Mohamed Khider University of Biskra
Faculty of Exact Sciences
Department of Computer Science

MASTER THESIS

Speciality : Master - Computer Science Applied to Business Management (Professional) - Ref : P4987/2026

TITLE

A Digital Platform for Requesting Maintenance Services via Video

Prepared and defended by

Kamel eddine LAHLOUH

Aymen abderraouf MORGHAD

Defense date : 07/06/2026

DEFENSE COMMITTEE

Full Name	Grade	Role
Rachida BENABDELAZIZ	Prof	**President**
Hanane OUAAR	MCB	**Supervisor**
Rima Hiouani	MCB	**Examiner**
Nadjoua HABBA	Prof	**Incubator Representative**

Academic Year : 2025/2026

Acknowledgments

First and foremost, we would like to express our deepest gratitude to Allah, the Most Gracious and the Most Merciful, for giving us the strength, patience, and perseverance to complete this work.

We would like to sincerely thank our beloved parents for their endless love, sacrifices, support, and prayers throughout our academic journey. Their encouragement has always been a source of strength and motivation for us.

We also extend our heartfelt thanks to our families, who have always stood by our side and supported us in every step. Their presence, care, and encouragement have played an important role in the completion of this work.

Our sincere gratitude goes to our supervisor, Dr. Hanane, for her guidance, valuable advice, patience, and continuous support during the preparation of this project. Her knowledge and encouragement helped us improve this work and overcome many difficulties.

We would also like to express our appreciation to Dr. Fatima for her kindness, help, and support. Her encouragement and valuable presence are deeply appreciated.

We would like to thank all our teachers for the knowledge, guidance, and efforts they provided throughout our studies. Their dedication and professionalism have greatly contributed to our academic development.

Finally, we would like to thank all our friends for their moral support, encouragement, and positive energy. Their presence made this journey easier and more meaningful.

Dedication

We dedicate this humble work to our dear parents, whose love, sacrifices, prayers, and support have always guided us and given us the strength to continue.

To our beloved families, for their encouragement, patience, and constant presence in our lives.

To our teachers, who shared their knowledge and guided us throughout our academic path.

To Dr. Hanane, with deep respect and gratitude for her valuable supervision and support.

To Dr. Fatima, with sincere appreciation for her kindness and encouragement.

To our dear friends, who supported us, motivated us, and shared unforgettable moments with us.

To everyone who believed in us and contributed, directly or indirectly, to the completion of this work.

Contents

Acknowledgments

Dedication

Abstract **1**

General Introduction **4**

1 State of the Art **8**

1.1 Introduction 8

1.2 Maintenance Service Platforms 8

1.2.1 Definitions 9

1.2.2 Objectives 9

1.3 Key Terminologies 10

1.3.1 Fundamental Concepts of Maintenance Services 11

1.3.2 Maintenance Service Categories 13

1.4 Review of Existing Works 14

1.4.1 Thumbtack (Global Platform) 14

1.4.2 ServiceMarket (Regional Platform) 16

1.4.3 Kandua (African Platform) 18

1.4.4 De9De9 (Local Algerian Platform) 19

1.4.5 Khelliha 3lina (Local Algerian Platform) 21

1.5 Proposed Solution: Fixoo Platform 23

1.5.1 Fixoo Definition 23

1.5.2 Main Objectives of Fixoo 24

1.5.3 Comparison with International Platforms 24

1.5.4 Comparison with Algerian Platforms . . . 26
1.6 Conclusion . . . 28

2 Analysis and Specifications **29**
2.1 Introduction . . . 29
2.2 Needs Analysis . . . 29
2.3 General Objective . . . 30
2.4 Specific Objectives . . . 31
2.5 Specifications . . . 34
2.5.1 System Description . . . 35
2.5.2 Identification of Actors . . . 35
2.5.3 Functional Requirements . . . 37
2.5.4 Non-Functional Requirements . . . 40
2.5.5 Constraints . . . 41
2.6 Conclusion . . . 41

3 *System Design* **43**
3.1 Introduction . . . 43
3.2 System Architecture . . . 43
3.2.1 Client-Server Architecture . . . 44
3.2.2 Multi-Tiered Architecture . . . 44
3.2.3 Clean Architecture Implementation . . . 46
3.3 UML Diagrams . . . 47
3.3.1 Use Case Diagrams . . . 47
3.3.2 Class Diagram . . . 53
3.3.3 Activity Diagrams . . . 58
3.3.4 System Modules . . . 60
3.4 Conclusion . . . 63

4 System Implementation **64**
4.1 Introduction . . . 64
4.2 Development Environment and Technologies . . . 64
4.2.1 Programming Languages . . . 65
4.2.2 Frameworks and Libraries . . . 69

4.2.3 Database Software Tools 74
4.2.4 Development Environment (IDE) 76
4.2.5 Software Development Tools 79
4.3 Database Structure 82
4.3.1 Logical Data Model (MLD) 83
4.3.2 Main Database Tables 84
4.3.3 Database Relationships 86
4.3.4 Mobile Application Interfaces 87
4.3.5 Administrative Web Panel (Blazor Server) 91
4.4 Conclusion 94

General Conclusion **95**

Bibliography **97**

List of Figures

1.1 Thumbtack platform interface. 14
1.2 ServiceMarket platform interface. 16
1.3 Kandua platform interface. 18
1.4 De9De9 platform interface. 20
1.5 Khelliha 3lina platform interface. 22

3.1 Three-Tier Architecture of the Fixoo Platform 46
3.2 Use Case Diagram – Fixoo Admin System 48
3.3 Use Case Diagram – Fixoo User and Provider System 51
3.4 Class Diagram of the Fixoo Platform. 55
3.5 Activity Diagram – Maintenance Service Request Process 59
3.6 Architectural Overview of Fixoo System Modules. 61

4.1 C# Language Logo. 65
4.2 Dart Language Logo. 66
4.3 SQL Logo. 67
4.4 HTML Logo. 68
4.5 CSS Logo. 69
4.6 JavaScript Logo. 69
4.7 ASP.NET Core Logo. 70
4.8 ADO.NET Logo. 71
4.9 Flutter Logo. 72
4.10 Blazor Server Logo. 72
4.11 Bootstrap Logo. 73
4.12 SignalR Logo. 74
4.13 Microsoft SQL Server Logo. 74

4.14 Azure Blob Storage Logo. 75
4.15 Visual Studio 2022 Logo. 76
4.16 Visual Studio Code Logo. 77
4.17 Android Studio Logo. 78
4.18 Postman Logo. 79
4.19 Swagger Logo. 80
4.20 Ngrok Logo. 80
4.21 GitHub Logo. 81
4.22 OpenStreetMap Logo. 81
4.23 PlantUML Logo. 82
4.24 OneSignal Logo. 82
4.25 USERS Table Structure from Microsoft SQL Server. 84
4.26 PROVIDERS Table Structure from Microsoft SQL Server. 85
4.27 VIDEO_REQUEST Table Structure from Microsoft SQL Server. 85
4.28 OFFERS Table Structure from Microsoft SQL Server. 86
4.29 ACCEPTED_PROJECTS Table Structure from Microsoft SQL Server. . . 86
4.30 Home Screen Showing Video Service Requests. 88
4.31 Video Request Submission Screen. 89
4.32 Offer Comparison and Selection Screen. 90
4.33 Provider Tracking Screen with Call and Message Options. 91
4.34 Administrative Dashboard – Main Statistics. 92
4.35 User Management Interface. 92
4.36 Provider Document Verification Interface. 93
4.37 Category Management Interface. 93
4.38 Payment Management Interface. 94
4.39 certificate 102

List of Tables

1.1 Fundamental Concepts in Maintenance Service Systems. 12
1.2 Common Categories of Maintenance Services. 13
1.3 Comparison between Fixoo and International Platforms 26
1.4 Comparison between Fixoo and Algerian Platforms 27
3.1 Description Table – Administrator Use Cases 50
3.2 Description Table – User and Provider 53

Abstract

The aim of this work is to design and develop an integrated digital platform called **Fixoo** to improve the process of requesting, managing, tracking, and evaluating maintenance services in the Algerian context. The platform aims to enhance coordination between service seekers and service providers through video-based maintenance requests, competitive offers, real-time tracking, provider evaluation, and administrative supervision. To achieve this, a multi-tier architecture was adopted, supported by UML modeling and Clean Architecture principles inside the business logic layer. The system was implemented using Flutter for the mobile application, ASP.NET Core Web API for backend services, Blazor Server for the administration interface, Microsoft SQL Server for data management, and Azure Blob Storage for storing videos, images, and documents. The results demonstrate that the proposed platform improves service accessibility, reduces misunderstandings between users and technicians, facilitates the selection of qualified providers, and strengthens trust through verification, ratings, notifications, and structured service follow-up. It also helps technicians increase their digital visibility, manage their work more efficiently, and build a professional reputation. Therefore, this work provides an integrated and scalable solution that supports the digital transformation of maintenance service management in Algeria and improves the overall experience for both clients and service providers.

Keywords:

Fixoo, maintenance services, digital platform, Algeria, service seeker, service provider, video request, competitive offers, real-time tracking, Clean Architecture, ASP.NET Core, Flutter, Blazor Server, Microsoft SQL Server, Azure Blob Storage.

Résumé

L'objectif de ce travail est de concevoir et de développer une plateforme numérique intégrée appelée **Fixoo**, afin d'améliorer le processus de demande, de gestion, de suivi et d'évaluation des services de maintenance dans le contexte algérien. La plateforme vise à renforcer la coordination entre les demandeurs de services et les prestataires à travers les demandes de maintenance basées sur la vidéo, les offres compétitives, le suivi en temps réel, l'évaluation des prestataires et la supervision administrative. Pour atteindre cet objectif, une architecture multi-couches a été adoptée, appuyée par la modélisation UML et les principes de la Clean Architecture au niveau de la couche logique métier. Le système a été implémenté en utilisant Flutter pour l'application mobile, ASP.NET Core Web API pour les services backend, Blazor Server pour l'interface d'administration, Microsoft SQL Server pour la gestion des données et Azure Blob Storage pour le stockage des vidéos, images et documents. Les résultats montrent que la plateforme proposée améliore l'accessibilité aux services, réduit les malentendus entre les utilisateurs et les techniciens, facilite le choix de prestataires qualifiés et renforce la confiance grâce à la vérification, aux évaluations, aux notifications et au suivi structuré des services. Elle permet également aux techniciens d'améliorer leur visibilité numérique, de gérer leur travail plus efficacement et de construire une réputation professionnelle. Ainsi, ce travail propose une solution intégrée et évolutive qui soutient la transformation numérique de la gestion des services de maintenance en Algérie.

Mots-clés :

Fixoo, services de maintenance, plateforme numérique, Algérie, demandeur de service, prestataire de service, demande vidéo, suivi en temps réel, Clean Architecture, ASP.NET Core, Flutter, Blazor Server, Microsoft SQL Server, Azure Blob Storage.

ملخص

يهدف هذا العمل إلى تصميم وتطوير منصة رقمية متكاملة تُدعى Fixoo لتحسين عملية طلب خدمات الصيانة وإدارتها وتتبعها وتقييمها في السياق الجزائري. تهدف المنصة إلى تعزيز التنسيق بين طالبي الخدمات ومقدمي الخدمات من خلال طلبات الصيانة المعتمدة على الفيديو، والعروض التنافسية، والتتبع الآني، وتقييم مقدمي الخدمات، والإشراف الإداري. ولتحقيق ذلك، تم اعتماد بنية متعددة الطبقات (Multi-Tier Architecture) مدعومة بنمذجة UML ومبادئ الهندسة النظيفة (Clean Architecture) داخل طبقة منطق الأعمال. تم تنفيذ النظام باستخدام Flutter لتطوير تطبيق الهاتف المحمول، وASP.NET Web Core API لتطوير خدمات الخلفية، وServer Blazor لتطوير واجهة الإدارة، وMicrosoft SQL Server لإدارة البيانات، وAzure Blob Storage لتخزين الفيديوهات والصور والوثائق. أظهرت النتائج أن المنصة المقترحة تُحسن من سهولة الوصول إلى الخدمات، وتقلل من سوء الفهم بين المستخدمين والتقنيين، وتُسهل عملية اختيار مقدمي الخدمات المؤهلين، كما تعزز الثقة من خلال التحقق، والتقييمات، والإشعارات، والمتابعة المنظمة للخدمات. كما تساعد التقنيين على زيادة حضورهم الرقمي، وإدارة أعمالهم بكفاءة أكبر، وبناء سمعة مهنية احترافية. وبالتالي، يوفر هذا العمل حلاً متكاملاً وقابلاً للتوسع يدعم التحول الرقمي لإدارة خدمات الصيانة في الجزائر، ويحسن التجربة العامة لكل من العملاء ومقدمي الخدمات.

الكلمات المفتاحية:

Fixoo، خدمات الصيانة، منصة رقمية، الجزائر، طالب الخدمة، مقدم الخدمة، طلب فيديو، العروض التنافسية، التتبع الآني، الهندسة النظيفة، ASP.NET Core، Blazor Flutter، SQL Microsoft Server، Blob Azure Server، Storage.

General Introduction

General Context

In recent years, maintenance services have become essential in daily life, as homes, workplaces, and public spaces require different interventions such as electrical repair, plumbing, appliance maintenance, cleaning, construction, and other technical services. However, many service seekers still face difficulties finding a reliable and available service provider at the right time.

Traditional methods such as phone calls, recommendations, social media groups, and informal contacts are still widely used. Although these methods may help in some cases, they often lack organization, transparency, and reliability. Moreover, some maintenance problems are complex, unusual, or difficult to describe through written text alone, which may lead to misunderstandings and inaccurate diagnosis.

From the provider's side, many skilled technicians also face difficulties in finding clients regularly and building a visible professional reputation. This situation highlights the need for a more organized and user-centered solution.

The Fixoo platform was proposed to answer this need. It aims to simplify the maintenance request process by allowing service seekers to present their problems clearly, receive offers from qualified providers, compare them, follow the service progress, and evaluate the provider after completion. The project is especially adapted to the Algerian context, taking into account local service needs, trust issues, language preferences, and geographic considerations.

Problem Statement

Despite the availability of many service providers, the process of finding a suitable technician remains difficult for many service seekers. This difficulty can be summarized through the following points:

- Service seekers often do not know which provider is qualified, available, nearby, or trustworthy.
- Explaining an uncommon or complex technical problem using only words can be insufficient, which may lead to misunderstandings, delays, wrong price estimations, or unnecessary visits.
- Traditional interactions often lack transparency, as service seekers cannot easily compare offers, verify provider reliability, track service progress, or keep a structured record of the intervention.
- Technicians also face challenges, since many qualified providers depend mainly on informal networks and do not have enough digital visibility.
- Service providers need a platform that allows them to receive relevant requests, submit offers, manage their work, and build a professional reputation.

Based on these observations, the main problem addressed by this project can be formulated as follows:

> *How can we design and implement a digital platform adapted to the Algerian context that improves the process of requesting, managing, tracking, and evaluating maintenance services while ensuring transparency, trust, and ease of use for both service seekers and service providers?*

Objectives

The main objective of this project is to design and develop Fixoo, a digital platform that connects service seekers with qualified maintenance service providers in a structured

and transparent way. The platform aims to replace disorganized traditional methods with a modern process that helps service seekers find suitable providers more easily and allows technicians to access service opportunities more efficiently.

The specific objectives of the project are:

- Facilitate the submission of maintenance requests by allowing service seekers to present their needs clearly.
- Enable providers to view available requests and submit competitive offers based on price, time, and service conditions.
- Allow service seekers to compare offers and choose the most suitable provider.
- Improve trust through provider verification, profiles, ratings, and reviews.
- Support service follow-up and progress monitoring.
- Provide an administrative environment for managing service seekers, providers, categories, complaints, payments, and reports.
- Adapt the platform to the Algerian market, languages, habits, and user expectations.

Through these objectives, Fixoo seeks to offer a practical solution for both sides. Service seekers gain a clearer, faster, and safer way to request maintenance services, while providers gain better visibility, organized opportunities, and a way to build credibility through completed services and client evaluations.

Organization of the Thesis

This thesis is organized into four chapters, followed by a general conclusion.

Chapter 01: State of the Art. This chapter introduces digital maintenance service platforms, defines their main concepts, and reviews existing solutions at global, regional, African, and local Algerian levels. It also highlights their limitations and presents Fixoo as a proposed solution.

Chapter 02: Analysis and Specifications. This chapter identifies the needs of the system, the main actors, and the functional and non-functional requirements. It also presents the objectives and constraints that guide the development of the platform.

Chapter 03: System Design. This chapter presents the design of the Fixoo platform, including the system architecture, UML diagrams, data structure, and main modules that organize the platform functions.

Chapter 04: System Implementation. This chapter describes the implementation phase, including the development environment, tools, database structure, and main interfaces of the mobile application and administrative panel.

Finally, the **General Conclusion** summarizes the work carried out, the challenges encountered, the skills acquired, and the future perspectives for improving the Fixoo platform.

Chapter 1

State of the Art

1.1 Introduction

The world is witnessing a continuous increase in demand for home and facility maintenance services that are fast, reliable, and easily accessible. Traditional methods of requesting maintenance services mainly rely on phone calls, word-of-mouth recommendations, or informal contacts, and are often slow, disorganized, and difficult to track.

The rapid advancement of web and mobile technologies, particularly video communication technologies, has created new opportunities to transform how clients describe problems and how technicians diagnose and respond to them.

This chapter aims to define the concept of digital maintenance service platforms and their core objectives. It also introduces the key terminologies that structure their operation and provides a critical review of existing solutions at the global, regional, African, and local Algerian levels.

Based on the identified gaps and limitations, the Fixoo platform is proposed as a modern solution designed to address these shortcomings by enabling video-based maintenance request submission and centralized service management.

1.2 Maintenance Service Platforms

Maintenance service platforms represent a fundamental category of digital systems

in the modern economy, and understanding their concept and objectives is essential before analyzing existing solutions or proposing new ones.

1.2.1 Definitions

In [1], maintenance is defined as the set of technical and administrative actions carried out to retain an item, equipment, or facility in, or restore it to, a state in which it can perform its required function. Maintenance activities include inspection, repair, replacement, and servicing operations aimed at ensuring reliability, safety, and operational efficiency.

In [2], maintenance services are described as organized processes that support the upkeep and repair of physical assets such as buildings, electrical systems, plumbing, and household equipment, ensuring that infrastructure continues to function correctly while minimizing downtime and operational risks.

In the context of digital transformation [3], a **Maintenance Service Platform** is defined as a web- or mobile-based system that connects customers with qualified technicians. It enables users to submit service requests, communicate issues, schedule interventions, and monitor service progress electronically. Such platforms replace traditional manual processes with structured, traceable, and efficient digital workflows.

Additionally, multimedia-based maintenance requests refer to service requests that incorporate supporting media such as images or videos to describe technical issues more clearly. This approach enhances problem understanding, improves diagnosis accuracy, and helps reduce unnecessary on-site visits [5].

1.2.2 Objectives

Digital maintenance service platforms in the modern economy constitute a fundamental pillar for achieving a set of operational objectives, which can be summarized as follows:

1. **Improving Service Accessibility:** Digital service platforms aim to make maintenance services easily accessible to any user, regardless of location or time, by providing online request submission interfaces and remote communication tools [3].

2. **Reducing Response Time:** By digitizing the request processing, technician assignment, and notification workflows, platforms significantly reduce delays between problem identification and technician intervention, improving overall operational efficiency [4].

3. **Enhancing Problem Diagnosis Accuracy:** Multimedia descriptions, particularly video, enable technicians to better understand the nature of technical issues before physical intervention, improving first-visit resolution rates and reducing unnecessary on-site visits [5].

4. **Centralizing Service Management:** Digital platforms integrate clients, technicians, and administrators into a single unified environment where requests, schedules, communications, and service histories are managed centrally [3].

5. **Ensuring Transparency and Traceability:** Electronic platforms maintain full service records, feedback logs, and status histories, ensuring accountability on all sides and enabling continuous quality monitoring and improvement [4].

In summary, digital maintenance platforms are not merely communication tools; they are structured information systems that integrate human, technical, and organizational resources to deliver faster, more reliable, and more transparent maintenance services.

1.3 Key Terminologies

A precise understanding of any service management system requires a clear definition of its fundamental concepts and terminology. In the context of maintenance and service operations, these terms help describe how services are requested, delivered, and evaluated across different domains.

1.3.1 Fundamental Concepts of Maintenance Services

Fundamental concepts refer to the essential elements that define how maintenance services are organized, managed, and delivered between clients and service providers. Table 1.1 presents the main concepts involved in maintenance service systems and their respective definitions:

Term	Definition
Service Request	A formal request submitted by a client describing a maintenance need, including the nature of the problem, location, and urgency level [3].
Client	An individual or organization that requires maintenance services and initiates service requests to ensure the proper functioning of systems or equipment [1].
Technician (Service Provider)	A qualified professional responsible for performing maintenance, repair, and diagnostic tasks based on specialized technical knowledge [2].
Diagnosis	The process of identifying faults or failures in a system through inspection and analysis to determine appropriate corrective actions [1].
Communication	The exchange of information between clients and service providers to ensure accurate understanding of problems and coordination of maintenance activities [4].
Service Offer	A proposal that includes estimated cost, required resources, and completion time for a maintenance task, enabling decision-making [3].
Service Execution	The process of carrying out maintenance operations, including repair, replacement, or adjustment of components to restore functionality [2].
Quality Evaluation	The assessment of maintenance outcomes based on performance, reliability, and user satisfaction [1].

Table 1.1: Fundamental Concepts in Maintenance Service Systems.

1.3.2 Maintenance Service Categories

Maintenance service categories represent the different areas in which maintenance activities are performed. These categories help classify service types and identify the required expertise.

Table 1.2 illustrates common categories of maintenance services:

Category	Specializations Covered
Electrical Maintenance	Installation, inspection, and repair of electrical systems in residential and industrial environments [1].
Plumbing & Sanitation	Maintenance of water systems, pipes, sanitation infrastructure [2].
Construction & Finishing	Activities related to building maintenance, renovation, painting, carpentry, welding, and tiling [1].
Automotive Maintenance	Diagnosis and repair of vehicles, including mechanical systems, bodywork, and tires [2].
Electronics Maintenance	Repair and maintenance of electronic devices such as computers, televisions, and household appliances [5].
Communication Systems	Installation and maintenance of networking systems, internet infrastructure, and surveillance technologies [5].
Cleaning & Hygiene Services	Maintenance of clean and safe environments through cleaning, pest control, and sanitation procedures [6].
Outdoor Maintenance	Gardening, landscaping, and irrigation system maintenance activities [6].
Security Systems	Installation and maintenance of safety equipment such as alarms, locks, and surveillance systems [6].

Table 1.2: Common Categories of Maintenance Services.

1.4 Review of Existing Works

The digital solutions proposed for maintenance service management vary widely in their approaches and capabilities. This section presents a critical review of the most prominent existing platforms at the global, regional, African, and local Algerian levels, examining their main functions and analyzing their advantages and limitations.

1.4.1 Thumbtack (Global Platform)

Thumbtack [7] is a large-scale American digital marketplace designed to connect customers with local professionals across hundreds of service categories, including home maintenance, electrical repair, plumbing, cleaning, and installation. Founded in 2008 and headquartered in San Francisco, the platform operates primarily within the United States and serves millions of users annually. Thumbtack focuses on simplifying the process of finding trusted professionals by digitizing service discovery, provider comparison, and booking workflows into a single integrated system accessible via web and mobile applications.

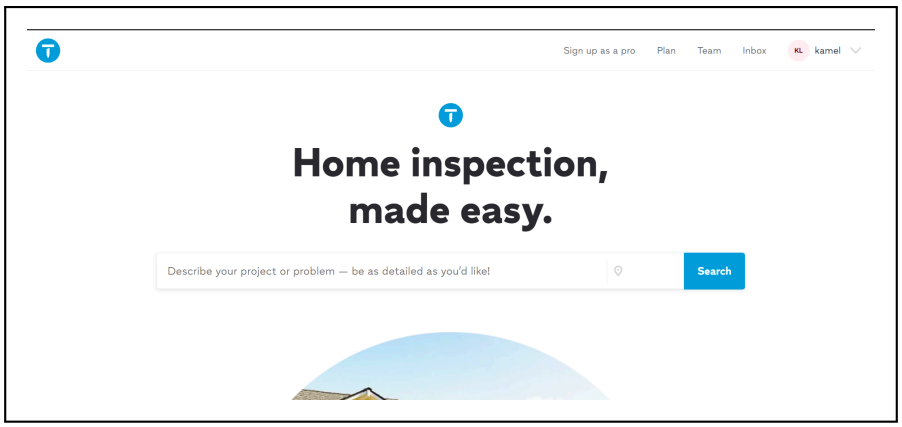


Figure 1.1: Thumbtack platform interface.

B. Main Functions of Thumbtack

Thumbtack provides several key functions for connecting clients with local service professionals [7]:

– **Smart Matching Algorithm:** Users describe their needs through a structured questionnaire, and the platform automatically suggests relevant professionals based

on location, expertise, availability, and past performance.

– **Detailed Professional Profiles:** Each service provider has a profile containing ratings, verified reviews, portfolio images, service descriptions, and pricing indicators, enabling informed decision-making.

– **Instant Booking and Requests:** Clients can send requests directly to professionals or schedule appointments through the platform interface.

– **Integrated Messaging System:** The platform provides built-in communication tools allowing clients and professionals to exchange details, clarify requirements, and coordinate service logistics.

– **Price Estimation Tools:** Thumbtack offers approximate price ranges based on historical service data, increasing financial transparency before confirming a request.

– **Review and Reputation System:** After service completion, customers evaluate professionals, contributing to a trust-based ecosystem.

C. Advantages and Limitations of Thumbtack

Key advantages and limitations of Thumbtack:

– **Advantages:**

– Wide coverage across hundreds of service categories.

– Smart matching reduces search time and improves provider relevance for clients.

– Strong review system builds trust and ensures provider accountability.

– User-friendly interface accessible via both web and mobile application.

– Price estimation tools provide financial transparency before service confirmation.

– **Limitations:**

– The platform does not support video-based problem submission.

– Geographically limited to the U.S. market and not adapted to Algerian, African, or developing-country service environments.

– Pricing structures and payment systems are tailored to U.S. financial infrastructure and do not align with local economic contexts elsewhere.

– No support for Arabic language or local currency, limiting accessibility for North African users.

1.4.2 ServiceMarket (Regional Platform)

ServiceMarket [8] is a UAE-based digital marketplace specialized in home services across the Middle East and North Africa (MENA) region. The platform allows customers to request maintenance or household services and receive quotations from multiple providers, supporting comparison-based decision-making and structured booking workflows. It aims to improve transparency in the home service sector by aggregating verified companies and enabling customers to evaluate offers before committing to a provider.

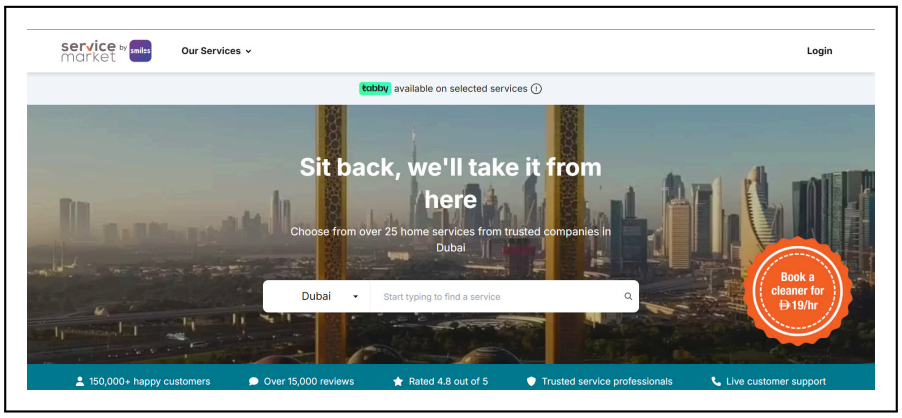


Figure 1.2: ServiceMarket platform interface.

