



ECOLE NATIONALE SUPÉRIEURE D'INFORMATIQUE ET D'ANALYSE DES
SYSTÈMES - RABAT

Major : Software Engineering

Second-Year Internship

Design and Development of a Multilingual Mobile/Web Application for Migrant Assistance in Morocco

Prepared by :

GHOUZALI AYOUB

Presented before the Jury :

Supervised by :

Mr. BEN ABDERRAHMANE
ILYASS

Pr. W. ETTAZI
Pr. M. FAKHOURI AMR

Academic year 2024/2025

Acknowledgments

I would like to express my sincere gratitude to all the people who contributed, directly or indirectly, to the completion of this internship and the preparation of this report. My deepest thanks go to my supervisor, **Mr. BEN ABDERRAHMANE Ilyass**, for his warm welcome, continuous guidance, and valuable advice throughout this experience. My appreciation extends as well to my professors at **ENSIAS** for the quality of their teaching and their continuous support during this academic year. Their knowledge and pedagogy have been a great source of learning and inspiration.

Thank you all.

Abstract :

This report presents the design and development of the *Migrant-e-s Help* application, carried out during a second-year internship at **DTR**. The project was conducted in collaboration with the Moroccan non-profit organization **Pionniers du Changement**, which works to promote human rights and support migrants in Morocco.

The objective of the internship was to create a multilingual mobile application (Arabic, French, English) that centralizes essential services, downloadable guides, and practical information for migrants, while enabling reporting of issues encountered. The system also integrates an AI assistant powered by a Retrieval-Augmented Generation (RAG) pipeline connected to the NGO's own content. A temporary section was added for visitors during the Africa Cup of Nations (CAN) : match schedules, stadium information, and city highlights.

The application was developed with **Flutter** for the frontend and **Supabase** as the backend platform (PostgreSQL, Auth, Storage, and Row Level Security). It also integrates **Mapbox** for geolocation, **OneSignal** for push notifications, and the **OpenAI API** for AI-powered assistance. The resulting solution offers an accessible, secure, and scalable digital tool that contributes to improving access to information and services for migrants in Morocco.

Keywords : Flutter, Supabase, Mapbox, Artificial Intelligence, RAG

Résumé :

Ce rapport présente la conception et le développement de l'application *Migrant-e-s Help*, réalisés dans le cadre d'un stage de deuxième année au sein de l'entreprise DTR, en collaboration avec l'association marocaine **Pionniers du Changement**, œuvrant dans les domaines des droits humains et de la migration.

L'objectif du projet est de concevoir une application mobile multilingue (arabe, français, anglais) destinée à faciliter l'accès des migrants aux services, guides et informations utiles, tout en leur permettant de signaler les difficultés rencontrées. L'application intègre également un assistant intelligent, basé sur une approche RAG (Retrieval-Augmented Generation) connectée aux ressources internes de l'association. Une section temporaire a été ajoutée pour les visiteurs de la Coupe d'Afrique des Nations : calendrier des matchs, informations sur les stades et lieux à visiter.

Développée avec le framework **Flutter** et la plateforme **Supabase** (PostgreSQL, Auth, Storage et RLS), l'application comprend des fonctionnalités de géolocalisation via **Mapbox**, des notifications push avec **OneSignal**, et un module d'assistance alimenté par l'**API OpenAI**. Le résultat final constitue un outil numérique fiable, inclusif et évolutif, contribuant à renforcer l'accès à l'information et aux services pour les migrants au Maroc.

Mots clés : Flutter, Supabase, Mapbox, Intelligence Artificielle, RAG

List of Abbreviations

Abbreviation	Meaning
API	Application Programming Interface
CRUD	Create, Read, Update, Delete
MVVM	Model–View–ViewModel
NGO	Non-Governmental Organization
RAG	Retrieval-Augmented Generation
RLS	Row-Level Security
SDK	Software Development Kit
SQL	Structured Query Language

Table des figures

1.1	GANTT diagram	20
2.2	Use Case Diagram of the <i>Migrant-e-s Help</i> system	29
2.3	Class Diagram of the core entities in the <i>Migrant-e-s Help</i> database	30
2.4	Sequence diagram of the RAG-based AI assistant interaction . . .	31
2.5	Sequence diagram of the report submission and validation process	32
2.6	Global architecture of the <i>Migrant-e-s Help</i> ecosystem illustrating the client–server model, backend components, and external service integrations.	33
2.7	Flutter Clean MVVM architecture with Riverpod.	35
4.8	Home page interfaces showing main navigation cards and multilin- gual layout.	42
4.9	Services module showing categorized service list and detailed ser- vice information.	43
4.10	AI Chat Assistant module demonstrating multilingual and contex- tual conversation.	44
4.11	News and article pages showing the list of recent updates and de- tailed article view.	45
4.12	Favorites module displaying saved items and quick access shortcuts.	46
4.13	Map module showing interactive service markers and offline map view.	47
4.14	Report module showing submission form and success confirmation screen.	48
4.15	Seasonal CAN section : matches, stadium details, and city highlights.	49
4.16	Admin web portal overview showing dashboard layout and content management interface.	50
4.17	Admin interfaces for managing services, reviewing reports, and pu- blishing news articles.	50

table of contents

General Introduction	10
Chapter I : Environment and Context of the Internship	13
1.1 Organization Presentation	14
1.2 Project Presentation	16
1.2.1 Context and Problem Statement	16
1.2.2 Project Objectives	17
1.2.3 Proposed Solution : The “Migrant-e-s Help” Mobile Appli- cation	18
1.3 Project Planning and Working Methodology	18
1.4 Conclusion	20
Chapter II : Analysis and Design	22
2.1 Introduction	23
2.2 Requirements Analysis	23
2.2.1 Identification of Actors and Use Cases	23
2.2.2 Functional Requirements	25
2.3 System Modeling	27
2.3.1 Use Case Diagram	27
2.3.2 Sequence Diagram – AI Assistant Interaction	30
2.3.3 Sequence Diagram – Reporting and Review	31
2.4 Application Architecture Design	32
2.4.1 Global Architecture (Client–Server Model)	32
2.4.2 Flutter Architecture (MVVM with Riverpod)	33
2.4.3 Integration of External Services (Supabase, Mapbox, One- Signal, OpenAI RAG)	35
2.5 Conclusion	37

Chapter III : Implementation and Technical Development	39
4.1 Development Environment	40
4.1.1 Software and Tools Used	40
4.2 Main Modules and Obtained Results	41
4.3 Conclusion	50
Conclusion générale	51

General Introduction

In today's rapidly evolving digital world, mobile technologies have become essential tools for improving access to information and services. Governments, organizations, and associations are increasingly adopting digital platforms to facilitate communication and support social inclusion. Within this context, the integration of technology into humanitarian and social fields represents a promising approach to solving real-world challenges.

The internship described in this report was carried out within the framework of developing a mobile application named “**Migrant-e-s Help**”, whose primary objective is to provide **migrants and refugees in Morocco** with simplified access to reliable information and local services. The application is designed to offer multilingual support, service search features, offline accessibility, and interactive functionalities that promote inclusion and autonomy for migrants within the Moroccan territory.

The project addresses a real societal problem : the **difficulty migrants face in accessing verified and up-to-date information** related to administrative procedures, legal rights, health services, education, and local opportunities. The *Migrant Help* app seeks to centralize this information in a user-friendly, accessible, and intelligent interface that can be used even without an internet connection.

To achieve these objectives, the project was developed using **Flutter**, a cross-platform framework developed by Google that enables the creation of performant

and visually consistent mobile applications for Android and iOS from a single codebase. The backend is powered by **Supabase**, that provides an integrated PostgreSQL database, authentication system, and file storage service. Additional technologies such as **Mapbox SDK** for interactive and offline maps, **OneSignal** for push notifications, **Hive** for offline data caching, and the **OPENAI API** for the AI-powered assistant were also integrated to ensure a rich and reliable user experience.

The development process followed an **Agile methodology**, allowing iterative improvements and continuous collaboration with the project supervisor. Each sprint focused on a functional module — including service catalog, guides, reporting, notifications, and AI assistant to ensure modular and scalable development.

This report presents the different stages of the internship and the realization of the project :

- The **first chapter** introduces the context of the internship, the host organization, and the general description of the project.
- The **second chapter** focuses on the analysis and design phases, including requirements identification, system modeling, and interface design.
- The **third chapter** details the technical implementation, covering the development environment, the main modules built, and the results obtained.

Finally, the report concludes with a reflection on the personal and professional skills acquired during the internship, the challenges encountered, and the potential directions for future improvements of the *Migrant-e-s Help* project.

Chapter I

Environment and Context of the Internship

Introduction

This chapter introduces the professional environment in which the internship took place and the overall context of the project. It presents the host company, DTR, its mission, and its role. Finally, it outlines the project objectives, methodology, and planning that guided the implementation phase.

1.1 Organization Presentation

DTR is a multidisciplinary company that transforms ideas into impactful realities. Founded with the initial goal of helping small businesses grow through affordable and effective marketing strategies, DTR has progressively expanded its expertise to include *information technology solutions* and *innovative digital products*. Today, DTR serves clients across multiple industries by combining branding, software engineering, and data-driven growth practices within a single, coherent offering.

Expertise by Sector

DTR supports organizations at different stages of maturity—startups, SMEs, non-profits, and enterprises—by tailoring strategy and execution to each sector :

- **Retail** : Customer engagement and loyalty through smart technologies and data-driven campaigns.

