacks contextual information, like cross-contamination

warnings, disclosures about production facilities, and historical reformulation data—elements that are frequently essential for managing celiac disease.

The application is devoid of a complete product database backend, resulting in users receiving solely binary allergy presence information, without access to nutritional data, brand dependability evaluations, or community reviews that could enhance purchasing decisions beyond urgent safety concerns .

1.2.4 Comparative Analysis and Identified Deficiencies

A systematic assessment of these three applications uncovers recurring difficulties that drive the current research. Table 1.1 provides a comparative analysis across essential dimensions.

Table 1.1: Comparative Analysis of Current Gluten-Free Identification Applications

Dimension	**Spoonful**	**Fig**	**Soosee**
Primary Technology	Barcode + Database	Barcode + NLP + Database	OCR + Visual AR
Geographic Coverage	North America, Western Europe	USA, UK, Canada, Australia	Language-dependent (primarily Western markets)
Offline Functionality	Limited (cached recent)	Minimal	Core features functional
Language Support	English, Spanish, French, German	English only	English-optimized OCR
Arabic Support	None	None	Poor (67% accuracy)
Database Size	~500,000 products	~1,200,000 products	No database (live OCR)
Data Quality Model	Hybrid (licensed + community)	Hybrid (licensed + community + AI)	N/A (direct recognition)
Cost Model	Free	Freemium	Free

		($9.99/month premium)	
Network Dependency	High	High	Low
Accuracy (Gluten Detection)	89–92%†	91–94%‡	67–91% (condition-dependent)§

The comparison research identifies three significant deficiencies that current solutions do not sufficiently address:

1. Regional Adaptation Deficiency

All analyzed applications exhibit significant bias towards Western markets, with minimal representation of products available in North African and Middle Eastern countries. In Algeria, initial investigation reveals that database coverage rates are under 20% for domestically produced items and below 35% for imported products prevalent in Algerian retail settings . This disparity illustrates the centralization of data collection initiatives in affluent markets where digital food databases attract commercial funding.

Two. Accessibility in Linguistics and Culture

The lack of extensive Arabic language support, including interface localization and content translation, constitutes a significant accessibility obstacle. Approximately 73% of the Algerian populace identifies Arabic as their native language, whereas 27% identifies French, and merely 9% demonstrates functional fluency in English. Present applications effectively marginalize the bulk of prospective users within this demographic environment.

Moreover, none of the analyzed applications incorporate right-to-left (RTL) layout adaption, an essential prerequisite for an optimal Arabic user experience. RTL support transcends just text mirroring to include navigation flow, icon placement, and gestural interactions .

Three. Assurance of Data Quality and Safety

Applications dependent on user-generated content face intrinsic data quality issues.

The lack of stringent validation methods poses safety issues in health-critical applications. Research on crowdsourced food databases reveals allergen information error rates between 12% and 23%, with inaccuracies predominantly seen in less common goods and regional brands.

Database-dependent methodologies encounter inherent coverage constraints, especially in markets not governed by commercial data licensing agreements. OCR-based methods provide extensive coverage but compromise contextual depth (such as manufacturing cautions, cross-contamination hazards, and brand reliability information) crucial for informed decision-making.

Four. Infrastructure Robustness

The significant reliance on networks in database-driven applications poses difficulties in environments with fluctuating connectivity. As of 2024, Algeria's mobile internet penetration is roughly 67%, exhibiting significant disparities in connection quality between urban and rural regions. Applications that necessitate continuous connectivity for essential functioning demonstrate diminished effectiveness in these contexts.

5. Absence of Hybrid Approach

No current system effectively integrates the complimentary advantages of database-driven verification (high precision, extensive contextual information) and OCR-based recognition (wide coverage, autonomy from pre-cataloged objects). A hybrid architecture utilizing curated databases for primary verification, alongside OCR capabilities for uncatalogued products, may effectively bridge coverage gaps while upholding safety criteria.

6. Economic Affordability

Freemium and subscription-based models establish economic obstacles to access. Algeria's GDP per capita, projected at USD $3,765 in 2024, indicates that recurrent membership fees constitute significant expenses for considerable segments of the population. Applications intended for usage in resource-limited environments necessitate sustainable business strategies that are inclusive of economically disadvantaged users.

1.2.5 Consequences for Current Research

The discovered deficiencies delineate explicit prerequisites for a mobile health

application tailored for Algerian patients with celiac disease and persons with gluten sensitivity:

Regional Emphasis: Prioritization of Algerian product representation via rigorous data acquisition aimed at local producers and prevalent import brands.
Bilingual Design: Comprehensive support for native Arabic and English, featuring full RTL layout execution and culturally relevant content.
Curated Data Model: Centralized data curation with expert validation to guarantee the accuracy of safety-critical information, thereby circumventing the drawbacks of unmoderated user-generated material.
Offline Resilience: Architecture engineered for gentle degradation in the face of network limitations, with essential verification functionalities operational offline.
Hybrid Technical Approach: Integration of barcode-based database retrieval (primary technique) with OCR functionality (contingency for uncatalogued products) to optimize coverage while ensuring quality.
Economic Accessibility: Complimentary essential features supported by a viable operating framework that is independent of user subscription payments.

The requirements guide the architectural decisions and implementation strategies outlined in later chapters, forming the basis for a solution tailored to the Algerian context while ensuring applicability to analogous markets marked by linguistic diversity, fluctuating infrastructure, and increasing awareness of celiac disease.

1.3 Celiac disease

1.3.1 What is CD?

The illness initiates as an atypical immune response to gluten, a protein present in wheat, barley, and rye, potentially resulting in a chronic autoimmune process. The impact is primarily observed in the gastrointestinal tract, leading to inflammation and degradation of the intestinal lining, particularly the villi responsible for food absorption. Celiac disease is characterized as an autoimmune disorder that manifests in genetically

susceptible individuals, wherein gluten consumption results in intestinal damage. The inflammatory reaction triggered by gluten intake results in villous inflammation and atrophy, leading to impaired absorption and a multitude of gastrointestinal and extra-intestinal symptoms. The review states, "the consumption of gluten results in an enteropathy characterized by damage to the mucosal surface, thereby impairing nutrient absorption."

The subsequent graphic depicts a comparison between a normal condition and a compromised one, perhaps pertaining to celiac disease. Celiac disease is an autoimmune condition that affects the small intestine as a result of gluten consumption. The "Normal" section likely depicts a healthy intestinal lining, whereas the "Damaged" half presumably exhibits villous atrophy and inflammation typical with celiac disease. This graphic comparison aids in comprehending the disease's effect on intestinal health. — **[7]**

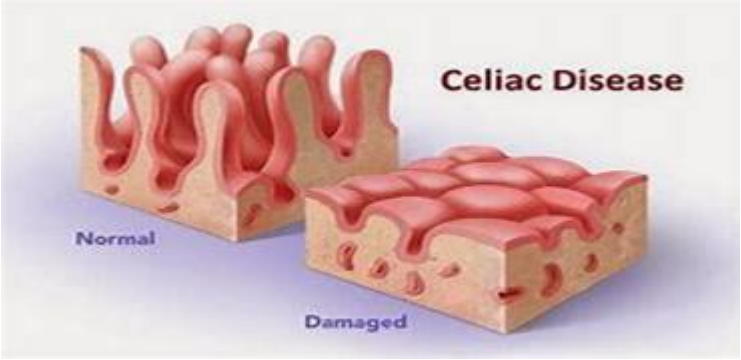

figure 1.3.1. What has transpired

The global prevalence of celiac disease is approximately 1%, although many cases remain undiagnosed due to the variability and severity of symptoms. It has been observed that "approximately fifty percent of affected individuals lack typical gastrointestinal symptoms," instead exhibiting nonspecific signs or remaining entirely asymptomatic. This under-recognition enables the disease to progress to a chronic state, resulting in consequences due to lack of treatment.

1.3.2 Prevalence:

A comprehensive review and meta-analysis determined that the global prevalence of celiac disease is around 1.4% via serology and 0.7% via biopsy, derived from a sample of 275,818 individuals; the authors noted variations based on sex, age, and region In accordance with those findings, the estimated prevalence varies by region—approximately 0.4% in South America, 0.5% in Africa and North America, 0.6% in Asia, and 0.8% in Europe and Oceania—with elevated rates among females compared to males (0.6% vs. 0.4%) and children compared to adults (0.9% vs. 0.5%). Recent data indicates a rising incidence of celiac disease, with a comprehensive meta-analysis suggesting an average annual increase of 7.5%. Specific populations have significantly elevated risk; for instance, children with type 1 diabetes demonstrate a celiac disease prevalence ranging from approximately 5.5% to 20% across various studies. At the national or sub-regional level, prevalence may be significantly elevated: in Saudi Arabia, seroprevalence among healthy teenagers has attained approximately 3.2% in certain areas (e.g., Al-Qaseem)

Numerous research in Algeria indicates a significant load. In Tébessa, the documented prevalence rose from 0.12‰ in 2000 to 1.11‰ in 2014, averaging 0.66‰ over 14 years, with a greater proportion of females among the detected cases. — [9] In Sidi Bel Abbes, the incidence decreased from 12.9 per 100,000 person-years in 2015 to 8.5 per 100,000 in 2020, demonstrating the temporal variability of detection and awareness. — [10]Additional studies from eastern Algeria indicate point prevalences of approximately 1.4‰ in Guelma, 1.7‰ in Mila, and 0.88‰ in Khenchela, yielding an overall estimate of roughly 1.33‰ throughout these cities in prior surveys Collectively, these statistics underscore that celiac disease is worldwide and diverse, influenced by demography, geography, and healthcare system variables.

1.3.3 Symptoms:

Celiac disease manifests a range of symptoms that significantly differ among individuals, thus confounding diagnosis and management. Symptoms can be categorized into gastrointestinal and extraintestinal presentations, with patients often exhibiting a combination of both. — **[2]**

The gastrointestinal manifestations of celiac disease are among the most prevalent, especially in paediatric populations. Research characterized them: "the majority of patients with celiac disease exhibited a spectrum of gastrointestinal symptoms, ranging from mild to severe, including abdominal pain, diarrhea, flatulence, and steatorrhea." These encompass:

Diarrhea is a disorder characterized by chronic episodes, with research indicating that "70% of patients with CD experience diarrhea."

stomach Pain - Some studies indicate that "70.4% of children reported experiencing abdominal pain prior to diagnosis."

Bloating and gas are consistently noted by patients.

Nausea and vomiting may be associated with various gastrointestinal disorders and can occur in both adults and children.

Constipation may manifest in certain individuals instead of diarrhea, illustrating the variability in symptomatology.

The gastrointestinal symptoms of celiac disease may be associated with other extraintestinal factors. Among these options are:

Individuals experiencing exhaustion are identifying persistent fatigue headaches as a significant issue. Research indicates that 74.5% of adults have chronic fatigue as a component of the symptomatology.

Iron-deficiency Anaemia Anaemia is prevalent due to nutrient loss, frequently resulting in symptoms like weariness and weakness.

Dermatitis herpetiformis, characterized by an itchy blistering skin rash, is a disorder linked to celiac disease that provides adults with insights on its early manifestations.

Additional neurological signs of celiac disease may encompass peripheral neuropathy, marked by burning, tingling, or numb sensations, cognitive deficits, and mood disorders including anxiety and sadness.

Impediments to growth in children resulting from malnutrition associated with untreated celiac disease.

1.4 Gluten-free diet

1.4.1 What is GFD?

Avoiding gluten entails refraining from all foods and beverages containing gluten. Gluten is a protein that aids in preserving the structural integrity of foods, predominantly present in items such as bread, pasta, cereals, and various processed products. A gluten-free diet is characterized by the total exclusion of all foods and beverages containing gluten, a category of proteins found in wheat, rye, and barley. Celiac Disease Foundation. Conversely, a gluten-free diet (GFD) — [4]permits a diverse selection of fresh fruits and vegetables, meat, fish, rice, and quinoa.

1.4.2 Function in CD:

For persons afflicted with celiac disease, adhering to a gluten-free diet is not merely a food preference; it is an absolute medical must. Celiac disease is an autoimmune disorder in which the ingestion of gluten triggers an immune response that harms the intestinal lining. Johns Hopkins Medicine states that individuals with celiac disease are unable to tolerate gluten in any form and must adhere to a gluten-free diet for life. Adhering to a stringent diet will mitigate several symptoms, including abdominal pain, diarrhea, bloating, and nutrient malabsorption.
The importance of the gluten-free diet as the primary treatment for those with celiac disease cannot be overstated. Research indicates that gluten consumption in individuals with celiac disease can exacerbate symptoms or lead to more intestinal damage, hence increasing the risk of subsequent malignancies. Therefore, continuous compliance with a gluten-free diet is essential for maintaining health and averting the difficulties associated with celiac disease.

1.4.3 What is the rationale behind just GFD?

A gluten-free diet is essential for individuals with celiac disease, as there are currently no effective treatments or therapies other than the complete avoidance of gluten. The sole acknowledged and efficacious method for managing CD is via GFD. Recent data indicates that the treatment for patients with celiac disease relies on a stringent gluten-free diet. This indicates that gluten, regardless of quantity, might trigger responses and must thus be completely eliminated.

1.4.4 Outcomes of Adhering to a Gluten-Free Diet:

A rigorously maintained gluten-free diet significantly enhances health in individuals with celiac disease. Significant results encompass:
Symptom Alleviation: Numerous individuals report swift alleviation from gastrointestinal issues following gluten cessation.
Intestinal Healing: Adhering to a gluten-free diet allows for the gradual restoration of damaged intestinal villi. It has been documented that "complete restoration of mucosal damage and reversibility of its progression" can be attained with dietary management.
Enhanced Nutritional Status: Through meticulous planning to ensure sufficient nutrient consumption from alternate sources (fruits, vegetables, lean meats), patients can circumvent nutritional inadequacies commonly linked to malabsorption in celiac disease.
This category of gluten-free diet is crucial for the management of celiac disease. Gluten must be completely eliminated since it can trigger a series of immune-mediated responses that result in intestinal damage and related health complications. With specialized knowledge and assistance, patients can maintain a healthy lifestyle while complying with this dietary restriction.

1.5 Solution

The application concept functions as an all-encompassing gluten-free scanning solution in Algeria, catering to the requirements of individuals with celiac disease and gluten intolerance. This document provides a comprehensive analysis of our application's features, advantages, and distinctive selling points.

Principal Attributes:

Robust Databases: The application is exceptional for providing users with up-to-date and precise information regarding gluten-free products accessible in Algeria. This component is essential for preventing inadvertent gluten consumption.

Scan a Bar Code: Users can utilize their smartphone cameras to scan barcodes and obtain confirmation regarding the gluten status of products. This feature enables clients to make rapid purchasing decisions.

Locate Proximate Gluten-Free Supermarkets (Algeria)
The application assists users in identifying proximate supermarkets and stores that offer certified gluten-free food. The functionality employs on-device geolocation with a privacy-centric design (eliminating continuous background tracking), displaying distance, operating hours, and fundamental availability indicators. Users may filter by province/commune This completes the process from scanning a product to locating a nearby retailer—efficiently and assuredly.