A. Main Functions of ServiceMarket

The main capabilities of ServiceMarket include the following key functions [8]:

– **Multi-Quote Comparison System:** Users submit a service request and receive multiple quotations from competing providers, allowing price and service comparison before confirmation.

– **Scheduling and Booking Interface:** Customers can choose preferred dates and times for service execution directly through the platform.

– **Verified Provider Profiles:** Companies are presented with ratings, reviews, certifications, and experience indicators to support trust and reliability.

– **Customer Support and Mediation:** The platform offers assistance for dispute resolution, service complaints, and booking management.

– **Service Lifecycle Tracking:** Users can monitor booking confirmation, provider assignment, and service completion status.

B. Advantages and Limitations of ServiceMarket

Key advantages and limitations of ServiceMarket:

– **Advantages:**

– Adapted to the MENA regional context with Arabic language support.

– Multi-quote system gives clients better control over pricing and provider selection.

– Covers a wide range of home service categories.

– Transparent provider profiles improve user trust and decision-making.

– **Limitations:**

– Does not include video-based diagnosis.

– Service requests may require additional communication steps before the provider fully understands the problem details.

– The platform mainly focuses on general booking and quotation workflows without advanced real-time interaction features.

– Provider selection depends heavily on manual browsing and comparison by the customer, which may reduce efficiency.

– Its services and operational model are primarily optimized for Gulf countries rather than the Algerian local market.

1.4.3 Kandua (African Platform)

Kandua [9] is a South African online platform connecting homeowners with vetted professionals specializing in home maintenance, renovation, and improvement services. The platform emphasizes trust and quality assurance by verifying service providers before allowing them to receive customer requests. It is particularly oriented toward ensuring reliability in markets where informal service providers are common, offering customers safer access to qualified technicians.

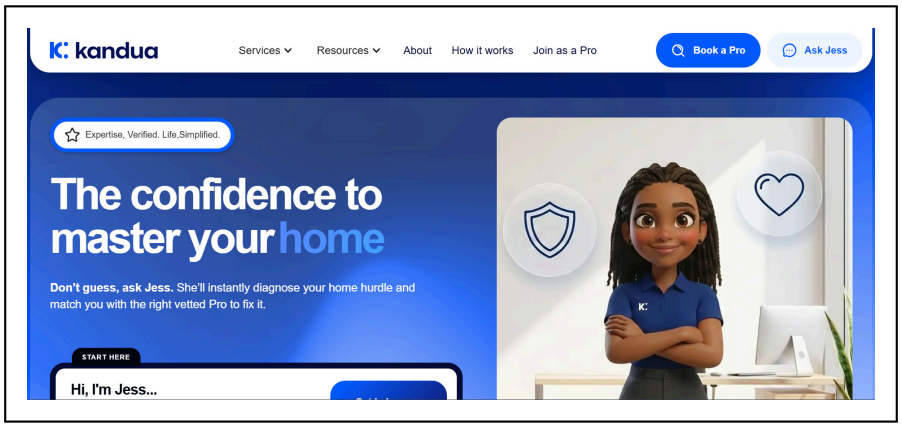


Figure 1.3: Kandua platform interface.

A. Main Functions of Kandua

Key functions of Kandua [9]:

– **Verified Professional Network:** Providers undergo screening and background checks before joining the platform, significantly increasing service reliability and user safety.

– **Structured Request Submission:** Customers submit maintenance needs using guided forms describing the problem, location, and service category.

– **Provider Responses and Quotes:** Professionals can respond to requests with availability and pricing information.

– **Messaging and Coordination Tools:** Clients and technicians communicate through the platform for clarification and scheduling.

– **Rating and Review Mechanism:** Post-service feedback contributes to provider reputation and ongoing service quality monitoring.

B. Advantages and Limitations of Kandua

Key advantages and limitations of Kandua:

– **Advantages:**

– Strong vetting process ensures high-quality and trustworthy service providers.

– Adapted to the African market context and local economic conditions.

– Covers a wide variety of home maintenance and improvement services.

– Background check mechanism builds user trust and improves service safety.

– **Limitations:**

– No support for video-based maintenance requests.

– Geographic coverage is concentrated mainly in South Africa with limited expansion elsewhere on the continent.

– The interface and workflows are not adapted to North African languages or administrative environments.

– No automated technician matching algorithm; client-provider connection depends on manual provider response.

1.4.4 De9De9 (Local Algerian Platform)

De9De9 [10] is an Algerian digital service platform developed to connect local users with technicians and service providers across several domains including maintenance, repair, and professional household services. It represents one of the earliest attempts to digitize service discovery within the Algerian market and improve visibility for local service providers. The platform focuses primarily on listing professionals and facilitating direct contact between customers and technicians.

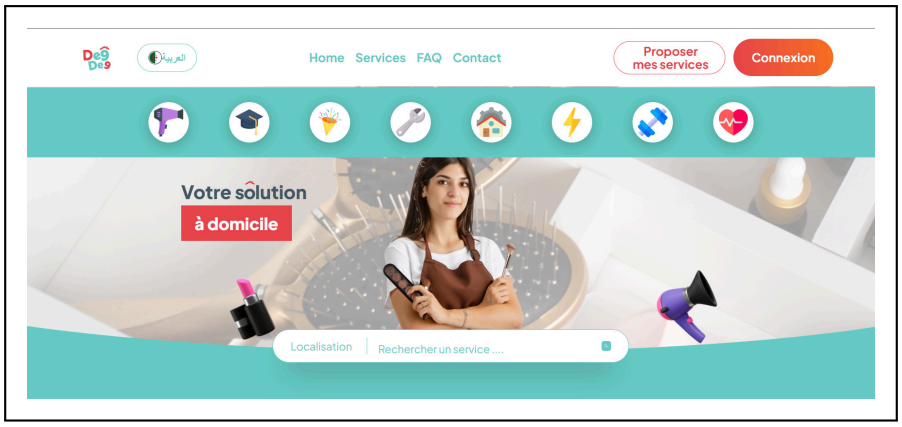


Figure 1.4: De9De9 platform interface.

A. Main Functions of De9De9

Key functions of De9De9 [10]:

– **Categorized Service Directory:** Users can browse technicians by specialization such as plumbing, electricity, repair, or installation, organized by category and region.

– **Request Posting System:** Clients can publish maintenance needs and wait for professionals to respond with availability and pricing.

– **Provider Profiles:** Basic profiles display service categories, region, and limited provider information to help users make initial contact decisions.

– **Regional Search Filters:** The platform allows filtering technicians by city or region to find nearby providers.

– **Rating and Commenting System:** Customers can provide feedback and ratings after service completion.

B. Advantages and Limitations of De9De9

Key advantages and limitations of De9De9:

– **Advantages:**

– Adapted to the Algerian market with Arabic and French language support.

– Covers local service needs specific to Algerian households and businesses.

– Provides visibility to small and informal service providers who lack digital presence.

– Accessible via mobile, making it convenient for Algerian users.

– **Limitations:**

– The platform does not support video-based problem description or visual diagnosis features.

– Although the platform includes in-app messaging, it does not provide real-time tracking of service providers during interventions.

– Provider profiles contain limited professional details, making service quality evaluation more difficult for users.

– The platform lacks advanced tools for managing and documenting service history and intervention records.

– The overall interface and workflow remain relatively simple compared to modern intelligent service platforms.

1.4.5 Khelliha 3lina (Local Algerian Platform)

Khelliha 3lina [11] is an Algerian digital platform designed to connect individuals with service providers for various everyday needs, including home maintenance, cleaning, delivery, and small repair tasks. The platform focuses on simplifying service access for users by allowing them to quickly find nearby providers and request services in a practical and user-friendly way. It represents a modern attempt to digitize local services and improve convenience in the Algerian urban environment.



Figure 1.5: Khelliha 3lina platform interface.

A. Main Functions of Khelliha 3lina

Key functions of Khelliha 3lina [11]:

– **Service Request Posting:** Users can submit service requests by specifying their needs and waiting for providers to respond.

– **Multi-Service Categories:** The platform covers various services such as maintenance, cleaning, transportation, and daily assistance.

– **Provider Listings:** Users can browse available service providers with basic information and contact details.

– **Location-Based Search:** Allows users to find nearby providers based on their geographical location.

– **Basic Communication:** Enables initial contact between clients and service providers.

B. Advantages and Limitations of Khelliha 3lina

Key advantages and limitations of Khelliha 3lina:

– **Advantages:**

  – Offers a wide range of daily services in one platform.

  – Adapted to Algerian users with simple and practical usage.

  – Facilitates quick access to nearby service providers.

– Helps informal workers gain visibility and job opportunities.

– **Limitations:**

– No support for video-based service requests or visual explanation of technical issues.

– The platform focuses mainly on connecting users with providers without advanced service management functionalities.

– Communication and coordination processes remain relatively basic and may require external contact methods.

– Limited mechanisms for provider verification and service quality assurance.

– The platform does not provide advanced tracking or real-time monitoring of service execution.

– The platform's interface design is relatively outdated and does not provide a modern or highly interactive user experience.

1.5 Proposed Solution: Fixoo Platform

To address the problems of traditional maintenance services in Algeria, this project introduces **Fixoo** a simple and modern digital platform. the platform allows clients to request maintenance services by recording a short video that clearly shows the problem. Technicians can watch the video, understand the issue remotely, it's designed specifically for the Algerian market to make finding reliable and fast maintenance services easier, faster, and more transparent for everyone..

1.5.1 Fixoo Definition

Fixoo is a modern web and mobile digital platform designed to revolutionize the way home and facility maintenance services are requested and managed in Algeria. The platform allows clients to easily record or upload a short video (maximum 60 seconds) that clearly describes their maintenance issue. Registered technicians can then watch the video remotely, understand the problem, and submit competitive service offers before making a physical visit. **Fixoo** centralizes everything in one secure and organized

environment: service requests, technician assignment, client-technician communication, real-time location tracking, and complete service history. By combining video-based diagnosis with smart request management and geolocation technology, Fixoo fills the gaps found in traditional maintenance platforms and delivers a fully localized, efficient, and transparent experience tailored specifically for the Algerian market.

1.5.2 Main Objectives of Fixoo

Fixoo has been designed with several key objectives in mind, all aimed at improving the efficiency, transparency, and quality of maintenance service delivery:

Enable video-based maintenance request submission: Allow clients to record or upload a short video (maximum 60 seconds) illustrating their technical problem, enabling more accurate remote diagnosis by technicians.

Support a competitive offer system: Allow technicians to browse published video requests and submit offers including pricing and estimated completion time, enabling clients to compare and select the best option.

Automate request management and status tracking: Centralize all requests in a unified system with clearly defined status stages (pending, in progress, completed, cancelled) visible to all parties.

Provide real-time geolocation tracking: Enable clients to track the technician's location on an interactive map from the moment the offer is accepted until arrival and task completion.

Guarantee service quality and accountability: Include a post-service rating and feedback system allowing clients to evaluate technicians, and enable clients to save or follow preferred service providers.

Adapt the platform to the Algerian context: Deliver a fully localized experience supporting Arabic, French, and English.

1.5.3 Comparison with International Platforms

In order to better position the proposed Fixoo platform within the global ecosystem of maintenance service solutions, a comparative study is conducted against well-known international platforms. This comparison focuses on identifying key functional differences and highlighting the advanced features introduced by Fixoo, particularly in terms of video-based interaction, offer management, complaint handling, service tracking, and localization.

Comparison Criteria	**Thumbtack**	**ServiceMarket**	**Kandua**	**Fixoo**
Geographic Coverage	USA only	MENA (UAE-focused)	South Africa	Algeria (Fully adapted)
Video-Based Request	Not available	Not available	Not available	Available (Video complaint/request upload)
Competitive Offers	Partial	Available	Not available	Fully supported with provider comparison
Provider Selection	Manual	Manual	Manual	Client selects the most suitable offer
Real-Time Tracking	Not available	Limited	Not available	Available
Complaint Management	available	available	Not available	Integrated complaint submission system
In-App Chat	Available	Limited	Limited	Integrated real-time chat

Comparison Criteria	**Thumbtack**	**ServiceMarket**	**Kandua**	**Fixoo**
Provider Verification	Partial	Company-level	Strong verification	Identity and document verification
Arabic Support	Not available	Available	Not available	Fully supported
Rating and Review System	Available	Available	Available	Available after task completion

Table 1.3: Comparison between Fixoo and International Platforms

The comparative analysis highlights Fixoo as a more advanced solution compared to international platforms such as Thumbtack, ServiceMarket, and Kandua. Unlike these platforms, Fixoo combines video-based requests, competitive offers, complaint management, real-time tracking, provider evaluation, and integrated communication within a single platform. In addition, Fixoo is specifically designed for the Algerian environment through Arabic language support, simplified workflows, and locally adapted service management mechanisms. These characteristics make Fixoo a modern, scalable, and user-centered platform in the maintenance service sector.

1.5.4 Comparison with Algerian Platforms

To evaluate the relevance of the Fixoo platform within the Algerian context, a comparative analysis is conducted with existing local platforms. This comparison highlights the limitations of current solutions and demonstrates how Fixoo introduces advanced features that significantly improve transparency, communication, service management, and user experience.

Comparison Criteria	**Da9Da9**	**Khalliha 3lina**	**Fixoo**
Geographic Coverage	Algeria	Algeria	Fully adapted to Algeria
Video Request and Complaint Upload	Not available	Not available	Available
Competitive Offers	Not available	Limited	Fully supported
Provider Selection	Direct contact	Basic provider listing	Comparison and selection of the best offer
Real-Time Tracking	Not available	Not available	Available
Complaint Management	Not available	available	Integrated complaint submission system
In-App Chat	available	Not available	Real-time chat system
Provider Verification	Advanced	Limited	Strong verification and validation
Rating System	Available	Available	Available after task completion

Table 1.4: Comparison between Fixoo and Algerian Platforms

The comparative analysis shows that Fixoo is a more advanced solution than existing Algerian platforms such as Da9Da9 and Khalliha 3lina. While current platforms mainly focus on connecting users with service providers, Fixoo introduces additional features including video-based requests, offer comparison, provider selection, task lifecycle management, complaint handling, integrated chat, and real-time tracking. These functionalities improve transparency, reliability, and interaction quality, making Fixoo a comprehensive and user-friendly platform that better addresses the needs of Algerian users.

1.6 Conclusion

This chapter presented the key concepts, objectives, and terminology related to digital maintenance service platforms, along with a comparative analysis of four platforms: Thumbtack, ServiceMarket, Kandua, and Da9Da9. The analysis highlighted common limitations, such as the absence of video-based request submission, limited real-time communication, lack of geolocation tracking, and poor adaptation to the Algerian context.

These shortcomings justify the need for the Fixoo platform, which aims to address these gaps through improved features and centralized service management. The next chapter focuses on the analysis and specification phase to define system requirements and guide the platform's design and development.

Chapter 2

Analysis and Specifications

2.1 Introduction

Developing any software application requires a rigorous preliminary phase of analysis and specification, which forms the foundation of the entire project. Without a clear and thorough understanding of needs and requirements, the risk of developing a system that fails to meet user expectations increases significantly.

This chapter is dedicated to this essential preparatory phase for the Fixoo platform. It begins with a comprehensive needs analysis that identifies the challenges faced by clients and technicians in the current maintenance service landscape. Based on this analysis, the general and specific objectives that the platform aims to achieve are defined.

The chapter then presents the complete system specifications, including a detailed description of the platform, identification of the actors who will interact with it, functional and non-functional requirements, as well as technical and organizational constraints. By the end of the chapter, a clear and comprehensive vision of the Fixoo platform is established, providing a solid foundation for the subsequent design and implementation phases.

2.2 Needs Analysis

To design the Fixoo system for effectively connecting clients with qualified main-

tenance service providers, it is first essential to identify the current challenges and actual needs of both sides. Many clients still rely on informal methods to find technicians, while technicians lack efficient tools to discover and respond to service opportunities. Therefore, there is a need for a unified digital platform that organizes the service request and response process in an orderly, transparent, and accessible manner for all stakeholders. The following are the key needs and challenges that the Fixoo system aims to address:

– Heavy reliance on phone calls and informal contacts, which slows down the process and makes it difficult to track requests or compare service providers.

– Inability to accurately describe maintenance problems through text alone, leading to misunderstandings and inefficient diagnosis before technician arrival.

– Lack of a competitive offer mechanism, preventing clients from comparing prices and delivery times before selecting a service provider.

– No real-time tracking of the technician's location after offer acceptance, creating uncertainty and reducing client confidence during service execution.

– Difficulty verifying technician qualifications and identity, increasing the risk of hiring unqualified or untrustworthy service providers.

– Absence of a structured service history, making it impossible for clients to refer to past interventions or for technicians to showcase their completed work and ratings.

– Lack of an integrated notification system, causing delays in communication between clients, technicians, and the platform administration.

– Poor adaptation of existing solutions to the Algerian geographic and administrative context, including the absence of Arabic language support and local address structures.

2.3 General Objective

The Fixoo project aims to design and develop an integrated mobile and web platform dedicated to connecting clients with qualified maintenance service providers in a

modern, efficient, and transparent manner. The system is based on video-based problem description, enabling clients to communicate technical issues more accurately, and on a competitive offer model, allowing technicians to propose their services with pricing and delivery time.

The system aims to replace traditional informal methods that rely on phone calls and word-of-mouth recommendations by providing an integrated solution that ensures organized request management, transparent offer comparison, and real-time service tracking. It also contributes to enhancing trust between clients and technicians through a rigorous identity verification process for service providers and a post-service rating mechanism.

The Fixoo system gives special attention to the geolocation dimension, enabling clients to share their location when submitting a request and to track the technician's movement on an interactive map from acceptance to arrival. This helps optimize response times and improves the overall service experience.

In addition, the system supports administrative oversight through a dedicated control panel that enables the platform management team to monitor user activity, manage service categories, oversee financial transactions, and generate analytical reports.

2.4 Specific Objectives

The main objectives of the Fixoo system are defined based on the analysis of system interfaces and functional services implemented within the platform architecture.

- **Authentication and Account Management**
  - Allow users (clients and technicians) to register and log in securely.
  - Support phone number verification using OTP during registration.
  - Enable password recovery and reset functionality.
  - Allow users to upload and confirm profile images.
  - Support session management (refresh token and logout).
- **Client Request (Video) Management**

  - Allow clients to record or upload a video (up to 60 seconds) describing the maintenance problem.
  - Enable adding descriptions, categories, and location details.
  - Allow clients to publish, edit, or delete service requests.
  - Manage request lifecycle (draft, published, in progress, completed, cancelled).

- **Competitive Offer System**
  - Allow technicians to browse video requests based on category and region.
  - Enable submission, update, and deletion of service offers.
  - Allow clients to view and compare offers (price and duration).
  - Allow clients to accept an offer and automatically assign the technician.

- **Project and Task Management**
  - Convert accepted offers into structured service projects.
  - Allow technicians to start and complete tasks.
  - Allow clients to monitor project progress.
  - Maintain project history for both client and provider.

- **Real-Time Geolocation Tracking**
  - Allow technicians to update their real-time location.
  - Enable clients to track technician location during service execution.
  - Calculate distance between client and provider.

- **Find Provider (Direct Search)**
  - Allow clients to search for registered providers by category, specialization, and location.
  - Display only providers who are subscribed and approved by the administrator.
  - Allow clients to view the provider's public profile and contact them directly .

- **Service Provider Profile Management**

  - Allow providers to register with professional information and documents.
  - Enable document upload and verification workflow.
  - Manage provider specializations and categories.
  - Display public profile including reviews and completed projects.

- **Rating and Review System**
  - Allow clients to rate and review providers after service completion.
  - Maintain review history and rating averages.
  - Use ratings for provider ranking and recommendation.

- **Notification and Communication System**
  - Send real-time notifications for key events (offers, acceptance, task updates).
  - Manage in-app notifications (read/unread, history).
  - Send email notifications for important actions (offers, approvals, payments).

- **In-Project Chat System**
  - Allow clients and providers to exchange messages within the context of an accepted project.
  - Enable real-time messaging).
  - Maintain full chat history linked to each project.
  - Restrict communication to project participants only.

- **Complaint and Dispute Management**
  - Allow clients to submit complaints during or after service execution.
  - Enable administrators to review and resolve complaints.
  - Support provider suspension based on complaint outcomes.

- **Payment and Financial Management**
  - Track provider earnings and platform commissions.
  - Manage provider debts and payments.

  - Allow administrators to record payments.
  - Generate financial summaries and revenue reports.

- **Category and Service Management**
  - Manage service categories and specializations.
  - Enable dynamic addition and modification of categories.
  - Support multilingual category names (Arabic / English).

- **Administrative Control Panel**
  - Manage users and providers (activation, suspension, verification).
  - Monitor platform activity and statistics.
  - Export reports (users, providers, financial data).
  - Manage commission rates and system configurations.

- **User Interface**
  - Provide a mobile application (Flutter) for clients and providers.
  - Provide a web-based admin panel (Blazor Server).
  - Ensure responsive and user-friendly UI design.
  - Support Arabic, French, and English languages.

2.5 Specifications

This section presents the main specifications of the Fixoo system by providing a general description of its nature, identifying the different categories of users who will interact with it and their respective roles, outlining the functional requirements that define the services and features the system must provide, and the non-functional requirements that ensure performance quality, security, and ease of use.

2.5.1 System Description

The Fixoo system is a digital platform composed of two main interfaces: a mobile application (Flutter) targeting clients and service providers, and a web-based administrative panel (Blazor Server) dedicated to platform management.

The mobile application enables clients to publish maintenance requests via video, receive and compare service offers from registered technicians, track the technician's location in real time, communicate with the service provider, and evaluate the service after completion. Technicians use the same application to browse requests matching their specialization, submit competitive offers, and manage their professional profile and completed project history.

The administrative panel allows the management team to supervise all platform activity, including user account management, service category administration, financial monitoring, and the generation of analytical reports and statistics.

2.5.2 Identification of Actors

Identifying actors is a fundamental step in the system analysis process, as it helps clarify how different users interact with the system and defines their roles and responsibilities within its various functions. In the Fixoo system, the main actors are as follows:

A. **Client**

- **Role:** A registered user who submits maintenance service requests via video and interacts with technicians through the mobile application.
- **Key interactions with the system:**
  - Create an account and log in securely.
  - Record or upload a video (up to 60 seconds) describing the maintenance problem.
  - Add a text description, select the service category, and choose the location.
  - Publish, edit, or delete a service request.
  - Receive and compare service offers (price and delivery time).

  - Accept an offer and notify the selected technician.
  - Exchange real-time messages with the assigned provider within the project chat.
  - Search for providers by location, category, and specialization..
  - Track the technician's real-time location on an interactive map.
  - Rate the technician.
  - Submit complaints when necessary.

- **Objectives:**
  - Submit maintenance requests accurately and efficiently using video.
  - Compare service offers and select the best option.
  - Monitor service execution and ensure task completion with confidence.

B. **Provider (Technician)**

- **Role:** A verified professional registered on the platform who browses client video requests and offers maintenance services.
- **Key interactions with the system:**
  - Create a professional account with full credentials (specialization, experience, national ID, certifications).
  - Browse published video requests filtered by category and region.
  - Submit, edit, or delete service offers including price and estimated completion time.
  - Communicate with the client through the in-project chat after offer acceptance.
  - Receive notification upon offer acceptance by a client.
  - Share real-time location with the client during service execution.
  - Start and complete service tasks.
  - View completed projects and received ratings on the personal profile.
  - Subscribe to the platform (via administrator approval) to become visible in the Find Provider search results.
- **Objectives:**

  - Discover relevant service opportunities efficiently.
  - Build a professional reputation through ratings and completed project history.
  - Manage offers and professional profile in an organized manner.

C. **Administrator**

- **Role:** A platform-level manager responsible for supervising all system operations through the web-based administrative panel.
- **Key interactions with the system:**
  - Manage client accounts (view, activate, deactivate, delete).
  - Review and approve or reject technician registration documents.
  - Manage service categories (add, edit, delete, reorder).
  - Monitor financial balances, commissions, and provider debts.
  - Manage provider subscriptions and control their visibility in the Find Provider search feature.
  - Send payment notifications and administrative messages to users.
  - Generate monthly, annual, and category-based reports exportable to Excel or PDF.
  - Handle complaints and disputes.
  - View an interactive statistics dashboard covering platform activity.
- **Objectives:**
  - Ensure proper functioning and supervision of all platform activities.
  - Maintain data quality, user trust, and service category structure.
  - Support decision-making through accurate analytical reports and statistics.

2.5.3 Functional Requirements

Functional requirements represent the set of functions and services that the Fixoo system must provide in order to meet the needs of different users according to their roles within the system.

- **Account Management**
  - Ensure that all system users have personal accounts with profile information.
  - Provide a secure password-based login system with error alerts.
  - Support phone number verification via OTP during registration.
  - Allow users to update their personal information and profile photo.
  - Enable secure password recovery in case it is forgotten.
  - Allow account deactivation or deletion while maintaining secure system records.
- **Video Request Management**
  - Allow clients to record or upload a video of up to 60 seconds describing the maintenance problem.
  - Enable clients to add a text description and select the appropriate service category.
  - Allow clients to define their location .
  - Support publishing, editing, and deleting service requests.
  - Manage the full request status lifecycle (pending, in progress, completed, cancelled).
- **Offer System**
  - Allow technicians to browse published requests filtered by category and geographic region.
  - Enable technicians to submit service offers including price and estimated completion time.
  - Allow technicians to edit or delete submitted offers before client selection.
  - Allow clients to view, compare, and accept the most suitable offer.
  - Automatically notify the selected technician upon offer acceptance.
- **Geolocation and Tracking**
  - Allow clients to share their location when publishing a service request.

  - Enable real-time tracking of the technician's location on an interactive map after offer acceptance.
  - Calculate the distance between the client and the technician.
  - Sending an automatic notification to the client when the task starts.

- **Rating and Review System**
  - Allow clients to rate technicians and leave written comments after service completion.
  - Display cumulative ratings and review history on each technician's public profile.
  - Enable clients to save or follow preferred technicians for future requests.

- **Find Provider**
  - Allow clients to search for available providers by category, specialization, and geographic location.
  - Display only providers who have an active subscription approved by the administrator.
  - Allow clients to view the provider's public profile and initiate direct contact outside the video request workflow.

- **Notification System**
  - Send real-time push notifications for all key events.
  - Key notification events include: new offer received, offer accepted, task started, task completed, and administrative message received.
  - Allow the administrator to send broadcast messages and payment reminders to service providers.

- **Administrative Control Panel**
  - Enable the administrator to manage all client and provider accounts.
  - Review technician registration documents and approve or reject accounts.
  - Manage service categories and subcategories (add, edit, delete, reorder).

- Monitor financial balances, commissions, and provider debts.
- Generate monthly, annual, and category-based analytical reports.
- Export reports in Excel or PDF format.
- Display an interactive real-time statistics dashboard.