-
- **E-Commerce** : Conversion-oriented web platforms and performance marketing for sustainable growth.
 - **Education** : Interactive solutions that enhance learning experiences and institutional communication.
 - **Startups** : Go-to-market acceleration, brand positioning, and scalable digital architectures.
 - **Corporations** : Enterprise-grade IT solutions that streamline operations and improve reliability.
 - **Non-Profit Organizations** : Digital strategies and tools that amplify mission impact and community reach.

Services

- **Branding Services** : Brand strategy, visual identity, and communication systems.
- **Ready-to-Go Brands** : Pre-designed, customizable brand kits to accelerate launch.
- **Digital Marketing** : Data-driven acquisition, content, SEO/SEM, and analytics.
- **Web Design & Development** : Modern, responsive websites and web apps focused on usability and performance.
- **AI-Powered Solutions** : Intelligent assistants, automation, and analytics leveraging state-of-the-art models.
- **IT Infrastructure** : Robust cloud architectures, integration, security

hardening, and ongoing support.

1.2 Project Presentation

1.2.1 Context and Problem Statement

Migration in Morocco has significantly increased over the last decade, with the country becoming both a point of transit and a host territory for migrants and refugees from various regions of Africa and the Middle East. Despite the efforts of national authorities and NGOs, many migrants still face major challenges in obtaining reliable, updated, and accessible information related to administrative procedures, health services, education, employment, and legal assistance.

Most available information remains scattered across different institutional websites, social media pages, and printed materials, which often leads to misinformation, communication barriers, and unnecessary travel. This lack of a centralized and user-friendly digital platform creates a real accessibility gap for vulnerable communities, especially for those with limited internet connectivity or language barriers.

In this context, the project “**Migrant-e-s Help**” was conceived as a technological solution aiming to centralize essential resources and facilitate communication between migrants and support organizations through a mobile platform. The application provides a simple and interactive interface that helps users quickly find nearby services, consult practical guides, and receive updates and announcements in real time.

This project was developed as part of my **internship at DTR Company**, in collaboration with the client organization **Pionniers du Changement**, a Moroccan non-profit association that will operate and manage the application once deployed. The objective of this collaboration was to create a digital tool that supports the association's mission to promote human rights and social inclusion of migrants across Morocco.

1.2.2 Project Objectives

The main objective of the project is to design and develop a **cross-platform mobile application** that simplifies access to information and services for migrants living in Morocco. More specifically, the project aims to :

- Provide a multilingual interface (Arabic, French, and English) with full right-to-left language support.
- Allow users to search for nearby services by category and location.
- Offer downloadable and offline-accessible guides containing practical information.
- Enable reporting of incidents or cases requiring assistance.
- Display interactive maps through the Mapbox SDK.
- Send push notifications about urgent news, events, or administrative changes.
- Integrate an intelligent assistant powered by the OpenAI API and enhanced with a Retrieval-Augmented Generation (RAG) system, allowing it to provide personalized and context-aware answers based on the application's internal data (guides, services, and news).

-
- Ensure data security, offline usage, and scalability through a robust backend architecture.
 - Seasonal Visitor Section (CAN) : Provide a temporary module for tournament visitors (match schedule, stadiums, places to visit), enabled only during the CAN period and planned for removal afterward.

1.2.3 Proposed Solution : The “Migrant-e-s Help” Mobile Application

The proposed solution is a mobile application developed with the **Flutter** framework, chosen for its ability to build high-performance cross-platform applications from a single codebase. The app communicates with a **Supabase** backend, which handles authentication, data storage, and secure API access using PostgreSQL and Row Level Security (RLS) policies.

1.3 Project Planning and Working Methodology

The development of the **Migrant-e-s Help** application followed an **Agile** methodology, allowing for iterative and incremental progress throughout the internship period, which extended from **June 20 to August 31, 2025**. This approach was chosen to ensure flexibility, continuous feedback, and frequent functional deliveries, which were essential given the evolving requirements of the client association **Pionniers du Changement**.

Project Management Approach

The project was organized around short and well-defined **sprints**, each lasting approximately two weeks. Each sprint aimed to deliver a concrete set of features that could be reviewed and tested before moving on to the next phase. Daily progress was tracked using the **Kanban method**, which provided a clear visualization of tasks through “To Do”, “In Progress”, and “Done” columns.

At the beginning of each sprint, objectives were discussed with the project supervisor from **DTR Company** and aligned with the client’s needs. This iterative process made it possible to quickly integrate new requirements, refine priorities, and ensure that the application evolved according to user expectations.

Task Scheduling and Gantt Chart

Given the internship duration of approximately ten weeks, the project was divided into five main sprints, each corresponding to a key development phase :

- **Sprint 1 (June 20 – June 30)** : Environment setup, project initialization (Flutter, Supabase, Mapbox), and UI design prototype creation.
- **Sprint 2 (July 1 – July 14)** : Implementation of admin authentication and multilingual interface (Arabic, French, English) with RTL support.
- **Sprint 3 (July 15 – July 28)** : Development of core modules : service directory, guides management, and offline data caching with Hive.
- **Sprint 4 (July 29 – August 14)** : Integration of Mapbox SDK for maps, reporting system, and push notifications using OneSignal.

-
- **Sprint 5 (August 15 – August 31) :** Integration of the OpenAI-based AI assistant with RAG personalization, global testing, bug fixing, and preparation for deployment.

Each sprint concluded with a short retrospective meeting, during which achievements and potential improvements were reviewed.

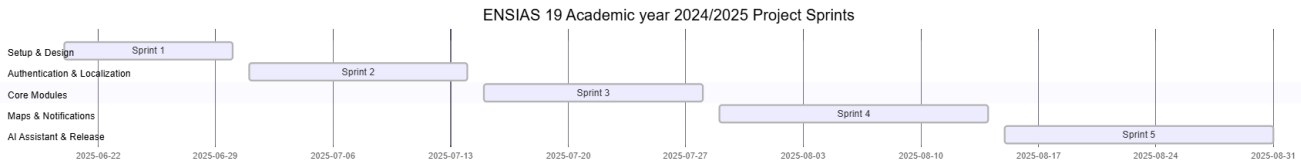


FIGURE 1.1 – GANTT diagram

1.4 Conclusion

This first chapter presented the overall framework of the internship and the context in which the *Migrant-e-s Help* application was developed. It outlined the social and technological motivations behind the project, the objectives pursued, and the methodological choices adopted.

Through a structured Agile workflow supported by collaborative tools, the project advanced iteratively and efficiently. Each sprint contributed to building a

functional component of the application while maintaining adaptability to client feedback.

This chapter thus laid the foundation for understanding the project's environment and execution methodology. The next chapter will focus on the **analysis and design phase**, detailing the functional requirements, system modeling, and architectural design of the *Migrant-e-s Help* application.

Chapter II

Analysis and Design

2.1 Introduction

After defining the general context, objectives, and methodology of the internship, this second chapter focuses on the analytical and design aspects of the *Migrant-e-s Help* project. The analysis phase represents a crucial step in the software engineering process, as it transforms the client's expressed needs into precise functional and technical specifications.

The design phase then aimed to model the system's core components through diagrams and data structures, laying the groundwork for the implementation stage.

Thus, this chapter details the essential steps of the analysis and design process, beginning with the identification of actors and requirements, followed by the modeling of system interactions, database design, and architectural conception.

2.2 Requirements Analysis

2.2.1 Identification of Actors and Use Cases

The first step in the analysis phase consisted of identifying the key actors who interact with the *Migrant-e-s Help* system. Each actor contributes to the proper functioning of the platform, which is composed of a public mobile application and an administrative dashboard.

The identified actors are summarized in Table 2.1.

Based on these actors, the following major use cases were identified :

Actor	Description and Role
Public User (Migrant or Visitor)	Any user can freely access the mobile application without authentication. They can search for services, read guides, view news, consult the interactive map, submit reports, and interact with the AI assistant. The interface supports Arabic, French, and English, and remains functional in offline mode.
Administrator (NGO Staff)	Member of the Pionniers du Changement association who accesses the secure web dashboard. Uses Supabase authentication to manage content such as services, guides, news, and reports. Also reviews reports submitted through the mobile app.

TABLE 2.1 – Main actors of the *Migrant-e-s Help* system

- **Information access** : users can explore services, read guides, and view news without login.
- **Search and filtering** : users can find nearby services by category or keyword, using the integrated Mapbox map.
- **Offline usage** : content (guides, services, and news) is cached locally via Hive for use without internet.
- **Reporting** : users can submit incident reports with text.
- **Notifications** : OneSignal sends updates and announcements to all users.
- **AI assistant** : users can ask questions and receive contextual responses generated by the RAG + OpenAI model.
- **Administrative management** : the NGO administrator logs into the web dashboard (Supabase-authenticated) to add, edit, or delete content and review reports.

2.2.2 Functional Requirements

Functional requirements describe what the system must do to meet the needs of its users. In the case of the *Migrant-e-s Help* project, the application ecosystem consists of two main components : the **public mobile application** accessible to all users without authentication, and the **admin web dashboard** restricted to authorized members of the **Pionniers du Changement** association.

The following list summarizes the essential functional requirements identified during the analysis phase.