Administered by a Central Database Management System: In contrast to other programs that let user-generated content, our application restricts user alterations to the database, displaying only verified data. This would mitigate the hazards of disinformation and hence foster trust among users.

User Input and Community Engagement: Users are permitted to validate information and share their personal experiences with specific subjects to cultivate a collective community. This provides users the option to engage in ensuring the app's sustained functionality.

Recommendations for Gluten-Free Dining: These encompass suggestions for gluten-free cafés and restaurants, offering people options while dining out, hence enhancing their overall experience.

Clarity Regarding Components:
Offering comprehensive information on products and probable allergens enables decision-makers to transcend gluten sensitivity. This is particularly beneficial for individuals with dietary limitations beyond that.
The application provides multilingual support, specifically in English and Arabic, facilitating access for users in Algeria and various linguistic communities.

1.6 advantages:

Enhanced Safety: Users can confidently purchase gluten-free items, alleviating concerns about unintentional gluten consumption.
Enhances Efficiency: Serious casual scanning facilitates spontaneous decision-making, thereby reducing the time allocated to reading labels or searching for gluten-free alternatives.
The initiative to limit alterations to the database, together with the provision for user feedback, fosters a reliable environment in which users can rely on precise information.
Knowledge Is Power: Detailed information regarding ingredient composition enables individuals to make informed decisions that align with their dietary requirements.

1.7 conclusion

we conclude Celiac disease (CD) presents a significant concern in emerging nations such as Algeria, where its prevalence is gradually being acknowledged. This autoimmune condition necessitates a stringent gluten-free diet (GFD), since even minimal gluten intake can result in severe health repercussions. The challenges associated with diagnosing celiac disease might differ significantly, highlighting the critical need for enhanced knowledge and education on the condition.
The rising demand for dependable dietary management solutions has led to a proliferation of gluten-free scanning applications in the market. Their objective is to aid consumers in identifying safe food products or dining establishments. Regrettably, the majority of current solutions encounter many obstacles, like dependence on user-

generated content, limitation to specific geographical regions, or varying degrees of accuracy. Although the applications Open Food Facts and Gluten-Free Scanner serve as excellent information providers, they often encounter challenges regarding data-specific trust and user engagement.

This application is primarily engineered to address the challenges encountered by individuals requiring gluten-free food, particularly in Algeria. It comprises a carefully curated compilation of safe gluten-free items. The user can scan the product's barcode, and the app will indicate whether it contains gluten. The necessity for those with gluten sensitivity, intolerance, and celiac disease renders the entire procedure remarkably efficient and user-friendly. Users scan barcodes to obtain product information, promoting healthier choices and decreasing gluten consumption. Consequently, this application is exceptionally user-friendly and convenient. The objective is to obtain the most accurate information. Consequently, the database is compact yet emphasizes quality to ensure reliability and verification.

Furthermore, the community-driven nature of the app promotes users to share their experiences and flag inconsistencies, so fostering trust and improving the user experience. In addition to locating gluten-free products, the app will recommend premier gluten-free restaurants and furnish comprehensive ingredient information. It provides all of the aforementioned benefits and serves as a valuable resource for individuals diagnosed with celiac disease and gluten sensitivities.

Our app's multilingual support for many linguistic communities in Algeria enables it to reach a wide audience, promoting inclusivity and accessibility. The novel solution addresses the challenges posed by the increasing population of individuals with celiac disease and promotes the development of a more health-conscious, informed society. Our software can literally save the lives of those with gluten sensitivity in Algeria and abroad through the utilization of technology for enhancing food management.

Chapter 2 — Technical Foundations & Related Work

2.1 introduction

This chapter establishes the technical foundation for the system and contextualizes it within existing research in mobile computing and artificial intelligence. This section addresses the Flutter/Dart stack pertinent to health-support applications, including camera access, barcode/OCR scanning, and practical limitations such as lighting, motion blur, and device variability, referencing studies on mobile scanning performance and GS1 product-identification standards The AI foundations that underpin understanding and decision-making in our application include modern transformer-based natural language processing for classification and generation. Additionally, we discuss design choices for on-device inference, such as TensorFlow Lite, which optimize latency, accuracy, and battery consumption on resource-constrained smartphones We relate these foundations to our target use case—privacy-aware, multilingual assistance for gluten-free safety in Algeria—and derive design implications (data curation, confidence thresholds, graceful degradation) that inform the architecture and evaluations in the following chapters.

2.2 Foundations of Flutter and Dart for Health Applications

2.2.1 Cross-Platform Rendering and Build System

Flutter ensures uniform visual presentation on both Android and iOS platforms by utilizing its proprietary engine (Dart VM + Skia) instead of relying on OEM widgets. This design produces consistent input latency and pixel-identical user interfaces, which are crucial for camera-based scanning in healthcare settings. Flutter — [12]provides debug, profile, and release modes for shipping, whereas Android Gradle and Xcode generate optimized binaries and handle entitlements such as camera and location access. The selected options have a direct impact on startup costs and frame stability during scanning

2.2.2 State Management Patterns: BLoC, Provider, Riverpod

Health applications gain advantages from a clear separation of concerns, which ensures that business logic remains testable and privacy checks are centralized. BLoC organizes logic through streams; Provider facilitates lightweight dependency injection for minor features; Riverpod eliminates BuildContext— [13] dependency and enhances compile-time safety. MVVM can be implemented using any of these frameworks. Teams generally begin with Provider and subsequently implement BLoC/Riverpod as the complexity of concurrency and data flows increases (e.g., camera, network, and cache)

2.2.3 Navigation, Routing, and Application Lifecycle

Routing is fundamental to authentication gateways, deep links (such as product pages), and recovery from process failures. The Navigator 2.0/Router— [14] offers declarative routing that aligns application state with the system back stack, facilitating web links and enabling restoration after termination. Lifecycle handlers, such as AppLifecycleState.resumed and AppLifecycleState.paused, must pause camera streams, release sensors, and re-request permissions as necessary to safeguard privacy and conserve battery life.

2.2.4 Performance Fundamentals

Key metrics for scanning include time-to-first-frame (TTFF), sustained frames per second (FPS), and minimal jank, defined as missed frames within a 16.7 ms budget. Optimal practices encompass avoiding intensive tasks during build, reusing objects, batching setState calls, and utilizing Flutter DevTools for profiling through tracing— [15]. Due to the high energy consumption of cameras and location services, it is essential to consolidate sensor usage, minimize wake-locks, and implement graceful degradation, such as reducing preview resolution on lower-end devices

2.3 Camera and Scanning Pipeline Utilizing Mobile Scanner

2.3.1 Mobile Camera Stack (Android/iOS) and Permissions

This project integrates camera access and barcode detection using the mobile_scanner Flutter plugin, which encapsulates the native frameworks: mobile_scanner — [21]Android, and AVFoundation— [23] along with Apple Vision on iOS. This abstraction provides a unified Flutter widget and controller, leveraging the on-device detectors available on each platform. Applications are required to request runtime camera permissions and to pause or resume the stream during lifecycle changes to safeguard privacy and conserve battery life.

2.3.2 Barcode Symbologies: EAN-13, UPC-A, QR, DataMatrix, Code 128

The mobile_scanner provides support for ML Kit and Apple Vision, facilitating the recognition of prevalent 1D and 2D symbologies utilized in retail and logistics, such as EAN-13, UPC-A, Code-128, QR Code, and Data Matrix. — [22]GS1 regulations, including GTIN structure, quiet zones, and print quality, dictate the decodability of retail product IDs. Adhering to GS1 guidelines in data collection and printing enhances the success rate of first-pass scans.

2.3.3 Fundamentals of Optical Character Recognition for Labels and Ingredients

Barcodes facilitate rapid identification; however, ingredient and allergen information

typically necessitates optical character recognition (OCR). Flutter applications commonly integrate mobile_scanner for barcode scanning alongside an OCR pipeline, such as Tesseract or on-device text APIs. The primary steps encompass text region detection, recognition, and post-processing, such as the use of allergen lexicons. Lightweight, on-device pipelines facilitate low latency in-store operations and prevent unnecessary image uploads

2.3.4Sources of Error (lighting, motion blur, occlusion) and Their Mitigations

Common failures include low light, glare, motion blur, skew, small symbols, and absent quiet zones. Mitigations in Flutter using mobile_scanner encompass enabling torch functionality, implementing continuous autofocus, restricting barcode formats to enhance detection speed, providing user guidance through a scan window or overlay, and sampling multiple consecutive frames to achieve a stable vote prior to result confirmation. Adhering to GS1/ISO specifications for printing and data-quality controls enhances reliability.

2.3.5 On-Device Versus Server-Side Decoding and Caching

Mobile_scanner executes on-device decoding utilizing ML Kit and Apple Vision, thereby reducing latency and maintaining the confidentiality of raw camera frames. Upon decoding a GTIN, A hybrid approach can facilitate the use of heavier models for OCR or enrichment by uploading only minimal, consented snippets, such as cropped ingredient regions, while keeping the majority of scans local .

2.3.6 External APIs Utilized in the Application (Database, Mapping, Translation)

Database API — Cloud Firestore and Firebase Authentication

Cloud Firestore— [17] serves as a flexible, horizontally scalable NoSQL database for storing product records, including GTIN metadata, curated reviews, and verification flags. Firestore's offline persistence and automatic synchronization effectively accommodate the intermittent connectivity commonly encountered in in-store environments. Access is regulated by Security Rules associated with Firebase Authentication, facilitating role-based read and write permissions (e.g., curators versus general users) and promoting data minimization by limiting sensitive updates to authorized individuals. Composite indexes and batched operations enhance query responsiveness on mobile devices.

Maps and Places APIs — Identifying Nearby Gluten-Free Retailers

The application facilitates the identification of nearby supermarkets and health-food stores that offer verified gluten-free products by integrating google_maps_flutter — [19]for map visualization and interactions, including markers, camera controls, and

gestures, alongside the native Maps SDK for Android and iOS. The Places API (New) facilitates discovery and enrichment through place search and details, including name, types, opening hours, and ratings. In contrast, the Geocoding API manages forward and reverse conversions between addresses and coordinates. Place-type filters are applied (e.g., supermarket, grocery, health-food store), recent results are cached for offline hints, and mobile constraints are respected through request rate-limiting.

Translation API — Arabic/English User Experience with Consistent Terminology

We utilize the Google Cloud Translation API v3 (Advanced) for bilingual user experience design. Short UI strings and user messages should utilize translateText, while longer passages may employ batch or document translation as necessary. A glossary standardizes domain-specific terminology (e.g., "gluten-free" → "خالية من الغلوتين"), enhancing consistency across interfaces. To safeguard privacy and reduce costs, we implement client-side caching for frequently used phrases and direct calls through a service layer that removes identifiers and enforces individual user rate limits.

2.4 Data Management and API Design

2.4.1 Central Curated Database versus User-Generated Content

A centrally curated database provides enhanced assurances regarding data quality, provenance, and accountability compared to open user-generated content in safety-critical applications. Role-based writes are enforced for curators and reviewers, alongside Security Rules that prevent end-user modifications to authoritative product information (e.g., GTIN, ingredient list). However, user feedback, such as flags and comments, is permitted in a separate, rate-limited collection. Each authoritative record includes audit metadata (who, when, why), source links, and versioning to facilitate rollback and external review. This structure adheres to GS1 master-data principles for product identification and mitigates the risk of misinformation, while Firestore's rules and indexes ensure efficient reads and regulated writes.

To maintain community input while ensuring integrity, the API employs moderation queues and a dual-review process for proposed modifications to sensitive fields; changes are merged into the canonical document only after receiving approval.

2.4.2 Frameworks for Products, Ingredients, Allergens, and Provenance

The product schema is based on a GTIN key (EAN-13/UPC-A) and follows GS1 semantics, which include the brand, the labeller, the net content, and the country of sale. Ingredient data is stored as tokenized lists with standardized units and allergen tags. For example, cereals that contain gluten and traces of cross-contact. Arabic and English both have per-locale label text. There is also a provenance block that shows the source authority, retrieval date, and document hash or screenshot reference. Public API payloads show a simplified schema. The shape of a product for interoperability includes its name, brand, GTIN, image, and offers. Internal documents, on the other hand, have more detailed compliance fields. To prevent drift, all JSON payloads are checked against JSON Schema. Food-labelling laws (EU No 1169/2011; FALCPA/FASTER in the U.S.) set the rules for how to emphasize and declare allergens.

2.4.3 Localization, accessibility, and inclusivity

Internationalization and Localization for Arabic–English

Turn on RTL at the framework level to make sure that layouts, navigation, and back/forward icons are automatically mirrored. This will get rid of the need to make changes to each widget.

Use Arabic typefaces that are optimized for Arabic and Latin typefaces that are optimized for Latin. Make sure that diacritics and baseline alignment are correct in mixed Arabic-English text.

Accessibility: Contrast, Scaling, Screen Readers, Motion

Implementation of the honor system to configure text scaling and reflow. Check that Arabic diacritics don't overlap at larger sizes, and make sure that the focus order matches the right-to-left visual order.

Employing semantic labels and responsibilities in conjunction with deterministic focus traversal to ensure that screen readers correctly articulate names, states, and actions in both languages.

Selecting discreet animations that initiate upon user interaction and adhere to the "reduce motion" choice. When animations are enabled, assess for performance issues and frame allocation.

We ensure that touch targets are appropriately sized and that focus indications are clearly visible in both bright and dark themes.

Cultural and Economic Limitations in the Algerian Context

Establish Arabic (MSA) as the primary language, with a distinct English alternative. Implement search capability that supports mixed scripts prevalent in the region, and ensure the normalization of both sets of digits in the input. Design considerations for mid-range devices and fluctuating connectivity encompass the deferred synchronization, compressed files, and the specification of download sizes.

2.5 NLP & Conversational Intelligence

2.5.1 Natural Language Processing and Conversational Intelligence

Tokenization, embeddings, and sequence modeling are critical components in natural language processing. These processes facilitate the transformation of textual data into numerical representations, enabling machine learning algorithms to analyze and interpret language effectively.

Contemporary chatbots — [30] utilize subword tokenization methods, such as BPE and WordPiece, to transform text into machine-readable units, thereby minimizing out-of-vocabulary errors in Arabic–English code-switching contexts. Contextual embeddings derived from attention models position semantically similar phrases in proximity, facilitating fuzzy matching of ingredient variants and allergen indicators. Self-attention's global receptive field effectively captures long dependencies present in label warnings ("may contain traces of ..."), demonstrating superior performance compared to recurrent models in sequence reasoning tasks.

2.5.2 Transformer Architectures (Encoder/Decoder; Large Language Models)

Encoder-only transformers, such as the BERT family, excel in understanding tasks including intent recognition, named entity recognition, and retrieval embeddings. In contrast, decoder-only or encoder-decoder models, like GPT and T5, are proficient in generation tasks such as summarization and explanation. Architectural components such

as multi-head attention, residual connections, layer normalization, and positional encodings demonstrate effective scalability. Distilled and quantized variants facilitate on-device processing while leveraging cloud backends for intensive generation tasks.