2.5.4 Non-Functional Requirements

Non-functional requirements are the general characteristics of the system that are not related to specific functions, but are essential to ensure that the system operates efficiently, reliably, and meets user expectations. These characteristics include:

- **Security:** Protect user data and credentials using JWT-based authentication, ensure secure access according to each user role, and apply HTTPS encryption for all data exchanges.
- **Reliability:** Ensure system stability with minimal errors, maintain continuous service availability, and handle unexpected failures gracefully.
- **Performance:** Provide fast response times for video uploads, offer processing, and map loading, optimized through efficient API design and cloud storage integration.
- **Usability:** Provide a clear, intuitive, and culturally adapted interface for all user types, with support for Arabic and French languages to ensure accessibility for Algerian users.
- **Scalability:** Enable the system to adapt to a growing number of users, requests, and geographic regions without affecting performance.
- **Maintainability:** Design the system architecture using clean code principles and modular structure (Clean Architecture) to facilitate future modifications and updates.
- **Compatibility:** Ensure the mobile application operates correctly on both Android and iOS devices, and the administrative panel functions on all modern web browsers.

2.5.5 Constraints

The Fixoo system is subject to a set of technical and organizational constraints to ensure efficient operation and meet the needs of different users. The main constraints are as follows:

- **Internet Connectivity:** Core system operations such as video upload, offer submission, real-time tracking, and push notifications require an active internet connection on both client and technician devices.
- **Video File Management:** Videos submitted by clients must be stored in a cloud-based file system (Azure Blob Storage or equivalent) to ensure fast streaming and minimize API server load.
- **Device Compatibility:** The mobile application must support a wide range of Android and iOS devices with varying screen sizes, ensuring a consistent user experience across all supported devices.
- **Security and Data Protection:** The system must ensure the confidentiality of users' personal data, including identity documents submitted by technicians, through secure storage and role-based access control.
- **Compliance with Algerian Regulations:** All data collection and processing operations must comply with applicable Algerian legal requirements regarding personal data protection and digital service provision.

2.6 Conclusion

This chapter concludes by providing a clear and comprehensive vision of the Fixoo system's needs and specifications, including functional and non-functional requirements as well as technical and organizational constraints. This vision provides a solid foundation for designing and implementing the Fixoo platform effectively and reliably, meeting all user expectations.

In the next chapter, we will move to the system design phase, covering the overall system architecture, UML diagrams, data modeling, and defining the main system modules.

Chapter 3

System Design

3.1 Introduction

This chapter presents the design of the Fixoo platform after defining the system requirements in the previous chapter. The purpose of this phase is to transform the functional needs of users, providers, and administrators into a structured technical model that can guide the implementation stage.

The design of Fixoo focuses on organizing the platform into clear components that support video-based service requests, offer management, provider task execution, user tracking, complaints, reviews, and administrative supervision. This chapter first explains the adopted system architecture, then presents the UML diagrams used to describe the main interactions and internal structure of the platform. Finally, it introduces the system modules that summarize the main services implemented in the application layer.

3.2 System Architecture

The Fixoo platform requires an architecture that can support several connected operations at the same time, such as uploading videos, browsing service requests, submitting offers, starting tasks, tracking providers, exchanging messages, and managing platform data. For this reason, the system was designed using a layered architecture where each part has a specific responsibility.

The architecture separates the user interfaces from the backend processing and the data storage mechanisms. This separation helps the system remain easier to develop, test, maintain, and extend. It also allows the mobile application, administration interface, API services, database, and cloud storage to communicate through well-defined interactions without mixing responsibilities.

3.2.1 Client-Server Architecture

The client-server model is based on a simple principle: a client application sends a request to a server, and the server processes this request and returns the appropriate response. The server is responsible for executing operations, applying rules, and accessing stored data, while the client is mainly responsible for presenting information and collecting user actions [12].

In the Fixoo platform, this model is applied through a Flutter mobile application used by requesters and providers, and an ASP.NET Core Web API that receives and handles their requests. The mobile application communicates with the backend through HTTP/HTTPS requests. The backend then processes the required operation, interacts with the database or storage services when needed, and returns the result to the application.

The administration side is implemented separately using a Blazor Server web interface. This interface allows the administrator to manage users, providers, categories, videos, payments, complaints, and reports. Therefore, the client-server organization provides a clear communication flow between the mobile application, the web administration panel, and the backend services.

3.2.2 Multi-Tiered Architecture

The Fixoo platform follows a Three-Tier Architecture composed of three main layers: the Presentation Layer, the Business Logic Layer, and the Data Layer. This organization helps separate the visual interface, the processing rules, and the storage mechanisms, which improves maintainability and scalability [15, 17].

- **Presentation Layer:** This layer represents the part of the system that users interact with directly. In Fixoo, it includes the Flutter mobile application used by requesters and providers, and the Blazor administration interface used by the administrator. Its role is to display data, collect inputs, and send requests to the backend services. It does not execute the core business rules of the platform, it only provides the interaction interface [16, 17].

- **Business Logic Layer:** This layer represents the central processing part of the Fixoo platform. It is implemented using ASP.NET Core Web API. It receives requests from the mobile and web interfaces, validates input data, checks user permissions, applies business rules, and coordinates operations such as creating video requests, submitting offers, accepting offers, starting tasks, completing projects, sending notifications, and handling complaints. It also manages communication with Azure Blob Storage for videos, thumbnails, profile images, and provider documents [12, 15, 17].

- **Data Layer:** This layer stores and manages the persistent data of the platform. Microsoft SQL Server is used to store structured data such as users, providers, offers, projects, complaints, reviews, notifications, categories, and transactions. Azure Blob Storage is used for unstructured files such as videos, thumbnails, profile images, certificates, identity cards, and self-employed cards. The Data Layer communicates only with the Business Logic Layer to keep the system secure and organized [15, 17].

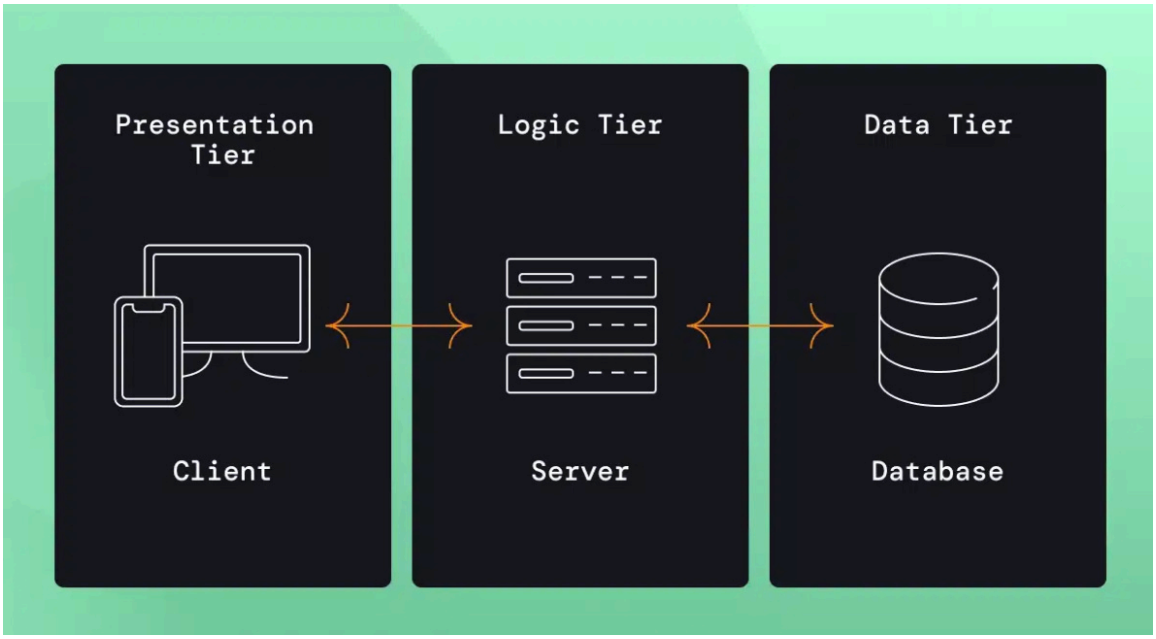


Figure 3.1: Three-Tier Architecture of the Fixoo Platform

While the overall Fixoo platform follows a Three-Tier Architecture, the internal organization of the Business Logic Layer is implemented using Clean Architecture principles. This combination improves separation of concerns, maintainability, scalability, and code organization inside the backend services [34, 17].

3.2.3 Clean Architecture Implementation

Inside the Business Logic Layer, the Fixoo backend is organized according to Clean Architecture principles. The solution is divided into separate projects, where each project has a specific responsibility. This separation makes the backend easier to maintain, test, and extend [34, 17].

- **Infrastructure Layer:** This layer is represented by the `SOVS.Infrastructure` project. It contains technical implementations such as data access, security, and services. It is responsible for communicating with Microsoft SQL Server, authentication mechanisms, storage services, and other external dependencies.
- **Domain Layer:** This layer is represented by the `SOVS.Domain` project. It contains the core elements of the system, such as entities, enums, interfaces, models,

and validation rules. This layer defines the main business concepts of the platform independently from the database, user interface, or external services.

- **Application Layer:** This layer is represented by the `SOVS.Application` project. It contains interfaces and services responsible for coordinating the main application operations. It connects the domain rules with the external infrastructure and manages workflows such as video requests, offers, projects, notifications, complaints, and provider operations.

- **Presentation Layer:** This layer is represented by the `SOVS.Users` and `SOVS.Admin` projects. The `SOVS.Users` project exposes API controllers for the mobile application, such as authentication, video requests, offers, payments, projects, notifications, and chat. The `SOVS.Admin` project represents the administration interface used to manage the platform.

Therefore, Clean Architecture is applied in Fixoo by separating the core business logic from the application services, technical infrastructure, and presentation projects. This organization reduces dependency between components and allows each part of the system to evolve independently.

3.3 UML Diagrams

This section presents the UML diagrams used to describe the behavior and structure of the Fixoo platform. These diagrams help clarify how the actors interact with the system, how the main entities are organized, and how the main video-based service request process flows from creation to completion.

3.3.1 Use Case Diagrams

In this project, use case diagrams are used to describe the services provided by Fixoo to each actor. They show what the Administrator, User, and Provider can do inside the platform and how some functions depend on or extend other functions [18].

- **Use Case Diagram – Administrator**

This diagram presents the main actions performed by the Administrator inside the Fixoo administration system (see Figure 3.2).

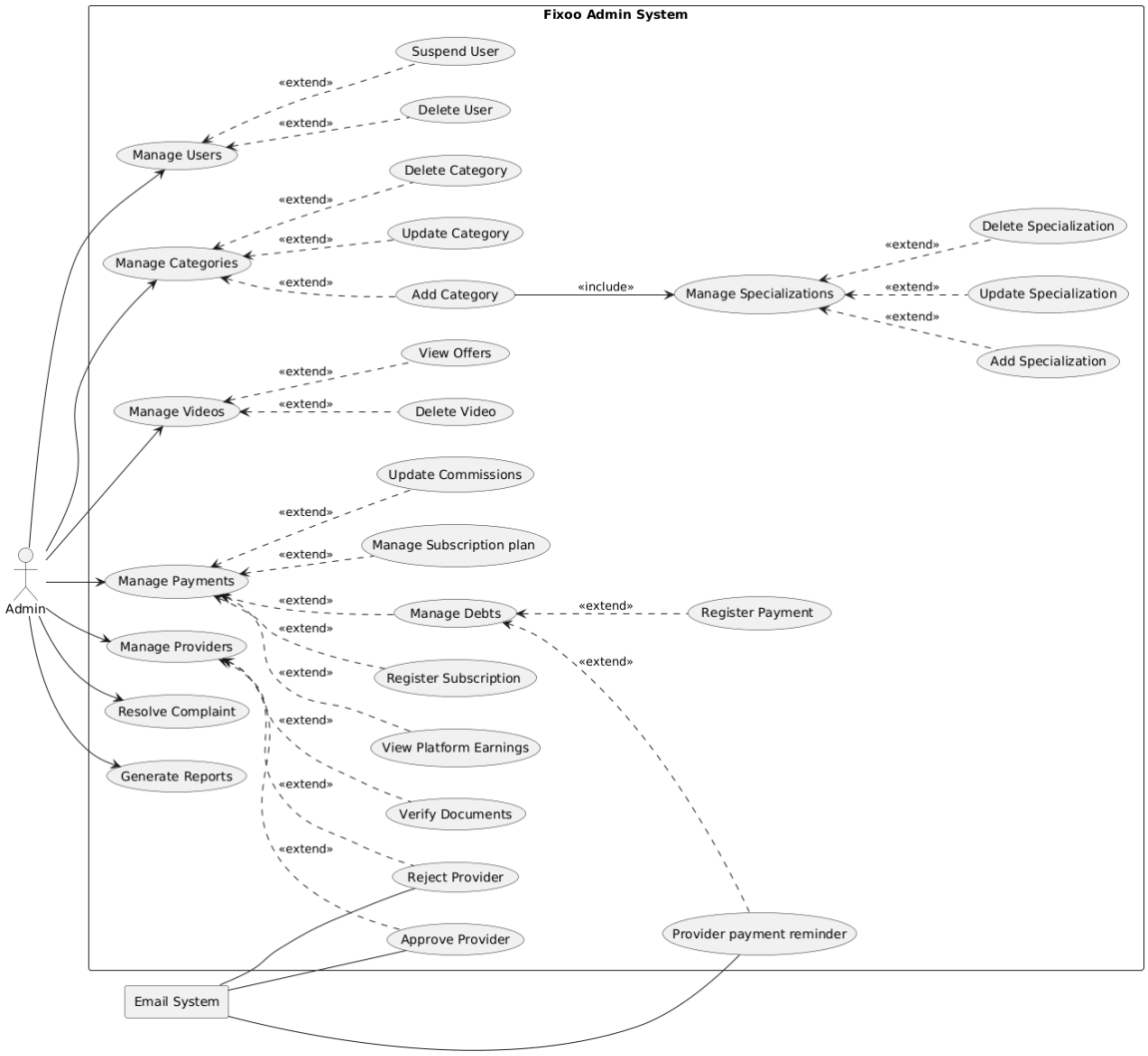


Figure 3.2: Use Case Diagram – Fixoo Admin System

Figure 3.2 shows how the administrator supervises the platform through several management functions. These functions include managing users, providers, categories, specializations, videos, payments, complaints, and reports. The diagram also includes the Email System as an external element used for sending provider approval, provider rejection, and payment reminder messages. The `<<extend>>` relations represent additional actions linked to a main administrative function, while the `<<include>>` relation indicates a required dependency between two use cases.

The following table (3.1) summarizes the main use cases of the Administra-

tor.

Use Case	Description	Relations
Manage Users	Enables the administrator to supervise user accounts and apply actions such as deleting or suspending accounts when required.	Extend: Delete User, Suspend User
Manage Providers	Enables the administrator to manage provider accounts by checking submitted documents and approving or rejecting provider registration.	Extend: Verify Documents, Approve Provider, Reject Provider
Manage Categories	Enables the administrator to organize service categories by adding new categories, updating existing ones, or deleting unused categories.	Extend: Add Category, Update Category, Delete Category
Manage Specializations	Enables the administrator to manage the specializations attached to service categories.	Extend: Add Specialization, Update Specialization, Delete Specialization
Manage Videos	Enables the administrator to supervise published video requests, remove inappropriate videos, and view the offers submitted for a video.	Extend: Delete Video, View Offers
Manage Payments	This feature also enables administrators to monitor financial operations, including supplier debts, platform profits, and commission and subscription settings.	Extend: Manage Debts, Update Commissions,Register Subscription,Manage Subscription plan, View Platform Earnings

Use Case	Description	Relations
Manage Debts	Enables the administrator to follow provider debt status and perform related actions such as registering a payment or sending a reminder.	Extend: Register Payment, Provider Payment Reminder
Resolve Complaint	Enables the administrator to examine complaints submitted by users and decide the appropriate resolution.	–
Generate Reports	Enables the administrator to generate reports related to users, providers, payments, and system activity.	–
Email System Interaction	Represents the external email system used to send approval, rejection, and payment reminder emails.	External System linked to Approve Provider, Reject Provider, Provider Payment Reminder

Table 3.1: Description Table – Administrator Use Cases

- **Use Case Diagram – User and Provider**

This diagram shows the main interactions between the User, the Provider, and the Fixoo system (see Figure 3.3).

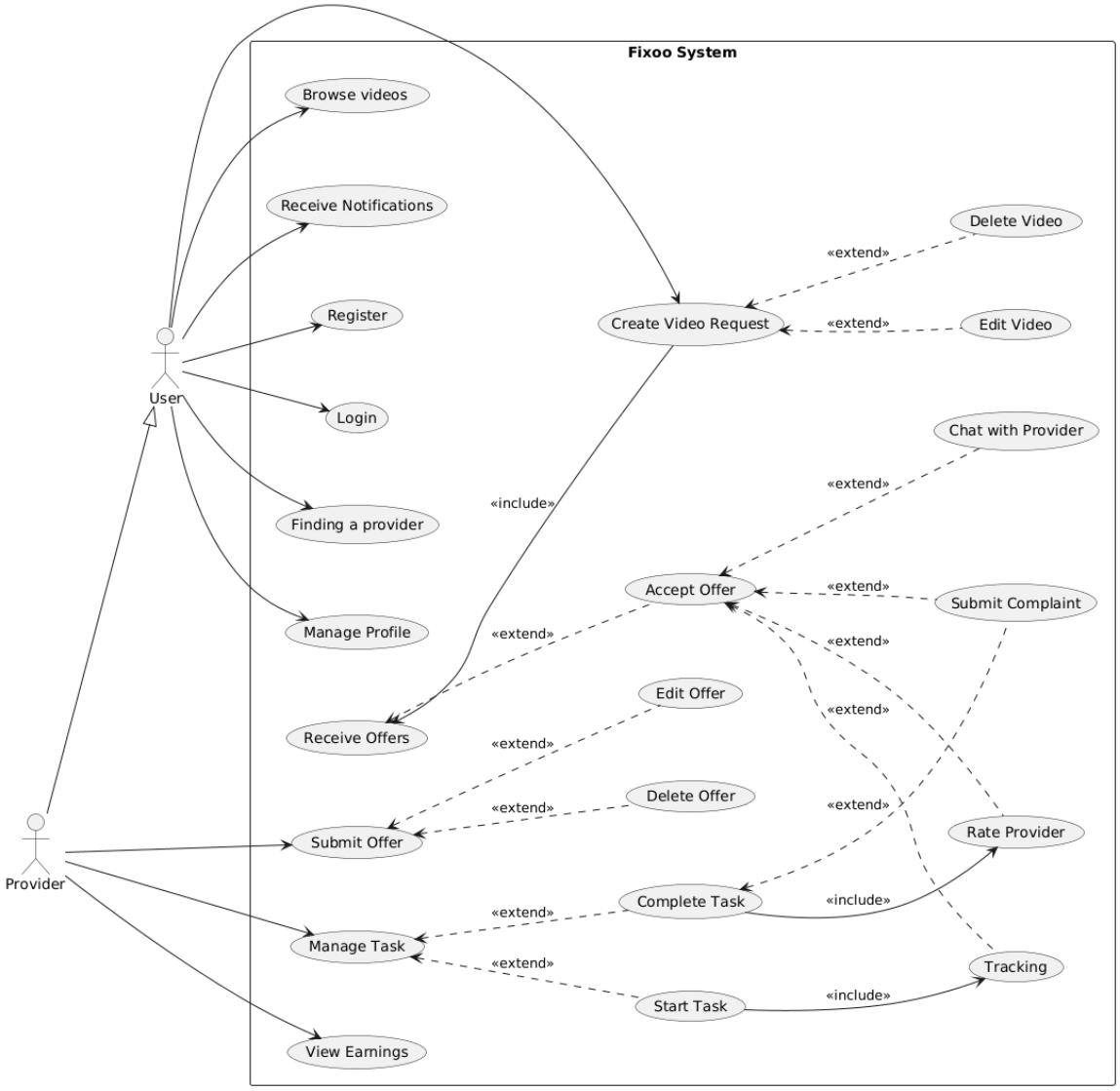


Figure 3.3: Use Case Diagram – Fixoo User and Provider System

Figure 3.3 presents the main actions available in the mobile part of the system. The User can create video requests, browse videos, receive and compare offers, accept an offer, communicate with the provider, follow task progress, submit complaints, and rate the provider,Finding a provider. The Provider can browse video requests, submit and manage offers, start and complete tasks, and view earnings. The `<<include>>` and `<<extend>>` relationships describe the logical links between these actions.

The following table (3.2) summarizes the main use cases of the User and the Provider.

Use Case	Description	Relations
Register / Login	Allows a User or Provider to access the platform by creating an account or signing in with valid credentials.	–
Manage Profile	Allows the connected actor to view and update personal profile information.	–
Finding a provider	Ability to search for a presenter by location, category, and specialization.	–
Create Video Request	Allows the User to create a maintenance request based on a video that explains the problem.	Include: Receive Offers; Extend: Edit Video, Delete Video
Edit Video	Allows the User to update the information of a created video request when changes are needed.	Extends: Create Video Request
Delete Video	Allows the User to remove a video request from the system if it should no longer be available.	Extends: Create Video Request
Browse Videos	Allows users and providers to view available video requests published in the platform.	–
Receive Offers	Allows the User to receive offers from providers after publishing a video request.	Extended by: Accept Offer
Accept Offer	Allows the User to choose one suitable offer from the received offers.	Extend: Rate Provider, Submit Complaint, Chat with Provider, Tracking
Rate Provider	Allows the User to evaluate the Provider after the service process.	Extends: Accept Offer, Included by: Complete Task
Submit Complaint	Allows the User to submit a complaint related to an accepted offer or a completed task.	Extends: Accept Offer, Complete Task

Use Case	Description	Relations
Chat with Provider	Allows the User to communicate with the Provider after accepting an offer.	Extends: Accept Offer
Tracking	Allows the User to follow the Provider after the task starts.	Extends: Accept Offer; Included by: Start Task
Receive Notifications	Allows the User to receive updates related to offers, tasks, and system actions.	–
Submit Offer	Allows the Provider to propose a service offer for a video request.	Extend: Edit Offer, Delete Offer
Edit Offer	Allows the Provider to update a submitted offer when needed.	Extends: Submit Offer
Delete Offer	Allows the Provider to delete a submitted offer before it is accepted.	Extends: Submit Offer
Manage Task	Allows the Provider to manage the task created after an offer is accepted.	Extend: Start Task, Complete Task
Start Task	Allows the Provider to begin the accepted service task.	Include: Tracking
Complete Task	Allows the Provider to mark the assigned task as completed.	Include: Rate Provider, Extend: Submit Complaint
View Earnings	Allows the Provider to view financial earnings related to completed work.	–

Table 3.2: Description Table – User and Provider

3.3.2 Class Diagram

In Fixoo, the class diagram is used to represent the main data entities of the system and the relationships between them. It helps describe how users, providers, video requests, offers, projects, complaints, reviews, categories, chat rooms, and transactions are connected in the platform [13].

Figure 3.4 presents the class diagram of the Fixoo platform. It gives a structural view of the entities used to support the main platform operations.

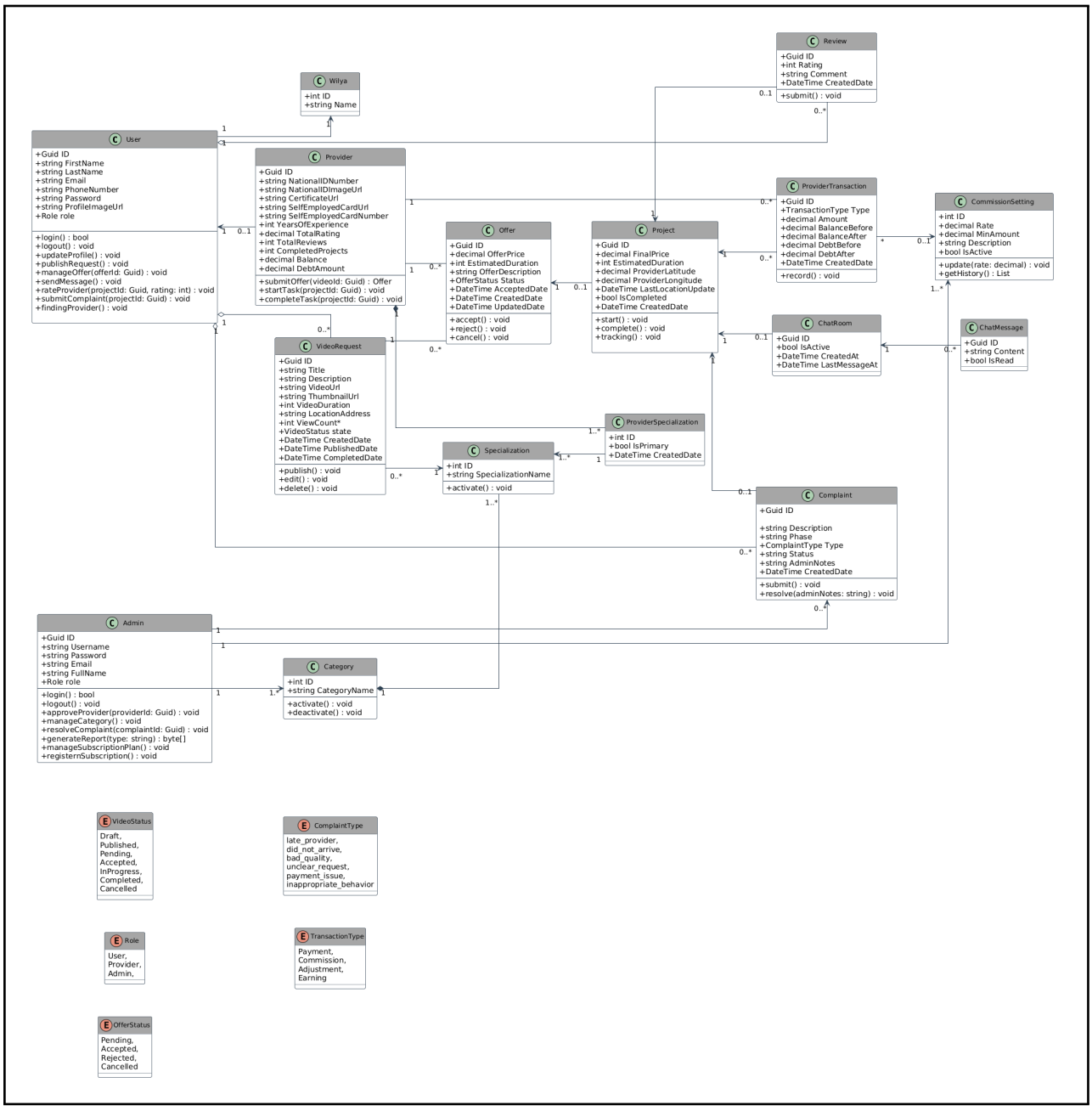


Figure 3.4: Class Diagram of the Fixoo Platform.

a. Class Diagram Description

This section explains the classes shown in the Fixoo class diagram and clarifies the role of each one in the system.

- The **User** class represents the primary account used on the platform. It stores shared account data such as first name, last name, email address, phone number, password, profile picture link, and role. Through this class, the user can perform actions such as logging in, updating their profile, posting video requests, searching for a shared provider, managing offers, sending messages, rating service providers, and submitting complaints.
- The **Wilya** class represents the wilaya linked to the User. It contains an identifier and a name, and each User is associated with one Wilya.
- The **Provider** class represents the professional profile connected to a User account. A User may have zero or one Provider profile, while each Provider belongs to one User. This class stores provider-related information such as national ID number, national ID image URL, certificate URL, self-employed card URL, self-employed card number, years of experience, total rating, total reviews, completed projects, balance, and debt amount. The Provider can submit offers, start tasks, and complete tasks.
- The **Admin** class represents the administrator account used to manage the platform. It is separated from the User class and contains administrator information such as username, password, email, full name, and role. The Admin can approve providers, manage categories,manage Subscription Plans,Subscription registration, resolve complaints, manage commission settings, and generate reports.
- The **VideoRequest** class represents a video-based maintenance request created by a User. It stores information such as title, description, video URL, thumbnail URL, video duration, location address, view count, status, creation date, publication date, and completion date. One User can create many video requests, and each VideoRequest is linked to one Specialization and can receive several Offers.
- The **Offer** class represents a proposal submitted by a Provider for a specific VideoRequest. It stores the proposed price, estimated duration, description, status, accepted date, created date, and updated date. A Provider can submit many offers, and one VideoRequest can receive many offers. When an offer is accepted, it can be associated with a Project.