A. Mobile Application

- **FR1 – Service Exploration** : The user can browse and filter available services by category, city, or proximity. Each service includes details such as name, description, contact information, and location on the map.
- **FR2 – Multilingual Interface** : The application supports Arabic (RTL), French, and English. Users can switch between languages seamlessly from the app bar.
- **FR3 – Guide Consultation** : The user can access informative guides provided by the NGO, download them for offline reading.
- **FR4 – News Feed** : The application displays the latest news and announcements from the organization, retrieved dynamically from the Supabase database.
- **FR5 – Map Visualization** : The system shows nearby services, markers in an interactive map.

-
- **FR6 – Reporting** : The user can report an incident or situation, attaching a description, and optional location data. Reports are securely stored in the Supabase database.
 - **FR7 – Push Notifications** : The application receives real-time notifications about important events or updates.
 - **FR8 – Offline Access** : The system stores data locally, allowing users to access services, guides, and news without an internet connection.
 - **FR9 – AI Assistant** : The user can interact with an intelligent assistant to obtain personalized and context-aware information derived from the app’s internal data.
 - **FR10 – CAN Visitor Section (Seasonal)** : Display match fixtures, stadium details, and recommended places to visit during the Africa Cup of Nations period. The module must be activatable/deactivatable via configuration without impacting core features.

B. Admin Dashboard

- **FR11 – Authentication and Access Control** : Admin users must log in through Supabase authentication to access the dashboard.
- **FR12 – Service Management** : The admin can add, edit, or delete service entries, including their geographic coordinates, category, and contact details.
- **FR13 – Guide Management** : The admin can upload, update, or delete guides in PDF format, assign them to specific categories, and provide

translations.

- **FR14 – News Management** : The admin can create and publish news posts with titles, descriptions, and optional images.
- **FR15 – Report Review** : The admin can view, validate, or archive reports submitted by users through the mobile app.
- **FR16 – Notifications Management** : The admin can compose and send broadcast notifications to all app users through the integrated One-Signal service.
- **FR17 – Data Synchronization** : All content updates made via the dashboard are automatically reflected in the mobile app via the Supabase backend.

2.3 System Modeling

The modeling phase aimed to represent the functional structure and dynamic behavior of the *Migrant-e-s Help* system in a formal and visual way. Using the UML language, several diagrams were created to illustrate the main components of the application, their relationships, and the interactions between users, the system, and external services.

2.3.1 Use Case Diagram

Figure 2.2 presents the global use case diagram of the *Migrant-e-s Help* system. It highlights the distinction between the two main environments of the

solution : the **public mobile application** (open to all users without authentication) and the **secure admin dashboard** reserved for NGO staff members.

Migrants and general users interact with the public app to explore services, read guides, consult news, visualize maps, submit reports, receive notifications, and communicate with the AI assistant. Meanwhile, NGO administrators authenticate through Supabase to manage all content and moderate user reports. External services such as **Mapbox** and **OneSignal** are also shown as actors interacting with specific use cases.

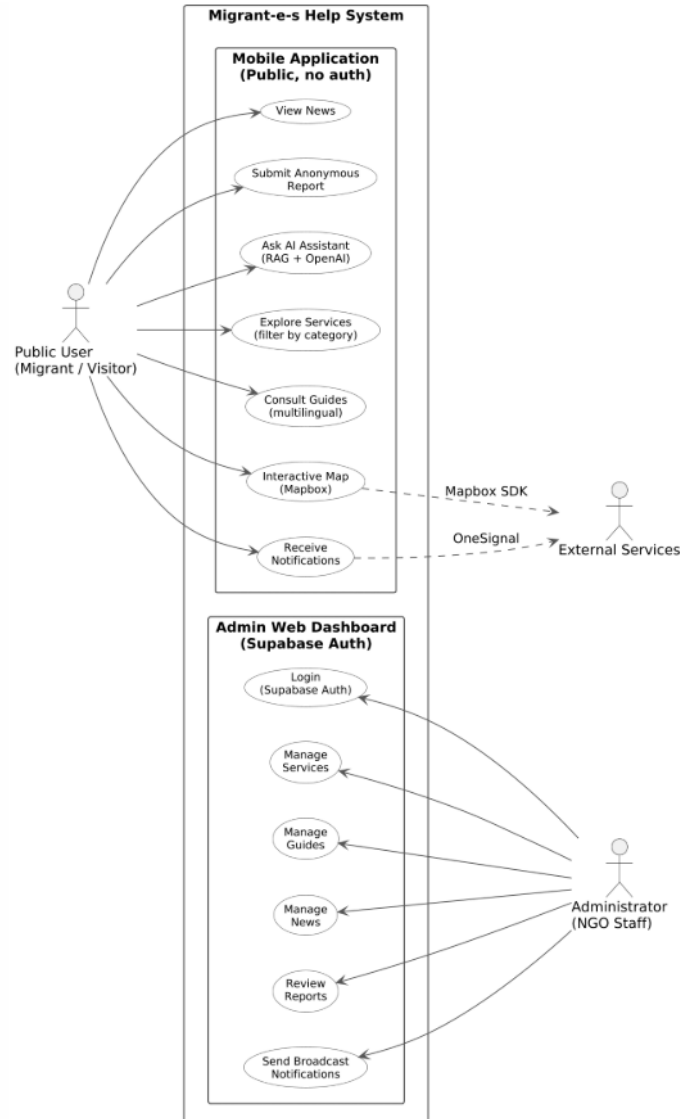


FIGURE 2.2 – Use Case Diagram of the *Migrant-e-s Help* system

Figure 2.3 depicts the main entities that compose the system’s data model and their relationships. The diagram reflects the structure implemented in the **Supabase PostgreSQL** database.

The central entities include *Service*, *Guide*, *News*, and *Report*. Each service is categorized and may be linked to one or more reports. The *AdminUser* class represents authenticated NGO staff members who manage these entities through the web dashboard. This diagram ensures that the logical model respects norma-

lization, integrity, and multilingual support.

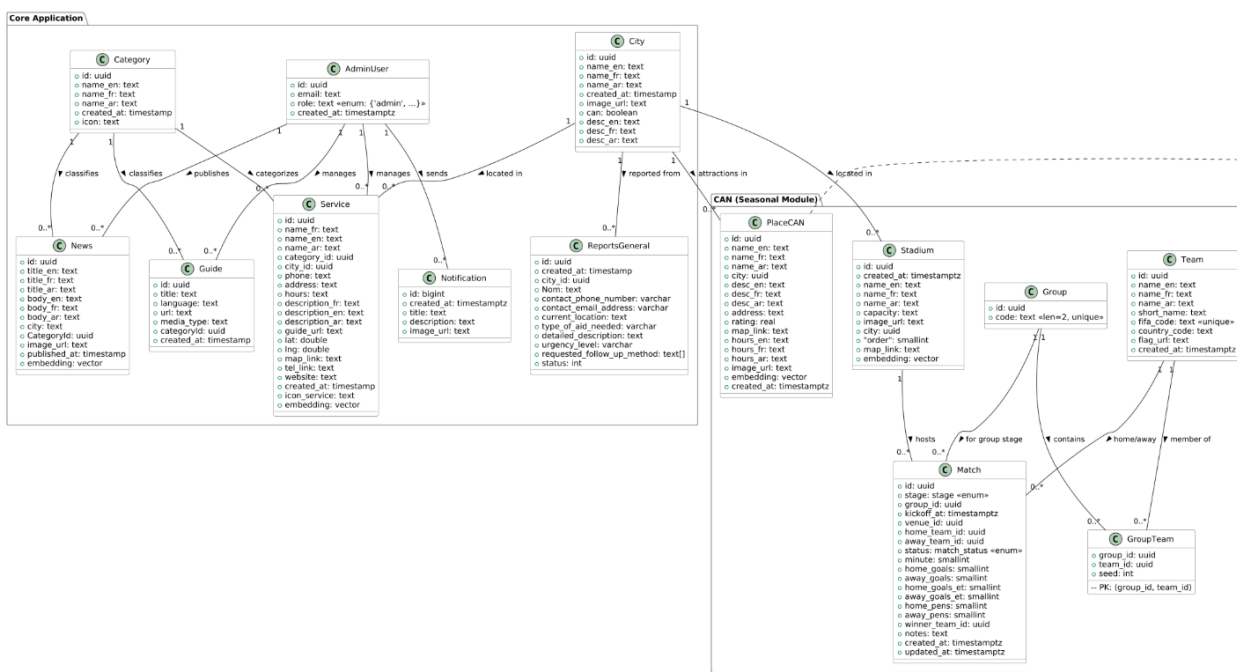


FIGURE 2.3 – Class Diagram of the core entities in the *Migrant-e-s Help* database

2.3.2 Sequence Diagram – AI Assistant Interaction

Figure 2.4 illustrates the sequence of interactions between a user and the AI assistant integrated into the application. When the user asks a question, the query passes through a **RAG orchestrator**, which retrieves the most relevant information from the embedded local dataset (guides, services, and news) before calling the **OpenAI API** to generate a contextualized answer. The resulting message is then displayed in the app interface with multilingual support and references to the underlying data. This process ensures that answers are both accurate and grounded in verified NGO content.