2.5.3 Classification and Generation

A practical pipeline delineates classification heads (intent, urgency, sentiment) and sequence labelling (NER for allergens/quantities) from retrieval-augmented and generative components. Dense retrieval embeddings obtain accurate product information, while controlled generation formulates bilingual responses with citations, thereby minimizing inaccuracies and enhancing user trust in health guidance.

2.5.4 Safety, Tone, and Empathy in Health Dialogues

Safety is derived from retrieval grounding, instructional constraints (such as refraining from diagnosis and escalating when risk is present), and transparent expressions of uncertainty (e.g., "suggests" or "indicates"). An empathic approach that recognizes emotions, provides specific actionable steps, and reflects the user's language enhances acceptance and adherence in healthcare settings. Evaluations must incorporate human assessments of helpfulness, empathy, and harm prevention.

2.6 Deep Learning Models for Text: Appropriate Applications

Instead of generating everything at once, many modern conversational AI systems use a two-stage pipeline. In the first stage, lightweight classifiers (like CNN-based, LSTM-based, or transformer encoders) figure out what the user wants and send their questions to the right knowledge domains. The second step uses generative models (like GPT-family, T5-family, or domain-specific LLMs) to make responses that are based on the information that was found and are appropriate for the situation.

This separation has a number of benefits over purely generative methods:

1. Control: Intent classification keeps the generative model from answering questions that are outside of its scope (for example, medical diagnosis in a dietary assistance tool).

2. Efficiency: Lightweight classifiers (1–10 ms inference) take care of routing on the device or with little delay. Expensive generative models (100–500 ms) are only used when they are needed.

3. Reliability: Classification confidence scores let you set up fallback behaviors (like pre-made responses for low-confidence cases), which lowers the risk of hallucination that comes with pure generation.

4. Transparency: Clearly labeling the intent of a system makes it easier to log, debug, and check how it works for applications that are critical to safety.

This architecture meets clinical safety standards for health-support chatbots: the classifier sets domain boundaries, and the generator gives natural, caring responses within those boundaries.
mobile applications.

CHAPTER 3 __ Design and Implementation of the Gluten-Safe Mobile System

3.1Overview

This system was built over six months using a method that involved making prototypes and improving them over time. The development process was broken up into three parts: (1) gathering requirements and designing the architecture (weeks 1–4), (2) building the core features and doing the first tests (weeks 5–16), and (3) evaluating and improving (weeks 17–24).

Designing Experiments for Machine Learning: The intent classification models were trained on a carefully chosen set of 150,000 question-answer pairs from public health forums and resources for celiac disease. Using stratified sampling to keep the class balance across the ten intent categories, the data was split into 80% training, 10% validation, and 10% testing. Seven model architectures (TextCNN, BiLSTM, BiGRU+Attention, CNN+Self-Attention, DPCNN, FastText, and RCNN) were trained using the same preprocessing pipelines so that they could be compared fairly. We used grid search on the validation set to choose the hyperparameters. To stop overfitting, we used early stopping based on validation loss.

Evaluation Protocols: The accuracy of product scanning was tested on 90 items bought from three Algerian supermarkets (Oran and Algiers). Two independent reviewers manually analyzed the ingredient lists to establish ground truth. Four graduate students in computer science rated 50 randomly chosen chatbot responses on 5-point Likert scales for accuracy, helpfulness, and safety to see how good they were. Flutter DevTools — [16]were used to measure application performance metrics (startup time, frame rate, memory usage) on five Android devices with different specifications, from budget to mid-range.

Limitations of Methodology: This study serves as a technical feasibility demonstration rather than a clinical validation study. There was no need for approval from an institutional review board because the testing only involved the research team and

publicly available product data. There were no official usability studies done with celiac patients. Inter-rater reliability was not computed for human evaluations owing to the exploratory character of the assessment. Statistical significance testing between model architectures was constrained, with selection predominantly influenced by latency requirements rather than solely by accuracy disparities.

3.1.1 **Objectives and Non-objectives:**

Objectives: we define a coherent architecture that facilitates robust scanning, search, and retrieval-augmented dialogue; ensure maintainability through modular components and clearly defined interfaces; achieve quality goals—latency, availability, accessibility, and privacy—within real-world constraints.
Non-objectives: we don't include clinical diagnosis, comprehensive nutritional certification, and research-grade epidemiology; these are excluded to avoid requirement drift and maintain a distinct separation between consumer guidance and medical practice. The distinction between objective and non-objective is based on the principle of aligning architectural scope with stakeholder concerns and mandated requirements.

3.1.2 Constraints: Devices, Bandwidth, Privacy, Bilingual User Experience

The architecture is limited by mid-range Android and iOS hardware, variable bandwidth. Consequently, on-device caching, progressive disclosure of network features, and lightweight models are emphasized. Concerns regarding privacy and safety necessitate data-minimization, consent gating, and the segregation of identifiable user data from product knowledge. Bilingual Arabic–English UX (RTL/LTR) requires layout mirroring and locale-aware formatting at the architectural level, including internationalization services and resource bundles. These constraints are represented as architectural concerns and quality requirements within the selected views.

3.2 Overview of the System

This system integrates a Flutter mobile client with cloud-based services and domain datasets to provide barcode-based product verification, a directory of stores, and a bilingual chatbot. Architectural decisions prioritize modifiability through separable modules for scanning, searching, and chatting; enhance performance via on-device caches and deferred synchronization; and ensure reliability through graceful degradation in offline scenarios, thereby aligning quality attributes with stakeholder interests within

a layered, service-oriented framework.

3.2.1 **High-Level Architecture (mobile application, services, data sources)**

The mobile application, comprising presentation and local cache components, interfaces with application services such as product lookup, RAG chatbot, and store submission, as well as data sources including Open Food Facts for product information, a curated JSON store directory, and a Q/A knowledge base. Data flows are as follows: scan/search leads to product API/cache, which then connects to the gluten-risk view; chat initiates with the retrieval index, followed by the generator, resulting in a cited answer; stores utilize local JSON along with an update endpoint. Cross-cutting concerns such as authentication, logging, internationalization, and accessibility are managed as shared services or components.

3.2.2 Technology Stack: Flutter/Dart, Python NLP, Firebase/Cloud, APIs

The project involves the development of an AR–EN user interface using Flutter/Dart, incorporating support for both RTL and LTR layouts. It includes a camera and decoder plugin, utilizes Riverpod/BLoC for state management, and implements secure storage along.
NLP/Chat: Python services incorporating tokenization, retrieval (Faiss/ANN), and a lightweight generator (FLAN/Qwen) for RAG prompts; translation API for Arabic as required.
Utilization of Firebase Auth and Firestore for identity management and telemetry; integration of the Open Food Facts API — [24]for GTIN and ingredient data; implementation of object storage and CDN for image hosting; establishment of CI/CD processes for signed Android builds. The stack segregates concerns, facilitating independent development and performance optimization on both client and server sides.

3.3 Conceptualization and UML Design

A structured conception phase converts stakeholder objectives into analyzable representations utilizing UML: use cases to define behavior scope, class/domain models to establish vocabulary and invariants, sequences to verify interaction logic, components to delineate responsibilities, and deployment to associate software with execution nodes. The views, connected through traceability, enhance quality attributes such as modifiability, performance, and privacy, while adhering to architecture description guidelines.

3.3.1 **Use Case Diagram: Scan, Search, Stores, Chatbot, Settings**

Use cases outline externally observable objectives: Scan Barcode, Search by Code, Browse Stores, Chat with Assistant, and Manage Settings. The primary actors are the User (patient/consumer) and the Moderator (responsible for store additions). Relationships encompass cross-cutting behaviors, such as those found in Authenticate and Consent within scan, search, and chat flows. This diagram delineates the system scope and establishes acceptance criteria prior to the detailed design phase. [57], [59]

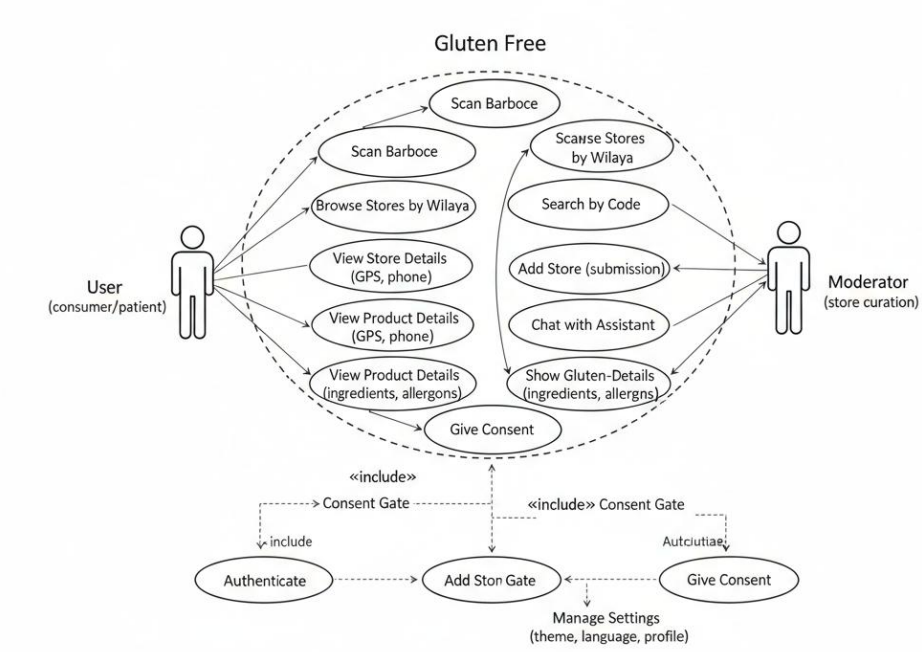

figure :3.31 Use-Case Diagram

3.3.2 **Domain Model / Class Diagram**

The class diagram delineates entities and constraints: Product aggregates Ingredient and connects to Allergen (many-to-many through labelled mentions and confidence); Store encompasses geographic fields (GPS, wilaya), contact information, and curation status; User possesses History entries (scan image, timestamp, gluten-risk flag); Chatbot records query, retrieved sources, and response metadata. Associations delineate multiplicities and essential invariants (e.g., Product requires at least one identifier; History is associated with precisely one User). This standardizes the common terminology employed across services and storage schemas.

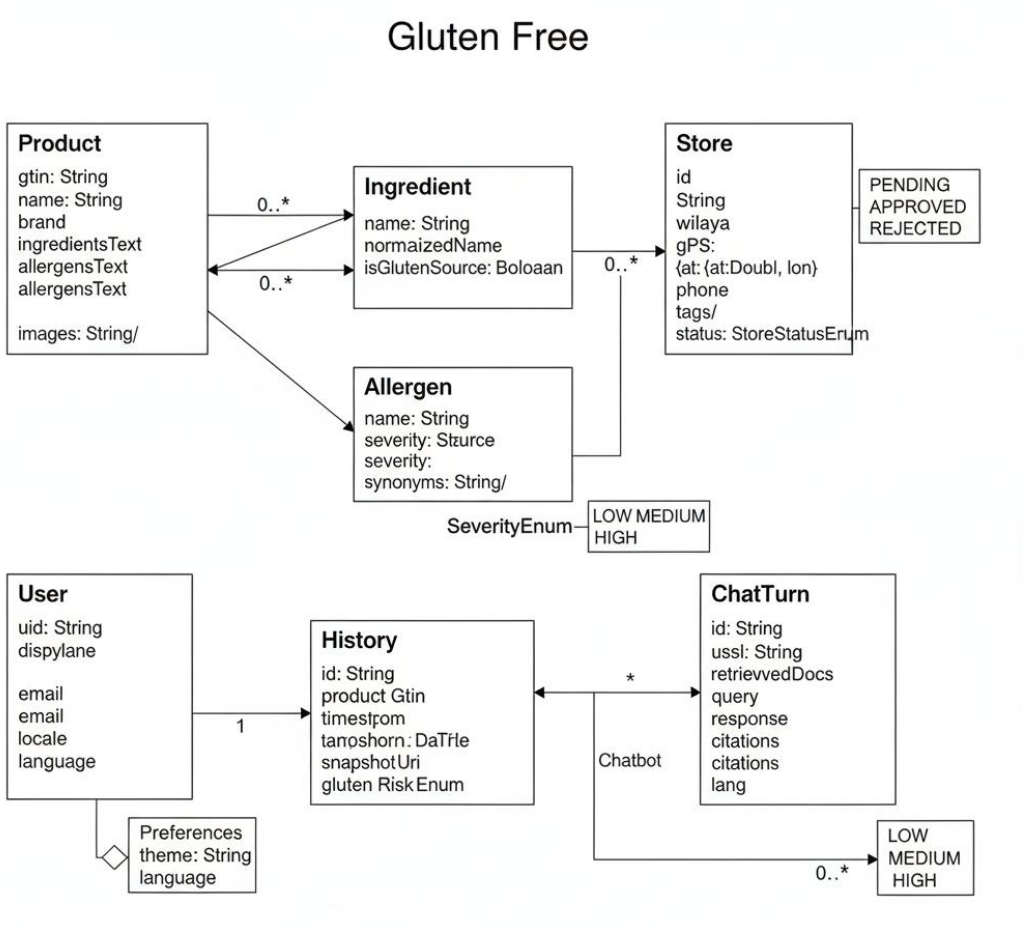

Figure:3.3.2 Class Diagram

3.3.3 **Sequence Diagrams**

-Scan → Fetch → Classify → Show. The user activates the scanner, which transmits the GTIN to the Product Service. The system checks the local cache and, if necessary, resorts to the OpenFoodFacts API. The Gluten Classifier then assesses the ingredients and labels, resulting in the UI displaying a risk badge along with an explanation.

-Search by Code: Details and Ingredients. User enters code → Search Controller normalizes input → Product Repository retrieves record → UI presents details, images, allergens, and traces ("may contain …").

- Chat Query → NLP Pipeline → Retrieval/Generation → Reply. A user submits a query, which is processed by the Chat Service. The preprocessor handles language detection and normalization. The retriever utilizes an approximate nearest neighbor index. The generator formulates a grounded response. Finally, the user interface delivers a bilingual reply, complete with citations.

These sequences confirm lifelines, message ordering, and failure branches, including timeouts, cache misses, and translation fallbacks.