- The **Project** class represents the accepted service task created after an offer is approved. It stores the final price, estimated duration, provider location coordinates, last location update, completion state, and creation date. It also includes operations to start, complete, and track the project. A Project can be linked to a Review, a Complaint, a ChatRoom, and provider transactions.

- The **Review** class represents the evaluation submitted by a User after a service process. It contains the rating value, written comment, and creation date. A User can submit multiple reviews, and each Review may be related to one Project.

- The **Complaint** class represents a complaint submitted by a User. It includes the complaint description, phase, type, status, admin notes, and creation date. A complaint may be linked to a Project, and the Admin can handle and resolve it by adding administrative notes.

- The **Category** class represents a general service category available in the platform. It stores the category identifier and category name. A Category can contain one or more Specializations, and the Admin can manage platform categories.

- The **Specialization** class represents a specific skill or service type under a Category. It contains a specialization identifier and specialization name. A VideoRequest belongs to one Specialization, while a Specialization can be linked to several video requests.

- The **ProviderSpecialization** class represents the link between Provider and Specialization. It stores whether the specialization is primary and the creation date. This class allows a Provider to have one or more specializations.

- The **ChatRoom** class represents the communication space associated with a Project. It contains the chat room identifier, active state, creation date, and last message date. A Project can have zero or one ChatRoom.

- The **ChatMessage** class represents a message exchanged inside a ChatRoom. It stores the message identifier, content, and read status. A ChatRoom can contain multiple messages.

- The **ProviderTransaction** class represents financial movements related to a Provider. It stores the transaction type, amount, balance before and after, debt before and after, and creation date. A Provider can have multiple transactions, and a Project can generate several provider transactions.

- The **CommissionSetting** class represents the commission configuration used by the platform. It stores the commission rate, minimum amount, description, and active state. The Admin manages these settings, and provider transactions may reference them.
- The diagram includes four enumeration types. The **Role** enumeration defines the roles: User, Provider, and Admin. The **VideoStatus** enumeration defines the status values of a video request: Draft, Published, Pending, Accepted, InProgress, Completed, and Cancelled. The **OfferStatus** enumeration defines the status values of an offer: Pending, Accepted, Rejected, and Cancelled. The **TransactionType** enumeration defines provider transaction types: Payment, Commission, Adjustment, and Earning.
- The **ComplaintType** enumeration defines the possible complaint categories: late_provider, did_not_arrive, bad_quality, unclear_request, payment_issue, and inappropriate_behavior.

3.3.3 Activity Diagrams

In this project, activity diagrams are used to describe the order of actions that occur during a specific process. They are useful for showing decisions, alternative paths, and the progression of the service request workflow in Fixoo [14].

• Activity Diagram – Maintenance Service Request Process

This activity diagram (Figure 3.5) describes the main process followed by a User to request a maintenance service through a video, receive offers, accept one offer, and follow the service until completion.

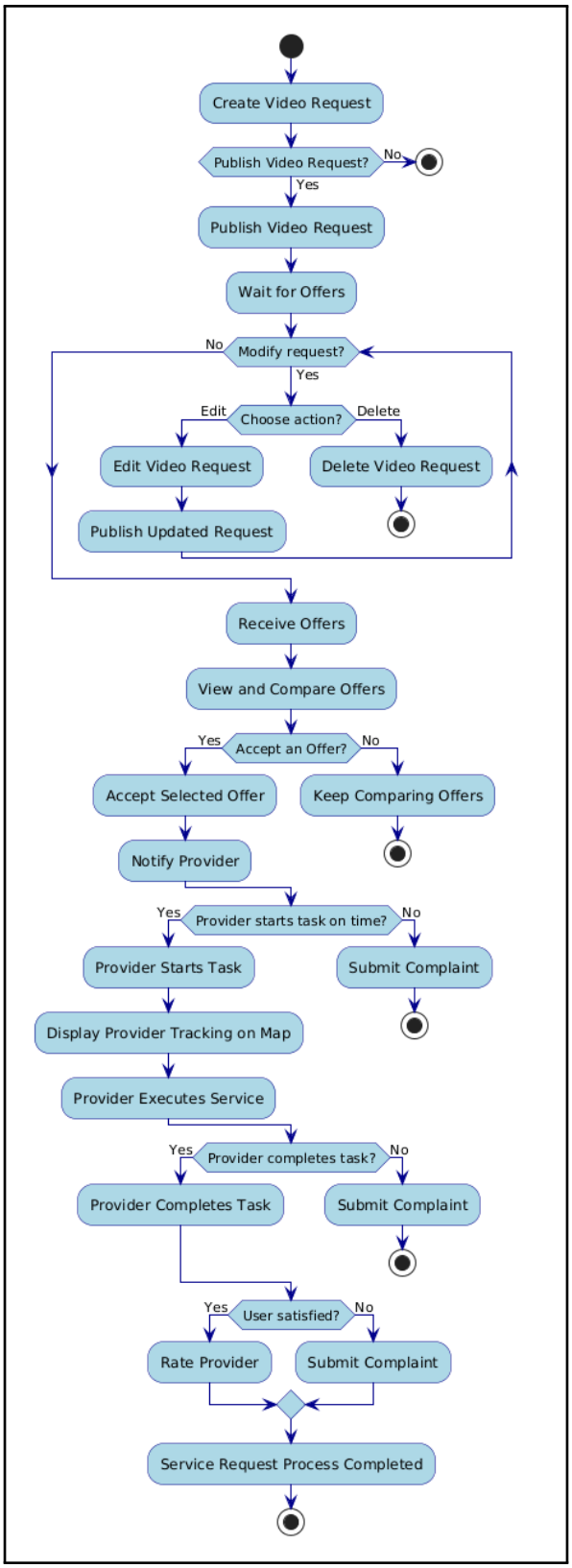


Figure 3.5: Activity Diagram – Maintenance Service Request Process

Figure 3.5 shows the service request workflow in Fixoo. The process starts when the

User creates a video request. Before publication, the User can decide whether to publish the request or stop the process. If the request is published, it becomes available for providers to view and respond to with offers.

a. Video Request Management After publication, the User waits for providers to submit offers. During this stage, the User still has control over the request and may choose to edit it or delete it. Editing allows the request to be updated and republished, while deleting the request ends the workflow. If no modification is needed, the process continues to the offer stage.

b. Offer Reception and Selection When offers are available, the User views and compares them according to price, duration, and provider information. If the User accepts one offer, the selected provider is notified and the service process continues. If the User does not accept any offer, the comparison step may continue without creating a project.

c. Task Execution and Tracking After the offer is accepted, the provider is expected to start the task. If the provider starts on time, the tracking phase becomes available and the User can follow the provider location on a map. If the provider does not start the task within the expected time, the User may submit a complaint and the process stops.

d. Task Completion and Satisfaction Once the provider performs the service, the system checks whether the task is completed. If the task is completed, the User can decide whether the service was satisfactory. A satisfied User can rate the Provider, while an unsatisfied User can submit a complaint. If the task is not completed, the User can also submit a complaint. The workflow ends after rating or complaint submission.

3.3.4 System Modules

The Fixoo platform is organized into modules that correspond to the services implemented in the application layer. Each module groups a set of operations that serve a specific part of the platform.

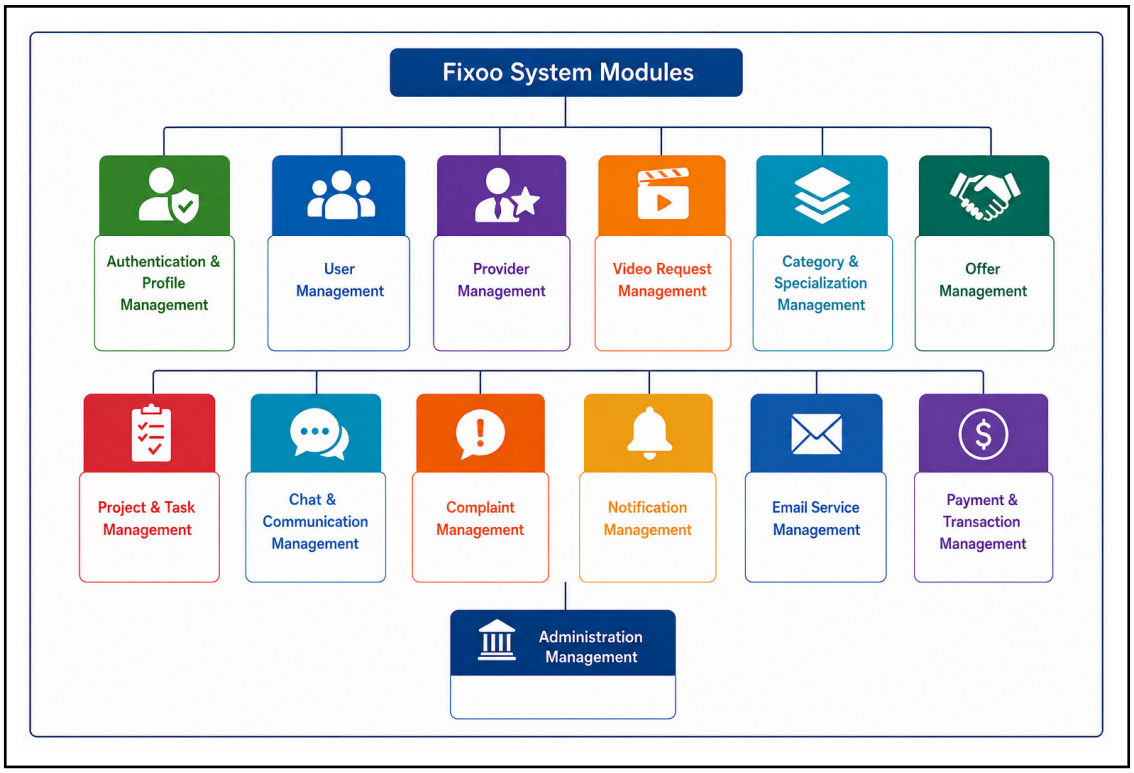


Figure 3.6: Architectural Overview of Fixoo System Modules.

- **Admin Module:** This module provides administrator access by retrieving administrator information through the admin service.
- **Authentication and Profile Management Module:** This module groups the operations related to registration, login, phone verification, OTP resend, password recovery, password reset, token refresh, logout, profile image upload, profile image confirmation, profile retrieval, profile update, email change, password change, and profile image deletion.
- **Complaint Management Module:** This module handles complaint operations. It supports creating complaints, retrieving complaints for a project, retrieving complaints with administration filters, reviewing complaints, and retrieving complaints submitted by a specific user.
- **Chat and Communication Module:** This module manages chat-related operations. It supports creating or retrieving a chat room, sending messages, retrieving messages with pagination, marking messages as read, listing user chat rooms, and verifying whether a user belongs to a chat room.
- **Category and Specialization Management Module:** This module contains the operations used to manage categories and specializations. It supports retrieving, creating,

updating, and deleting categories, as well as retrieving, creating, updating, and deleting specializations.

- **Email Service Module:** This module handles the email operations used by the platform. These operations include provider registration emails, approval emails, rejection emails, offer submission emails, offer acceptance emails, offer rejection emails, new offer notifications to clients, payment confirmation emails, debt reminders, complaint submission emails, complaint resolution emails, and provider suspension emails.

- **Notification Module:** This module manages internal notifications. It supports notification creation, notification retrieval, unread notification counting, marking one notification as read, and marking all notifications as read.

- **Offer Management Module:** This module manages offer operations. Providers can create, update, delete, and retrieve their offers, while users can retrieve offers related to a video request and accept one offer.

- **Payment and Transaction Module:** This module manages the financial side of the platform. It supports retrieving provider financial summaries, retrieving provider transactions, updating commission rates, retrieving commission history, retrieving the active commission rate, adding subscription plans, registering subscriptions,retrieving provider financial data for administration, retrieving providers with debt, recording manual payments, retrieving platform revenue summaries, retrieving monthly revenue, retrieving top providers, sending a debt reminder, and sending reminders to overdue providers.

- **Project and Task Management Module:** This module manages accepted offers as projects. It supports retrieving project details, retrieving client projects, retrieving provider location, creating reviews, retrieving provider projects, starting tasks, completing tasks, and updating provider location.

- **Provider Management Module:** This module manages provider operations. It supports provider registration, provider profile retrieval, provider profile update, document upload URL generation, specialization management, provider public profile retrieval, provider review retrieval, provider pagination for administration, provider retrieval by identifier, provider activation, deactivation, verification, unverification, and provider export to Excel.

- **User Management Module:** This module manages user-related administration operations. It supports retrieving users with pagination and filters, retrieving a user by identifier, activating users, deactivating users, suspending users, and exporting user data to Excel.

- **Video Request Management Module:** This module manages video request operations. It supports upload URL generation, thumbnail upload, video creation, video retrieval, video update, video deletion, administration video listing, video status change, video publication, video feed retrieval, user video retrieval, trending video retrieval, view count increment, stream URL generation, video streaming, blob information retrieval, active category retrieval, and active specialization retrieval.

3.4 Conclusion

This chapter presented the design of the Fixoo platform from several perspectives. The architecture section explained how the system is organized into client-side interfaces, backend services, and data storage components. The use case diagrams described the interactions of the Administrator, User, and Provider with the system. The class diagram presented the main entities and relationships used to structure the platform data. The activity diagram showed the main workflow of a video-based maintenance request, from publication to offer selection, task execution, tracking, rating, or complaint submission.

The chapter also summarized the system modules derived from the implemented services. These modules show how the platform functions are grouped into authentication, user management, provider management, video request management, offer management, task management, notifications, complaints, chat, payment, and administration. The next chapter will move from design to implementation by presenting the technologies, tools, and practical development environment used to build the Fixoo platform.

Chapter 4

System Implementation

4.1 Introduction

This chapter describes how the proposed Fixoo platform was implemented in practice after completing the analysis and design phases. The implementation stage transforms the previously defined requirements, architecture, and UML models into a working software solution composed of a mobile application, a backend API, an administration interface, and supporting storage services.

The chapter presents the main technologies used to build Fixoo, including the programming languages, frameworks, database tools, development environments, and software tools adopted during development. Each technology is explained according to its role in the project and how it contributes to implementing the platform features such as video request creation, offer management, real-time communication, tracking, notifications, and administrative supervision.

4.2 Development Environment and Technologies

This section presents the technical environment used to implement the Fixoo platform. The selected technologies were chosen according to the needs of the system, especially the need for a reliable backend, a cross-platform mobile application, structured data management, secure media storage, real-time communication, and a maintainable administration dashboard.

4.2.1 Programming Languages

Different programming languages were used in Fixoo according to the role of each system layer. The backend was implemented using C#, the mobile application was developed using Dart, the database logic relied on SQL, and the administration interface used standard web technologies.

A. C# (C-Sharp)

Definition:

C# is a programming language created by Microsoft and used mainly within the .NET ecosystem. It supports object-oriented development and is suitable for building backend services, APIs, enterprise applications, and cloud-connected systems [19].

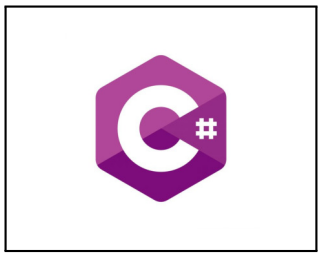


Figure 4.1: C# Language Logo.

Why We Use It:

C# was used as the main backend language of Fixoo because it fits the structure of the ASP.NET Core API and supports the business operations required by the platform.

- **Backend Logic Implementation:** C# was used to implement services related to authentication, video requests, offers, projects, complaints, payments, and notifications.
- **ASP.NET Core Compatibility:** It works naturally with ASP.NET Core Web API, which represents the server-side layer of Fixoo.
- **Clear Code Organization:** Its object-oriented approach helped separate controllers, services, models, and database access logic.
- **Asynchronous Operations:** It supports asynchronous programming, which is useful for database calls, file storage operations, email sending, and notification delivery.

- **Reliability:** Its type system helps reduce programming mistakes when handling important data such as user identifiers, offers, transactions, and project states.
- **Project Maintainability:** C# made it easier to organize the backend into services such as AuthService, VideoService, OfferService, ProjectService, PaymentService, and NotificationService.
- **Integration with Microsoft Tools:** It integrates well with Visual Studio, SQL Server, Azure Blob Storage, and the .NET libraries used in the project.

B. Dart

Definition:

Dart is a programming language developed by Google and used with Flutter to build applications for several platforms from one codebase. It is designed for client-side development and supports modern mobile application features [20].



Figure 4.2: Dart Language Logo.

Why We Use It:

Dart was used to implement the Fixoo mobile application used by both users and providers.

- **Single Codebase:** It allowed the same application logic and interface structure to be shared for Android and iOS development.
- **API Communication:** Dart was used to send HTTP requests to the ASP.NET Core backend and process JSON responses.
- **Mobile Interface Logic:** It was used to manage screens such as login, video feed, offer submission, tracking, chat, profile, and notifications.
- **Asynchronous Processing:** Futures and Streams were useful for calling APIs, loading videos, receiving chat updates, and managing notifications.

- **Fast Development Cycle:** Dart with Flutter helped test interface changes quickly during the implementation of the mobile application.
- **Package Support:** It allowed the integration of packages needed for maps, media handling, authentication storage, and push notifications.

C. SQL (Structured Query Language)

Definition:

SQL is a language used to define, query, and manipulate relational data. It provides commands for creating tables, retrieving records, inserting new data, updating existing data, and enforcing relationships between stored information [21].



Figure 4.3: SQL Logo.

Why We Use It:

SQL was used to manage the structured data of the Fixoo platform.

- **Platform Data Storage:** SQL was used to store users, providers, video requests, offers, projects, complaints, reviews, categories, specializations, and financial transactions.
- **Relationship Management:** It helped define relations between the main entities of the system, such as users and videos, providers and offers, offers and projects, and projects and reviews.
- **Querying System Data:** It was used to retrieve filtered and paginated data for mobile screens and administration pages.
- **Data Consistency:** SQL constraints and transactions helped protect important operations such as accepting offers, creating projects, recording payments, and updating task states.
- **Stored Logic:** Stored procedures and database functions were used to centralize several database operations used by the backend services.

- **Reporting Support:** SQL queries supported statistics and summaries used by the administration dashboard.
- **Controlled Updates:** It allowed precise updates to important records such as video status, offer status, provider balance, and notification read state.

D. HTML (HyperText Markup Language)

Definition:

HTML is a markup language used to describe the structure of web pages. It organizes page elements such as text blocks, forms, tables, links, and images [22].



Figure 4.4: HTML Logo.

Why We Use It:

HTML was used in the administration dashboard to structure the pages that display users, providers, categories, videos, payments, complaints, and reports. It provides the base layout of the web interface and organizes the content shown to the administrator.

E. CSS (Cascading Style Sheets)

Definition:

CSS is a styling language used with HTML to control the visual presentation of web pages, including layout, spacing, colors, typography, and responsive behavior [23].



Figure 4.5: CSS Logo.

Why We Use It:

CSS was used to improve the visual appearance of the Fixoo administration dashboard. It helped create clean pages, readable tables, consistent spacing, and a more organized interface for managing platform data.

F. JavaScript

Definition:

JavaScript is a scripting language used in web development to add interactive behavior to web pages. It can react to user actions, update content dynamically, and support client-side logic [24].



Figure 4.6: JavaScript Logo.

Why We Use It:

JavaScript was used where client-side interactions were needed in the administration interface. It helped improve page behavior, handle dynamic actions, and support interactive elements in the dashboard.

4.2.2 Frameworks and Libraries

Frameworks and libraries were used to accelerate development and provide stable components for backend services, mobile interfaces, web dashboards, real-time communication, and responsive design.

A. ASP.NET Core

Definition:

ASP.NET Core is a Microsoft framework used to build web applications, APIs, and backend services. It is designed to support modern server-side development and works across different operating systems [25].



Figure 4.7: ASP.NET Core Logo.

Why We Use It:

ASP.NET Core was used to build the backend API of Fixoo.

- **API Layer:** It provided the controllers consumed by the Flutter application and the administration dashboard.
- **Business Services:** It hosted services such as AuthService, VideoService, OfferService, ProjectService, ChatService, and PaymentService.
- **Authentication:** It supported JWT authentication to protect endpoints according to the connected user.
- **Rate Limiting:** It allowed limiting sensitive actions such as login, OTP verification, offer creation, video creation, and complaint submission.
- **Middleware Support:** It was used for request processing, error handling, CORS configuration, HTTPS redirection, and authorization.
- **Integration:** It connected the mobile application, the database, Azure Blob Storage, SignalR, OneSignal, and the admin dashboard.

B. ADO.NET

Definition:

ADO.NET is a data access technology for .NET applications. It allows applications to connect to databases, execute SQL commands, call stored procedures, and read returned data [27].



Figure 4.8: ADO.NET Logo.

Why We Use It:

ADO.NET was used in Fixoo to communicate directly with the database through a controlled database helper layer.

- **Stored Procedure Execution:** It was used to execute procedures for operations such as registration, video creation, offer submission, project updates, and notifications.
- **Database Functions:** It helped retrieve paginated and filtered data through SQL functions.
- **Direct Control:** It gave direct control over SQL parameters and database commands.
- **Performance:** It supported fast data access for backend services that frequently interact with the database.

C. Flutter Framework

Definition:

Flutter is a UI framework created by Google for building applications from a single codebase. It provides widgets, rendering tools, and development features for mobile and other platforms [28].



Figure 4.9: Flutter Logo.

Why We Use It:

Flutter was used to build the Fixoo mobile application for users and providers.

- **Mobile Application:** It was used to implement the main interface used for video requests, offers, tracking, chat, and notifications.
- **Shared Codebase:** It reduced the need to write separate code for Android and iOS.
- **User Interface:** It provided widgets for building responsive screens adapted to the platform workflow.
- **Media Features:** It supported the implementation of video upload, thumbnail display, and video streaming screens.
- **Map Integration:** It allowed displaying maps and provider tracking inside the mobile application.
- **Backend Communication:** It was used to consume the REST APIs provided by the ASP.NET Core backend.

D. Blazor Server

Definition:

Blazor Server is a Microsoft framework used to create interactive web applications using C#. In this model, the application logic runs on the server while the browser communicates with it through a real-time connection [29].



Figure 4.10: Blazor Server Logo.

Why We Use It:

Blazor Server was used to implement the Fixoo administration dashboard.

- **Administration Interface:** It was used to manage users, providers, categories, videos, complaints, payments, and reports.
- **C# Reuse:** It allowed the administration side to be developed using the same main language as the backend.
- **Interactive Pages:** It supported dynamic pages for filtering, viewing, updating, and managing platform data.
- **Backend Compatibility:** It worked well with the ASP.NET Core environment used in the system.

E. Bootstrap

Definition:

Bootstrap is a popular open-source CSS framework used to design responsive and modern web interfaces. It provides ready-made components such as buttons, forms, navigation bars, cards, and grids [42].

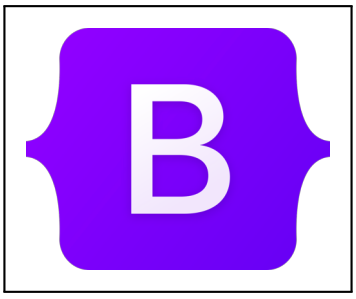


Figure 4.11: Bootstrap Logo.

Why We Use It:

Bootstrap was used in Fixoo to style and organize the administration interface using responsive layouts and ready-made UI components.

F. SignalR

Definition:

SignalR is an ASP.NET Core library that enables real-time communication between the server and connected clients [30].



Figure 4.12: SignalR Logo.

Why We Use It:

SignalR was used in Fixoo to implement the real-time chat between users and providers and to support live updates in the administration interface.

4.2.3 Database Software Tools

Database and storage tools were used to manage the two main types of data in Fixoo. Structured data, such as users, offers, and projects, was stored in a relational database, while media files and verification documents were stored in cloud object storage.

A. Microsoft SQL Server

Definition:

Microsoft SQL Server is a relational database management system developed by Microsoft. It is used to store, secure, query, and manage structured data for software applications [31].



Figure 4.13: Microsoft SQL Server Logo.

Why We Use It:

- **Core Database:** SQL Server was used as the main database engine of the Fixoo backend.

- **Structured Records:** It stored users, providers, categories, specializations, video requests, offers, accepted projects, reviews, complaints, notifications, and transactions.
- **Relations Between Tables:** It helped maintain relationships between the main system tables.
- **Stored Procedures:** It supported stored procedures used by the backend services to execute database operations.
- **Data Security:** It protected important information such as user accounts, provider documents metadata, and financial records.
- **Backend Integration:** It worked with the ASP.NET Core API through the database helper and ADO.NET.

B. Azure Blob Storage

Definition:

Azure Blob Storage is a cloud object storage service from Microsoft used to store unstructured files such as videos, images, documents, and other binary data [32].



Figure 4.14: Azure Blob Storage Logo.

Why We Use It:

- **Video Request Files:** It was used to store the videos uploaded by users when creating maintenance requests.
- **Video Thumbnails:** It stored thumbnails related to video requests to make video browsing easier in the application.
- **Profile Images:** It stored profile images for users and providers.

- **Provider Documents:** It stored verification documents uploaded by providers, including specialization certificates, national identity cards, and self-employed cards.
- **Server Load Reduction:** It prevented the API server from directly storing large media files.
- **Controlled Access:** It allowed secure access to private media and documents through generated URLs.
- **Media Retrieval:** It supported retrieving videos, images, and documents from both the mobile application and the administration dashboard.

4.2.4 Development Environment (IDE)

The development environment includes the tools used to write, run, test, and debug the Fixoo source code. These tools were selected according to the different parts of the system, especially the ASP.NET Core backend, the Blazor administration dashboard, and the Flutter mobile application.

A. Visual Studio

Definition:

Visual Studio is an integrated development environment provided by Microsoft. It offers tools for writing code, debugging applications, managing projects, connecting to databases, and developing .NET-based systems [33].



Figure 4.15: Visual Studio 2022 Logo.

Why We Use It:

Visual Studio was used mainly for the backend and the administration side of Fixoo.

- **Backend Development:** It was used to create and debug the ASP.NET Core Web API project.
- **Blazor Development:** It supported the development of the administration dashboard.
- **Solution Management:** It helped manage the different projects of the backend solution, including application, domain, infrastructure, and API layers.
- **Debugging:** It provided breakpoints, watches, and exception tracking during backend testing.
- **Database Work:** It helped during database connection testing and SQL Server integration.
- **Microsoft Stack Support:** It worked efficiently with C#, ASP.NET Core, Blazor, and SQL Server.

B. Visual Studio Code

Definition:

Visual Studio Code is a lightweight source code editor developed by Microsoft. It supports many languages and development workflows through extensions, integrated terminal, Git tools, and debugging features [35].



Figure 4.16: Visual Studio Code Logo.

Why We Use It:

Visual Studio Code was used mainly during the development of the Flutter mobile application.