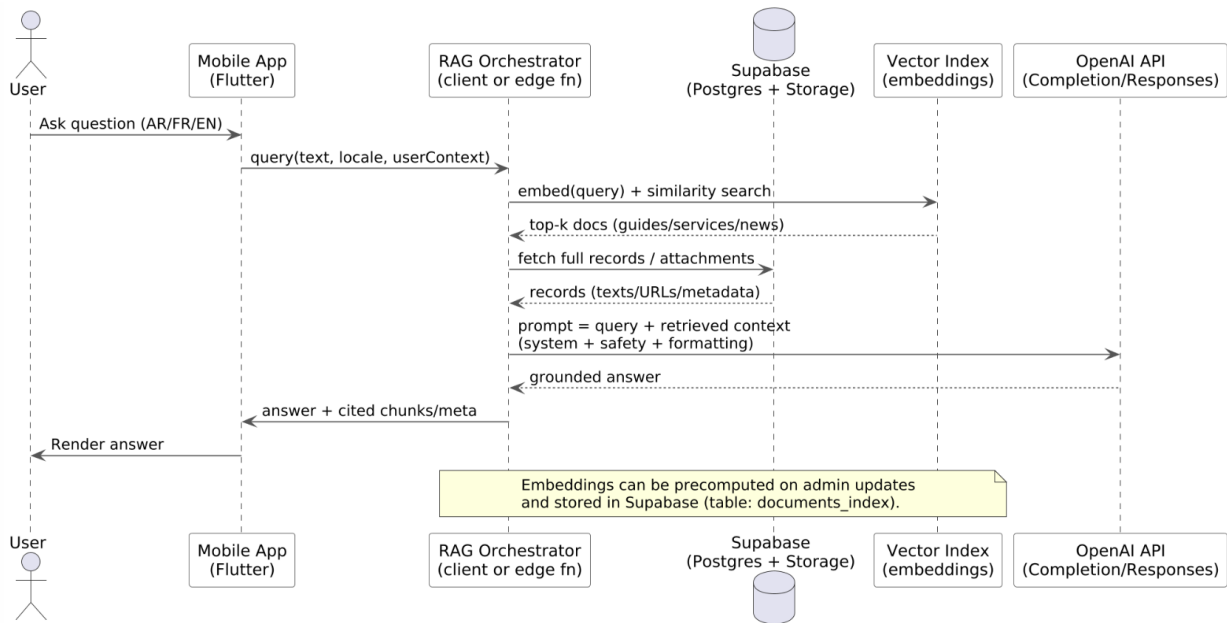


FIGURE 2.4 – Sequence diagram of the RAG-based AI assistant interaction

2.3.3 Sequence Diagram – Reporting and Review

Figure 2.5 represents the workflow of submitting and managing a report. The user fills out the report form in the mobile app. The information is securely stored in Supabase, and a notification is sent to the NGO administrators through OneSignal. Once logged into the dashboard, the administrator reviews the report, changes its status (e.g., *in review*, *resolved*, or *archived*), and the updated state is synchronized across the system.

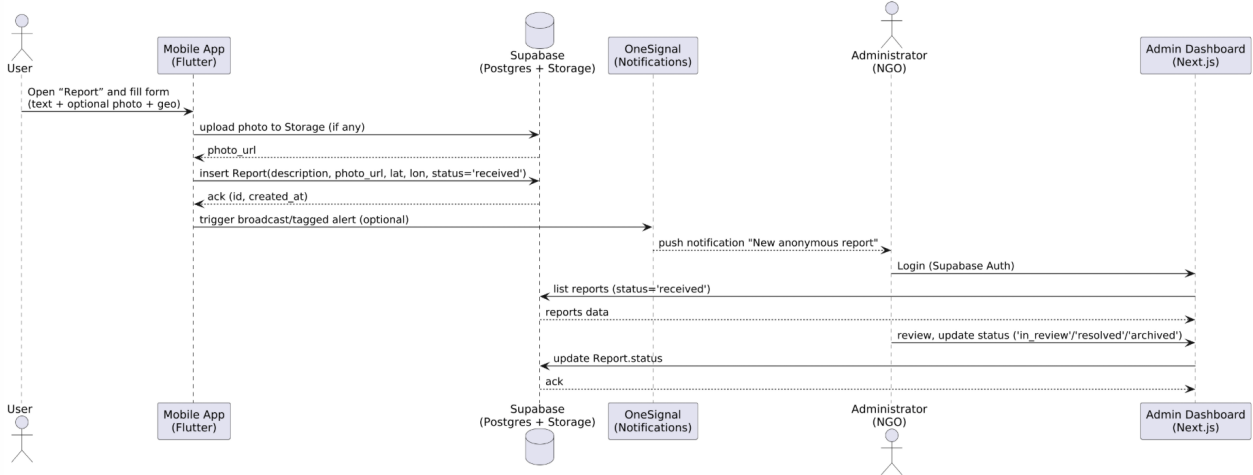


FIGURE 2.5 – Sequence diagram of the report submission and validation process

2.4 Application Architecture Design

2.4.1 Global Architecture (Client–Server Model)

The *Migrant-e-s Help* ecosystem adopts a classic **client–server** architecture with a **public mobile client** and a **secure admin web dashboard** consuming the same backend services.

- **Mobile Client** : reads published content (services, guides, news), submits reports, receives push notifications, displays offline maps, and queries the AI assistant. No end-user authentication is required.
- **Admin Dashboard (web, Supabase Auth)** : authenticated NGO staff manage content and validate reports. All writes pass through RLS-protected endpoints.
- **Backend (Supabase)** : PostgreSQL database with Row Level Security (RLS), REST/Realtime APIs, Storage for media, and SQL functions for consistency. Data is normalized and timestamped for audit and sync.

- **CDN/Storage** : Supabase Storage hosts PDFs/images ; public, cached reads from the app ; admin writes are authenticated.
- **RAG Services** : a lightweight *documents_index* table (text chunks + embeddings) enables semantic retrieval ; the app (or an edge function) composes prompts to the OpenAI API.
- **Third-party integrations** : Mapbox SDK and OneSignal (push).

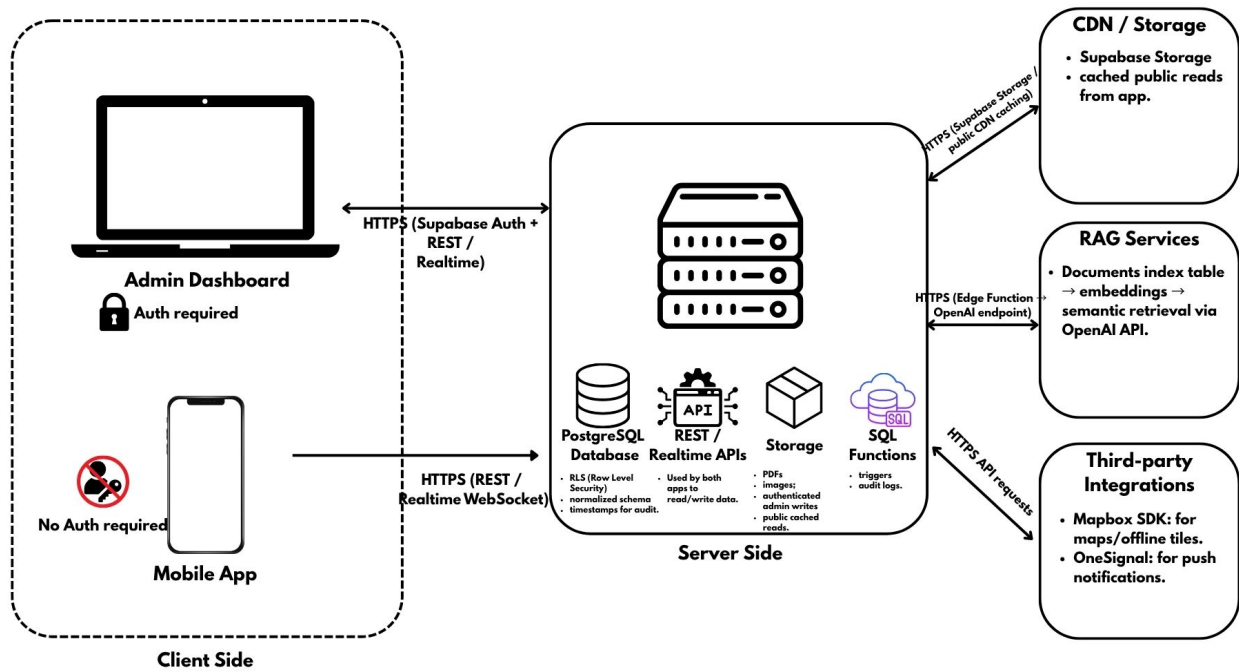


FIGURE 2.6 – Global architecture of the *Migrant-e-s Help* ecosystem illustrating the client-server model, backend components, and external service integrations.

2.4.2 Flutter Architecture (MVVM with Riverpod)

On the client side, the app follows a **Clean MVVM** layering and uses **Riverpod** for dependency injection and state management. The goal is to isolate UI concerns from business logic, simplify testing, and guarantee offline-first behavior.

Layering

- **Presentation (Views & Widgets)** : screens, responsive layout, i18n (AR/FR/EN with RTL), accessibility (contrast, scalable text). Views are *stateless* where possible ; state is provided via Riverpod.
- **ViewModels (State Notifiers)** : orchestrate use cases, expose immutable UI state (loading/data/error), debounce search, paginate lists, and coordinate online/offline strategies.
- **Domain (Use Cases & Models)** : service objects encapsulate operations such as *FetchServices*, *SearchGuides*, *SubmitReport*, *AskAssistant*. Entities are pure Dart models.
- **Data (Repositories)** : each module exposes a repository interface (*ServicesRepository*, *GuidesRepository*, *NewsRepository*, *ReportsRepository*, *RagRepository*). Implementations combine :
 - *Remote Data Source* (Supabase REST/SQL, Storage, Realtime),
 - *Local Data Source* (Hive boxes for services/guides/news, attachment cache),
 - a *Sync Policy* (stale-while-revalidate, backoff, conflict-free inserts for reports).