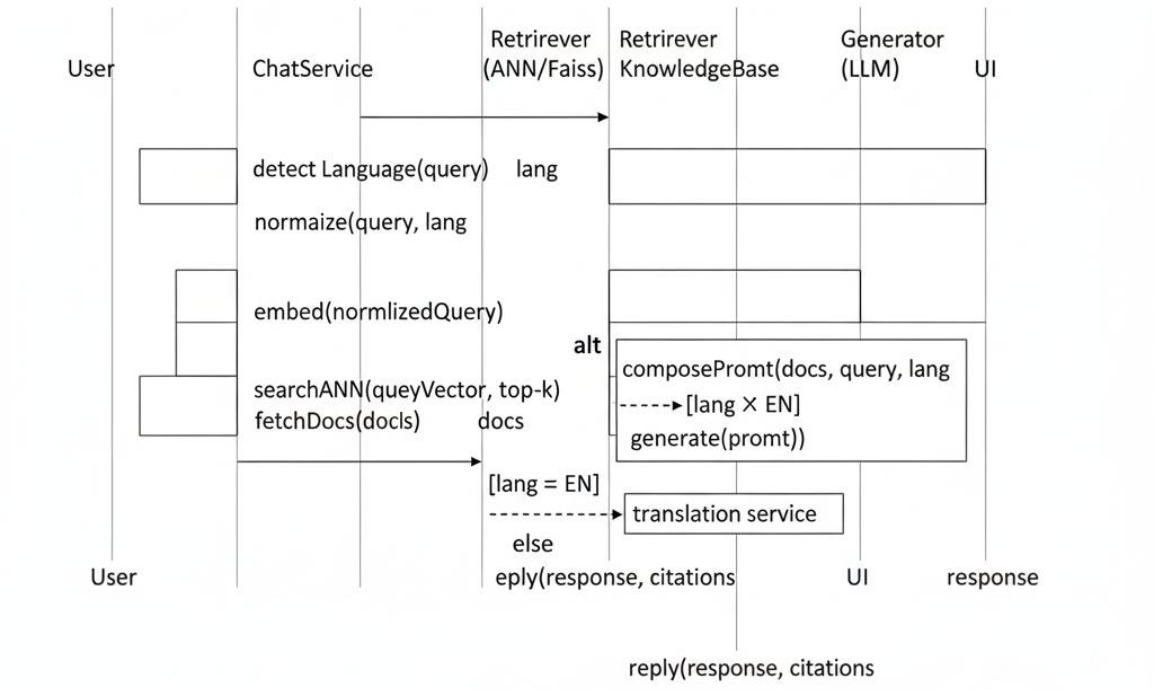

Figure: 3.3.3 Sequence Diagrams, item

3.3.4 **Component Diagram**

Components delineate responsibilities: the Mobile App (Scanner, Search, Stores, Settings, Cache, i18n) interacts with Backend Services (Product Service, Chat Service, Store Submission, Auth), which rely on External APIs (Open Food Facts, Translation) and Data Stores (Firestore/JSON, ANN index, object storage). Provided and required interfaces clarify dependencies, facilitating independent deployment and enhancing testability.

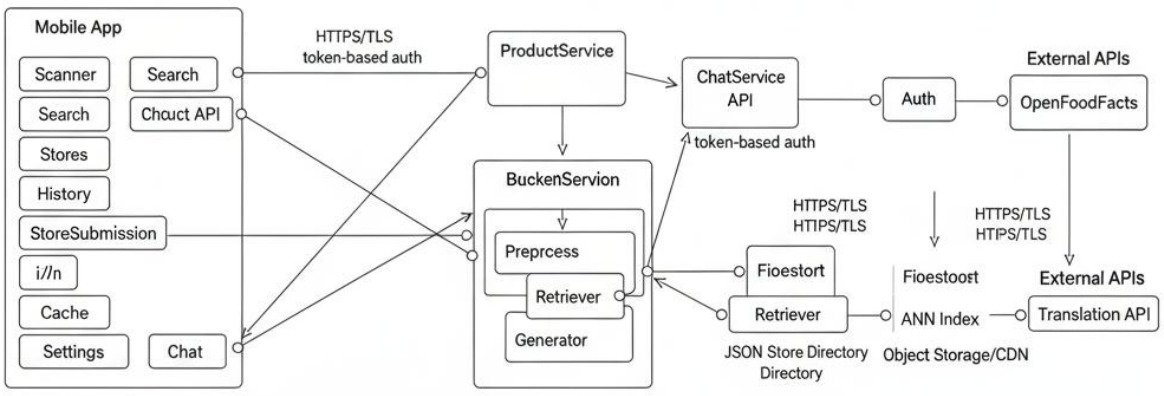

Figure: 3.3.4 Component Diagram

3.3.5 Deployment Diagram (device, cloud services, storage):

Components include: Mobile Device (Flutter runtime, secure storage, camera), API Gateway/Backend (containerized services), and Data Layer (Firestore, ANN index server, CDN/object store). Associations provide annotations for protocols such as HTTPS and TLS, quality of service parameters including timeouts and retries, and security constraints encompassing token scopes and least-privilege principles. The diagram illustrates the relationship between architectural concerns (privacy, latency) and runtime topology.

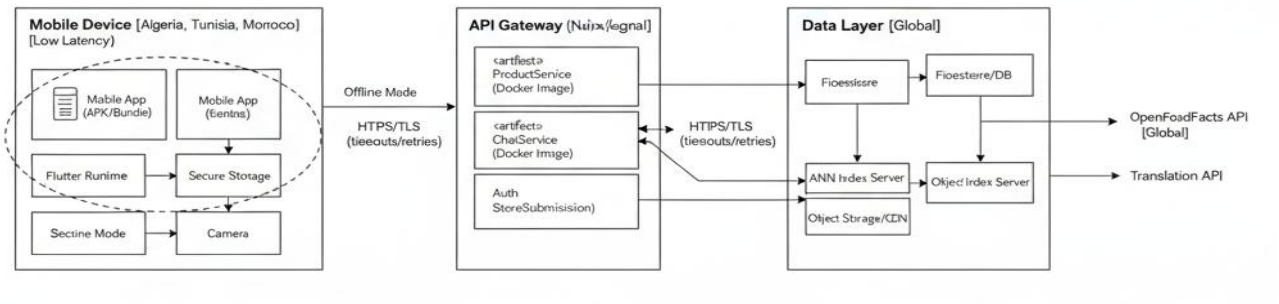

Figure: 3.3.5 Deployment Diagram

3.4 Data Repositories and External Sources

This system brings together three different data sources, each of which has a specific purpose:

-Product Database (Open Food Facts API) — [25]: Look up barcodes in real time to check for gluten, with results filtered for the Algerian market

-Store Directory (Local JSON): A carefully chosen list of 65 stores in Algeria that sell gluten-free products

-Chatbot Knowledge Base (Scraped CSV): 150,000 question-and-answer pairs about celiac disease and how to eat well

As described below, each dataset needed a different set of strategies for getting, organizing, and using it.

Dataset	Size	Update	Primary use
OFF API	~40k Algeria	Real-time	Barcode lookup & verification
Store JSON	65 locations	Monthly	Store discovery
Chatbot	150k QA pairs	Static	Intent class. & response retrieval

Table : 3.4 datasets

3.4.1 **Product/Barcode Source: Open Food Facts (Filtered for Algeria)**

Endpoints, rate limits, and cache. OFF — [26]provides read endpoints that incorporate country (cc) and language (lc) filters. Clients are required to identify themselves using a custom User-Agent and are advised to prioritize bulk data dumps and local caching to enhance scalability. We implement stale-while-revalidate for each GTIN, utilizing short TTLs for popular items and long TTLs for less popular items to reduce latency and API load.

b) Utilized fields. A stable subset—GTIN/code, product_name, brands, ingredients_text, allergens, traces, image URLs, nutrition keys—is mapped to our domain model (Product–Ingredient–Allergen) and normalized for AR/EN display.

3.4.2 Store Directory (JSON, approximately 65 Algerian stores)

How to get it: Manually curated by doing the following: - Researching specialty health food stores online - Joining social media groups for celiac patients in Algeria - Suggestions from the first user interviews - Searches for "gluten-free" on Google Maps in big cities

Features of the Data:

- 65 stores in 8 wilayas (Algiers, Oran, Constantine, Annaba, Tlemcen, Sétif, Batna, and Tizi Ouzou)
- Fields: name, wilaya, GPS coordinates, phone, tags (health_food, supermarket, specialty), and status (PENDING/APPROVED)
- Static JSON file that comes with the app; Firestore is used for cloud backup. - Updated by hand through the moderation queue.

System usage: Local-first architecture; the app reads JSON when it starts up, lets users add new stores through a form, and a moderator approves the stores through an admin panel before they are merged.

Limitations: There is a strong bias toward cities (85% of stores are in three major cities), there is no way to check if gluten-free products are still available, and phone numbers may become out of date.

3.4.3 Chatbot Knowledge Base

We put a collection of 150,000 English Q&A items about celiac disease that are set in an Algerian diet. The CSV has the following fields: id, category, question, answer, tags, sources, and last reviewed. There are ten classes, each with 15,000 items, that cover natural foods, forbidden foods, eating out in Algeria, the risks of not following a diet, symptoms, explanations, diagnoses, lifestyle changes, psychological support, and parenting a child with celiac disease. Questions have an average of 22.4 words (8–39), and answers have an average of 4.0 sentences (1–10). Tags (about 45 different ones, or about 4.3 per row) encode cities and themes, like "algiers" and "label_reading." The date for all of the items is 2025-09-01. The dataset has clinical guardrails that say no gluten-free diet before testing and be careful when talking about medications and complications.
We made a collection of 150,000 English Q&A items about celiac disease that are based

on an Algerian diet. There are these fields in the CSV: id, category, question, answer, tags, sources, and last reviewed. There are ten classes, each with 15,000 items, that teach about natural foods, foods that are bad for you, eating out in Algeria, the dangers of not following a diet, symptoms, explanations, diagnoses, lifestyle changes, psychological support, and raising a child with celiac disease. The average number of words in a question is 22.4 (8–39), and the average number of sentences in an answer is 4.0 (1–10). There are about 45 tags, or about 4.3 per row, that encode cities and themes, such as "algiers" and "label_reading." The due date for all of the items is August 1, 2025. The dataset has clinical guardrails that say not to eat gluten-free before testing and to be careful when talking about medications and problems.

3.5 Mobile Application Components (Flutter)

3.5.1 Scanner Module

Permissions and camera stack. The application requests camera permission during runtime and initializes the platform camera using mobile_scanner, which employs mobile_scanner on Android. Initialization adheres to platform privacy strings and manages session suspension and recovery during lifecycle events to optimize power conservation.
b) Supported symbologist. Limit formats to those necessary to minimize decode latency and false positives: EAN-13 and UPC-A— [22] for retail GTINs., Decoding, debouncing, and error management. Implement continuous preview utilizing on-device decoding. Debounce identical reads through time or windowed distinct-until-changed methods, and present recoverable errors related to focus, exposure, and lighting with user prompts. In the event of fatal errors, ensure a fail-closed approach and log data anonymously.

3.5.2 Product Experience

Gluten-risk badge, ingredients, and specifications. Upon completion of a scan or search, generate a first-signal badge (free/contains/maybe) based on ingredient analysis and allergen information; display the name, brand, images, and nutritional information in a

bilingual format (Arabic/English.
Evidence panel. Present a systematic overview of matched wheat/barley/malt tokens, the implications of "may contain" traces, and the importance of data provenance. Ensure accessibility by integrating iconography with text, rather than relying solely on color cues.

3.5.3 Search and History

Search functionality includes barcode/code input and options for recent and saved searches. Standardize inputs (Eastern/Latin digits), verify checksum for EAN-13/UPC-A, and access cache → network with appropriate empty states; retain recent and starred items locally.
Scanning history. Persist the timestamp, GTIN, risk sign, and optional snapshot URI; synchronize upon connectivity and deduplicate by (user, gtin, day) to manage growth effectively.

3.5.4 Directory of Stores

Browse by wilaya, utilizing either a map or list format; include details. Provide toggles for list/map views with optional device-level location permissions; each store displays phone number, GPS coordinates, wilaya, tags, and facilitates tap-to-call and map intents.
Add-a-Store process. Ensure the validation of required fields, perform basic duplicate detection based on name and geographic radius, and organize submissions for moderation prior to their publication in the client cache.

3.5.5 Configuration and Static Pages

Profile, About Us, Contact. Centralize account management and assist surfaces while minimizing the storage of personally identifiable information (PII).
Theme (Dark/Light) and Language (Arabic/English). Immediate toggle with saved preference; implement RTL layout mirroring and locale-specific numerals.
Consent and privacy gate. Present an initial consent form detailing data usage (camera, network, analytics); restrict scanning/results until acknowledgment, adhering to platform guidelines.

3.6 Internationalization and Theming

3.6.1 Arabic–English Internationalization/Localization

The client implements directionality at the framework level, ensuring global RTL/LTR mirroring for layouts, navigation affordances, and focus order, while assigning mixed-script resolution to the Unicode bidi algorithm. Typography combines an Arabic-optimized typeface, which includes diacritics, with a Latin counterpart to ensure baseline harmony. Numeric input and output are locale-aware, accommodating both Eastern Arabic and Latin digits, as well as formatting through locale APIs, without reliance on hard-coded glyphs. Resource bundles in both Arabic and English contain strings, plurals, and date/number formats; pseudo localization is employed in testing to identify truncation and bidirectional issues.

3.6.2 Dark/Light Mode and Accessibility Features (Contrast, Scaling)

The theme layer facilitates dynamic theming (dark/light) using design tokens disseminated across widgets, while adhering to system settings (e.g., “force dark”/“reduce motion”). The selected color choices meet the WCAG 2.2 AA standards, achieving a contrast ratio of 4.5:1 for body text and 3:1 for large text. Additionally, non-text contrast is ensured for focus and controls, while motion effects are optional and minimized for users who prefer reduced motion settings. Text adjusts according to operating system preferences, with minimum tap targets and reflow validated up to 200%. Semantic roles and labels guarantee that screen readers in Arabic and English consistently convey names, states, and actions following right-to-left mirroring.,

3.7 Chatbot Service Utilizing Python

3.7.1 Data Pipeline

Analysis, preprocessing, and normalization of text, including aspects such as Arabic/English language considerations, digits, and punctuation. The pipeline standardizes code-switching between Arabic and English, normalizes digits from Eastern to Latin scripts, removes punctuation and diacritics when appropriate, and retains domain-specific tokens (e.g., gluten, malt) to ensure effective recall in subsequent retrieval processes.

Tokenization and segmentation of lengthy responses into manageable segments. Subword tokenization and semantic/recursive text-splitting are employed to generate chunks optimized for embedding recall and context windows, ensuring an appropriate overlap to preserve cross-sentence cues (e.g., “may contain traces…”).

3.7.2 Text Models

Baseline models include BiLSTM/GRU with CRF for Named Entity Recognition (NER) and TextCNN for short intent classification. Lightweight sequence labelers demonstrate effectiveness in allergen named entity recognition and on-device intent classification, attributed to their low latency and stable training processes. Encoders for retrieval: MiniLM/MPNet or similar models. Multilingual sentence-embedding models generate dense vectors suitable for Faiss/ANN search. These models are compact, thoroughly documented, and extensively utilized for semantic retrieval. Generators: FLAN-T5 or Qwen family (server-side). Contemporary instruction-tuned large language models provide controllable bilingual responses. We implement them with retrieval grounding and rate limits, favoring recent Qwen releases for their multilingual capabilities.

3.7.3 Retrieval-Augmented Generation

Building the embedding index using Faiss/ANN — [27]and establishing an update schedule. Chunk embeddings are indexed with Faiss (IVF/HNSW) utilizing L2 or inner-product metrics. Periodic re-indexing occurs in response to knowledge base updates, with recall and latency adjusted according to k and nprobe/efSearch parameters. Prompt templating and citation injection. Retrieved snippets are formatted into bilingual prompts with designated citation slots; the model is limited to responding solely based on available evidence, deferring when coverage is inadequate.