- **Flutter Coding:** It was used to write and organize Dart files.
- **Flutter Commands:** Its terminal was used to run commands such as `flutter pub get`, `flutter run`, and `flutter build`.
- **Extension Support:** Flutter and Dart extensions helped with completion, analysis, formatting, and debugging.
- **Fast Editing:** It provided a lightweight environment for quick code changes.
- **Git Usage:** It supported source code version control during mobile application development.
- **Cross-Platform Work:** It can be used on different operating systems, which makes the development workflow flexible.

C. Android Studio

Definition:

Android Studio is the official development environment for Android applications. It provides Android SDK management, emulator creation, testing tools, and profiling utilities [36].



Figure 4.17: Android Studio Logo.

Why We Use It:

Android Studio was used to support Android testing for the Fixoo mobile application.

- **Android SDK:** It managed the SDK tools required to build and run the Flutter application on Android.
- **Emulator Testing:** It provided virtual devices for testing the application when a physical phone was not available.

- **Device Management:** It helped manage connected Android phones and virtual devices.
- **Flutter Support:** It supported Flutter and Dart plugins for mobile development.
- **Screen Testing:** It allowed testing the application on different Android versions and screen sizes.
- **Performance Tools:** It helped inspect application behavior during development and testing.

4.2.5 Software Development Tools

Software development tools were used to test APIs, expose local services, manage source code, generate UML diagrams, display maps, and send mobile push notifications. These tools supported the implementation and testing workflow of the Fixoo platform.

A. Postman

Definition:

Postman is a tool used by developers to test, organize, and document APIs through HTTP requests and responses [37].



Figure 4.18: Postman Logo.

Why We Use It:

Postman was used to test and validate Fixoo backend API endpoints before integrating them with the Flutter mobile application.

B. Swagger

Definition:

Swagger provides tools for describing, documenting, and testing REST APIs through an interactive web interface [46].



Figure 4.19: Swagger Logo.

Why We Use It:

Swagger was used to display and test the available Fixoo API endpoints directly from the browser during backend development.

C. Ngrok

Definition:

Ngrok is a tunneling tool that makes a local development server accessible through a public URL over the internet [47].



Figure 4.20: Ngrok Logo.

Why We Use It:

Ngrok was used to expose the local ASP.NET Core backend so the Flutter application could access it during testing on a real device.

D. Git and GitHub

Definition:

Git is a version control system used to track changes in source code, while GitHub is a hosting platform for Git repositories and development collaboration [38].



Figure 4.21: GitHub Logo.

Why We Use It:

Git and GitHub were used to manage the Fixoo source code, track changes, and keep a remote backup of the project.

E. OpenStreetMap

Definition:

OpenStreetMap is an open geographic data project that provides map data used by applications to display locations and map-based information [48].



Figure 4.22: OpenStreetMap Logo.

Why We Use It:

OpenStreetMap was used in Fixoo to display maps and support provider location tracking during task execution.

F. PlantUML

Definition:

PlantUML is a tool that generates UML diagrams from textual descriptions [49].



Figure 4.23: PlantUML Logo.

Why We Use It:

PlantUML was used to create and update the UML diagrams of the Fixoo system, including use case, class, and activity diagrams.

G. OneSignal

Definition:

OneSignal is a messaging platform used to send push notifications to mobile and web applications. It provides APIs and SDKs that allow applications to deliver notifications to users' devices [44, 45].



Figure 4.24: OneSignal Logo.

Why We Use It:

OneSignal was used to send mobile push notifications to users and providers about offers, task updates, complaints, and important system events.

4.3 Database Structure

The database represents the main foundation of the Fixoo platform, since it stores and organizes all information related to users, providers, video requests, offers, accepted projects,

financial transactions, and administrators.

The Fixoo platform uses a relational database implemented with Microsoft SQL Server. The database tables are linked through primary keys and foreign keys in order to ensure data consistency, integrity, and efficient access to information.

4.3.1 Logical Data Model (MLD)

The logical data model of the Fixoo platform can be represented as follows:

- **USERS**(ID, FirstName, LastName, Email, Role, PhoneNumber, PasswordHash, ProfileImageUrl, #WilayaID, Address, Latitude, Longitude, IsActive, IsEmailVerified, IsPhoneVerified, Createdate) See Figure 4.25
- **PROVIDERS**(ID, #UserID, NationalIDNumber, NationalIDImageUrl, SelfEmployedCardNumber, SelfEmployedCardUrl, CertificateUrl, YearsOfExperience, TotalRating, TotalReviews, CompletedProjects, Balance, DebtAmount, ArrivalTimeMinutes, AdminNotes, RejectedDate, VerifiedDate, IsActive, IsVerified, Createdate) See Figure 4.26
- **VIDEO_REQUEST**(ID, #UserID, VideoUrl, ThumbnailUrl, VideoDuration, Title, Description, #CategoryID, #SpecializationID, #StatusID, LocationAddress, Latitude, Longitude, ViewCount, IsActive, Createdate, Publishedate, Completedate) See Figure 4.27
- **OFFERS**(ID, #VideoID, #ProviderID, OfferPrice, EstimatedDuration, OfferDescription, IsAccepted, IsRejected, Acceptedate, Createdate, Updatedate, ArrivalTimeMinutes, IsActive, RejectedDate) See Figure 4.28
- **ACCEPTED_PROJECTS**(ID, #VideoID, #OfferID, #ProviderID, #UserID, FinalPrice, EstimatedDuration, LastLocationUpdate, IsCompleted, Createdate, ProviderLatitude, ProviderLongitude, TaskStartedDate, TaskCompletedDate, Status) See Figure 4.29

In this model, underlined attributes represent primary keys, while attributes preceded by the symbol # represent foreign keys.

4.3.2 Main Database Tables

The following figures present the main tables of the Fixoo database as implemented in Microsoft SQL Server.

A. USERS Table

The `USERS` table stores the basic account information of users, including identity, authentication data, contact information, verification status, and location.

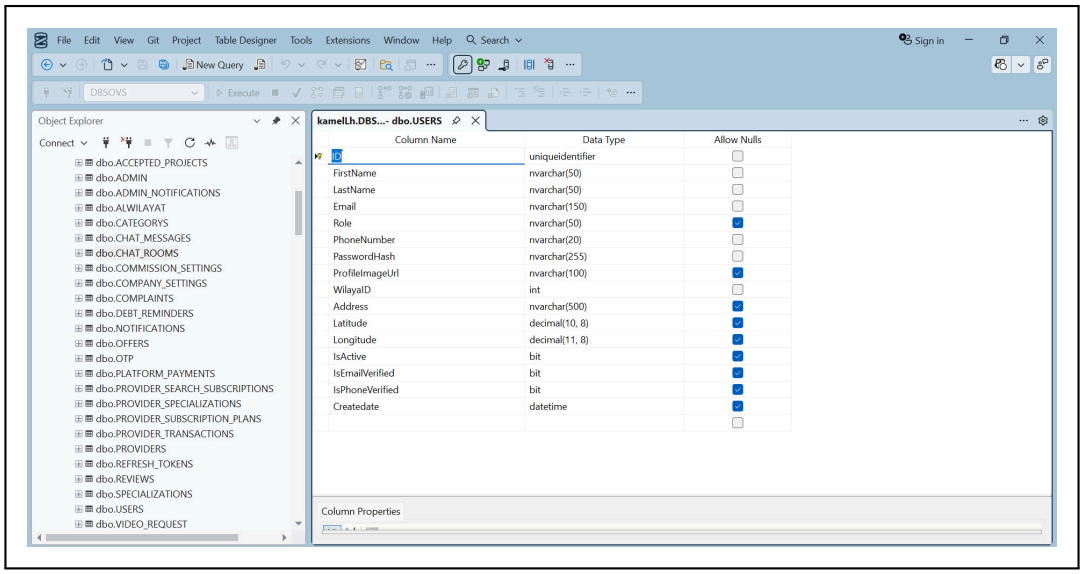


Figure 4.25: USERS Table Structure from Microsoft SQL Server.

B. PROVIDERS Table

The `PROVIDERS` table stores information related to service providers, such as verification documents, experience, ratings, balance, debt, and account status.

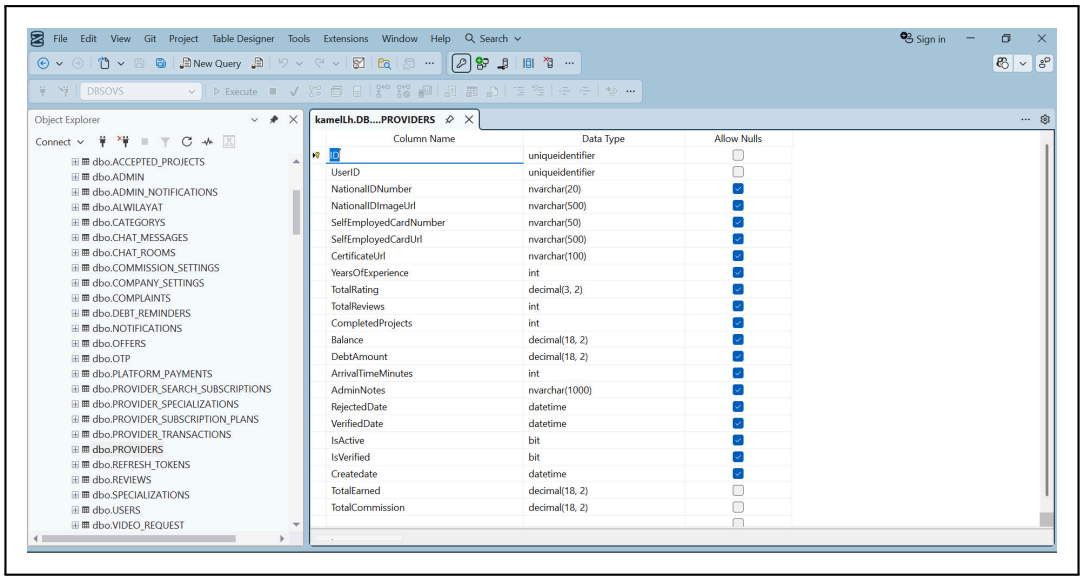


Figure 4.26: PROVIDERS Table Structure from Microsoft SQL Server.

C. VIDEO_REQUEST Table

The `VIDEO_REQUEST` table stores maintenance requests submitted by users through videos. It contains video information, category, specialization, status, location, and request dates.

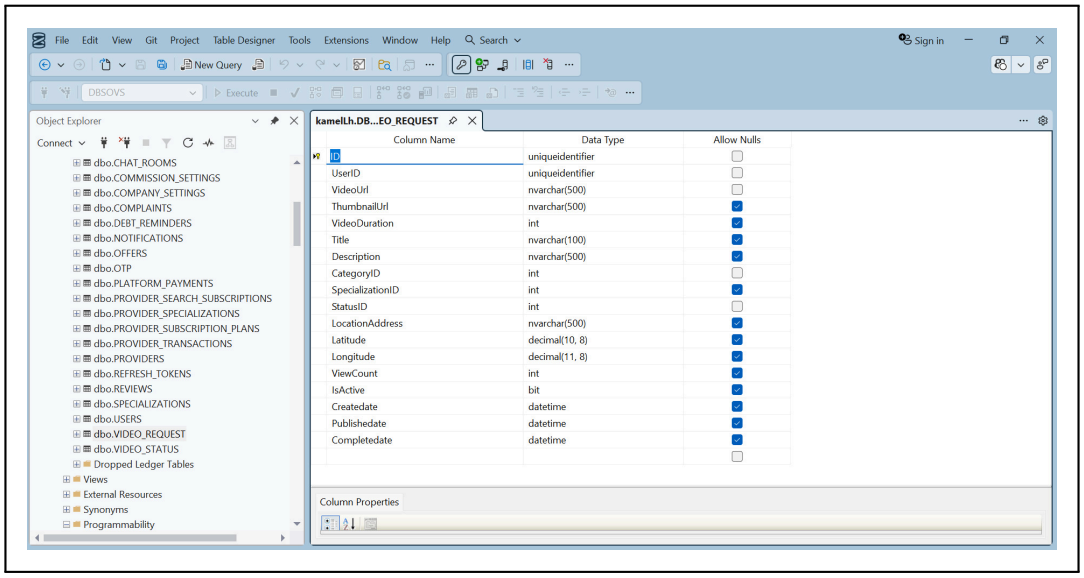


Figure 4.27: VIDEO_REQUEST Table Structure from Microsoft SQL Server.

D. OFFERS Table

The `OFFERS` table stores the offers submitted by providers for each video request. It includes price, estimated duration, offer description, acceptance status, and dates.

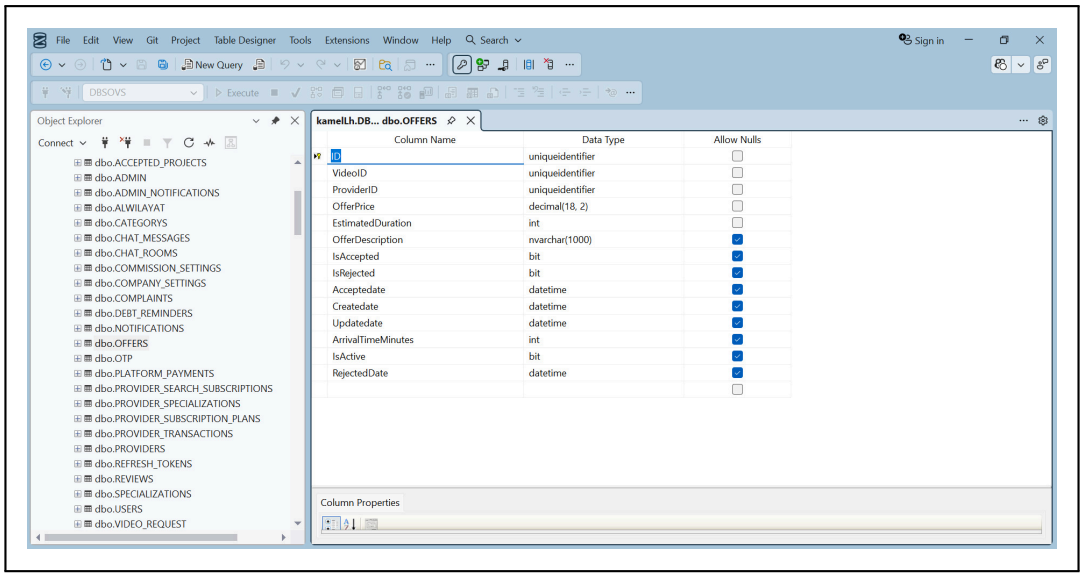


Figure 4.28: OFFERS Table Structure from Microsoft SQL Server.

E. ACCEPTED_PROJECTS Table

The `ACCEPTED_PROJECTS` table stores projects created after a user accepts an offer. It includes the selected provider, accepted offer, final price, task status, and tracking information.

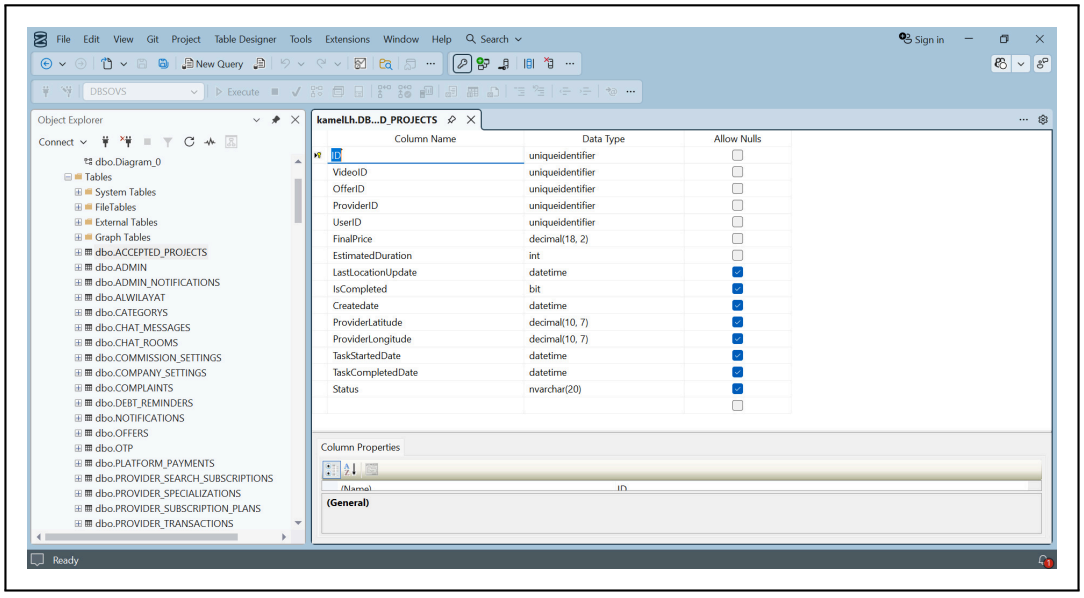


Figure 4.29: ACCEPTED_PROJECTS Table Structure from Microsoft SQL Server.

4.3.3 Database Relationships

The Fixoo database follows a relational model where tables are connected through foreign key relationships to maintain referential integrity and enable complex queries. The main relationships include:

- **One-to-Many:** A single `USERS` record can submit multiple `VIDEO_REQUEST` entries; a single `VIDEO_REQUEST` can receive multiple `OFFERS`; a single `PROVIDERS` record can have multiple `PROVIDER_TRANSACTIONS`.
- **Many-to-One:** Multiple `VIDEO_REQUEST` entries belong to one category via `CategoryID`; multiple `OFFERS` are submitted by one `PROVIDERS` record.
- **One-to-One:** Each `USERS` record of provider type has exactly one `PROVIDERS` profile; each accepted `OFFERS` record converts to exactly one `ACCEPTED_PROJECTS` entry.

4.3.4 Mobile Application Interfaces

The mobile application provides the main interface for service seekers and providers. It supports the complete service process, from publishing video requests to receiving offers, selecting a provider, and tracking the service execution.

A. Home Screen

The home screen displays a scrollable list of video service requests. Providers can watch the published videos to understand the problem and use the **Fix** button to send a service offer.

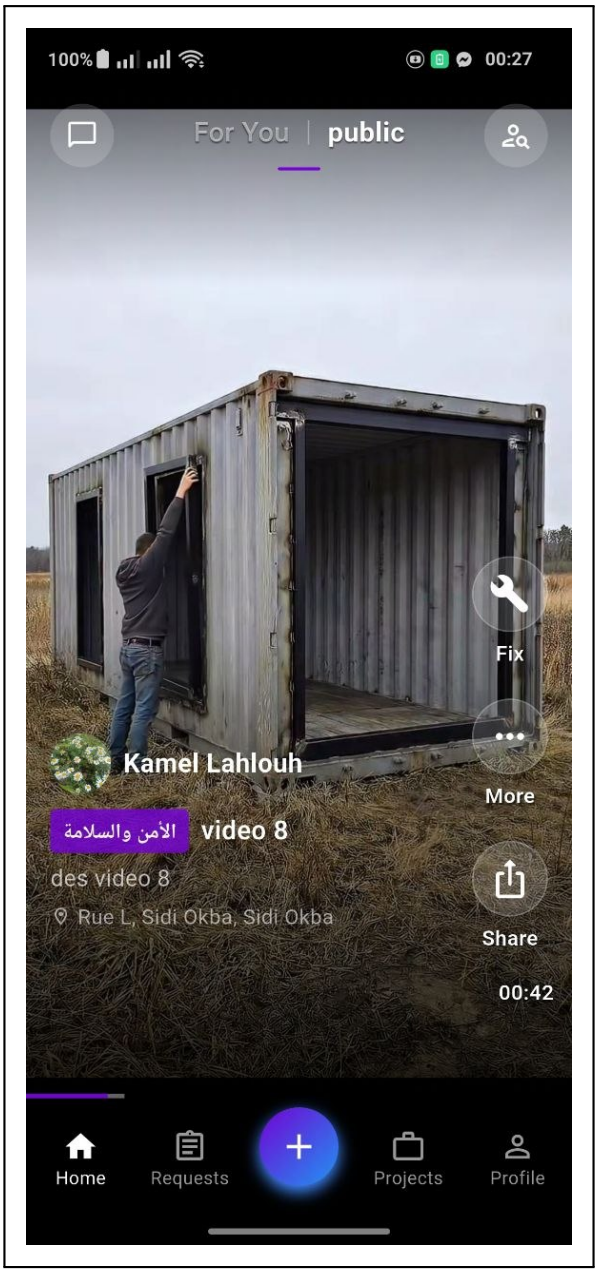


Figure 4.30: Home Screen Showing Video Service Requests.

B. Video Request Submission Screen

This screen allows the service seeker to create a request by recording a video using the camera or uploading one from the gallery. The user can also add request details such as description, category, and location.

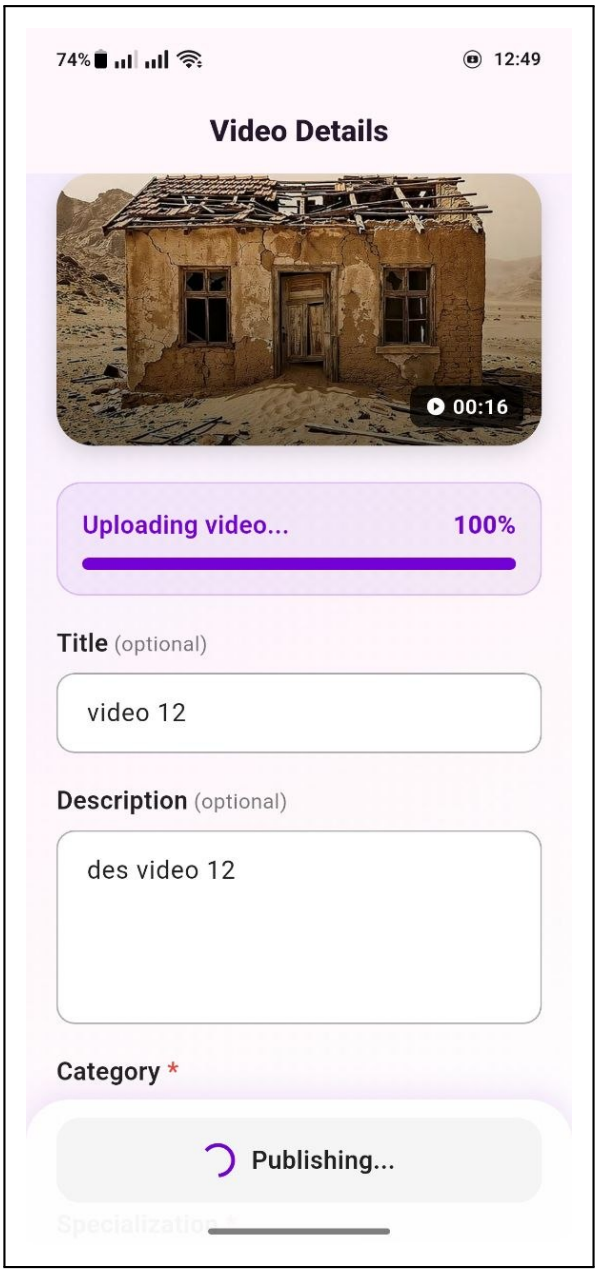


Figure 4.31: Video Request Submission Screen.

C. Offer Comparison Screen

This screen allows the service seeker to compare received offers according to price, duration, provider information, and rating, then choose the most suitable provider.

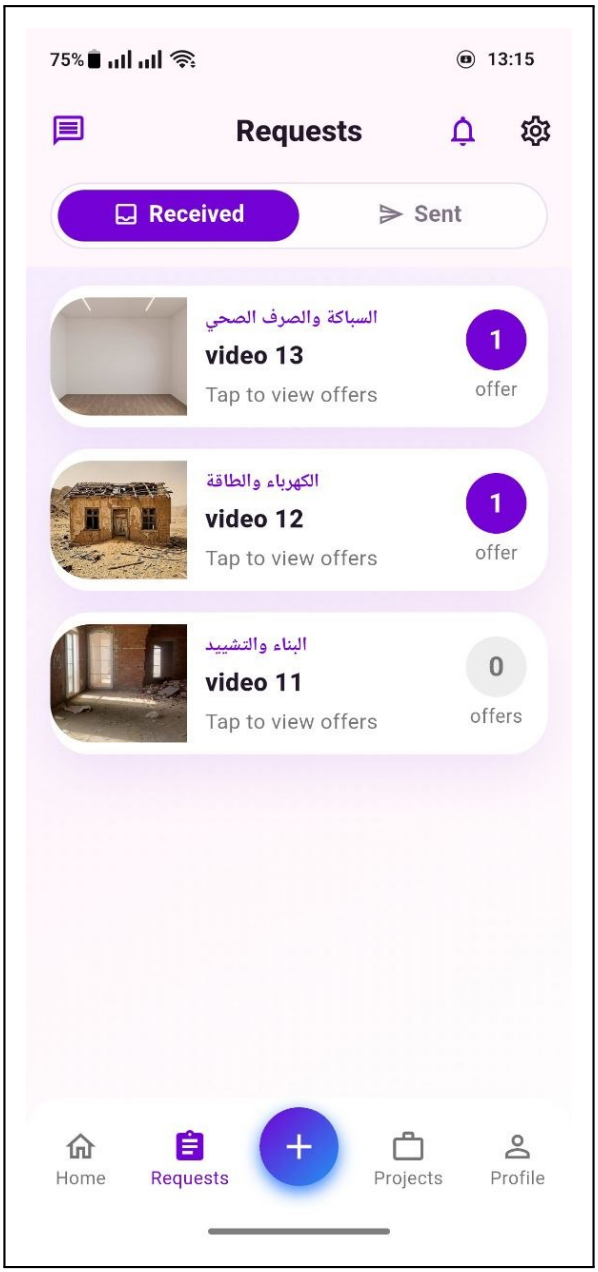


Figure 4.32: Offer Comparison and Selection Screen.

D. Tracking Screen

The tracking screen allows the service seeker to follow the provider's location after accepting an offer. It also includes call and message buttons to facilitate communication during the service process.

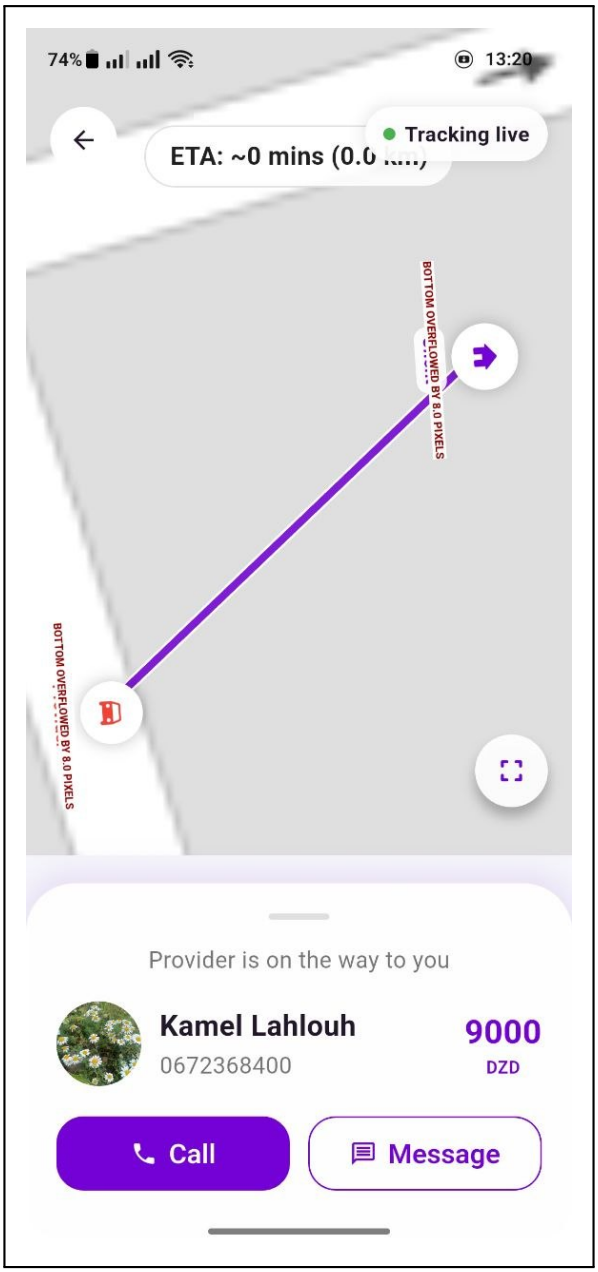


Figure 4.33: Provider Tracking Screen with Call and Message Options.