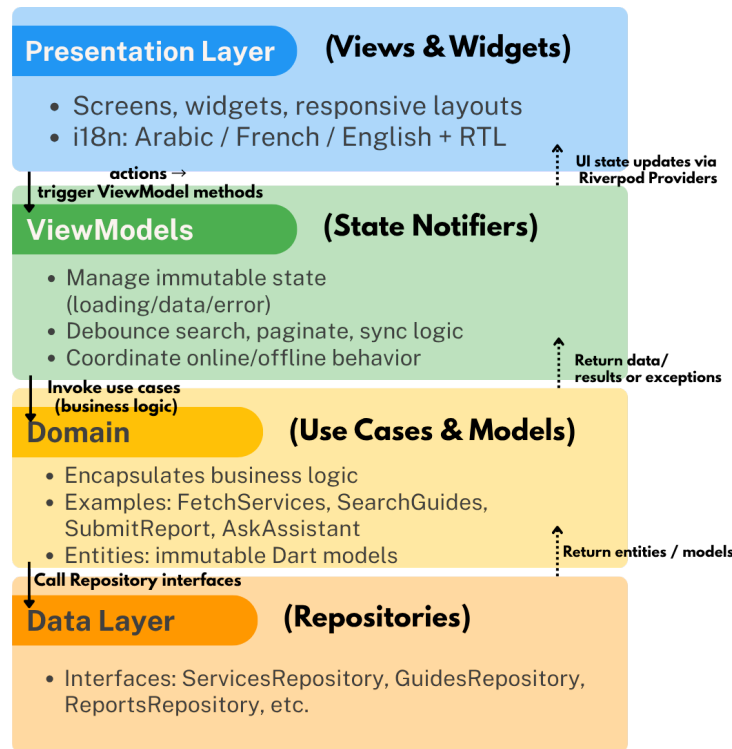


FIGURE 2.7 – Flutter Clean MVVM architecture with Riverpod.

2.4.3 Integration of External Services (Supabase, Mapbox, OneSignal, OpenAI RAG)

Supabase (DB, Auth, Storage, Realtime)

- **Auth** : only for the admin dashboard. JWT claims (email) checked in RLS policies to authorize writes.
- **DB** : normalized schema ; read-only public policies for services/guides/news ; insert-only policy for reports ; admin CRUD via authenticated policies.
- **Storage** : buckets for *guides/*, *images/*, *reports/* ; public read for published assets ; signed URLs if needed.
- **Sync** : mobile uses light REST/SQL queries ; Realtime optional for live

news or report counters.

Mapbox SDK

- **Features** : service markers, clustering, user geolocation (when permitted), basic routing.
- **UX** : semantic zoom levels, category filters, tap-to-detail ; fallback map when tiles unavailable.

OneSignal (Push Notifications)

- **Registration** : device token capture at first launch (with permission prompt).
- **Segmentation** : optional tags (city/category) for targeted broadcasts.
- **Flows** : admin composes a message in dashboard → server triggers OneSignal REST → client displays and deep-links to the relevant screen.

OpenAI API with RAG

- **Indexing** : server/admin job splits services/guides/news into chunks, stores text + embeddings in `documents_index`.
- **Querying** : app sends user question → similarity search (top-k) → compose prompt with citations/metadata → OpenAI generates grounded answer.
- **Safety/Privacy** : no PII in prompts ; language routed to the user's locale ; capped context window ; graceful degradation when offline (serve guide excerpts).

This architecture keeps the **public app frictionless**, confines **writes to authenticated admins**, and ensures **resilience offline**. Using MVVM + Riverpod improves modularity and testability, while Supabase, Mapbox, OneSignal, and OpenAI RAG provide robust building blocks with minimal operational overhead.

Seasonal Module Configuration (CAN) The CAN visitor section is implemented as an optional module guarded by a configuration flag (remote or build-time). Data (fixtures, stadiums, POIs) is stored in separate tables/collections and accessed through dedicated endpoints to ensure clean isolation and safe removal post-event.

2.5 Conclusion

This chapter outlined the analytical and design foundations of the *Migrant-e-s Help* project. Through a structured functional and non-functional analysis, we identified the system's key actors, use cases, and requirements to ensure accessibility, reliability, and inclusiveness for all users.

The proposed architecture—combining a public mobile application and a secure admin dashboard—relies on a clear client-server model supported by Supabase services. The Flutter-based client follows an MVVM architecture with Riverpod for robust state management and modular design, ensuring an efficient offline-first experience.

Additionally, the integration of external services such as Mapbox for geo-location, OneSignal for push notifications, and the OpenAI API for AI-powered assistance demonstrates the scalability and modernity of the solution. These de-

sign choices provide a strong foundation for the next stage of the project : the technical realization and development of the complete system.

Chapter III : Implementation and Technical Development

Introduction

After establishing the conceptual and architectural foundations of the app, this chapter focuses on the concrete implementation phase. It details the development environment, the tools and technologies used, and the process followed to translate the designed models into a functional and user-friendly application.

4.1 Development Environment

4.1.1 Software and Tools Used

Flutter SDK (v3.24) : Cross-platform development framework used to build the mobile application for Android and web using a single Dart codebase.

Dart : Object-oriented language optimized for Flutter ; ensures reactive UI and high performance on mobile.

Visual Studio Code : Lightweight and extensible development environment used for code editing, version control, and debugging.

Supabase : Provides PostgreSQL database, authentication, file storage, and Row-Level Security (RLS) through a managed cloud backend.

Mapbox SDK : Handles interactive maps, geolocation, and offline tiles used in the “Services” module.

OneSignal : Delivers push notifications to users and manages notification segmentation and targeting.

OpenAI API (GPT-4) : Powers the multilingual AI assistant through

Retrieval-Augmented Generation (RAG) with indexed content.

Hive : Lightweight NoSQL storage solution for offline-first caching of guides, news, and services.

Git + GitHub : Manages source code, branches, and collaborative development using a private GitHub repository.

Supabase Console : Used for monitoring database tables, RLS policies, and storage buckets during development.

4.2 Main Modules and Obtained Results

The following figures present the main interfaces of the *Migrant-e-s Help* mobile application. Each module is designed with simplicity, accessibility, and multilingual support in mind (Arabic, French, English).

Home Page

The home screen serves as the main entry point of the application, providing quick access to the different modules such as Services, News, Guides, Reports, and the Chat Assistant. It also displays dynamic cards and shortcuts depending on the user's language preference.

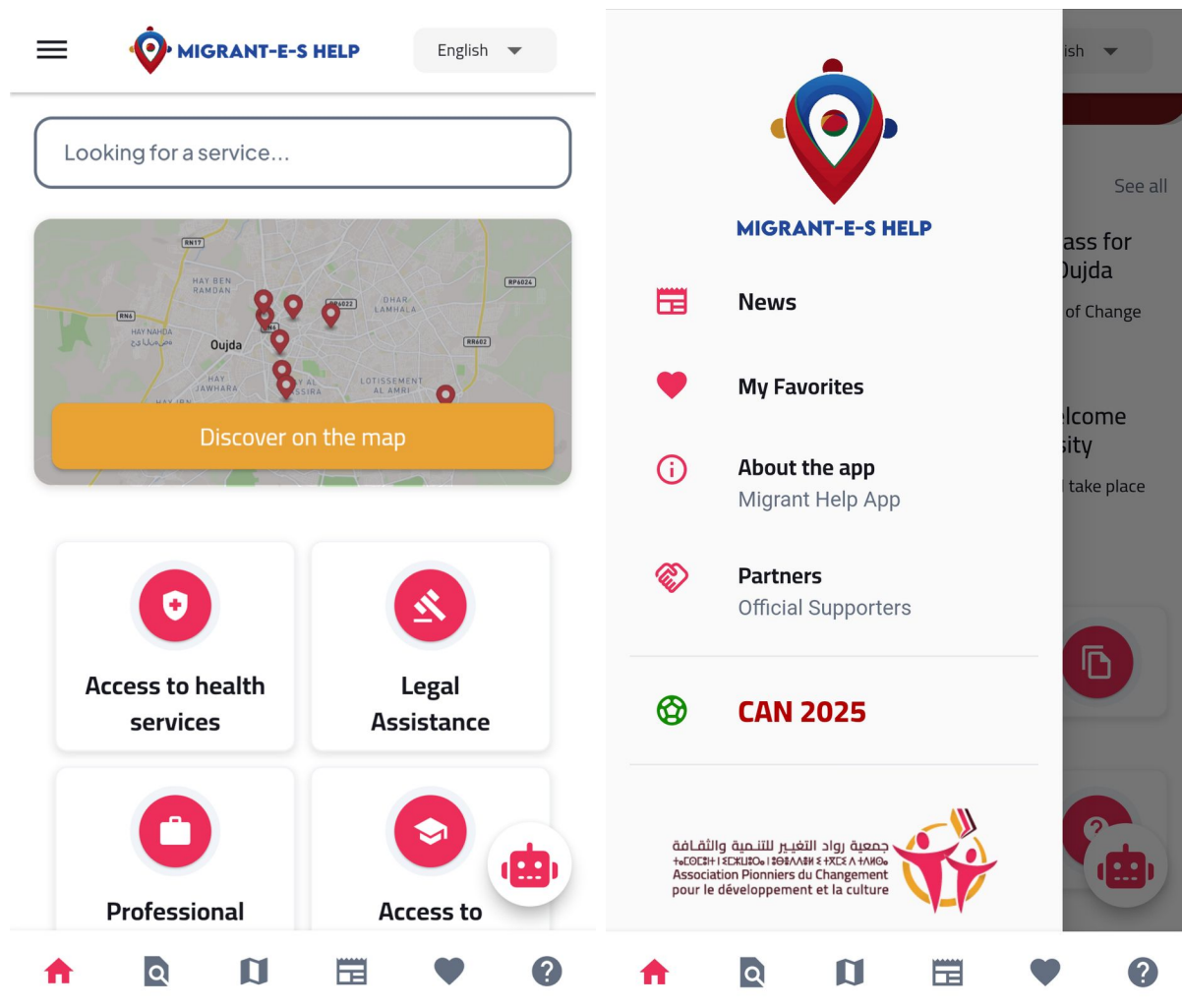


FIGURE 4.8 – Home page interfaces showing main navigation cards and multilingual layout.