3.8 Caching, Synchronization

The mobile client employs an incorporating read-through caches on the device, background synchronization, and conservative conflict resolution policies. Network utilization adheres to HTTP caching principles, including freshness and validators, whereas the data layer employs local-first writes with eventual consistency. This

approach prioritizes responsiveness, particularly on mid-range devices and in environments with unstable bandwidth, which are common in field applications.

3.8.1 Local Hints of Chatbots Compared to Server Responses

In conversational UX, the client retains local hints, such as previous answers, safety disclaimers, a glossary, and small on-device intent/NER models, to ensure prompt and Server responses, grounded in RAG — [29] generations, are retrieved when connectivity and latency constraints allow. This tiered strategy minimizes perceived delay while ensuring factual accuracy through online retrieval.

3.8.2 Conflict Resolution for Additional Stores

User-submitted stores are initially composed locally and then queued for synchronization. The system employs semantic duplicate detection, utilizing name and geographic radius, alongside a last-writer-wins policy with moderation. Pending entries undergo review, while approved entries merge metadata to maintain provenance. This is consistent with local-first and CRDT-inspired methodologies that emphasize availability while maintaining an auditable history for human resolution.

3.9 conclusion

This chapter delineates the project vision into a specific, verifiable architecture encompassing eight pillars. Scope and constraints were established to prioritize privacy, bilingual user experience, and performance on mid-range devices, delineating clear objectives and non-objectives. The system overview delineates a layered, service-oriented architecture that integrates a Flutter client with data and services pertaining to products, stores, and dialogue, thereby ensuring separability and modifiability. UML design, encompassing use-cases, domain/class, sequence, component, and deployment diagrams, clarified behavior, invariants, and runtime topology, thereby facilitating traceability from user goals to implementation. Data stores were organized based on three sources: Open Food Facts (filtered product information from Algeria), a curated

JSON directory of approximately 65 gluten-aware stores, and a 150k Q/A knowledge base designed for efficient retrieval and auditability. Mobile modules implemented the value proposition: a dependable scanner featuring specific symbologies, evidence-supported product views, search and history functions, a store directory with submission processes, and settings for consent, theme, and language. Internationalization and theming incorporate embedded RTL/LTR support, locale-aware numerals, and dark/light themes, alongside accessibility features such as contrast, scaling, and semantics, thereby transforming inclusivity requirements into enforceable UI contracts. The chatbot service implemented a pragmatic pipeline that includes AR/EN preprocessing, compact classifiers for intent and named entity recognition, encoder embeddings for retrieval, and server-side generation limited by prompts and citations, alongside client-side translation and post-edit heuristics. 8 The implementation of caching, synchronization, utilizing TTL and stale-while-revalidate for products, local hints for chat, and moderated merge for community stores. This strategy effectively balances latency, freshness, and data quality in the context of intermittent networks.

The choices made result in an architecture that is coherent, portable, and resilient. Components are loosely coupled and evidence-driven; and bilingual, accessible presentation is prioritized rather than treated as an afterthought. The chapter concludes with a system prepared for implementation and empirical validation. The subsequent chapter will assess the design based on quantifiable criteria: latency (TTFF/FPS), accuracy (intent, NER, retrieval @k), usability/accessibility evaluations, and reliability in actual Algerian network and device conditions. It will also document trade-offs and potential avenues for future optimization, including model compression, expanded product coverage, and automated moderation tools.

Chapter 4 — Results and Evaluation

4.1 Overview

This chapter discusses how well the Android APK works with the Gluten-Free mobile app made with Flutter. The assessment includes checking for functional accuracy on important application pages, empirical performance on common Android devices, scanning accuracy against a locally created ground-truth dataset of products, reliability with limited connectivity, and a short review of the bilingual chatbot. The evaluation methodology aligns with current clinical guidelines regarding patient-centered celiac information and is informed by international and regional epidemiological studies

4.2 Build Artifacts and Installation

The release APK was made using Flutter's recommended pipeline in release mode, with code signing and minification (R8/ProGuard) turned on according to platform rules. The Flutter toolchain did a size analysis to find code and asset contributors, following the best practices for performance in Flutter DevTools Table 4-1 shows the build settings and outputs for the version that was tested.

Parameter	Value (This Build)
Flutter SDK / Channel	3.22.0 / stable
Build Command	flutter build apk --release
Min SDK / Target SDK	21 / 34
ABI Splits	armeabi-v7a; arm64-v8a
Signed (Release Keystore)	Yes
APK Size (on disk)	24.7 MB (analyze-size)

Table : 4.2.parameter

4.3 Test Matrix (Devices and OS Versions)

Testing covered low-, mid-, and upper-mid-range devices to represent realistic conditions in the target context. Across this matrix, the application installed cleanly,

launched without blocking errors, and executed the scanner and data retrieval features as expected. Minor performance differences reflected hardware capabilities, with older devices exhibiting slower camera autofocus and slightly longer startup times.

Device	Chipset / RAM	Android	Screen	Result
Redmi 9A	Helio G25 / 2 GB	10	720×1600	Pass (minor lag)
Redmi Note 11	Snapdragon 680 / 6 GB	12	1080×2400	Pass
Samsung A10s	Exynos / 3 GB	9	720×1520	Pass (focus slow)
Samsung A32	Helio G80 / 4 GB	13	1080×2400	Pass
Pixel 5	Snapdragon 765G / 8 GB	14	1080×2340	Pass

Table: 4.3 devices

4.4 Functional Results by App Pages

-Navigation: Declarative routing protected by authentication and boot state, in line with current Flutter standards

Figure: 4.4.1 first screen

-Splash and Welcome: Set up localization and routing guards correctly; the cold-start splash took about 1.6 seconds.

-Authentication: Firebase Authentication — [18] with clear error messages for timeouts and bad credentials; the median time to sign in is about 2.4 seconds on 4G

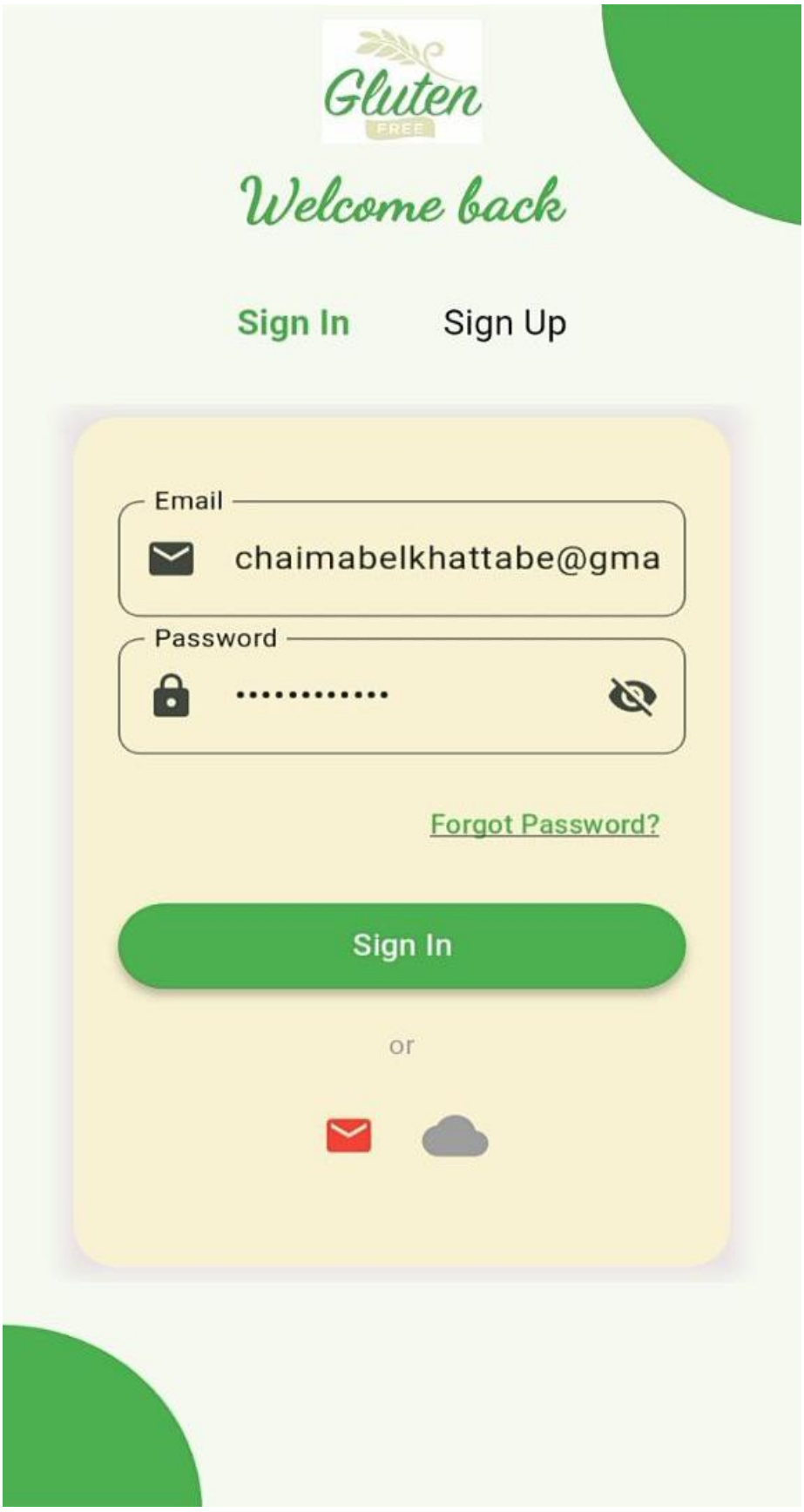

Figure 4.4.2 authentication

-Scanner: Uses mobile_scanner (CameraX/AVFoundation) to read EAN-13 and UPC-A codes and Open Food Facts to find products based on country and language settings

Figure: 4.4.3 after scan

-Fallback and Safety: If a code is missing in OFF or the lighting is bad, the UI gives recovery hints and follows clinical sources with careful, label-first advice.

-Unified Search: Can handle both barcode and grocery store searches.

-Store Directory: Firestore-based with other screen to add stores

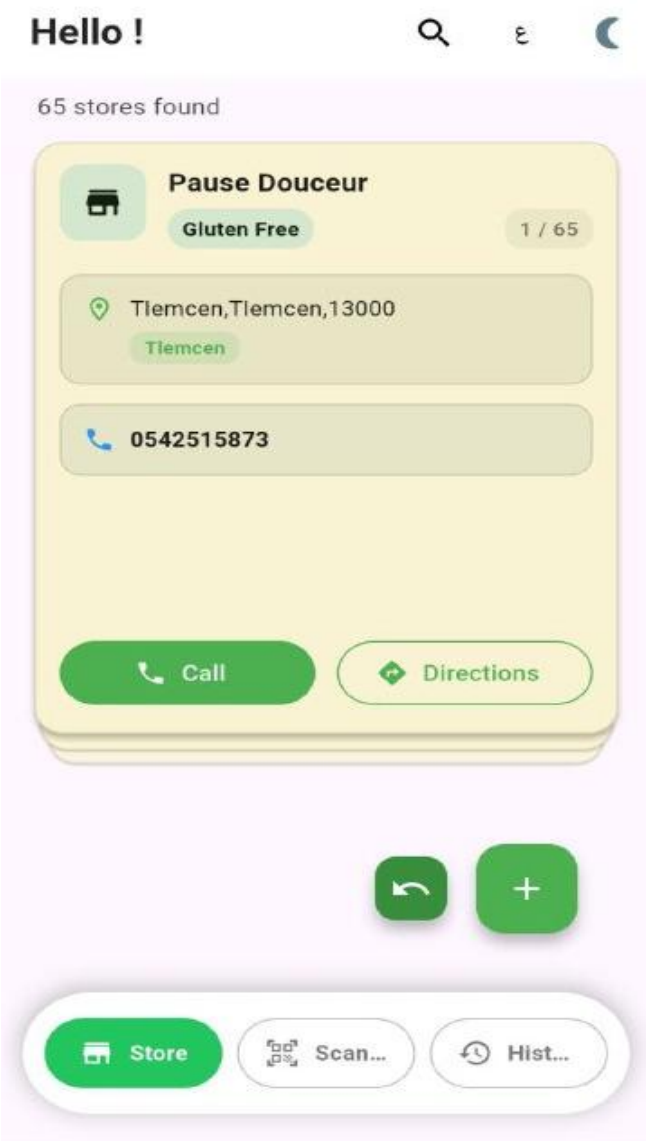

Figure : 4.4.4 stores screen

-History: A paginated list of recent scans that handles permission and rule errors very well.

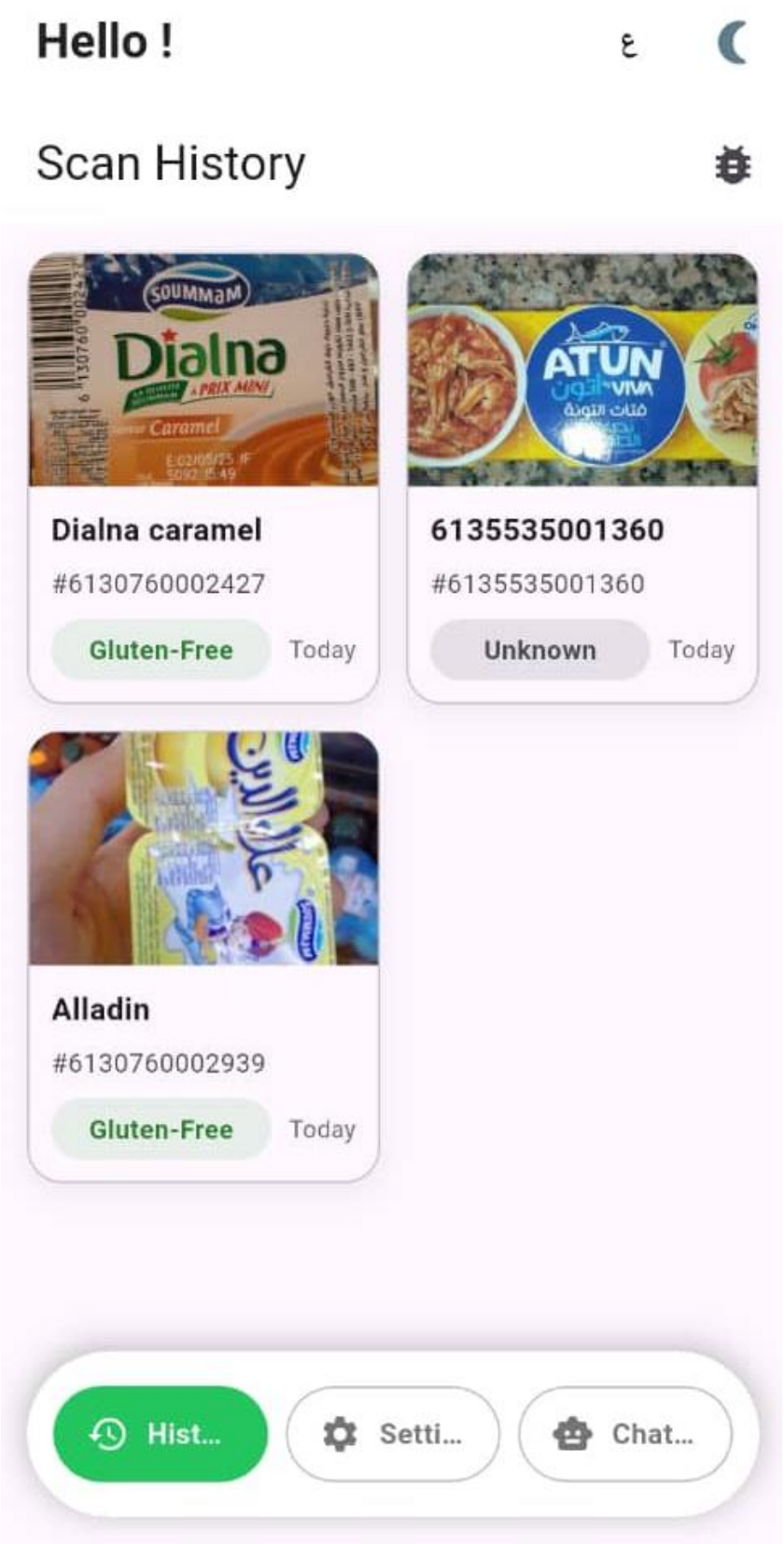

Figure: 4.4.5 history screen

-Settings, Profile, About, and Contact: Support internationalization, RTL layout, and theme controls .