4.3.5 Administrative Web Panel (Blazor Server)

The administrative panel provides comprehensive tools for platform management, monitoring, and reporting.

A. Admin Dashboard

The main dashboard displays key performance indicators including total users, active

requests, completed projects, revenue statistics, and recent activity.

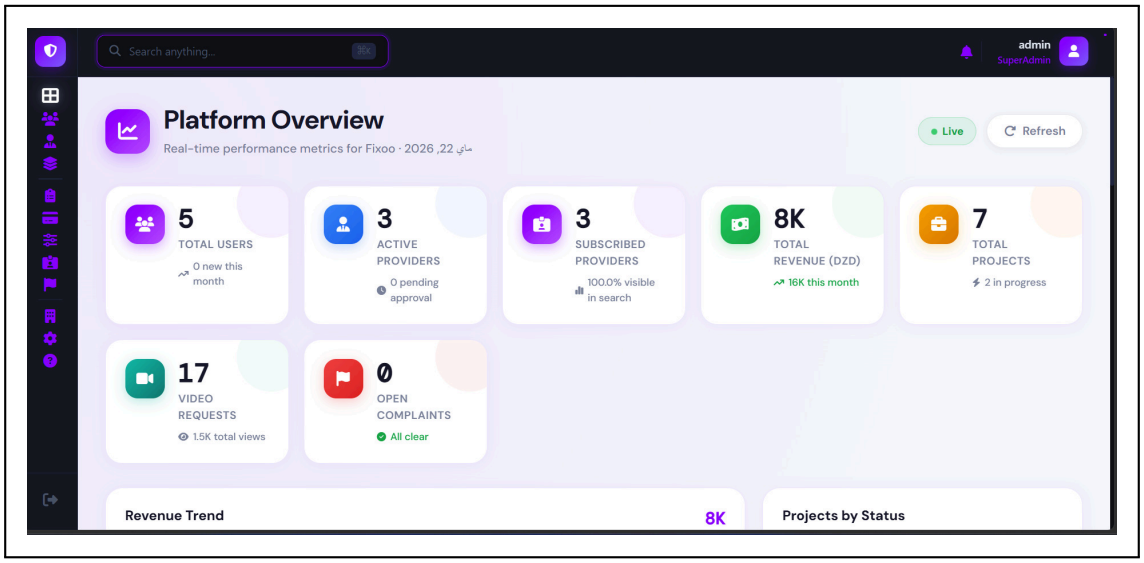


Figure 4.34: Administrative Dashboard – Main Statistics.

B. User Management Interface

Administrators can view, search, filter, activate, suspend, or delete user accounts.

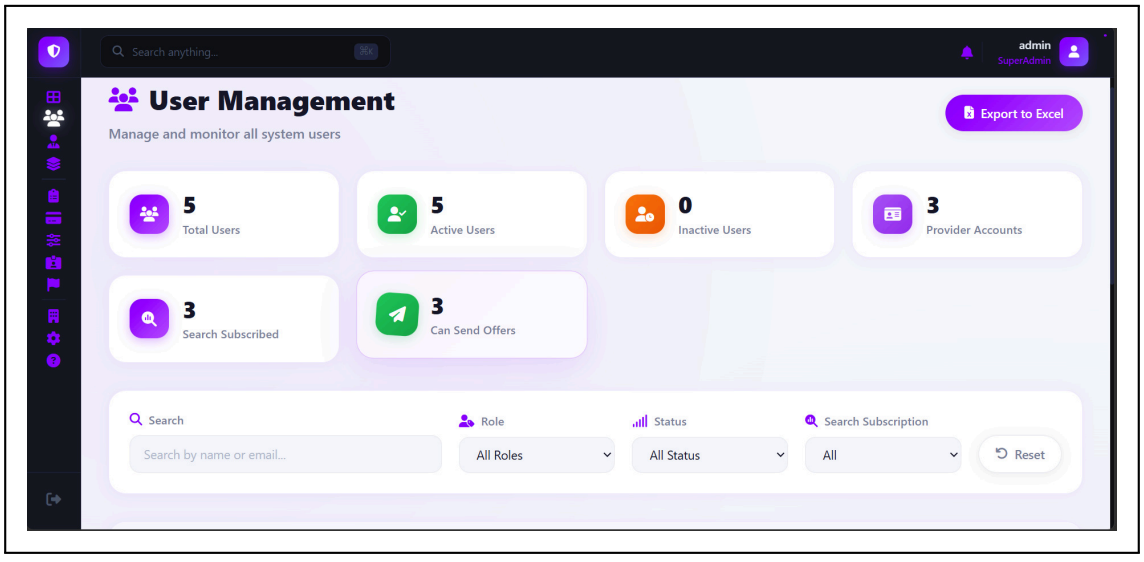


Figure 4.35: User Management Interface.

C. Providers Management Interface

The Providers Management screen displays pending applications with uploaded documents, allowing administrators to approve or reject registrations.

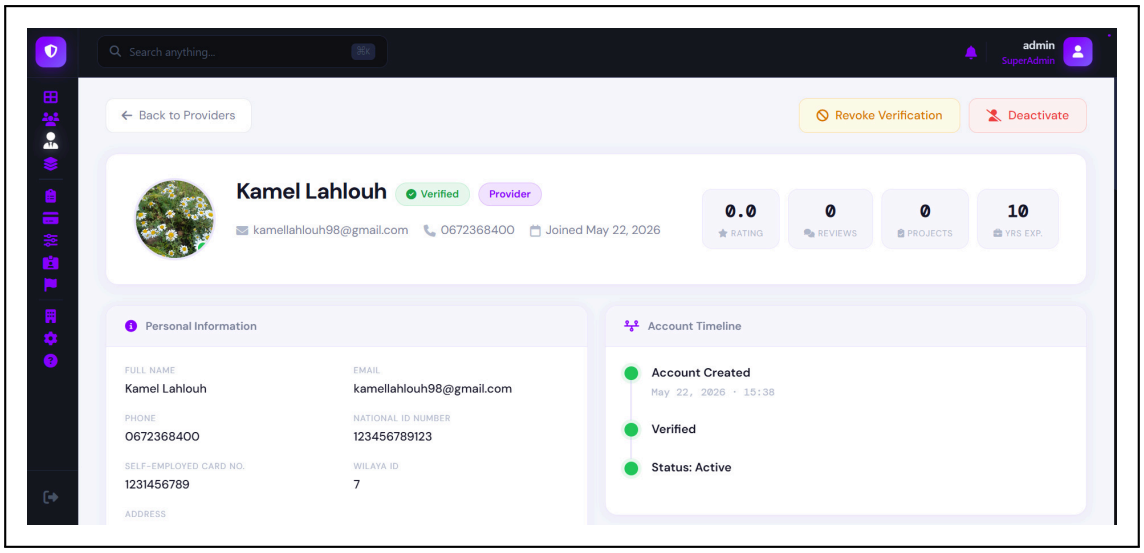


Figure 4.36: Provider Document Verification Interface.

D. Category Management Interface

Administrators can add, edit or delete service categories

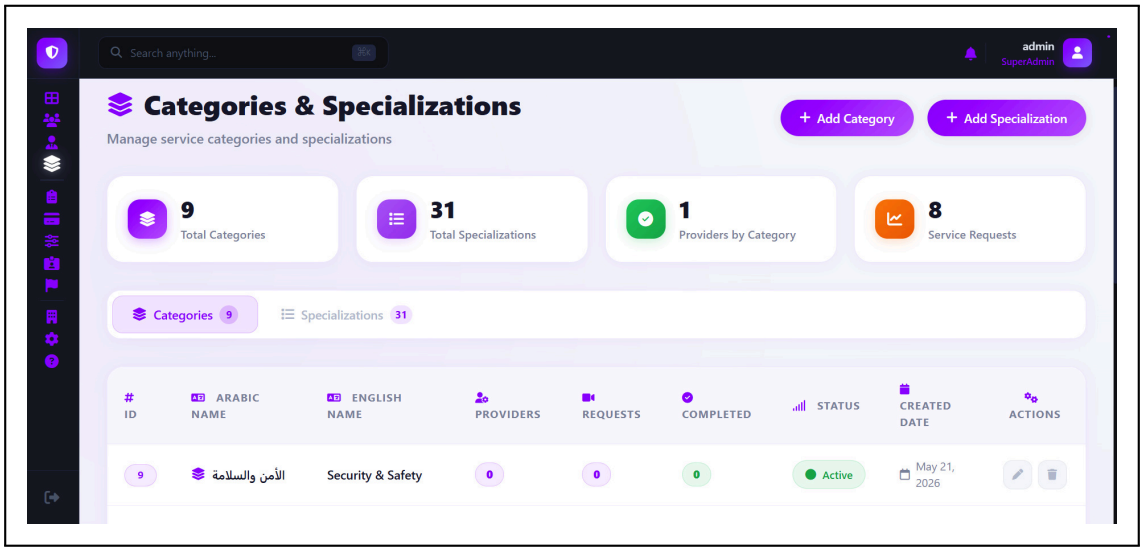


Figure 4.37: Category Management Interface.

E. Payment Management Interface

his interface displays platform earnings, commission settings, technician debts, and payment history

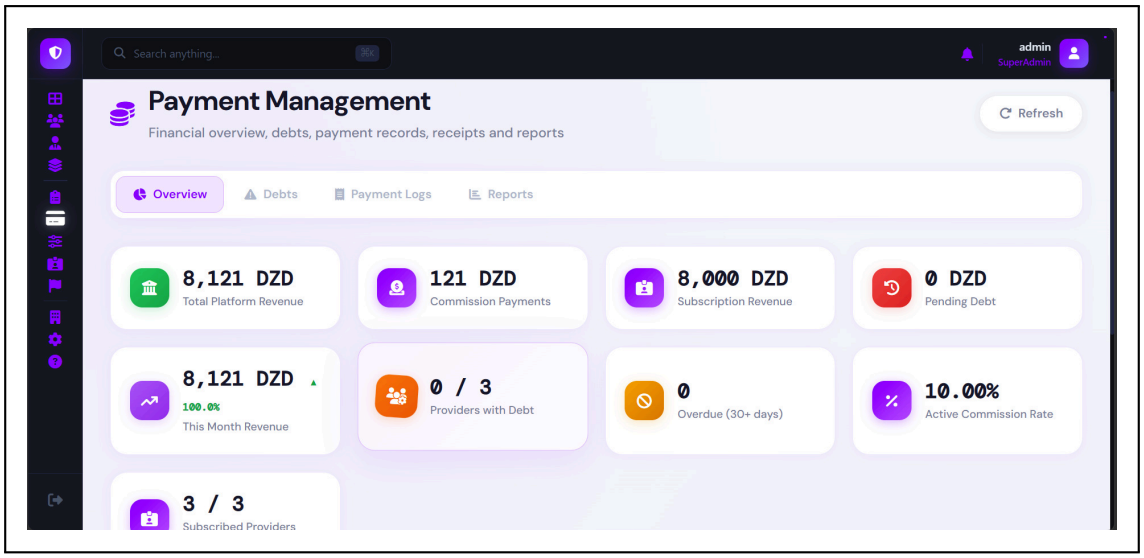


Figure 4.38: Payment Management Interface.

4.4 Conclusion

This chapter presented a comprehensive overview of the implementation phase of the Fixoo platform. It detailed the development environment, technologies, programming languages, frameworks, and tools used throughout the project. The database structure was thoroughly explained, showing how data is organized and managed within the system. Finally, representative screenshots of the main system interfaces demonstrated the practical implementation of the designed functionalities.

The next chapter will focus on testing and evaluation, where the implemented system will be tested for functionality, performance, security, and usability to ensure it meets all specified requirements and provides a reliable service to end users.

General Conclusion

Summary of the Work Conducted

This project focused on the design and implementation of Fixoo, a digital platform that improves the way maintenance services are requested, managed, and evaluated. The main goal was to replace informal methods with a more organized, transparent, and reliable solution adapted to the Algerian context.

The work started with a study of existing maintenance service platforms at international, regional, African, and local levels. This study helped identify their main limitations and justify the need for a more complete solution. Then, the analysis phase defined the system needs, actors, and requirements. After that, the design phase presented the architecture, UML diagrams, data entities, and main workflows of the platform.

Finally, the implementation phase transformed the proposed design into a functional system composed of a mobile application, an administrative web panel, a backend layer, and a relational database. The platform includes key features such as request publication, offer comparison, task follow-up, provider evaluation, complaint handling, payment management, and reporting.

Overall, this project achieved its objective by proposing and developing a practical platform that makes maintenance service management more organized, transparent, and efficient.

Challenges Encountered

Several challenges were encountered during the development of this project. The first challenge was understanding the real needs of both clients and service providers and designing a workflow that remains simple, clear, and useful for both sides.

Another challenge was organizing the complete service lifecycle, from request publication

and offer reception to provider selection, task completion, rating, and complaint submission. Building trust was also important, which required mechanisms such as provider verification, ratings, reviews, complaints, and administrative supervision.

In addition, managing different types of data and integrating the mobile application, backend services, administrative panel, and database required careful design, testing, and coordination.

Skills Acquired

This project helped improve several skills, especially problem analysis, requirements specification, software design, and system modeling. The preparation of the architecture, UML diagrams, and system modules improved the understanding of how complete applications are structured.

From a practical side, the project provided experience in backend development, mobile application development, database design, administrative dashboard creation, and integration between different system components.

It also strengthened problem-solving, project organization, technical documentation, and academic writing skills.

Future Perspectives

Several improvements can be considered in future work. The platform could include a better recommendation system to guide clients toward the most suitable providers based on location, availability, ratings, and completed work.

A secure online payment system could also be integrated to protect both clients and providers and automate commission management. In addition, the platform may be extended with advanced communication options, more service categories, wider regional coverage, and improved analytics for administrators.

Finally, stronger quality control mechanisms could be added, such as detailed provider verification, badges for highly rated technicians, automatic detection of repeated complaints, and improved dispute resolution workflows.

In conclusion, Fixoo represents a step toward modernizing maintenance service access in Algeria. It provides a structured relationship between clients, providers, and administrators,

while future improvements can make it more secure, scalable, intelligent, and impactful.

Bibliography

[1] R. Keith Mobley, *Maintenance Engineering Handbook*, 7th Edition, McGraw-Hill, 2008.

[2] Terry Wireman, *Total Productive Maintenance*, Industrial Press, 2010.

[3] Kenneth C. Laudon and Jane P. Laudon, *Management Information Systems: Managing the Digital Firm*, 16th Edition, Pearson, 2020.

[4] Brian Fitzgerald and Klaas-Jan Stol, *Continuous Software Engineering and Digital Service Systems*, Springer, 2014.

[5] W. Tsai, X. Bai, and Y. Huang, *Smart Service Systems and Digital Transformation*, IEEE Press / Wiley, 2019.

[6] J. Schallmo, *Digital Transformation of Business Models*, Springer, 2018.

[7] Thumbtack Inc., *Thumbtack Official Website*. Available at: https://www.thumbtack.com/. Accessed: February 2026.

[8] ServiceMarket, *ServiceMarket Official Website*. Available at: https://www.servicemarket.com/. Accessed: February 2026.

[9] Kandua, *Kandua Official Website*. Available at: https://www.kandua.com/. Accessed: February 2026.

[10] De9De9 Algeria, *De9De9 Service Platform*. Available at: https://de9de9.dz/. Accessed: February 2026.

[11] Khelliha 3lina, *Khelliha 3lina Service Platform*. Available at: https://kheliha3lina.com/. Accessed: February 2026.

[12] A. S. Tanenbaum and M. Van Steen, *Distributed Systems: Principles and Paradigms*, 3rd Edition, Pearson, 2017.

[13] M. Fowler, *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, 3rd Edition, Addison-Wesley Professional, 2003.

[14] Object Management Group (OMG), *Unified Modeling Language (UML) Specification Version 2.5.1*. Available at: https://www.omg.org/spec/UML/2.5.1/. Accessed: February 2026.

[15] R. Orfali and D. Harkey, *Client/Server Programming with Java and CORBA*, 2nd Edition, Wiley, 1998.

[16] M. Napoli, *Flutter for Beginners*, 2nd Edition, Packt Publishing, 2021.

[17] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice*, 4th Edition, Addison-Wesley Professional, 2021.

[18] Object Management Group (OMG), *Unified Modeling Language (UML) Version 2.5.1 Specification*, Available at: https://www.omg.org/spec/UML/2.5.1/, Accessed: May 2026.

[19] Microsoft, *C# documentation*. Available at: https://learn.microsoft.com/en-us/dotnet/csharp/. Accessed: May 2026.

[20] Google, *Dart Programming Language*. Available at: https://dart.dev/. Accessed: May 2026.

[21] International Organization for Standardization, *ISO/IEC 9075: Database languages SQL*. Available at: https://www.iso.org/standard/76583.html. Accessed: May 2026.

[22] MDN Web Docs, *HTML: HyperText Markup Language*. Available at: https://developer.mozilla.org/en-US/docs/Web/HTML. Accessed: May 2026.

[23] MDN Web Docs, *CSS: Cascading Style Sheets*. Available at: https://developer.mozilla.org/en-US/docs/Web/CSS. Accessed: May 2026.

[24] MDN Web Docs, *JavaScript*. Available at: https://developer.mozilla.org/en-US/docs/Web/JavaScript. Accessed: May 2026.

[25] Microsoft, *ASP.NET Core documentation*. Available at: https://learn.microsoft.com/en-us/aspnet/core/. Accessed: May 2026.

[26] Microsoft, *Entity Framework Core documentation*. Available at: https://learn.microsoft.com/en-us/ef/core/. Accessed: May 2026.

[27] Microsoft, *ADO.NET documentation*. Available at: https://learn.microsoft.com/en-us/dotnet/framework/data/adonet/. Accessed: May 2026.

[28] Google, *Flutter documentation*. Available at: https://flutter.dev/docs. Accessed: May 2026.

[29] Microsoft, *Blazor documentation*. Available at: https://learn.microsoft.com/en-us/aspnet/core/blazor/. Accessed: May 2026.

[30] Microsoft, *SignalR documentation*. Available at: https://learn.microsoft.com/en-us/aspnet/core/signalr/. Accessed: May 2026.

[31] Microsoft, *SQL Server documentation*. Available at: https://learn.microsoft.com/en-us/sql/sql-server/. Accessed: May 2026.

[32] Microsoft, *Azure Blob Storage documentation*. Available at: https://learn.microsoft.com/en-us/azure/storage/blobs/. Accessed: May 2026.

[33] Microsoft, *Visual Studio documentation*. Available at: https://learn.microsoft.com/en-us/visualstudio/. Accessed: May 2026.

[34] Robert C. Martin, *Clean Architecture: A Craftsman's Guide to Software Structure and Design*, Prentice Hall, 2017.

[35] Microsoft, *Visual Studio Code documentation*. Available at: https://code.visualstudio.com/docs. Accessed: May 2026.

[36] Google, *Android Studio documentation*. Available at: https://developer.android.com/studio. Accessed: May 2026.

[37] Postman, *Postman documentation*. Available at: https://learning.postman.com/docs/. Accessed: May 2026.

[38] Git, *Git documentation*. Available at: https://git-scm.com/doc. Accessed: May 2026.

[39] diagrams.net, *draw.io documentation*. Available at: https://www.drawio.com/. Accessed: May 2026.

[40] Figma, *Figma documentation*. Available at: https://help.figma.com/. Accessed: May 2026.

[41] Google, *Firebase Cloud Messaging documentation*. Available at: https://firebase.google.com/docs/cloud-messaging. Accessed: May 2026.

[42] Bootstrap Team, *Bootstrap Documentation*. Available at: https://getbootstrap.com/docs/ (Accessed: May 08, 2026).

[43] Google, *Google Maps Platform documentation*. Available at: https://developers.google.com/maps. Accessed: May 2026.

[44] OneSignal, *OneSignal Documentation*. Available at: https://documentation.onesignal.com/docs/en/home. Accessed: May 2026.

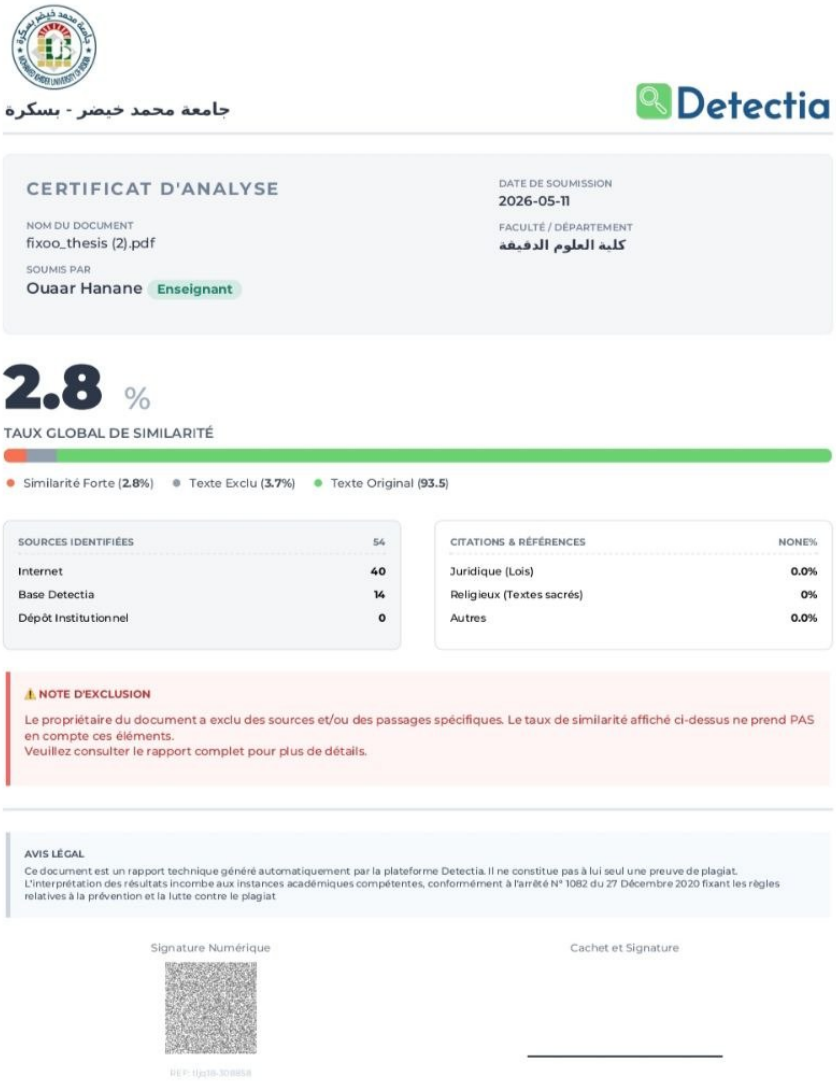
[45] OneSignal, *Flutter SDK Setup – OneSignal Documentation*. Available at: https://documentation.onesignal.com/docs/en/flutter-sdk-setup. Accessed: May 2026.

[46] SmartBear Software, *Swagger Documentation*. Available at: https://swagger.io/docs/. Accessed: May 2026.

[47] ngrok, *ngrok Documentation*. Available at: https://ngrok.com/docs/. Accessed: May 2026.

[48] OpenStreetMap, *OpenStreetMap Documentation*. Available at: https://www.openstreetmap.org/help. Accessed: May 2026.

[49] PlantUML, *PlantUML Documentation*. Available at: https://www.plantuml.com/. Accessed: May 2026.



جامعة محمد خيضر - بسكرة

Detectia

CERTIFICAT D'ANALYSE

NOM DU DOCUMENT
fixoo_thesis (2).pdf

SOUMIS PAR
Ouaar Hanane Enseignant

DATE DE SOUMISSION
2026-05-11

FACULTÉ / DÉPARTEMENT
كلية العلوم الدقيقة

2.8 %

TAUX GLOBAL DE SIMILARITÉ

Similarité Forte (2.8%) Texte Exclu (3.7%) Texte Original (93.5)

SOURCES IDENTIFIÉES	54
Internet	40
Base Detectia	14
Dépôt Institutionnel	0

CITATIONS & RÉFÉRENCES	NONE%
Juridique (Lois)	0.0%
Religieux (Textes sacrés)	0%
Autres	0.0%

⚠ NOTE D'EXCLUSION

Le propriétaire du document a exclu des sources et/ou des passages spécifiques. Le taux de similarité affiché ci-dessus ne prend PAS en compte ces éléments.
Veuillez consulter le rapport complet pour plus de détails.

AVIS LÉGAL

Ce document est un rapport technique généré automatiquement par la plateforme Detectia. Il ne constitue pas à lui seul une preuve de plagiat. L'interprétation des résultats incombe aux instances académiques compétentes, conformément à l'arrêté N° 1082 du 27 Décembre 2020 fixant les règles relatives à la prévention et la lutte contre le plagiat

Signature Numérique

Cachet et Signature

Figure 4.39: certificate

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي
اللجنة الوطنية لحاضنات الأعمال الجامعية

مشروع مؤسسة ناشئة

منصة رقمية لطلب خدمات الصيانة

منصة ذكية لربط طالبي الخدمات بمقدمي الخدمات عبر الفيديو

صورة العلامة التجارية

Fixoo

الاسم التجاري

Fixoo

إعداد:
مرغاد أيمن عبد الرؤوف
كمال الدين لحلوح

إشراف:
د. واعر حنان
د. واعر فطيمة

جامعة محمد خيضر - بسكرة
كلية العلوم الدقيقة والحياة / قسم الإعلام الآلي

السنة الجامعية 2025 / 2026

المحتويات

مقدمة **4**

1 المحور الأول: تقديم المشروع **5**
1.1 فكرة المشروع 5
2.1 القيم المقترحة 6
3.1 فريق العمل 7
4.1 أهداف المشروع 7
5.1 الجدول الزمني 8

2 المحور الثاني: الجوانب الابتكارية **9**
1.2 طبيعة الابتكار 9
2.2 مجالات الابتكار 9

3 المحور الثالث: التحليل الاستراتيجي للسوق **11**
1.3 السوق المستهدف 11
2.3 تحليل المنافسة 12
3.3 الاستراتيجية التسويقية 12
4.3 تحليل البيئة الداخلية والخارجية للمشروع 13
1.4.3 تحليل SWOT 14
2.4.3 تحليل PORTER 15

4 المحور الرابع: خطة الإنتاج والتنظيم **16**
1.4 عملية الإنجاز 16
2.4 الموارد 17
3.4 الشراكات 17

5 المحور الخامس: الخطة المالية **19**
1.5 الاستثمار الأولي 19
2.5 تقدير رقم الأعمال 20
3.5 التكاليف التشغيلية الشهرية والسنوية 20
4.5 الكتلة الأجرية 21
5.5 حساب النتيجة السنوية 21
6.5 نقطة التعادل ومؤشرات الاسترداد 22
7.5 الملخص التنفيذي لدراسة التكاليف والعوائد 23

6 المحور السادس: النموذج الأولي التجريبي **24**
1.6 وصف النموذج الأولي 24
2.6 التقنيات والأدوات المستخدمة 24
3.6 مراحل تطوير النموذج الأولي 25
4.6 النتائج الأولية 25
5.6 صور منصة الويب الإدارية 25
6.6 صور تطبيق الموبايل 30
1.6.6 واجهات التسجيل والدخول 30
2.6.6 الواجهة الرئيسية 31
3.6.6 نشر طلب الخدمة بالفيديو 32
4.6.6 إدارة العروض 33
5.6.6 تنفيذ المشروع وتتبع الخدمة 34
6.6.6 التتبع الجغرافي 35
7.6.6 نظام الدردشة 36
8.6.6 واجهات مزود الخدمة 37
9.6.6 التقييمات والشكاوى 37
10.6.6 الملف الشخصي والإعدادات 39

عرض الاستبيان **40**

الملحق رقم 03: نموذج العمل التجاري BMC **44**

خاتمة **45**

مقدمة

تشهد المشاريع الناشئة في السنوات الأخيرة تطوراً متسارعاً بفضل التقدم التكنولوجي والتحول الرقمي الذي أصبح يشمل مختلف مجالات الحياة، خاصة في قطاع الخدمات. وقد ساهم هذا التحول في ظهور حلول رقمية مبتكرة تهدف إلى تحسين جودة الخدمات وتسهيل الوصول إليها، مع تعزيز الكفاءة والشفافية في التعاملات اليومية.