Services Module

This module lists organizations and facilities that provide legal, social, educational, and medical services to migrants. Users can filter by category or city and view detailed information, including location and contact details.

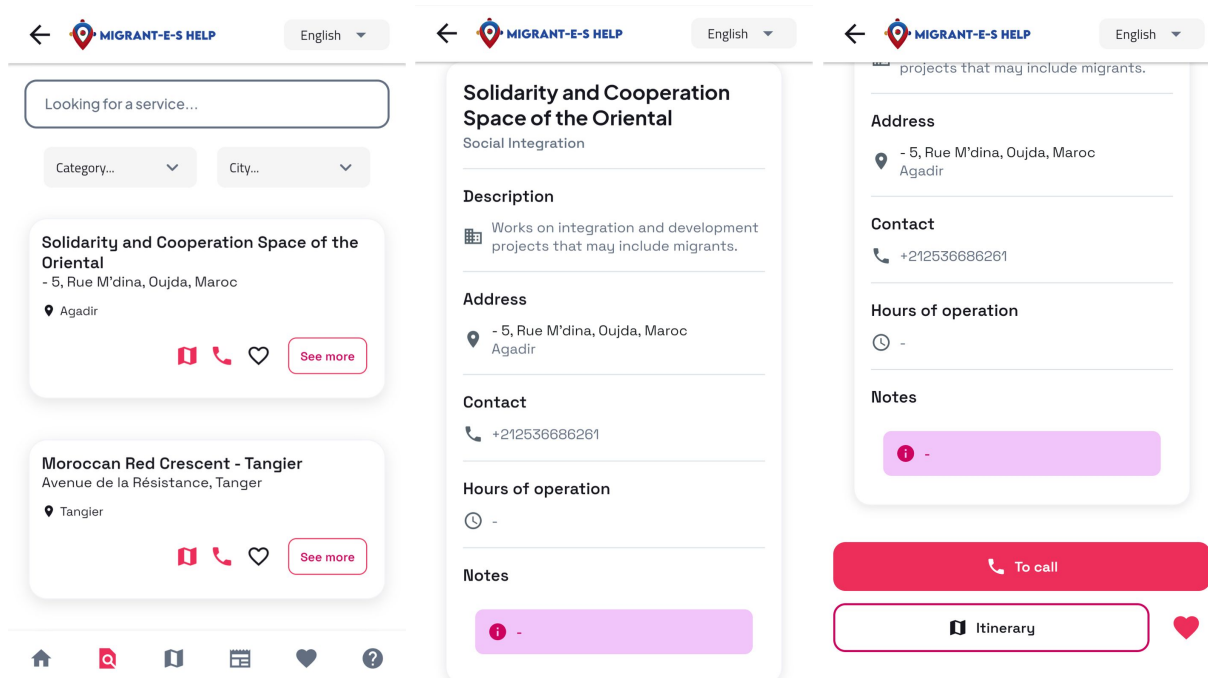


FIGURE 4.9 – Services module showing categorized service list and detailed service information.

AI Chat Assistant

The AI assistant provides contextual answers to user questions. It retrieves relevant information from the NGO's verified content (services, guides, and news) and delivers answers in the user's selected language.

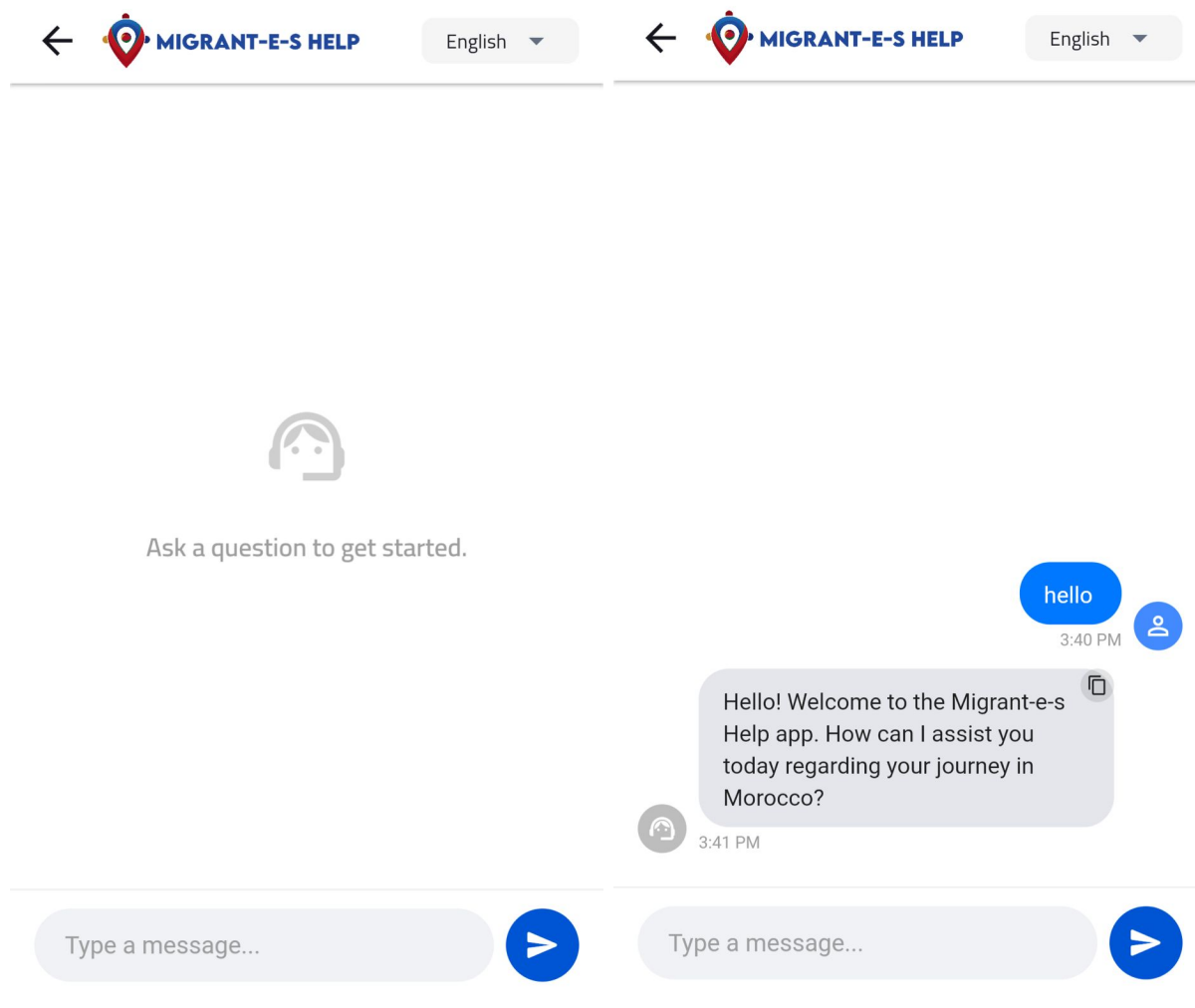


FIGURE 4.10 – AI Chat Assistant module demonstrating multilingual and contextual conversation.

News and Articles Module

This module presents official announcements and news articles published by the NGO. Each article contains a title, date, image, and full text. Users are notified of new updates through push notifications sent via OneSignal.