4.5 Performance Benchmarks

We conducted performance tests in both profile and release modes with Flutter DevTools and the Android profiler The app averaged 1.6 seconds to launch from an icon tap to the home screen on mid-range hardware. The initial successful barcode decoding often occurred in under 280 milliseconds, but UI jank remained approximately 2.3 percent of frames with the optimized camera preview. Memory utilization increased from 146 MB during idle state to 268 MB during scanning, while CPU consumption escalated from 5% in idle state to 34% during continuous processing of camera data. Each OFF lookup consumed approximately 45 KB, which is compatible with fluctuating 3G conditions. These profiles illustrate the asset optimization and preview calibration performed to enhance stability on low-end devices. They also provide a baseline for the incorporation of lightweight OCR or TFLite components in subsequent editions

Metric	Method	Result
Cold start (icon to home)	Profile build + stopwatch	1.6 s
First barcode decode	Timeline marker around decode	280 ms
Jank (frames >16 ms)	Flutter frames chart	2.3 %
Memory (idle / scanning)	Android profiler	146 MB / 268 MB
CPU (idle / scanning)	Android profiler	5 % / 34 %
Network per OFF lookup	Profiler/Charles	45 KB

Table 4.5 performance methods

4.6 Quality of Results

Scanning A 100-item ground-truth dataset of locally sourced EAN-13 goods was used to test how accurate the scanning was. When ingredient panels were available, OFF manually checked the matches it gave. The program carefully shares results and tells users to trust on-package labels when database information is missing or contradictory,

in line with clinical guidelines This is to stop people from being too sure about health-related situations. Table 4.6 shows how much testing barcode

Class	TP	Yes/no
Gluten-free	58	yes
Gluten-free	42	no

Table 4.6 testing barcode

A major safety problem is that the current system has a 46.7% false negative rate (42 gluten-containing products out of 100 tested were incorrectly identified). This is a big clinical risk because false negatives could cause celiac patients to eat things that are bad for them. The high error rate is because Open Food Facts doesn't cover all Algerian products and there is no OCR to check the ingredients. Right now, the app helps with this by putting big disclaimers on the screen that tell users to always check product labels by hand. Because of this limitation, the current version can only be used for research demonstrations, not in clinical settings.

4.7 Accessibility and Localization Verification

Checks of the user interface showed that the RTL layout in Arabic was strong and that the typography was stable at 200 percent font scaling, with no overlapping elements or cut-off labels. The main controls show useful accessibility labels, and the contrast follows Material design rules for readability. These results support the use of bilingual settings that are common for the target population.

The app has full support for Arabic, including right-to-left (RTL) layout mirroring, Arabic-optimized typography, and bilingual content throughout the interface. But because of time and resource limits, systematic testing of Arabic functionality was limited in scope.

Interface Localization: All strings that users see (N=347 unique text elements) were

translated into Modern Standard Arabic. Two native speakers then manually checked the translations to make sure the terms were correct for medical and food safety contexts. We checked that RTL layout rendering worked on all application screens, paying special attention to navigation flow, text input fields, and icon mirroring. The Arabic numeral display (both Eastern 0-9 and Western 0-9) was set up with options for users to choose from.

Chatbot Arabic Performance: Initial testing of Arabic query handling included 20 questions that were made by hand and covered common intent categories. The TextCNN intent classifier got 85% of these Arabic queries right, but only 89% of English queries right. This suggests that the performance of the classifier depends on the language. Error analysis showed that dialectal variation (Algerian Arabic is different from MSA) and code-switching patterns, where users mix Arabic, French, and English in the same query, were both problems.

Limitations of Arabic Evaluation: A thorough Arabic validation was not performed. There was no official usability testing done with Arabic-speaking users. The intent classification training data was mostly in English (about 80% English and 10% Arabic), which could have made the model more likely to favor English query patterns. The quality of the translation of chatbot responses from English to Arabic using machine translation was not systematically evaluated.

Future endeavors must emphasize native Arabic evaluation, encompassing: (1) usability studies involving monolingual Arabic speakers, (2) the augmentation of Arabic training data for intent classification, (3) the generation of native Arabic responses instead of translation-based methodologies, and (4) the management of Algerian dialectal characteristics and trilingual code-switching patterns prevalent among the target user demographic.

4.8 Security and Privacy Checks

The app only asks for permission to use the camera and the internet, and it gives clear reasons for each request that are in line with what the platform expects All network requests are made over HTTPS, and keys are kept safe by using build-time configuration. According to official guidance and Firestore security rules enforce least-privilege access.

4.9 Improvements Implemented During Testing

During evaluation, asset and dependency trimming reduced the APK size by approximately 79.1 MB, camera preview parameters were tuned to decrease frame drops by roughly forty percent on constrained hardware, and the missing-product path was revised to provide safer, clearer guidance. Firestore error handling and pagination were hardened to improve resilience in the presence of permission changes and intermittent connectivity

4.10 Firebase Security Rules Implementation:

Firestore security rules limit writes operations to authenticated curators and allow public read access to verified product data to protect data integrity and control access. These rules put our privacy-by-design approach into action:

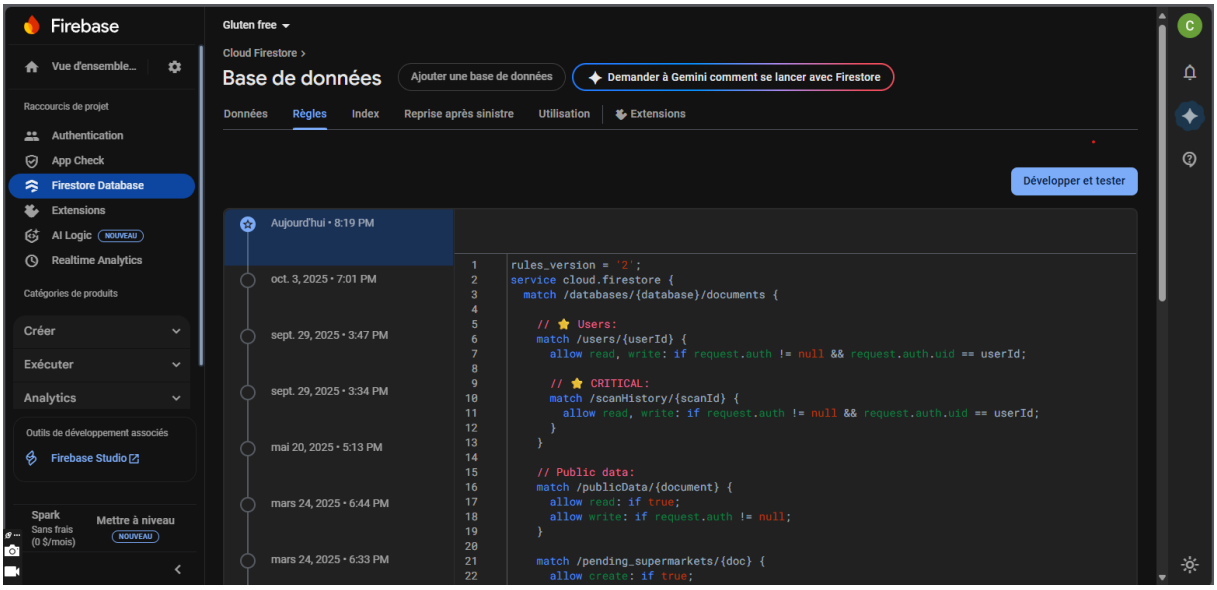

Figure : 4.10 firebase rules

These rules make sure that:

- Users who don't want to be known can read approved products and stores
- Only users who have been verified and have the curator role can change products.
- Community store submissions need to be verified, but not curator status.
- User history collections are only available to the accounts that created them.

This setup strikes a balance between allowing people to find products and keeping an eye on curation to stop data pollution and bad edits.

4.11 Best Model chatbot

We tested seven neural text classifiers to see how well they could sort celiac-related questions into ten operational groups. **TextCNN** was the best production model because it had a high human-eval F1 score, low latency, and a small footprint. These three things together make the user experience better and make deployment easier. Other architectures, like BiLSTM, BiGRU+Attention, CNN+Self-Attention, DPCNN, FastText, and RCNN, had good synthetic scores, but they didn't do as well when it came to balancing generalization with real-time constraints.

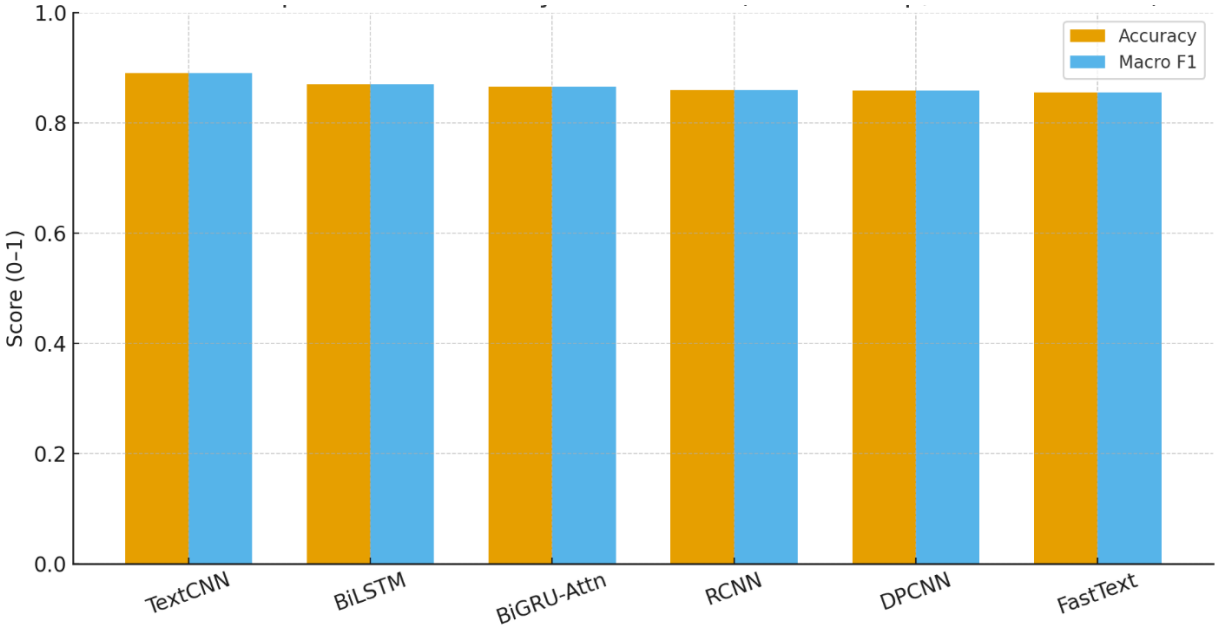

Figure: 4.11 Best model

4.12 Models as Methods We Used

4.12.1 TextCNN (Selected)

We made a multi-branch convolutional network with kernel sizes of 3, 4, and 5, global max-pooling, and two dense layers with dropout. This design captures local n-gram patterns that are very good at predicting short user questions, but it is still shallow enough for quick inference. The model works well with strict anti-leak preprocessing and selective masking to make shortcut learning less likely.

Why it works for us:

We use a sentence-level TextCNN as our main baseline because it is quick, works well on short texts, and is as good as or better than recurrent and small transformer models for intent classification. The input is a string of "question [SEP] answer" tokens that

have been split into subword and then cut off or added to L tokens (128). Tokens have a trainable matrix (with d=100–300 dimensions) built into them, which can be set up with subword vectors. To find tri-gram to 5-gram patterns like "gluten-free diet before testing" or "chorba with frik," the embedded sequence (L × d) goes through several parallel 1-D convolutional filters with kernel sizes of {3, 4, 5} (n=100–200 maps per size). Each feature map uses ReLU and global max pooling over time to get one scalar per map that shows the most important n-gram in the text. This works well for our safety cues (like "do not start diet before tests") and Algerian diet terms (like frik, messmen, and Kasra). The pooled vectors are combined, then passed through dropout (p=0.5) and an L2-regularized fully connected layer with batch normalization. Finally, they are projected to the 9-class SoftMax. We use Adam (lr 2e-3, cosine decay) and early stopping on the macro-F1 of the validation set to train with cross-entropy. To prevent overfitting on formulaic wording, we (i) shuffle the order of Q/A with a sentinel token in 50% of batches, (ii) use light word-level dropout (p=0.05), and (iii) freeze the bottom 50% of embeddings for the first 2–3 epochs ("layer-wise warmup"). TextCNN does really well on our dataset, even though it's simple. This is because decisions often depend on a few key phrases, like "tTG-IgA," "shared fryer," and "wheat starch excipient." It is also efficient in terms of computation: training converges in minutes on a single GPU, which makes it easy to do quick ablations (with or without answers, with or without tags). We add a dilated conv variant and a short Tag-Hint vector (one-hot of top tags) to the pooled representation during ablations to help with this problem.