في هذا السياق، برزت الحاجة إلى تطوير منصات ذكية قادرة على ربط مختلف الفاعلين في سوق العمل، خصوصاً في مجالات الخدمات والحرف، بطريقة أكثر تنظيماً وفعالية. إذ لا يزال العديد من المستخدمين يواجهون صعوبات في العثور على مزودي خدمات موثوقين، في حين يفتقر مقدمو الخدمات إلى قنوات حديثة تتيح لهم عرض مهاراتهم والوصول إلى فرص عمل بشكل مباشر. وانطلاقاً من هذه الإشكالية، يهدف هذا المشروع إلى اقتراح حل رقمي مبتكر يعتمد على تقنيات حديثة لتسهيل عملية التواصل بين طالبي الخدمات ومقدميها، مع التركيز على تحسين تجربة المستخدم، تقليل الوقت والجهد، وتعزيز مبدأ الشفافية من خلال آليات تفاعلية متقدمة.

كما يسعى المشروع إلى خلق بيئة رقمية ديناميكية تساهم في دعم الاقتصاد المحلي، وتشجيع روح المبادرة، وفتح آفاق جديدة لفرص العمل، مع قابلية التطوير والتوسع مستقبلاً وفق متطلبات السوق.

1 المحور الأول: تقديم المشروع

1.1 فكرة المشروع

يتمثل مشروع **Fixoo** في تطوير منصة رقمية مبتكرة تندرج ضمن مجال **الخدمات عند الطلب On-demand services** ، وتهدف إلى ربط طالبي الخدمات بمقدميها بطريقة حديثة تعتمد على المحتوى المرئي (الفيديو) بدل الأساليب التقليدية القائمة على الوصف النصي أو الاتصال المباشر.

جاءت فكرة المشروع استجابةً للتطور الملحوظ في سلوك المستخدمين في العصر الرقمي، خاصة مع الانتشار الواسع للتطبيقات المعتمدة على الفيديوهات القصيرة، حيث أصبح المستخدم يميل إلى الفهم البصري السريع بدل القراءة. في المقابل، لا تزال أغلب المنصات الحالية في مجال الخدمات تعتمد على طرق تقليدية لا تعكس بدقة طبيعة الطلب، مما يؤدي في كثير من الأحيان إلى سوء فهم بين العميل ومقدم الخدمة.

وانطلاقاً من هذه الإشكالية، تم تصميم منصة **Fixoo** التي تمكّن المستخدم من تسجيل ونشر فيديو قصير يشرح فيه الخدمة المطلوبة بشكل واضح (المشكلة). يتم عرض هذه الطلبات ضمن واجهة تفاعلية تعتمد على أسلوب التمرير (Scroll) مما يتيح لمقدمي الخدمات تصفحها بسهولة واختيار الطلبات المناسبة لهم.

عند العثور على طلب مناسب، يقوم مقدم الخدمة بإرسال عرض يتضمن السعر المقترح والمدة الزمنية اللازمة لإنجاز المهمة. ويتلقى العميل بدوره عدة عروض، مما يتيح له المقارنة بينها واختيار الأنسب وفق معايير متعددة مثل السعر، الوقت، وكفاءة مقدم الخدمة.

بعد قبول العرض، يتم تفعيل مرحلة تنفيذ الخدمة، حيث توفر المنصة وسائل تواصل مباشرة بين الطرفين، بالإضافة إلى خاصية التتبع الجغرافي المباشر التي تتيح معرفة موقع مقدم الخدمة إلى غاية الوصول. كما تعتمد المنصة على نظام تقييم شفاف يمكّن العملاء من تقييم مقدمي الخدمات بعد انتهاء المهمة، مما يساهم في تعزيز الثقة وتحسين جودة الخدمات.

يعتمد المشروع كذلك على نموذج اقتصادي قائم على العمولة، حيث تقوم المنصة بتحصيل نسبة من كل عملية ناجحة.

ومن الناحية التقنية، يتم تطوير النظام من خلال:

- تطبيق موبايل موجه للمستخدمين (العملاء ومقدمي الخدمات).
- منصة ويب مخصصة للإدارة (لوحة تحكم).
- بنية خلفية (Back-end) لربط مكونات النظام عبر واجهات برمجية .(API)
- الاعتماد على خدمات سحابية لتخزين ومعالجة البيانات، خاصة الفيديوهات.

يتم إنجاز هذا المشروع في إطار أكاديمي جامعي، مع قابلية تطويره مستقبلاً ليصبح مؤسسة ناشئة (Startup) قادرة على التوسع في السوق الرقمية للخدمات.

مجال النشاط:

ينشط مشروع **Fixoo** في مجال **الخدمات الرقمية عند الطلب**، حيث يجمع بين تكنولوجيا التطبيقات المحمولة والخدمات الحرفية والمهنية، بهدف رقمنة وتسهيل الوصول إلى مختلف الخدمات اليومية في السوق الجزائري.

الخدمات المقدمة:

- تمكين المستخدمين من نشر طلبات خدمات عبر فيديوهات قصيرة توضح المشكلة أو الحاجة بشكل دقيق.
- توفير منصة تفاعلية لمقدمي الخدمات (حرفيين، تقنيين، عمال...) لعرض مهاراتهم والتفاعل مع الطلبات المتاحة.
- نظام استقبال العروض، حيث يقدم مقدمو الخدمات عروضهم (السعر + مدة الإنجاز) لكل طلب.
- إمكانية المقارنة بين العروض واختيار الأنسب من طرف العميل.
- توفير نظام تواصل مباشر بين العميل ومقدم الخدمة بعد قبول العرض.
- خاصية التتبع الجغرافي المباشر لمقدم الخدمة إلى غاية الوصول.
- نظام تقييم ومراجعات يساهم في تعزيز الثقة وتحسين جودة الخدمات.
- توفير لوحة تحكم إدارية لإدارة المستخدمين والعمليات داخل المنصة.

2.1 القيم المقترحة

- **الابتكار:** تعتمد منصة Fixoo على نموذج حديث قائم على الفيديو القصير بدل الوصف النصي، مما يوفر طريقة أكثر وضوحًا لفهم الطلب.
- **الأداء:** تتيح المنصة تجربة سريعة وفعالة من خلال تقليل الوقت اللازم للعثور على الخدمة المناسبة.
- **التخصيص:** يمكن لمقدمي الخدمات تقديم عروض مخصصة حسب كل طلب من حيث السعر ومدة التنفيذ.
- **إنجاز المهمة:** تساعد المنصة المستخدمين على إنجاز احتياجاتهم اليومية بسهولة من خلال ربطهم مباشرة بمقدمي خدمات مؤهلين.
- **التصميم:** تعتمد Fixoo على واجهة حديثة وسلسة تعتمد على التمرير مما يجعل تجربة الاستخدام بسيطة وجذابة.
- **السعر:** توفر المنصة بيئة تنافسية تسمح للعميل بالحصول على أفضل عرض من حيث السعر.
- **تقليل التكاليف:** تقلل المنصة من التكاليف المرتبطة بالبحث عن الخدمة أو التعامل مع وسطاء.
- **تقليل المخاطر:** توفر Fixoo نظام تقييم وتتبع مباشر مما يعزز الثقة بين المستخدمين ويقلل من المخاطر.

- **سهولة الوصول:** تمكن المنصة المستخدمين من الوصول إلى عدد كبير من مقدمي الخدمات في مختلف المجالات.
- **سهولة الاستخدام:** تتميز المنصة بواجهة بسيطة تعتمد على التفاعل البصري مما يجعلها مناسبة لجميع الفئات.

3.1 فريق العمل

العضو	المهمة
مرغاد أيمن عبد الرؤوف	تطوير تطبيق الموبايل وتصميم الواجهات وتجربة المستخدم
كمال الدين لحلوح	تطوير النظام الخلفي وقاعدة البيانات ولوحة التحكم الإدارية

4.1 أهداف المشروع

1. إطلاق منصة Fixoo في السوق المحلي بداية بمدينة بسكرة واستقطاب ما بين 500 إلى 5000 مستخدم خلال المرحلة الأولى.
2. جذب ما بين 200 إلى 1000 مقدم خدمة نشط على المنصة خلال الأشهر الأولى من الإطلاق.
3. بناء قاعدة مستخدمين نشطة وتحقيق معدل تفاعل مرتفع من خلال نشر الطلبات واستقبال العروض بشكل يومي.
4. تقليل متوسط وقت العثور على مقدم خدمة إلى أقل من 24 ساعة بفضل نظام العروض المباشر.
5. تحقيق معدل رضا المستخدمين يفوق 80% من خلال نظام التقييم والمتابعة.
6. توسيع نطاق المنصة لتشمل على الأقل 10 ولايات جزائرية خلال المرحلة المتوسطة.
7. تطوير خصائص إضافية مثل نظام الترتيب والدفع الإلكتروني لرفع معدل الاستخدام بنسبة 30%.
8. الوصول إلى أكثر من 50,000 مستخدم و 10,000 مقدم خدمة خلال السنوات الأولى.
9. تحقيق حصة سوقية معتبرة في مجال الخدمات عند الطلب داخل الجزائر.
10. ضمان الاستدامة المالية عبر تحقيق دخل منتظم من نظام العمولة ونماذج الاشتراك.
11. التوسع مستقبلاً نحو أسواق عربية مشابهة بعد إثبات نجاح النموذج محلياً.

5.1 الجدول الزمني

رقم	النشاط	المدة
1	تحليل الفكرة ودراسة السوق	2 أسابيع
2	تحديد المتطلبات الوظيفية والتقنية	2 أسابيع
3	تصميم النظام وقاعدة البيانات	4 أسابيع
4	تصميم واجهات تطبيق الموبايل	1 أسبوع
5	تطوير النظام الخلفي وواجهات API	6 أسابيع
6	تطوير تطبيق الموبايل	6 أسابيع
7	تطوير لوحة التحكم الإدارية	5 أسابيع
8	اختبار النظام وتصحيح الأخطاء	3 أسابيع

2 المحور الثاني: الجوانب الابتكارية

1.2 طبيعة الابتكار

يعتمد مشروع **Fixoo** على توظيف التكنولوجيا الحديثة من أجل تقديم نموذج مبتكر في مجال الخدمات الرقمية، حيث يجمع بين المحتوى المرئي والتفاعل المباشر والتتبع الجغرافي لتحسين تجربة المستخدم وتسهيل عملية الوصول إلى الخدمات.

ويتميز المشروع باعتماده على مجموعة من الابتكارات المرتبطة بالسوق والتكنولوجيا، والتي تهدف إلى تحسين كفاءة التواصل بين طالبي الخدمات ومقدميها، مع توفير بيئة أكثر شفافية وسهولة مقارنة بالحلول التقليدية.



شكل 1: طبيعة الابتكارات المعتمدة في مشروع Fixoo

2.2 مجالات الابتكار

- **عمليات جديدة:** يعتمد المشروع على رقمنة عملية طلب الخدمات من خلال الفيديو والتفاعل المباشر، مما يساهم في تحسين كفاءة العمليات وتقليل الوقت اللازم للوصول إلى مقدم الخدمة المناسب.
- **تجارب جديدة:** يوفر Fixoo تجربة استخدام حديثة تعتمد على التصفح بأسلوب التمرير ومشاهدة الفيديوهات القصيرة بدل الطرق التقليدية المعتمدة على النصوص والإعلانات.
- **ميزات جديدة:** يقدم المشروع خصائص مبتكرة مثل نشر طلب الخدمة عبر الفيديو، استقبال عروض متعددة، التتبع الجغرافي المباشر، ونظام التقييمات.
- **عملاء جدد:** يستهدف المشروع فئة واسعة من المستخدمين تشمل طالبي الخدمات اليومية والحرفيين والتقنيين الذين لا يملكون وسائل رقمية فعالة للوصول إلى العملاء.

- **عروض جديدة:** يقدم المشروع نموذجًا جديدًا للخدمات الرقمية عند الطلب يعتمد على التفاعل المرئي والمنافسة بين مقدمي الخدمات.
- **نماذج جديدة:** يعتمد المشروع على نموذج اقتصادي قائم على العمولة والتفاعل الرقمي المباشر بدل الوساطة التقليدية.
- **الجوانب الابتكارية للمشروع:** يتمثل الجانب الابتكاري في Fixoo في كونه من أوائل المشاريع في الجزائر التي توظف الفيديو القصير والتتبع المباشر في مجال الخدمات عند الطلب، مع التركيز على تسهيل التواصل وبناء الثقة بين المستخدمين.

3 المحور الثالث: التحليل الاستراتيجي للسوق

1.3 السوق المستهدف

يستهدف مشروع **Fixoo** سوق الخدمات الرقمية عند الطلب، والذي يشمل الأفراد الذين يحتاجون بشكل يومي إلى خدمات الصيانة، الإصلاح، أو الخدمات المنزلية المختلفة، إضافة إلى مقدمي الخدمات والحرفيين الباحثين عن فرص عمل جديدة وزيادة عدد الزبائن.

ويتمثل السوق المحتمل للمشروع في:

- الأفراد الذين يواجهون مشاكل منزلية أو تقنية ويبحثون عن حلول سريعة وموثوقة.
- أصحاب السيارات الذين يحتاجون إلى خدمات إصلاح أو صيانة متنقلة.
- أصحاب الورشات والمنازل الذين يحتاجون إلى حرفيين أو عمال بشكل مستمر.
- الحرفيون ومقدمو الخدمات مثل البنائين، السباكين، الكهربائيين، الميكانيكيين، وعمال الصيانة الباحثين عن فرص عمل إضافية.

أما السوق المستهدف فيتمثل أساسًا في:

- فئة الشباب والأشخاص المعتادين على استخدام التطبيقات الرقمية والمحتوى المرئي.
- الأفراد الذين يفضلون شرح مشاكلهم عبر الفيديو بدل الوصف النصي التقليدي.
- مقدمي الخدمات المستقلين الراغبين في الوصول إلى عدد أكبر من العملاء بطريقة حديثة وسريعة.
- المستخدمين داخل المدن الجزائرية ذات الكثافة السكانية العالية مثل الجزائر العاصمة، وهران، قسنطينة، وبسكرة.

وقد تم اختيار هذه الشريحة السوقية نظرًا لعدة عوامل أهمها:

- الانتشار الكبير لاستخدام الهواتف الذكية والتطبيقات الرقمية في الجزائر.
- تزايد الاعتماد على الفيديوهات القصيرة والمحتوى المرئي في الحياة اليومية.
- وجود صعوبة لدى الكثير من العملاء في إيجاد حرفيين موثوقين بسرعة.
- حاجة الحرفيين إلى منصات حديثة تساعدهم على الحصول على فرص عمل بشكل مستمر.
- غياب منصة جزائرية تعتمد على الفيديو والتفاعل المباشر في مجال الخدمات عند الطلب.

كما يسعى المشروع مستقبلًا إلى بناء شراكات وعقود تعاون مع بعض المؤسسات الصغيرة، وورشات الصيانة، وشركات الخدمات المنزلية، بهدف توسيع قاعدة المستخدمين وتعزيز انتشار المنصة داخل السوق الجزائرية.

2.3 تحليل المنافسة

المنافس	نقاط القوة	نقاط الضعف
دقدق	• الأقدمية في السوق الجزائري • قاعدة مستخدمين موثوقة • قوة العلامة التجارية • انتشار جيد في الجزائر العاصمة • تسويق رقمي قوي	• اعتماد الطريقة التقليدية في طلب الخدمات • غياب نظام الفيديو التفاعلي • عدم توفير حرية مقارنة العروض بشكل مرن • غياب خاصية تتبع مقدم الخدمة • الاعتماد على الوصف النصي فقط
خليها علينا	• حضور معروف في السوق المحلي • سهولة الوصول إلى بعض الحرفيين • امتلاك قاعدة أولية من المستخدمين • خبرة سابقة في مجال الخدمات	• غياب المحتوى المرئي داخل المنصة • طرق تواصل تقليدية بين العميل والحرفي • محدودية التفاعل داخل التطبيق • عدم وجود نظام تتبع مباشر • محدودية خيارات مقارنة الأسعار والخدمات

3.3 الاستراتيجية التسويقية

- **الخدمة المقدمة:** يقدم مشروع Fixoo منصة رقمية حديثة تربط بين طالبي الخدمات ومقدميها بطريقة مبتكرة تعتمد على الفيديو القصير، مع توفير نظام عروض، تتبع مباشر، وتقييمات لتحسين تجربة المستخدم.
- **سياسة التسعير:** يعتمد المشروع على نموذج اقتصادي قائم على العمولة، حيث يتم اقتطاع نسبة من كل خدمة مكتملة من طرف مقدم الخدمة. كما يخطط المشروع مستقبلًا لإضافة نظام اشتراكات شهرية وسنوية للحرفيين بدل نظام العمولة.
- **الدفع:** يتم حاليًا الدفع بشكل مباشر بين العميل ومقدم الخدمة، مع التخطيط مستقبلًا لإدماج نظام دفع إلكتروني يسمح بإتمام العمليات المالية داخل التطبيق.
- **السوق المستهدف:** يستهدف المشروع في مرحلته الأولى الأفراد داخل مدينة بسكرة، ثم التوسع تدريجيًا نحو الورشات الصغيرة والشركات ومختلف ولايات الجزائر.

- **قنوات التوزيع:** يعتمد المشروع على تطبيق موبايل كوسيلة أساسية لتقديم الخدمات وإدارة الطلبات والتواصل بين المستخدمين.
- **طرق الترويج:** سيتم الاعتماد على التسويق الرقمي عبر منصات Facebook و Instagram و TikTok من خلال الفيديوهات القصيرة والإعلانات الممولة، إضافة إلى التعاون مع المؤثرين والحرفيين المعروفين لزيادة انتشار المنصة.
- **الميزة التنافسية:** تتمثل أهم نقطة قوة تسويقية في Fixoo في اعتماد نظام الفيديو والتصفح بأسلوب التمرير، مع إمكانية مقارنة العروض والأسعار، مما يمنح المستخدم تجربة أكثر وضوحًا ومرونة مقارنة بالحلول التقليدية.
- **استراتيجية جذب الحرفيين:** يعتمد المشروع على تحفيز الحرفيين ومقدمي الخدمات النشطين من خلال تحسين ظهورهم داخل المنصة، إضافة إلى توفير اشتراكات شهرية اختيارية تمنحهم أولوية الظهور في نتائج البحث وزيادة فرص الحصول على العملاء.
- **التوسع المستقبلي:** يخطط المشروع لإضافة خصائص متقدمة مثل الدفع الإلكتروني، الذكاء الاصطناعي للتحقق من المحتوى والوثائق، وتطوير نظام إعلانات وشراكات مع شركات الصيانة والخدمات.
- **علاقة المؤسسة بالزبائن:** يتيح المشروع للمستخدمين تقديم الشكاوى والتقييمات والمقترحات عبر التطبيق، مع العمل على معالجتها وتحسين جودة الخدمات بشكل مستمر.

4.3 تحليل البيئة الداخلية والخارجية للمشروع

يتطلب نجاح مشروع **Fixoo** فهم البيئة الداخلية والخارجية المؤثرة عليه من أجل تحديد نقاط القوة والفرص والمخاطر المحتملة، إضافة إلى دراسة شدة المنافسة داخل السوق المستهدف.
يعتمد التحليل على الأداتين التاليتين:

- تحليل SWOT لتقييم البيئة الداخلية والخارجية للمشروع.
- تحليل PORTER لدراسة المنافسة داخل سوق الخدمات الرقمية.

1.4.3 تحليل SWOT

نقاط القوة	نقاط الضعف
• نموذج مبتكر يعتمد على الفيديو بدل الوصف النصي. • واجهة تفاعلية تعتمد على التصفح بالتمرير. • إمكانية مقارنة العروض والأسعار بين عدة مزودي خدمات. • وجود نظام تتبع مباشر وتقييمات. • منصة موجهة خصيصًا للسوق الجزائري. • قابلية إضافة الذكاء الاصطناعي والدفع الإلكتروني مستقبلاً.	• المشروع ما يزال في مرحلة الإطلاق الأولية. • الحاجة إلى تمويل للتسويق الرقمي والتوسع. • الاعتماد الحالي على الدفع التقليدي خارج المنصة. • الحاجة إلى بناء قاعدة مستخدمين واسعة لزيادة النشاط. • ارتفاع متطلبات تخزين ومعالجة الفيديوهات.
الفرص	**التهديدات**
• التوجه المتزايد نحو الرقمنة والخدمات الرقمية. • نمو استخدام تطبيقات الفيديو القصير. • توسع سوق الخدمات عند الطلب داخل الجزائر. • إمكانية التعاون مع شركات الصيانة والورشات. • فرص التوسع نحو ولايات وأسواق جديدة.	• دخول منافسين جدد إلى السوق. • صعوبة تغيير سلوك المستخدمين التقليدي. • ارتفاع تكاليف التسويق واكتساب المستخدمين. • ظهور منصات مشابهة تعتمد على نفس الفكرة. • تحديات الثقة في المعاملات الرقمية.

2.4.3 تحليل PORTER

العامل	التحليل
شدة المنافسة	تعتبر المنافسة متوسطة داخل السوق الجزائري، حيث توجد بعض المنصات مثل دقدق وخليها علينا، إلا أنها تعتمد على أساليب تقليدية ولا توفر نموذج الفيديو والتصفح التفاعلي الذي تعتمد عليه Fixoo.
تهديد دخول منافسين جدد	يعد متوسطًا بسبب نمو سوق الخدمات الرقمية، لكن بناء قاعدة مستخدمين قوية واعتماد نموذج الفيديو يمكن أن يشكل حاجزًا تنافسيًا مستقبلاً.
القوة التفاوضية للمستخدمين	مرتفعة نسبيًا بسبب توفر بدائل تقليدية مثل البحث المباشر عبر مواقع التواصل أو العلاقات الشخصية، مما يتطلب تحسين تجربة المستخدم باستمرار.
القوة التفاوضية للشركاء ومقدمي الخدمات	متوسطة، لأن المنصة تعتمد على الحرفيين والورشات كمزودي خدمات، مع إمكانية جذبهم عبر زيادة فرص العمل والظهور داخل التطبيق.
تهديد الخدمات البديلة	توجد بدائل تقليدية مثل صفحات التواصل الاجتماعي والمجموعات المحلية، لكنها تفتقر إلى التتبع والتقييم ومقارنة العروض.

وبناءً على التحليل السابق، يتمتع مشروع **Fixoo** بميزة تنافسية واضحة تعتمد على المحتوى المرئي والتفاعل المباشر، مع وجود فرص نمو قوية داخل السوق المحلي رغم بعض التحديات المتعلقة بالتسويق وبناء الثقة.

4 المحور الرابع: خطة الإنتاج والتنظيم

1.4 عملية الإنجاز

تمر عملية تقديم الخدمة داخل منصة **Fixoo** بعدة مراحل مترابطة تهدف إلى تسهيل عملية التواصل بين العميل ومقدم الخدمة، وضمان السرعة والوضوح في تنفيذ الطلبات. وتعتمد المنصة على نموذج حديث قائم على الفيديو والتفاعل المباشر بين مختلف الأطراف.

1. **نشر طلب الخدمة:** يقوم العميل بتسجيل ونشر فيديو قصير يشرح فيه المشكلة أو الخدمة المطلوبة بشكل واضح.
2. **عرض الطلبات داخل المنصة:** يتم عرض الفيديوهات داخل واجهة تفاعلية تعتمد على التمرير، مما يسمح لمقدمي الخدمات بتصفح الطلبات بسهولة.
3. **إرسال العروض:** يقوم مقدمو الخدمات بإرسال عروضهم التي تتضمن السعر المقترح، مدة الإنجاز، وبعض التفاصيل المتعلقة بالخدمة.
4. **اختيار العرض المناسب:** يتلقى العميل عدة عروض ويقوم بالمقارنة بينها لاختيار العرض الأنسب حسب السعر، الوقت، والتقييمات.
5. **بدء تنفيذ الخدمة:** بعد قبول العرض، يتم تفعيل التواصل بين الطرفين مع إمكانية تتبع مقدم الخدمة عبر الخريطة إلى غاية الوصول.
6. **إنهاء الخدمة والتقييم:** بعد انتهاء المهمة، يمكن للعميل تقييم مقدم الخدمة وتقديم ملاحظات أو شكاوى لتحسين جودة الخدمات داخل المنصة.



شكل 2: مخطط يوضح مراحل عمل منصة Fixoo

2.4 الموارد

- **الموارد البشرية:** يعتمد مشروع Fixoo في مرحلته الأولى على فريق عمل يتكون من مطوري تطبيقات موبايل مطوري Back-end ومشرف إداري لإدارة المنصة ومتابعة المستخدمين والشكاوى.
- **عدد مناصب العمل:** من المتوقع أن يساهم المشروع مستقبلاً في خلق عدة مناصب عمل مباشرة في مجالات التطوير، الإدارة، والدعم التقني، إضافة إلى مناصب عمل غير مباشرة لفائدة الحرفيين ومقدمي الخدمات المستعملين للمنصة.
- **نوعية اليد العاملة:** يحتاج المشروع إلى مطوري تطبيقات، مختصين في قواعد البيانات، مطوري Web وBack-end، إضافة إلى مشرفين لإدارة المنصة ومراقبة المحتوى.
- **الوسائل التقنية:** يعتمد المشروع على حواسيب للتطوير، هواتف ذكية لاختبار التطبيق، خوادم سحابية، قواعد بيانات، وخدمات خاصة بالتتبع الجغرافي والإشعارات.
- **البرمجيات المستخدمة:** يعتمد المشروع على Flutter لتطوير تطبيق الموبايل، وتقنيات NET. و C# لتطوير منصة الويب والنظام الخلفي، إضافة إلى SQL Server لإدارة قواعد البيانات، مع استعمال واجهات API لربط مختلف مكونات النظام وتبادل البيانات بينها، إلى جانب خدمات Azure لمعالجة وتخزين الفيديوهات.
- **الموردون والخدمات الخارجية:** يتعامل المشروع مع مزودي خدمات سحابية وتقنيات رقمية مثل خدمات التخزين السحابي، الخرائط، والإشعارات، مما يسمح بضمان استقرار المنصة وتحسين جودة الخدمة.
- **سياسة الدفع والاستلام:** يعتمد المشروع حاليًا على الدفع المباشر بين العميل ومقدم الخدمة، مع التخطيط مستقبلًا لإضافة وسائل دفع إلكترونية داخل التطبيق.
- **المناولة:** يمكن للمشروع مستقبلاً الاستعانة بجهات خارجية في بعض الجوانب مثل التسويق الرقمي، التحقق من الوثائق، أو خدمات الصيانة التقنية.