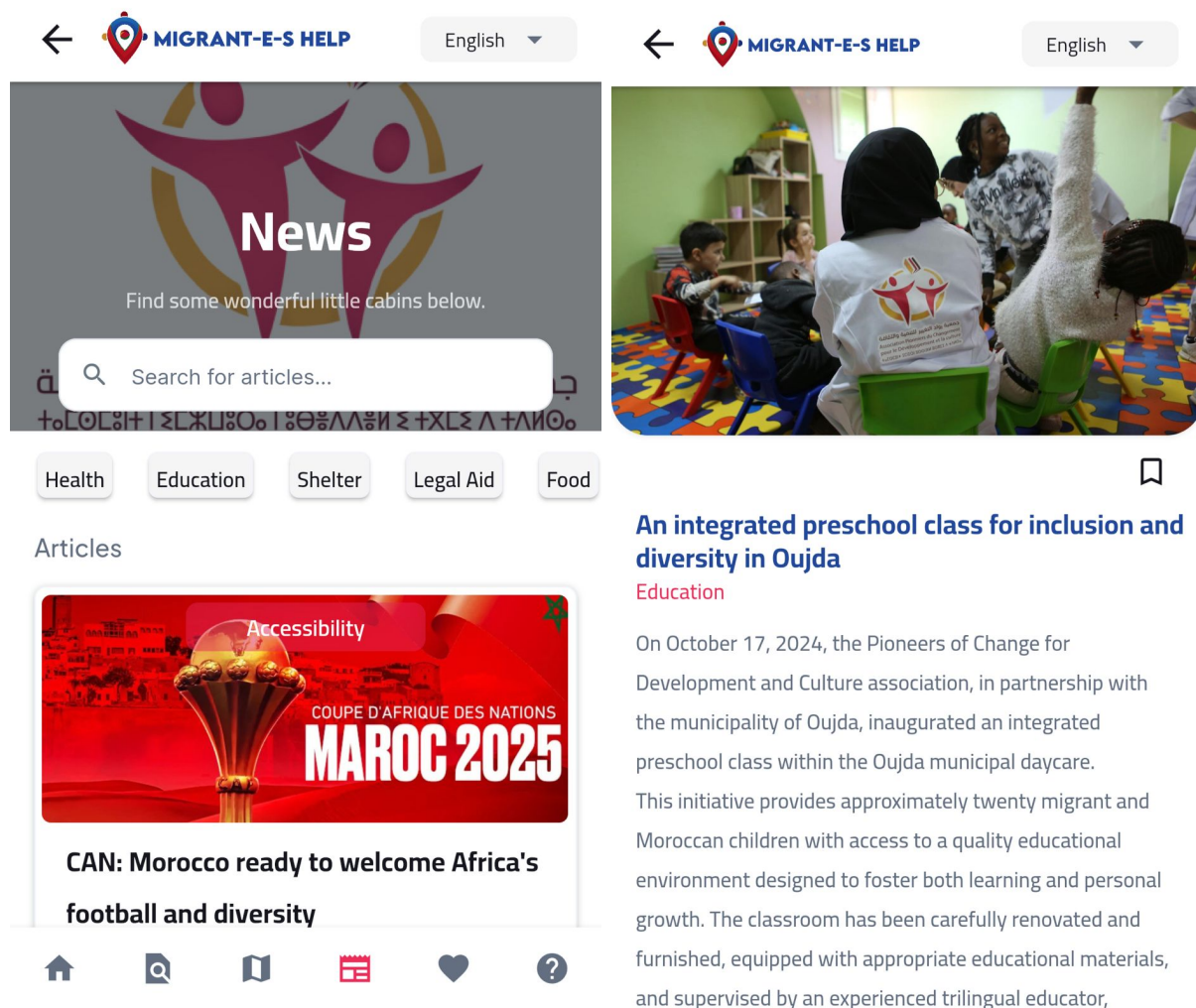


FIGURE 4.11 – News and article pages showing the list of recent updates and detailed article view.

Favorites Module

Users can bookmark services, or articles for quick access later, even when offline. This feature improves personalization and encourages users to revisit frequently used content.

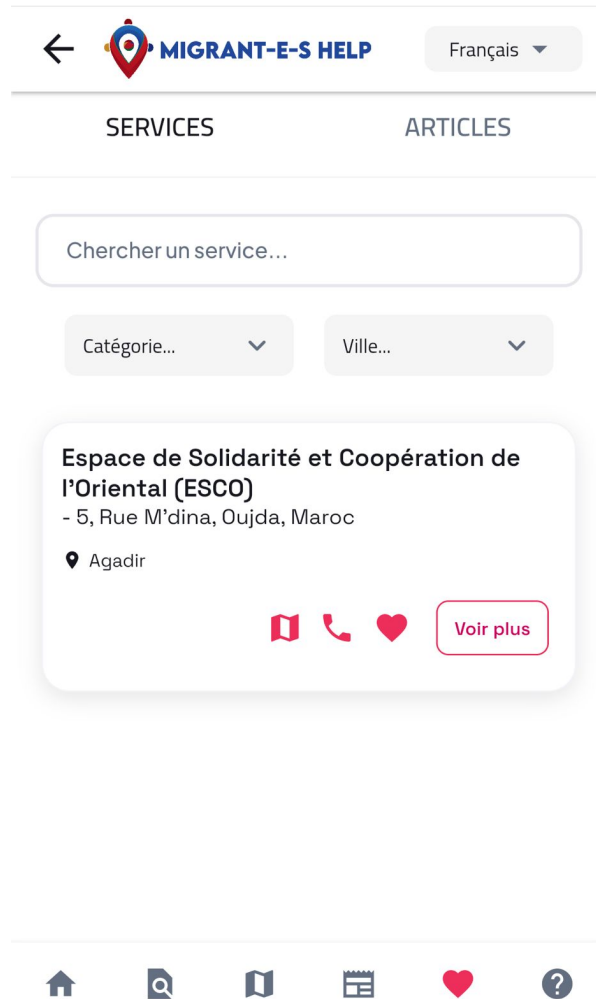


FIGURE 4.12 – Favorites module displaying saved items and quick access shortcuts.

Map and Geolocation Module

The interactive map, powered by Mapbox, displays the locations of nearby services. It supports zoom levels, category-based filtering, and offline map usage for low-connectivity areas.

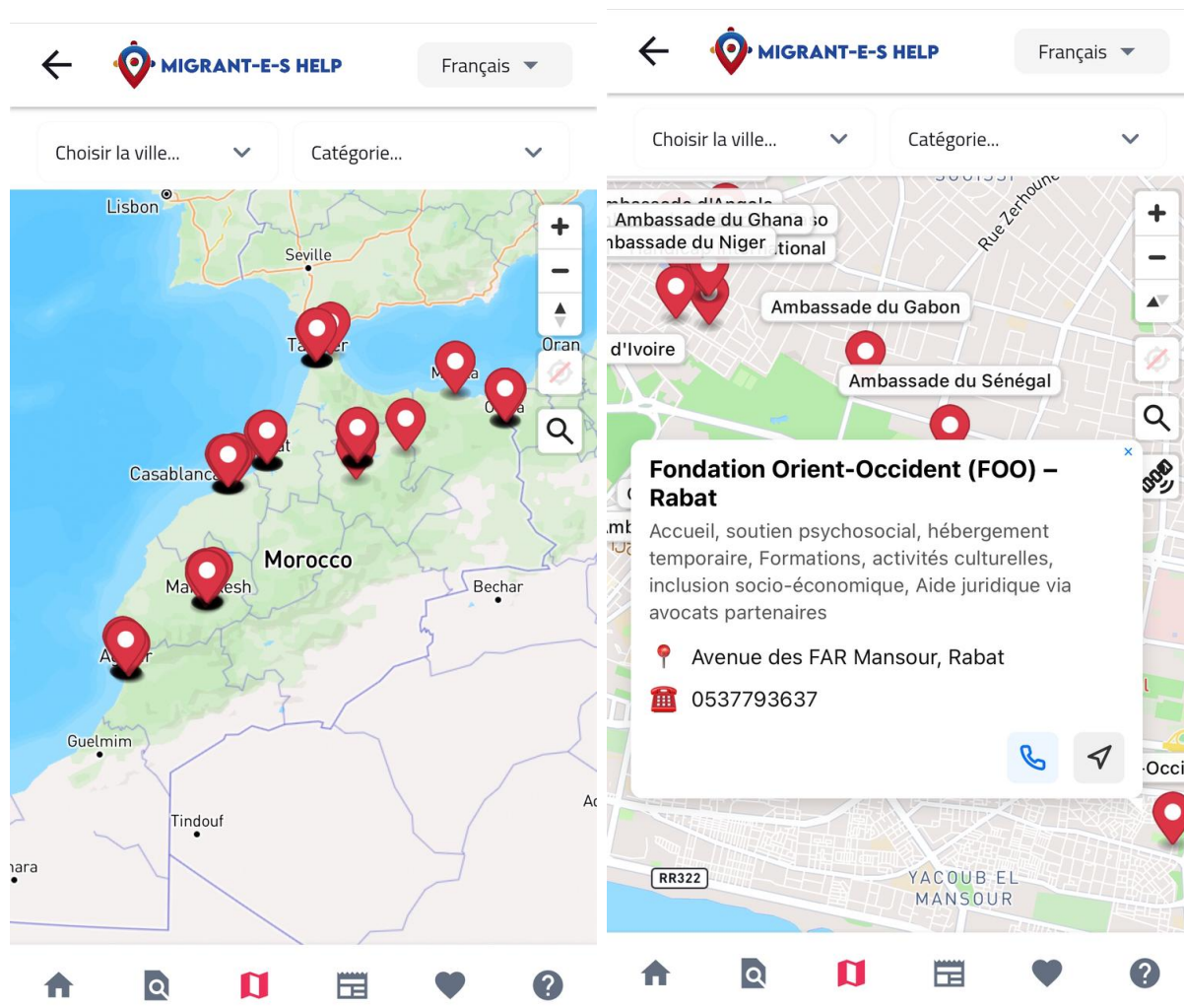


FIGURE 4.13 – Map module showing interactive service markers and offline map view.

Report Module

The **Report Module** allows users to securely submit incidents or technical issues without providing any personal information.

Reports are inserted directly into the Supabase database using a secure, public endpoint governed by **Row Level Security (RLS)** rules to ensure privacy and data integrity.