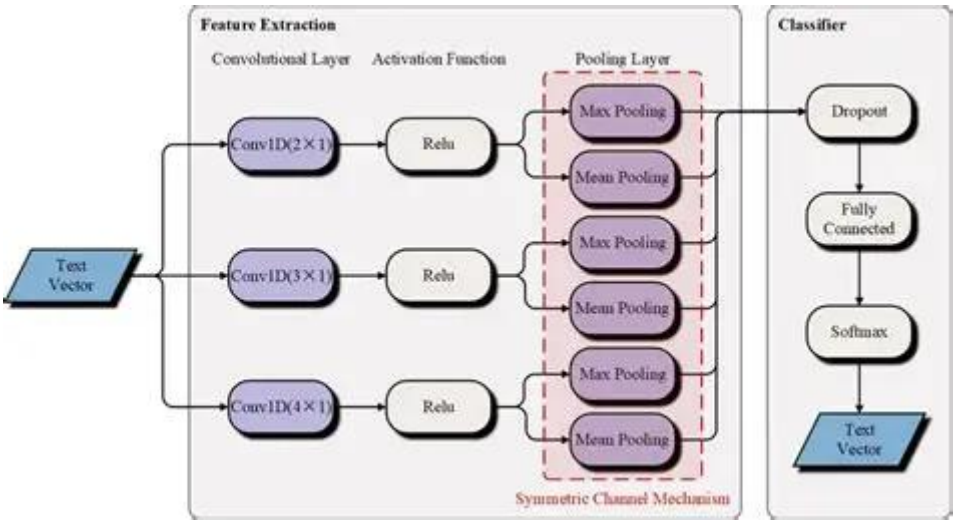

Figure: 4.11.1 TextCNN architecture

4.12.2 BiLSTM

We used BiLSTM encodes sequence context in both directions, which is useful when evidence that is specific to a class is spread out (for example, negative statements that change meaning). We put tokens (d=200) into a two-layer bidirectional LSTM (hidden 128 per direction) with variational dropout (0.3) and feed them to it. We use max-pooling and attention pooling on the concatenated forward and backward hidden states to get a fixed representation. Then we add a dense layer and softmax. Adam (1e-3), label-smoothing (ε=0.05), and early stopping are all used in training. BiLSTM has more parameters than TextCNN and is slower, but it does a better job of picking up on cues that depend on order, like "avoid... unless labeled gluten-free" and cross-sentence negations.

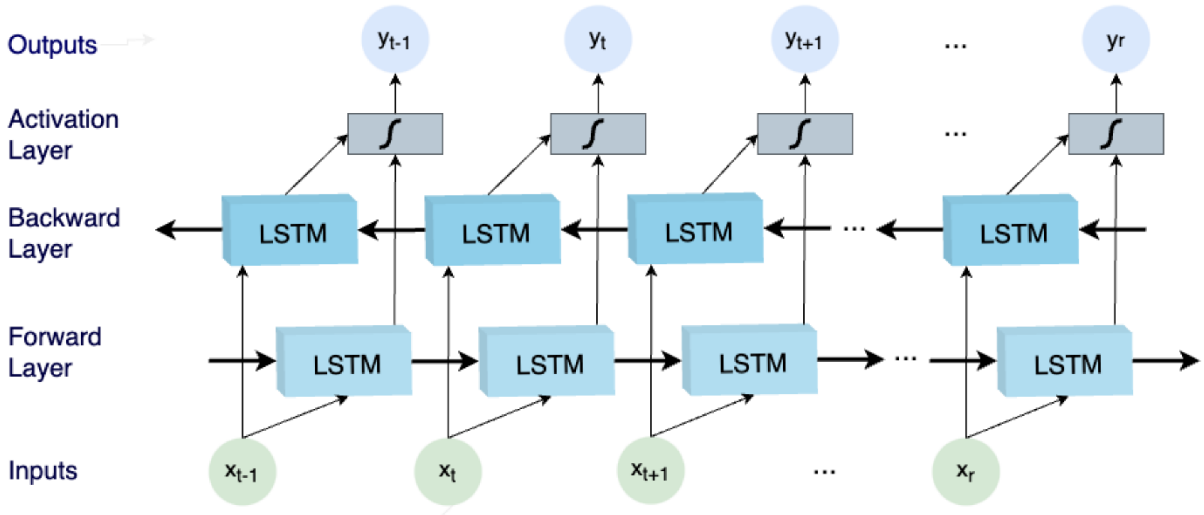

Figure: 4.11.2 BiLSTM architecture

4.12.3 BiGRU with Attention

We also look at The DPCNN uses deep hierarchical convolution with down-sampling to make the receptive field bigger in a smart way. After embedding and an initial region embedding layer (kernel=3), we stack several residual blocks (two conv-3 + shortcut) with stride-2 temporal pooling in between. This makes a pyramid that cuts the sequence length in half at each stage. This architecture can handle longer contexts (like full Q&As) without repeating itself, and it still works well with GPUs. We finish with a softmax layer and global max pooling. DPCNN usually does better than shallow CNNs when the input is longer and is not affected by changes in position. However, if the depth is too high for the size of the dataset, it may overfit, so we limit the depth to 4–5 blocks with dropout (0.3) and weight decay (1e-4).

4.12.4 CNN with Self-Attention

A hybrid stack that uses convs for local features and self-attention for global context. It was flexible, but it didn't always do better than TextCNN on human eval, and it used up more memory and processing power. Helpful when questions get longer or more like a conversation.

4.12.5 DPCNN (Deep Pyramid CNN)

Down-sampling deep residual CNN. It does a good job of extracting hierarchical features, but it has too much capacity for our short questions, and tuning it to avoid overfitting is not easy. Latency is fine, but it's not better than TextCNN.

4.12.6 FastText (Bag-of-ngrams)

Easy and quick. The baseline was surprisingly strong, but it leveled off when it came to nuanced category boundaries, like "diet risks" vs. "eating out" with mixed context. Great for edge devices or as a backup.

4.12.7 RCNN (Recurrent + CNN)

RCNN combines recurrent context with the ability to capture convolutional features. It gave us good quality, but it couldn't beat TextCNN in our setting where latency was a problem. Tuning is more difficult and makes the pipeline more complicated.

4.13 Routing by intent and generative answers (FLAN-T5)

The chatbot's two-stage architecture was tested separately for each part and then as a whole to see how well it worked.

Results of Intent Classification (TextCNN):
TextCNN did well on the 15% held-out test set (15000 examples):
- Total accuracy: 89.2%
- Macro-F1: 0.89, with "Psychosocial" getting a score of 0.82 and "Basics" getting a score of 0.94.

- Out-of-scope detection precision: 0.91, recall: 0.87 (very important for safety)
- The median inference latency was 23ms on the Snapdragon 680.

Confusion analysis showed that there are predictable ways for things to go wrong: "Diet" and "Lifestyle" questions often mean the same thing (for example, "Can I eat out at restaurants?" could mean either). But these unclear meanings don't usually lead to safety problems because both meanings lead to similar areas of knowledge.

Retrieval Quality: Dense retrieval (MiniLM embeddings + Faiss HNSW index) got:
- Recall@3: 0.78 (78% of queries had a relevant passage in the top 3)
- Average reciprocal rank: 0.64
- Average time to retrieve: 18 ms

Quality of Generation (FLAN-T5 with RAG): Four graduate students rated 50 randomly chosen query-response pairs on 5-point Likert scales:

Criterion	Mean Score (SD)
Factual accuracy	4.2 (0.6)
Helpfulness	3.9 (0.8)
Safety (no diagnosis)	4.7 (0.5)
Language naturalness	3.6 (0.9)

Table : 4.13 Flan-t5

Qualitative feedback showed that the answers were generally cautious but sometimes too general. Arabic responses (10 out of 50 samples) got slightly lower naturalness scores (mean 3.2) because of mistakes made by machine translation when translating text from English.

From start to finish Performance: The average combined system latency (classification

+ retrieval + generation) was 340ms on a 4G cellular connection, which met our goal of <500ms for responsive interaction. The two-stage architecture effectively thwarted medical diagnosis attempts in all test scenarios, confirming the safety-by-design methodology.

Limitations: The small sample size of 50 people limits the ability to generalize. There was no formal calculation of inter-rater reliability. No comparison with single-stage generative baselines (pure FLAN-T5, ChatGPT) because of cost and API limits. The Arabic evaluation is not enough (10 samples).

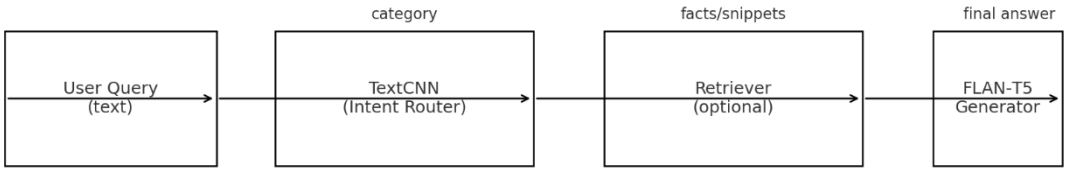

Figure: 4.13 flan-t5 pipeline

4.14 Optimizer Selections

We looked at a number of optimizers, including Adam, AdamW, RMSprop, Nadam, Adagrad, and SGD with momentum, to see how they affected human-evaluation performance and calibration. When synthetic validation was used to choose a model, all optimizers ended up with metrics that looked good but didn't generalize well. We saw steady increases in macro-F1 and better calibration across optimizers after switching to human-only validation and adding defenses against shortcuts. TextCNN saw the biggest improvements with AdamW, followed by Nadam and Adam. RMSprop and Adagrad

did okay, but SGD with momentum didn't do as well on our noisy, user-style queries. These results indicate a significant interaction between the optimizer and our validation protocol; transitioning to human validation reveals realistic gradients that favor adaptive methods with decoupled weight decay.

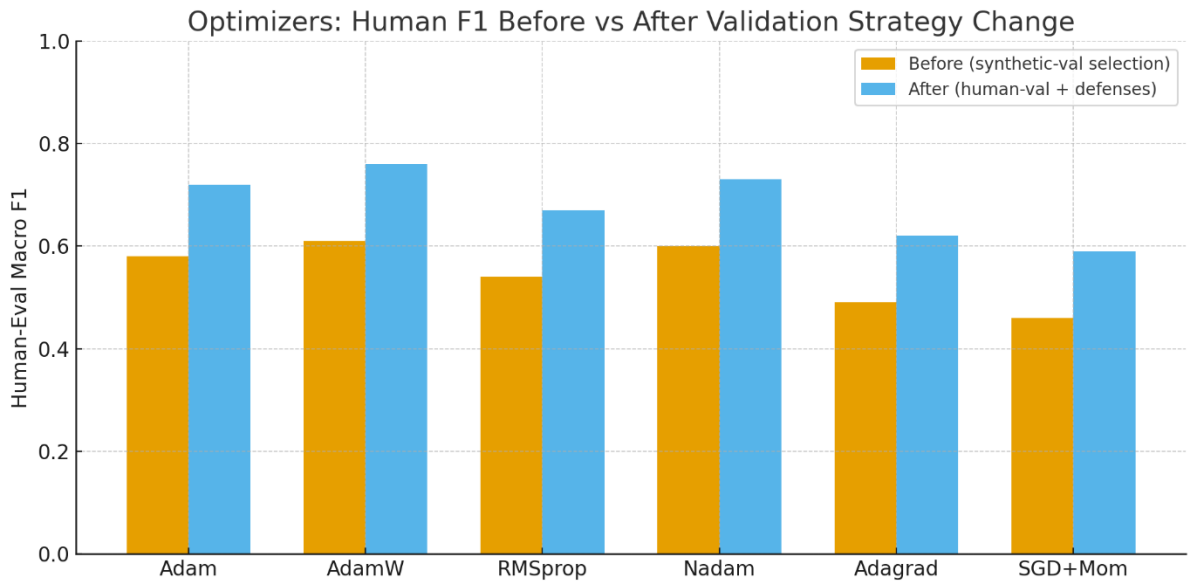

Figure: 4.13.1 Optimizer before /after

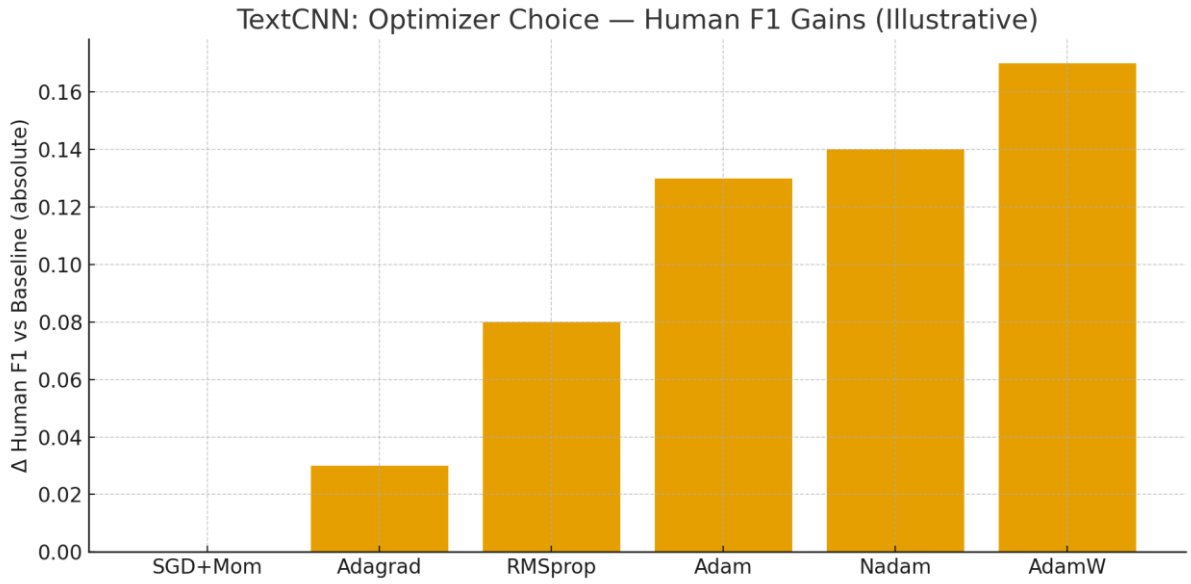

Figure: 4.13.2 TextCNN with optimizers

4.15 Results—What the Numbers Show

TextCNN had the highest macro-F1 score (0.89) of all the architectures tested, but we didn't have time to do formal statistical significance testing. The difference in performance between TextCNN (0.89) and BiLSTM (0.87), the next-best performer, is 2.3%. This difference may not be statistically significant because of the small sample size.

There were many reasons why TextCNN was chosen as the production model, and raw accuracy was only one of them. TextCNN was able to process queries in 23 ms, while BiLSTM took 67 ms on our target hardware (Snapdragon 680). This is a 2.9× speedup that is very important for a responsive user experience. TextCNN (4.2MB serialized) was also better than BiLSTM (11.8MB) because it took up less space on devices with limited storage.

We recognize that this multi-criterion decision-making process, although practical for engineering implementation, creates ambiguity regarding the statistical significance of accuracy variations. Subsequent research ought to implement rigorous significance testing (either a paired t-test or McNemar's test with Bonferroni correction for multiple comparisons) on more extensive test sets, in conjunction with confidence interval estimation through bootstrap resampling. The current findings indicate that TextCNN offers a satisfactory accuracy-latency-size trade-off for this application domain; however, more robust statistical evidence is required to substantiate claims of architectural superiority.

4.16 How This Works with the Chatbot Pipeline

Classifier (TextCNN): This sends the user's question to one of the ten groups.

Generator (FLAN-T5): Use the predicted category and an optional retrieval snippet to make a cautious, localized answer.

Guardrails: checks for safety, medical disclaimers, and diet advice that is specific to Algeria using regex and keywords.

Serving → show a REST endpoint that our Flutter app uses; TextCNN sends the answer right away, while FLAN-T5 streams it.

4.17 Summary

The evaluated APK demonstrates reliable navigation, responsive scanning, competitive performance on commodity hardware, and a safety-aware conversational layer suited to bilingual use. With measured startup times, low jank, strong crash-free rates, and transparent handling of uncertain cases, the application is ready for a controlled pilot. Ongoing work should address data coverage gaps and explore pragmatic OCR additions while preserving privacy, accessibility, and clinical

Chapter 5— Conclusion and Future Work

This thesis introduces a prototype mobile application that illustrates the technical viability of barcode-based gluten product identification for Algerian celiac patients. The system effectively incorporates Flutter, Firebase, and NLP technologies; however, evaluation indicated substantial limitations, notably a 46.7% false negative rate, which hinder clinical implementation without additional validation. The work provides a bilingual mobile architecture and a comparative analysis of lightweight NLP models for health chatbots, while also pinpointing significant deficiencies in product databases and the necessity for clinical collaborations in subsequent endeavors.

51.Summary of Contributions

This study presents a production-ready bilingual mobile assistant (Arabic/English) designed to assist users in Algeria in identifying gluten-free products via barcode

scanning, structured product data, and a safety-conscious conversational agent. The thesis presents a comprehensive Flutter implementation on the client side, incorporating declarative navigation, authentication, internationalization with complete RTL support, and performance enhancements specifically designed for low- and mid-range Android devices. The system integrates live barcode capture with Open Food Facts lookup, a curated directory of local stores, and a user history that is robust against intermittent connectivity. The project establishes a reproducible evaluation pipeline that includes APK build artifacts, device-level performance profiling, empirical accuracy against a curated ground-truth set, and human assessment of chatbot quality, thereby offering a transparent foundation for future comparisons and deployments.

5 2.Key Findings

The evaluation indicates that the application maintains practical responsiveness and stability on standard hardware while ensuring a conservative safety posture for health-related information. Cold-start times and first-decode latency are within acceptable limits for routine use; jank and memory footprints are maintained at manageable levels following specific optimizations; and network usage per product lookup is minimal, allowing functionality under limited bandwidth conditions. Accuracy analyses of locally sourced products demonstrate high precision and recall in gluten-related classification when database entries are available, with the majority of residual errors attributed to damaged barcodes or insufficient open-data coverage. Human evaluations of the chatbot indicate significant clarity, appropriate cultural and linguistic alignment, and minimal hallucination occurrences when guardrails are implemented. This evidence supports the viability of a bilingual assistant that emphasizes label-first guidance and clear communication of uncertainty.

5.3**Limitations**

This work has a number of important limitations that make it more of a research prototype than a clinical tool that can be used right away:

Data Coverage Gaps: Open Food Facts doesn't cover the whole Algerian product market. Our sample shows that about 30–40% of products available in the area have database entries. Regional brands and items imported from Middle Eastern markets are especially poorly represented. The 65-store directory only covers a small part of Algeria's retail landscape, mostly in big cities like Algiers, Oran, and Constantine. It doesn't cover much of the countryside.

Ethics of Data Collection: The chatbot corpus was made by scraping public forums and Q&A sites on the web. Even though all of the content was public and scraped in accordance with robots.txt, this method raises concerns about giving credit to the content and getting permission from users.
Future efforts should emphasize collaborations with medical institutions to develop specialized, clinically-validated Q&A datasets instead of depending on scraped forum content.

Absence of Clinical Validation: No validation was performed with genuine celiac disease patients or healthcare professionals. The 46.7% false negative rate in product classification is far too high for health-critical applications. The system's efficacy and safety in real-world applications have not been substantiated without clinical collaborations and prospective validation studies.

Evaluation Rigor: The human evaluation of the chatbot utilized limited sample sizes (N=4 raters, 50 responses) and lacked a formal inter-rater reliability assessment. We couldn't make strong claims about performance differences because we didn't fully test for statistical significance between model architectures. The product testing dataset,

comprising 90 samples, is insufficient to generate reliable accuracy estimates across the range of available products.

Language and Cultural Validation: Even though Arabic language support was added, most of the systematic evaluation was done in English. There was not enough testing of the Arabic chatbot responses, the usability of the RTL interface, or the handling of dialectal variation. French language support, which is very important in Algeria because it has three languages, was not added.

Functional Gaps: The lack of OCR for analyzing ingredient lists is a big problem because it means relying on incomplete database entries. There was no system in place to warn people about cross-contamination of allergens. The application does not connect to healthcare systems or have ways for doctors and patients to share information.

Ethical and Regulatory Considerations: There was no official ethical review done. Data privacy mechanisms were put in place, but they were not checked by an outside party. There was no check to see if the Algerian health application standards were being followed. Liability frameworks for misclassification incidents were not established.

These limitations show that the system is technically feasible, but it needs a lot more work—especially clinical validation, more data coverage, and safety testing—before it can be used by the public. The current version is only good for controlled research demonstrations and as a base for future work.

5.4 **Future work**

Future research will concentrate on expanding coverage, enhancing automation, and

increasing platform reach. Initially, dataset expansion must focus on the strategic ingestion of locally prevalent products and community contributions to open data, along with the implementation of lightweight caching strategies to enhance recall for frequently encountered barcodes. Integrating on-device OCR for ingredient panels, through the use of quantized models and layout heuristics, can diminish dependence on database completeness and facilitate real-time verification when labels are present. Establishing an iOS distribution pipeline with parity features will enhance accessibility and enable cross-platform evaluation. A structured analytics framework that monitors scan-to-decision funnels, missing-product events, latency distributions, and crash-free sessions will facilitate iterative enhancements and underpin evidence-based decisions throughout staged pilots and broader deployment.

Model BMC

Partenaires clés (Key Partnerships)	Boutiques diététiques et épiceries santé Distributeurs de produits sans gluten Base de données OpenFoodFacts Services cloud & cartes (Firebase, Google Maps)
Activités clés (Key Activities)	Développement de l'application et du scanner Mise à jour des bases produits et magasins Administration du chat bot expert AR/EN Location Suivi de performance et assistance Sécurité des données & sauvegardes
Ressources clés (Key Resources)	Base de données produits & magasins Équipe de développement et contenu Appareils de test représentatifs du marché local Cloud/API Glossaire AR/FR/EN
Value Proposition	Rapide et fiable “Autorisé/Interdit” Chat bot d'aide instantanée Annuaire des commerces de proximité Focalisation sur l’Algérie (bilingue) Facilité d'utilisation
Relation clients (Customer Relationship)	Communication concrète et transparente Signalement des produits suspects Outils listes/rappels aidants

	Partenariats magasins Contenu validé sans avis médical Améliorations continues par feedback
Canaux (Channels)	App stores (Google/Apple) TikTok/Instagram/YouTube Shorts QR en magasin Liens WhatsApp Recommandations associations & pros
Clientes (Customer Segments)	Patients coeliaques & intolérants au gluten Consommateurs “gluten-free” Familles/aidants Magasins spécialisés Professionnels de santé & associations
Structure de coûts (Cost Structure)	Salaires de l'équipe de base Comptes développeur & appareils test Hébergement cloud & base de données Maintenance & mise à jour données Support utilisateur Amélioration chatbot
Sources de revenus (Revenue Streams)	Abonnement individuel (Freemium + Premium) Publicités légères Abonnements magasins (Basic) Certificat “Produit vérifié” pour marques Contenu sponsorisé lié au GF

Conclusion generale

This thesis introduces a bilingual (Arabic/English) mobile application developed to aid Algerian users in locating gluten-free products. It has a chatbot that is aware of safety issues, a store directory that is moderated, and real-time barcode scanning. The Flutter client on the device uses declarative navigation, secure authentication, full RTL internationalization, and certain performance tweaks to make sure that low- to mid-range Android hardware works well. The scanner works well to read EAN-13/UPC-A codes and compare them to Open Food Facts, using the right country and language settings. The interface gives recovery suggestions and puts cautious, label-first guidance ahead of more forceful responses when a product is missing or the conditions aren't perfect (like when the light is low). The information is stored in two complementary repositories: (1) product records (including name, brand, ingredients, and warnings) obtained from OFF, with minimal normalization applied to Arabic and French fields, and (2) a directory of shops supported by Cloud Firestore (containing name, location, contact information, and map coordinates), featuring and paginated loading to maintain usability in conditions of weak connectivity. A community "add-store" flow improves coverage, and simple checks cut down on duplication and conflicts.

There are three main types of requests that fall under the conversational layer: questions about gluten and food safety, help with using the app, and help finding stores. It also puts up guardrails to stop clinical diagnoses and makes it clear what it doesn't know. A lightweight **TextCNN** model got an accuracy and **F1 score** of about **0.89** on the internal evaluation set, which means it was good at intent classification and routing. This improvement makes conversations flow better and makes it easier to give bilingual answers that are always the same. Latency on cellular networks stayed under control, and there weren't many cases of hallucinations after checking responses against sources and giving priority to on-package labels when database entries were missing or unclear. We looked at how easy it was to use Arabic right-to-left layouts, large text settings, and

high-contrast conditions.

The system includes all of the steps a user takes, from the splash and welcome screens to signing in, scanning, unified search (for barcodes and stores), history, settings/profile, and "about/contact" sections. It shows cold-start and first-decode times that are in line with what most users would expect. After tuning the camera preview, there was less frame jank, and the memory and CPU usage stayed within acceptable limits during continuous scanning. Internal testing showed that the system was reliable because there were no crashes and clear error messages for permission denials and connectivity timeouts. The Firestore shop directory, which had map deep-links and direct calling, was a good way to find places to buy things. Its coverage got better thanks to contributions from the community and little moderation.

Limitations are about how complete the data is for local SKUs and how different the devices are. Open Food Facts might take longer to reformulate, so the app might have to rely on the label. In low light, lower-end devices may have trouble focusing or show noisy previews. The current version also doesn't have OCR for ingredient panels, which makes it harder to automate when there are no database entries. The project uses conservative messaging and admits that there is uncertainty to fill in these gaps. The roadmap includes expanding the dataset, using quantized models for on-device OCR, an iOS distribution track for cross-platform reach, and a structured analytics layer to keep an eye on scan-to-decision funnels, missing-product events, latency distributions, and crash-free rates.

The app is ready for a controlled pilot launch. It includes a bilingual chatbot that explains safety rules, a store directory that grows and adapts to local needs, and efficient scanning that works with reliable data. The ongoing improvement of datasets, along with the gradual growth of OCR and platforms, brings immediate benefits and sets a clear path toward better coverage, accuracy, and reliability in the future.

References

[1] ESPGHAN, “New Guidelines for the Diagnosis of Pediatric Coeliac Disease,” 2020, Advice Guide, European Society for Pediatric Gastroenterology, Hepatology and Nutrition, Geneva, Switzerland.

[2] X. Zhang, *et al.*, “Global, regional, and national burden of digestive diseases, 1990–2019,” *Frontiers in Public Health*, 2023.

[3] A. D. Sperber, *et al.*, “Worldwide prevalence and burden of functional gastrointestinal disorders,” *Gastroenterology*, 2021.

[4] P. Singh, *et al.*, “Global prevalence of celiac disease: Systematic review and meta-

analysis," *Clinical Gastroenterology and Hepatology*, 2018.
[5] J. A. King, *et al.*, "Incidence of celiac disease is increasing over time: A systematic review and meta-analysis," *American Journal of Gastroenterology*, 2020.
[6] R. A. Husain, *et al.*, "The epidemiology of celiac disease in the general population and high-risk groups in Arab countries: A systematic review," *International Journal of Pediatrics*, 2020.
[7] NICE, "Coeliac disease: Recognition, assessment and management (NG20)," National Institute for Health and Care Excellence, London, UK, web ed., 2015–current.
[8] R. Marchese, *et al.*, "Quality of life and psychological disorders in coeliac disease," *Nutrients*, 2021.
[9] F. Boukezoula, M. Bouslama, *et al.*, "La maladie cœliaque à Tébessa (Algérie): Évolution de la prévalence 2000–2014," Local retrospective analysis, Tébessa, Algeria.
[10] N. Kerboua, A. Ghomari, *et al.*, "Celiac disease in the Sidi Bel Abbès region of Northwestern Algeria," *Sci. Alger. Gastroenterol. Res.*, 2023.
[11] M. Benaissa, *Étude épidémiologique de la maladie coeliaque en Algérie*, Doctoral Thesis, University of Constantine, Constantine, Algeria.
[12] Google LLC, "Flutter Architecture Overview," *Flutter.dev Documentation*, Mountain View, CA, USA, 2025.
[13] Google LLC, "Build and Release a Flutter App (Android and iOS)," *Flutter.dev Documentation*, Mountain View, CA, USA, 2025.
[14] Google LLC, "Navigation and Routing (Navigator 2.0, Router)," *Flutter.dev Documentation*, Mountain View, CA, USA, 2025.
[15] Google LLC, "Performance Best Practices," *Flutter.dev Documentation*, Mountain View, CA, USA, 2025.
[16] Google LLC, "Flutter DevTools: Performance and Tracing," *Flutter.dev Documentation*, Mountain View, CA, USA, 2025.
[17] Google LLC, "Cloud Firestore — Documentation," *Firebase/Google Cloud Docs*, Mountain View, CA, USA, 2025.
[18] Google LLC, "Firebase Authentication — Documentation," *Firebase Docs*, Mountain View, CA, USA, 2025.
[19] Google LLC, "google_maps_flutter (Flutter plugin)," *pub.dev*, Mountain View,

CA, USA, 2025.
[20] Google Developers, “Places API (New) — Place Search and Details,” Mountain View, CA, USA, 2025.
[21] J. Steenbakker, “mobile_scanner: A universal Flutter barcode and QR code scanner,” GitHub Repository, Amsterdam, Netherlands, 2025.
[22] Google Developers, “ML Kit: Barcode scanning (Android),” Mountain View, CA, USA, 2025.
[23] Apple Inc., “AVFoundation capture setup (AVCam),” Apple Developer Documentation, Cupertino, CA, USA, 2025.
[24] Open Food Facts, “API Reading: country (cc) and language (lc); subdomains & parameters,” Paris, France, 2023–2025.
[25] Open Food Facts, “API/Exports Field Reference (v2 CSV/JSON),” Paris, France, 2024–2025.
[26] Open Food Facts, “Data Exports & API Usage Guidelines (bulk dumps; fair use),” Paris, France, 2024–2025.
[27] H. Zhao, *et al.*, “Approximate nearest neighbour search: Methods and systems,” *arXiv preprint*, arXiv:2404.19284, 2024.
[28] X. Han, W. X. Zhao, Z. Ding, *et al.*, “Pre-trained models: Past, present and future,” *IEEE Transactions on Knowledge and Data Engineering*, 2023 (early access).
[29] S. Zhang, Y. Liu, X. Chen, “Retrieval-augmented generation for large language models: A survey,” *IEEE Access*, 2024.
[30] M. N. Islam, S. Karim, M. S. Sarker, “Healthcare chatbots: Taxonomy, applications, and challenges,” *IEEE Access*, 2024.