Once received, each report automatically appears in the **admin dashboard's Kanban view**, where NGO staff can review and update its status (*Re-*

ceived, In Review, Resolved, Archived). Additionally, the system triggers an **automated email notification** to the responsible administrator, who can then respond to the user or take appropriate follow-up actions.

The figure displays two screenshots of the MIGRANT-E-S HELP application interface. The left screenshot shows a French-language submission form titled "Veillez remplir le formulaire". It includes a header with a back arrow, the logo, and a language dropdown set to "Français". The form fields are: "Numéro de téléphone de contact", "Adresse e-mail de contact", "Type d'aide nécessaire" (with a dropdown arrow), "Localisation actuelle", and "Description détaillée de la situation / des besoins". At the bottom, there is a "Niveau d'urgence" label. The right screenshot shows an English-language success confirmation screen titled "Contact us". It features a header with a back arrow, the logo, and a language dropdown set to "English". Below the header, there is a "Need more help?" section with a sub-header "Our support team is available to help you. You can contact us in several ways:". A white modal box displays three options: "Voluntary departure", "Problem in the application", and "Request for help". Below the modal, there is an "Admin area" section with a dark blue button containing a circular arrow icon. At the bottom, there is a navigation bar with icons for home, search, documents, calendar, heart, and a red question mark.

FIGURE 4.14 – Report module showing submission form and success confirmation screen.

CAN Visitor Section (Seasonal)

A temporary section designed for Africa Cup of Nations visitors provides :

- (i) match schedules by city/stadium, (ii) stadium information (access, capacity, map), and (iii) curated places to visit. The section is activated only during the tournament and will be removed afterwards.

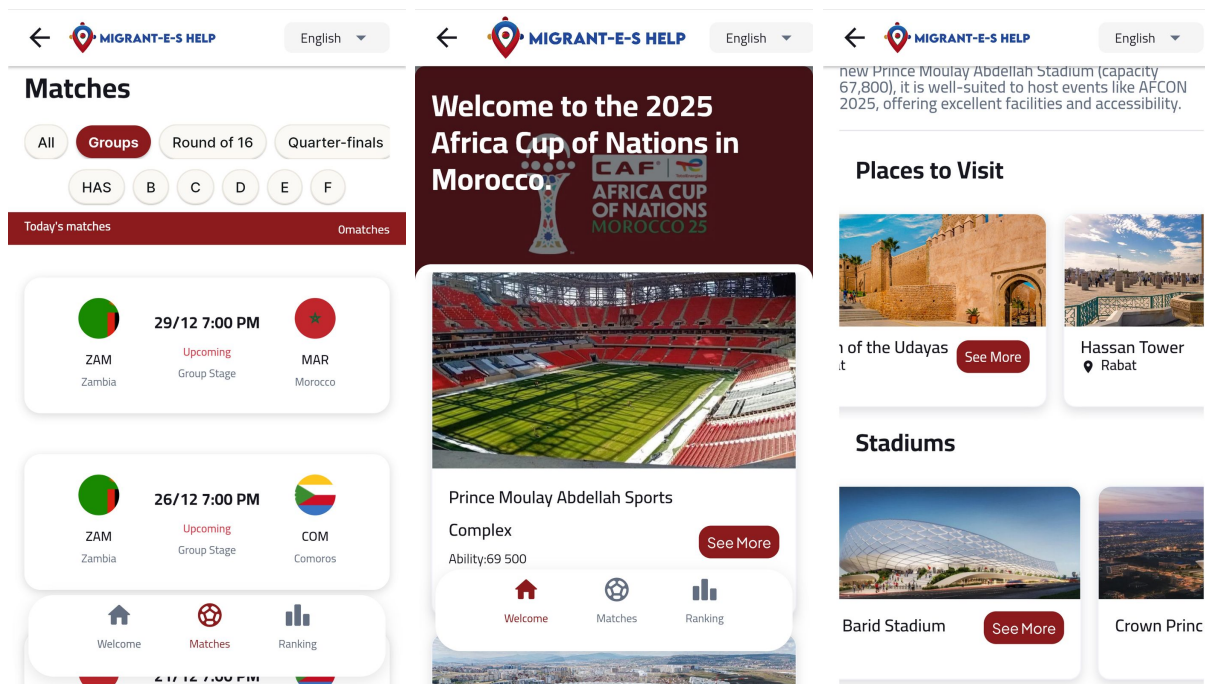


FIGURE 4.15 – Seasonal CAN section : matches, stadium details, and city highlights.

Admin Web Portal

The administration dashboard was designed for NGO staff to manage the content displayed in the mobile application. It is accessible only to authenticated users through Supabase Auth and follows a secure role-based access model (admin, editor, reviewer). From this portal, administrators can add, edit, or remove services, news, guides, and monitor reports submitted by users.

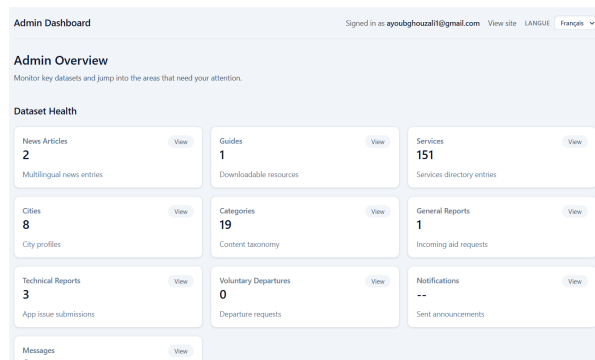
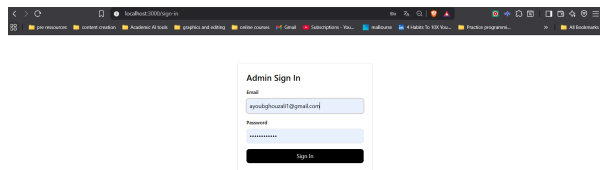


FIGURE 4.16 – Admin web portal overview showing dashboard layout and content management interface.

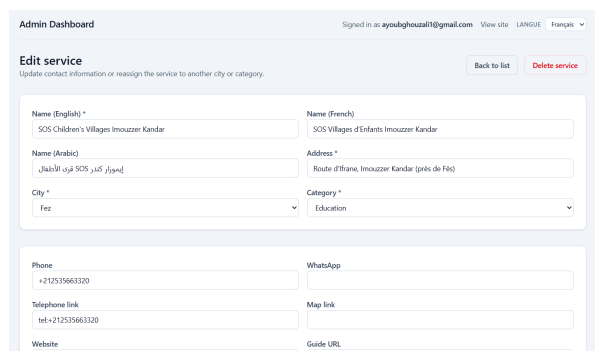
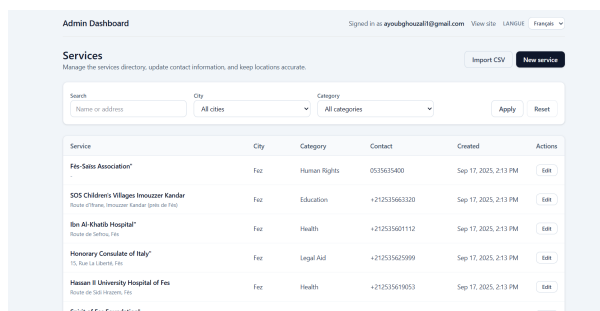


FIGURE 4.17 – Admin interfaces for managing services, reviewing reports, and publishing news articles.

4.3 Conclusion

This chapter presented the practical implementation of the *Migrant-e-s Help* project, from the development environment to the realization of its main functional modules.

General Conclusion

This internship represented an enriching professional experience that combined technical learning, creativity, and social impact. Hosted by **DTR**, the internship focused on the design and development of the *Migrant-e-s Help* application—an innovative mobile solution created for the NGO **Pionniers du Changement** to support migrants in Morocco.

Throughout the internship, the project followed a structured and iterative methodology based on the Agile approach, enabling efficient progress from analysis and design to implementation and testing. The chosen technological stack—**Flutter** for cross-platform development and **Supabase** for backend management—ensured a modern, secure, and scalable system. The integration of third-party services such as **Mapbox**, **OneSignal**, and the **OpenAI API** contributed to a rich, intelligent, and user-friendly experience.

From a functional perspective, the application successfully met its objectives : providing multilingual access to services, downloadable guides, real-time news, reporting, and an AI assistant for personalized support. These features make the platform a concrete digital tool for inclusion and information accessibility, aligning with the humanitarian mission of the NGO partner.

On a personal level, this internship allowed me to strengthen my technical skills in mobile development, database management, and API integration, while also deepening my understanding of project management and user-centered de-

sign. I also learned to work within professional constraints—deadlines, version control, and iterative feedback—which are essential aspects of real-world software engineering.

Despite some challenges, particularly in managing offline data synchronization and integrating AI retrieval logic, these difficulties proved valuable opportunities to develop problem-solving skills and adaptability.

In conclusion, this internship not only contributed to my professional growth as a software engineer but also allowed me to participate in a socially meaningful project that promotes inclusion and digital accessibility.

WEBOGRAPHIE

Flutter Documentation : <https://docs.flutter.dev/>

Supabase Documentation : <https://supabase.com/docs>

Hive Documentation : <https://docs.hivedb.dev/>

Mapbox Developer Documentation : <https://docs.mapbox.com/>

OneSignal Documentation : <https://documentation.onesignal.com/>

OpenAI API Documentation : <https://platform.openai.com/docs>