3.4 الشراكات

- **حاضنة الأعمال الجامعية:** تمثل حاضنة الأعمال الجامعية جهة داعمة للمشروع من خلال التوجيه، المرافقة، وتوفير بيئة مناسبة لتطوير المشروع وتحويله إلى مؤسسة ناشئة.
- **مزودو الخدمات السحابية:** يعتمد المشروع على شراكات مع مزودي الخدمات السحابية مثل Azure لتخزين الفيديوهات ومعالجتها وتحسين أداء المنصة.
- **شركات الصيانة والخدمات:** يسعى المشروع مستقبلاً إلى بناء شراكات مع شركات الصيانة والحرفيين المحترفين من أجل توسيع قاعدة مقدمي الخدمات وتحسين جودة العروض داخل المنصة.
- **شركات التسويق الرقمي:** يمكن التعاون مع وكالات أو مؤثرين في مجال التسويق الرقمي للمساعدة في الترويج للمنصة وزيادة انتشارها داخل السوق الجزائرية.

- **خدمات الخرائط والإشعارات:** يعتمد المشروع على خدمات خارجية خاصة بالخرائط، التتبع الجغرافي، والإشعارات لتحسين تجربة المستخدم وضمان فعالية التواصل داخل التطبيق.

5 المحور الخامس: الخطة المالية

1.5 الاستثمار الأولي

يعرض هذا الجزء مختلف التكاليف الأولية اللازمة لإطلاق المشروع.

أ.1. الاستثمار الأولي — منصة فيكسو FIXOO					
سعر الصرف: $1 = 132.37 دج					
#	البيان	التكلفة (دج)	التكلفة ($)	اهتلاك (سنة)	اهتلاك سنوي (دج)
التجهيزات المكتبية والإدارية — 1.1					
1	حاسوب محمول للمدير	80,000 دج	$604.37	3	26,667 دج
2	حاسوب محمول للمطور (احتياطي)	75,000 دج	$566.59	3	25,000 دج
3	طابعة متعددة الوظائف	25,000 دج	$188.86	5	5,000 دج
4	راوتر + شبكة إنترنت احتياطية	15,000 دج	$113.32	5	3,000 دج
5	أثاث مكتبي	30,000 دج	$226.64	10	3,000 دج
6	كاميرا ومعدات توثيق	20,000 دج	$151.09	5	4,000 دج
إجمالي التجهيزات		245,000 دج	$1,850.87		66,667 دج

المصاريف التأسيسية والقانونية (مرة واحدة) — 1.2					
1	رسوم التسجيل التجاري (CNRC)	15,000 دج	$113.32	مرة واحدة	
2	الرسوم القانونية والتوثيق	30,000 دج	$226.64	مرة واحدة	
3	تسجيل العلامة التجارية	20,000 دج	$151.09	مرة واحدة	
4	مصاريف محاسب قانوني (سنة أولى)	60,000 دج	$453.27	مرة واحدة	
إجمالي المصاريف التأسيسية		125,000 دج	$944.32		
إجمالي الاستثمار الأولي		370,000 دج	$2,795.20		

شكل 3: الاستثمار الأولي

2.5 تقدير رقم الأعمال

يعرض هذا الجزء التقديرات الأولية للإيرادات المتوقعة.

أ.2. تقدير رقم الأعمال — منصة فيكسو FIXOO					
مصادر الدخل: عمولة على الطلبات + اشتراك ظهور المزودين					
الافتراضات الرئيسية					
#	البيان	القيمة	الوحدة	السنة 2	السنة 3	ملاحظة
1	متوسط سعر الخدمة الواحدة	4,000	دج/طلب	4,500	5,000	
2	نسبة عمولة المنصة	10.0%	%	0	0	نسبة العمولة المنصة تختلف حسب اختلاف سعر الخدمة
3	عدد الطلبات شهرياً	200	طلب/شهر	1,000	2,500	
4	سعر اشتراك المزود (شهري)	2,000	دج/مزود	2,000	2,000	يمكن أيضا عمل اشتراك سنوي للمزودين
5	عدد المزودين المشتركين	200	مزود	1,000	2,000	
حساب الإيرادات السنوية						
#	مصدر الإيراد	السنة 1 (دج)	السنة 1 ($)	السنة 2 (دج)	السنة 3 (دج)	ملاحظة
1	عمولات الطلبات	960,000 دج	$7,252.40	4,800,000 دج	12,000,000 دج	عمولة على كل طلب مكتمل
2	اشتراكات ظهور المزودين	400,000 دج	$3,021.83	2,000,000 دج	4,000,000 دج	اشتراك شهري لكل مزود
إجمالي رقم الأعمال		1,360,000 دج	$10,274.23	6,800,000 دج	16,000,000 دج	

شكل 4: تقدير رقم الأعمال

3.5 التكاليف التشغيلية الشهرية والسنوية

يعرض هذا الجزء المصاريف التشغيلية الخاصة بالمشروع.

أ.3. التكاليف التشغيلية الشهرية والسنوية — منصة فيكسو FIXOO					
(بدون أجور)					
#	البيان	الشهري (دج)	الشهري ($)	السنوي (دج)	السنوي ($)
التكاليف التقنية (خدمات سحابية) — 1					
1	Azure App Service (خادم التطبيق)	2,648 دج	$20.00	31,776 دج	$240.06
2	Azure SQL Database (قاعدة البيانات)	1,989 دج	$15.03	23,868 دج	$180.31
3	Azure Blob Storage (تخزين الملفات والفيديو)	662 دج	$5.00	7,944 دج	$60.01
4	معالجة الفيديو (Azure Media Services)	1,989 دج	$15.03	23,868 دج	$180.31
5	Mapbox Maps (خرائط)	1,324 دج	$10.00	15,888 دج	$120.03
6	SMS OTP — infobip (رسائل التحقق)	2,648 دج	$20.00	31,776 دج	$240.05
7	Google Play Store (اشتراك سنوي ÷ 12)	275 دج	$2.08	3,300 دج	$24.93
8	App Store (اشتراك سنوي ÷ 12)	1,092 دج	$8.25	13,104 دج	$99.00
9	Domain Name (اسم النطاق ÷ 12)	110 دج	$0.83	1,320 دج	$9.97
إجمالي التكاليف التقنية		12,737 دج	$96.22	152,844 دج	$1,154.68

المصاريف التشغيلية العامة — 2					
1	الإنترنت (خط احترافي)	4,000 دج	$30.22	48,000 دج	$362.62
2	إيجار المكتب	20,000 دج	$151.09	240,000 دج	$1,813.10
3	تسويق رقمي (Meta + Google Ads)	26,487 دج	$200.10	317,844 دج	$2,401.18
4	تصميم ومحتوى بصري	13,243 دج	$100.05	158,916 دج	$1,200.54
5	مصاريف قانونية / إدارية شهرية	2,649 دج	$20.01	31,788 دج	$240.15
6	صيانة المعدات والأنظمة	3,000 دج	$22.66	36,000 دج	$271.96
7	تأمين المعدات	2,000 دج	$15.11	24,000 دج	$181.31
8	مصاريف طوارئ متنوعة	5,000 دج	$37.77	60,000 دج	$453.27
إجمالي المصاريف العامة		76,379 دج	$577.01	916,548 دج	$6,924.14
إجمالي التكاليف التشغيلية (بدون أجور)		89,116 دج	$673.23	1,069,392 دج	$8,078.81

شكل 5: التكاليف التشغيلية الشهرية والسنوية

4.5 الكتلة الأجرية

أ.4. الكتلة الأجرية — منصة فيكسو FIXOO								
#	المنصب	الأجر الصافي (دج/شهر)	عدد (س1)	الكتلة الشهرية (س1)	الكتلة السنوية (س1)	عدد (س2)	الكتلة السنوية (س2)	عدد (س3)	الكتلة السنوية (س3)
1	موظف إداري	30,000 دج	1	30,000 دج	360,000 دج	2	720,000 دج	2	720,000 دج
2	مطور برمجيات Front-end	65,000 دج	1	65,000 دج	780,000 دج	1	780,000 دج	2	1,560,000 دج
3	مطور برمجيات Back-end	65,000 دج	1	65,000 دج	780,000 دج	2	1,560,000 دج	3	2,340,000 دج
4	منسق رقمي	40,000 دج	1	40,000 دج	480,000 دج	2	960,000 دج	4	1,920,000 دج
الإجمالي				200,000 دج	2,400,000 دج		4,020,000 دج		6,540,000 دج

شكل 6: الكتلة الأجرية

5.5 حساب النتيجة السنوية

ب.1. حساب النتيجة السنوية — منصة فيكسو FIXOO					
قيم السنة 2 و3 باستخدام افتراضات النمو					
#	البيان	السنة 1 (دج)	السنة 2 (دج)	السنة 3 (دج)	ملاحظة
1	رقم الأعمال (إيرادات)	1,360,000.00 دج	6,800,000.00 دج	16,000,000.00 دج	من ورقة الإيرادات
2	التكاليف التشغيلية (بدون أجور)	1,069,392.48 دج	1,069,392.48 دج	1,069,392.48 دج	ثابتة
3	الأجور والمرتبات	2,400,000.00 دج	4,020,000.00 دج	6,540,000.00 دج	من ورقة الأجور
الربح الإجمالي (قبل الأجور)		290,607.52 دج	5,730,607.52 دج	14,930,607.52 دج	إيرادات - تكاليف تشغيلية
صافي الربح (بعد الأجور)		(2,109,392.48 دج)	1,710,607.52 دج	8,390,607.52 دج	الربح الإجمالي - الأجور

شكل 7: حساب النتيجة السنوية

6.5 نقطة التعادل ومؤشرات الاسترداد

ج.1. نقطة التعادل ومؤشرات الاسترداد — منصة فيكسو FIXOO				
يساعد هذا الجدول على تحديد: كم طلباً نحتاج شهرياً لتغطية التكاليف وما هو اشتراك المزود المطلوب				
التكاليف الإجمالية الواجب استردادها				
#	البيان	المبلغ ($)	المبلغ (دج)	ملاحظة
1	الاستثمار الأولي (تجهيزات + تأسيس)	**$2,795.20**	370,000 دج	بدون تطوير المنصة
2	إجمالي التكاليف التشغيلية السنوية (بدون أجور)	$8,078.81	1,069,392 دج	
3	الأجور السنوية (السنة 1)	$18,131.00	2,400,000 دج	
إجمالي التكاليف السنوية (تشغيل + أجور)		**$26,209.81**	**3,469,392 دج**	

سيناريوهات نقطة التعادل — عمولة المنصة: 400 دج/طلب (10% × 4,000 دج) | اشتراك فعلي: 2,000 دج/مزود				
السيناريو	عدد الطلبات / شهر	دخل العمولات (دج/شهر)	عدد المزودين المطلوبين (بـ 2,000 دج/مزود)	أشهر لاسترداد الاستثمار
سيناريو 1 — 100 طلب/شهر (متحفظ)	100	40,000.00	125	2.45
سيناريو 2 — 200 طلب/شهر (هدف السنة 1)	200	80,000.00	105	1.94
سيناريو 3 — 500 طلب/شهر (متوسط)	500	200,000.00	45	1.19
نقطة التعادل الكاملة — عمولات تغطي كل التكاليف بدون اشتراكات	**723**	**289,200.00**	**0**	**0.92**

فترة استرداد التكاليف — 1 / 2 / 3 سنوات				
الفترة	إيرادات متوقعة (دج)	تكاليف إجمالية (دج)	الربح التراكمي (دج)	الحكم
السنة 1	1,360,000.00 دج	3,839,392.48 دج	**(2,479,392.48 دج)**	لم يتم بعد ✗
السنة 2	6,800,000.00 دج	6,938,784.96 دج	**(2,618,177.44 دج)**	لم يتم بعد ✗
السنة 3	16,000,000.00 دج	10,408,177.44 دج	**2,973,645.12 دج**	تم الاسترداد ✓

شكل 8: نقطة التعادل ومؤشرات الاسترداد

7.5 الملخص التنفيذي لدراسة التكاليف والعوائد

الملخص التنفيذي — دراسة التكاليف والعوائد — منصة فيكسو FIXOO				
سعر الصرف: 1$ = 132.37 دج				
أ — الاستثمار الأولي				
#	البيان	المبلغ ($)	المبلغ (دج)	ملاحظة
1	التجهيزات المكتبية والإدارية	**$1,850.87**	245,000 دج	
2	المصاريف التأسيسية والقانونية	**$944.32**	125,000 دج	
إجمالي الاستثمار الأولي		**$2,795.20**	**370,000 دج**	
ب — التكاليف التشغيلية الشهرية				
#	البيان	المبلغ ($)	المبلغ (دج)	ملاحظة
1	التكاليف التقنية (سحابية)	$96.22	12,737 دج	
2	المصاريف التشغيلية العامة	$577.01	76,379 دج	
3	الأجور والرواتب (شهري)	$1,510.92	200,000 دج	
إجمالي التكاليف الشهرية		**$2,184.15**	**289,116 دج**	

ج — رقم الأعمال المتوقع والنتيجة				
السنة	الإيرادات (دج)	التكاليف الإجمالية (دج)	صافي الربح (دج)	الحكم
السنة 1	1,360,000.00 دج	3,469,392.48 دج	**(2,109,392.48 دج)**	خسارة ✗
السنة 2	6,800,000.00 دج	5,089,392.48 دج	**1,710,607.52 دج**	ربح ✓
السنة 3	16,000,000.00 دج	7,609,392.48 دج	**8,390,607.52 دج**	ربح ✓

شكل 9: الملخص التنفيذي

6 المحور السادس: النموذج الأولي التجريبي

يمثل النموذج الأولي التجريبي لمنصة **Fixoo** النسخة الأولى القابلة للتجربة من المشروع، حيث تم تطوير مجموعة من الخصائص الأساسية التي تسمح بمحاكاة طريقة عمل المنصة بشكل واقعي، بداية من نشر طلب الخدمة عبر الفيديو إلى غاية تنفيذ المهمة وتتبع مقدم الخدمة.

وقد تم تطوير هذا النموذج بهدف اختبار الفكرة عمليًا، ودراسة تجربة المستخدم، والتحقق من إمكانية تطبيق النظام داخل السوق الجزائرية، مع التركيز على سهولة الاستخدام وسرعة التفاعل بين العميل ومقدم الخدمة.

يعتمد النموذج الحالي على تطبيق موبايل موجه للمستخدمين، بالإضافة إلى منصة ويب إدارية تسمح بمتابعة البيانات وإدارة المستخدمين والخدمات.

1.6 وصف النموذج الأولي

يتكون النموذج الأولي الحالي من:

- تطبيق موبايل يسمح للعملاء بنشر فيديوهات لشرح الخدمات المطلوبة.
- نظام عروض يتيح لمقدمي الخدمات إرسال عروض تتضمن السعر والمدة الزمنية.
- نظام دردشة مباشر بين العميل ومقدم الخدمة بعد قبول العرض.
- نظام تتبع مباشر يسمح للطرفين بمتابعة موقع بعضهما عبر الخريطة.
- لوحة تحكم إدارية خاصة بإدارة المستخدمين والخدمات والشكاوى.
- نظام تقييم وشكاوى لتحسين جودة الخدمات وتعزيز الثقة داخل المنصة.

2.6 التقنيات والأدوات المستخدمة

- Flutter لتطوير تطبيق الموبايل.
- Core ASP.NET و C# لتطوير النظام الخلفي ومنصة الويب.
- Server SQL لإدارة قواعد البيانات.
- API REST لربط تطبيق الموبايل بالنظام الخلفي.
- Azure لتخزين الفيديوهات والخدمات السحابية.
- Maps Google لتوفير خاصية التتبع الجغرافي.

3.6 مراحل تطوير النموذج الأولي

1. تحليل فكرة المشروع ودراسة احتياجات المستخدمين.
2. تصميم واجهات التطبيق وتجربة المستخدم.
3. تصميم قاعدة البيانات وربط النظام الخلفي.
4. تطوير تطبيق الموبايل.
5. تطوير لوحة التحكم الإدارية.
6. اختبار الخصائص وتصحيح الأخطاء التقنية.
7. تجربة النموذج الأولي وتحسين الأداء.

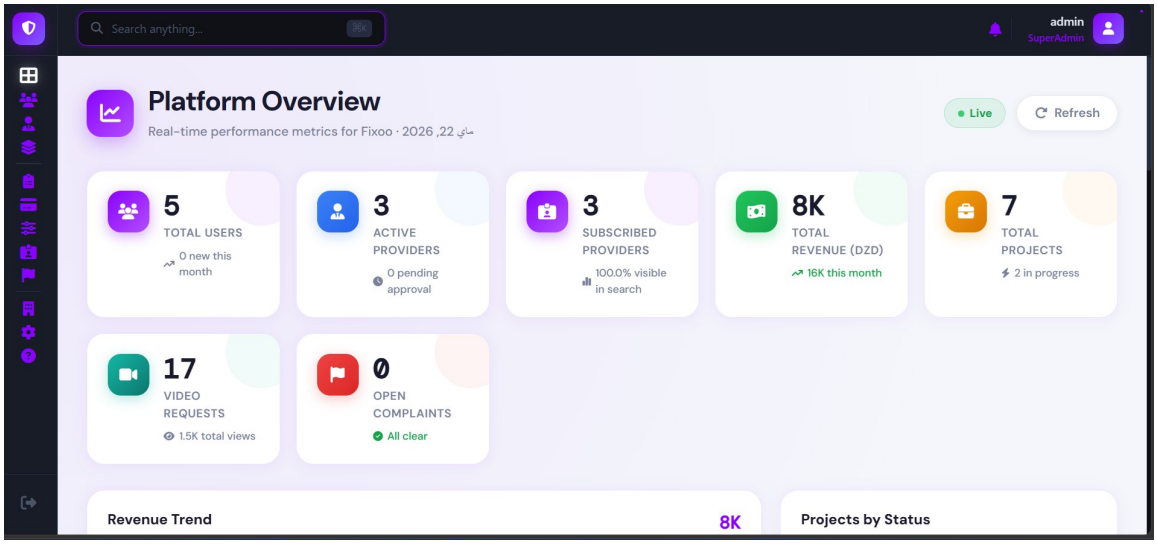
4.6 النتائج الأولية

سمح النموذج الأولي الحالي بالوصول إلى مجموعة من النتائج المهمة، من بينها:

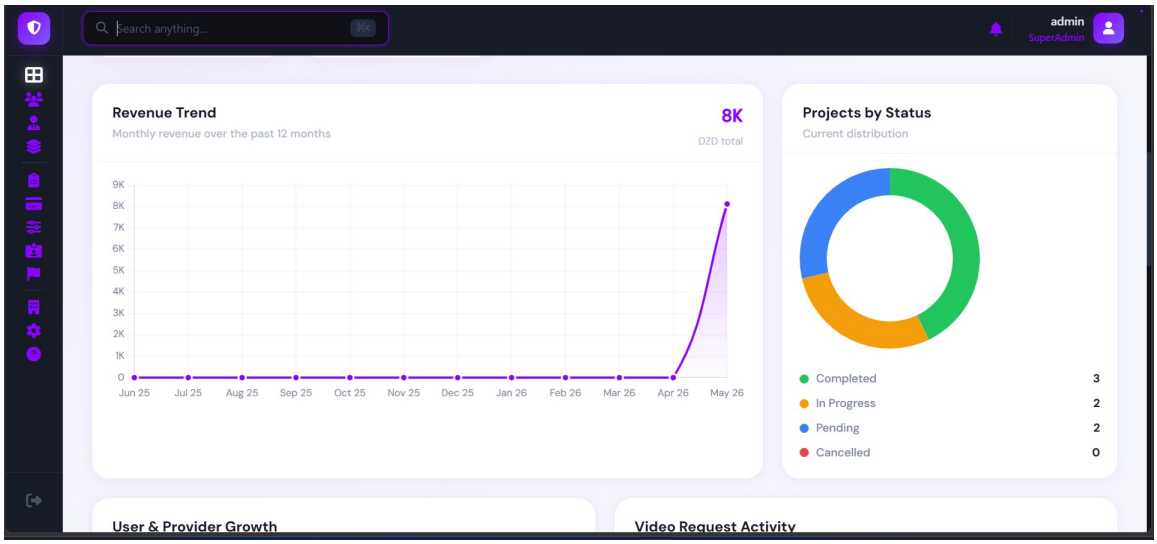
- نجاح عملية نشر الفيديوهات داخل التطبيق.
- نجاح إرسال واستقبال العروض بين المستخدمين.
- نجاح ربط التطبيق بالنظام الخلفي وقاعدة البيانات.
- تفعيل خاصية الدردشة بين العميل ومقدم الخدمة.
- نجاح تجربة التتبع الجغرافي المباشر عبر الخريطة.
- الوصول إلى نسخة أولية قابلة للتجربة والتطوير مستقبلًا.

5.6 صور منصة الويب الإدارية

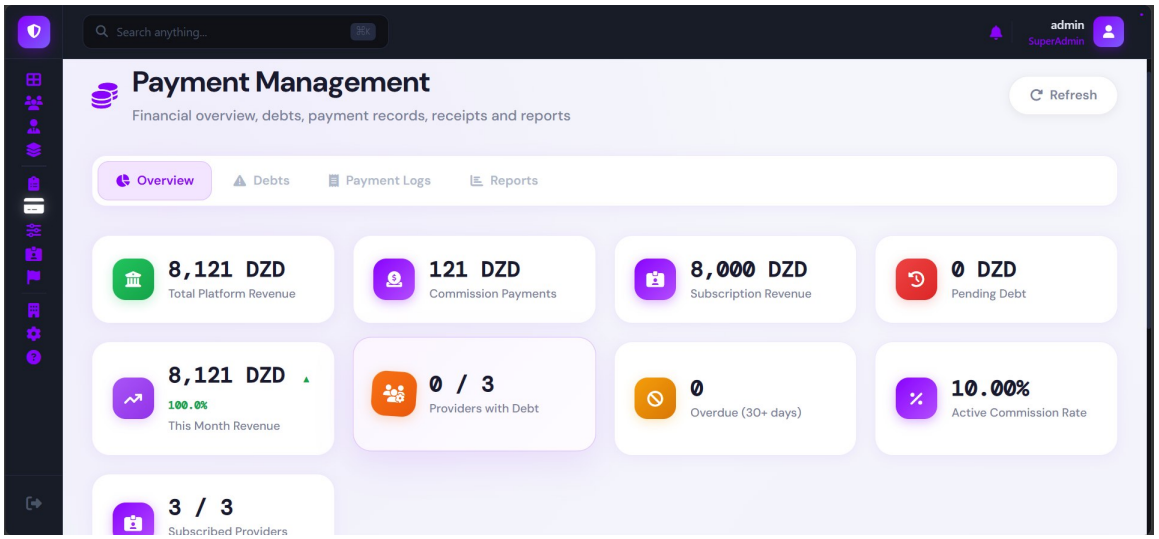
توضح الصور التالية بعض الواجهات الخاصة بمنصة الويب الإدارية المستعملة لإدارة النظام ومتابعة المستخدمين والخدمات.



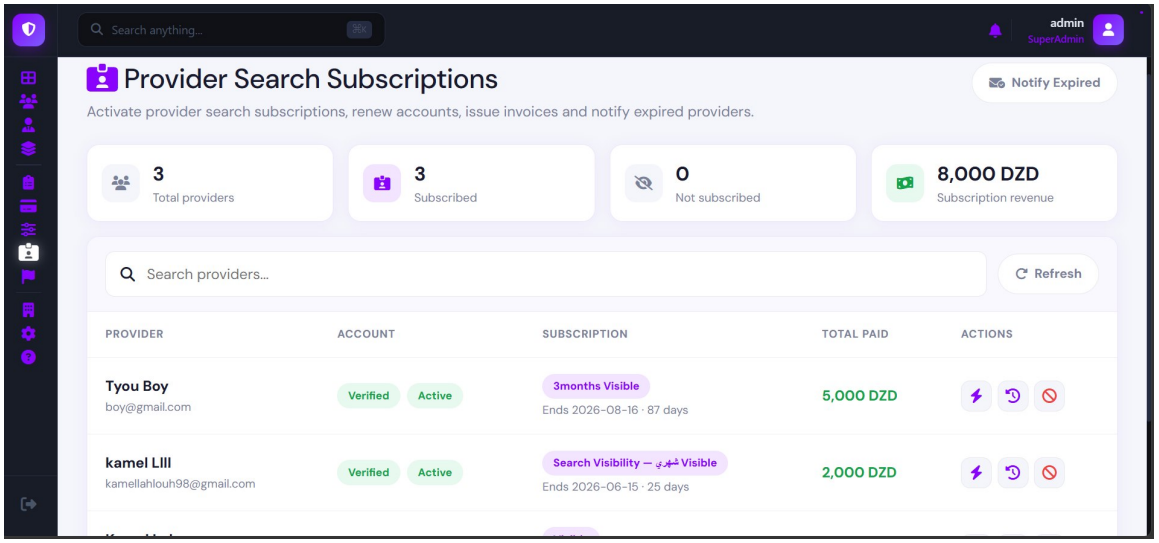
شكل 10: نظرة عامة على منصة الإدارة



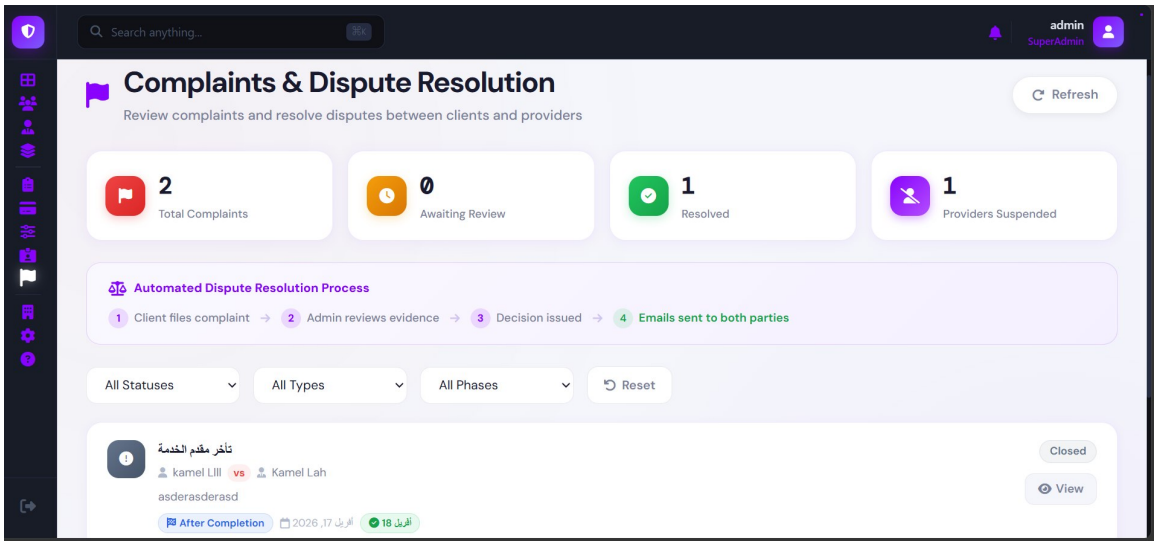
شكل 11: صفحة الإحصاءات والتحليلات



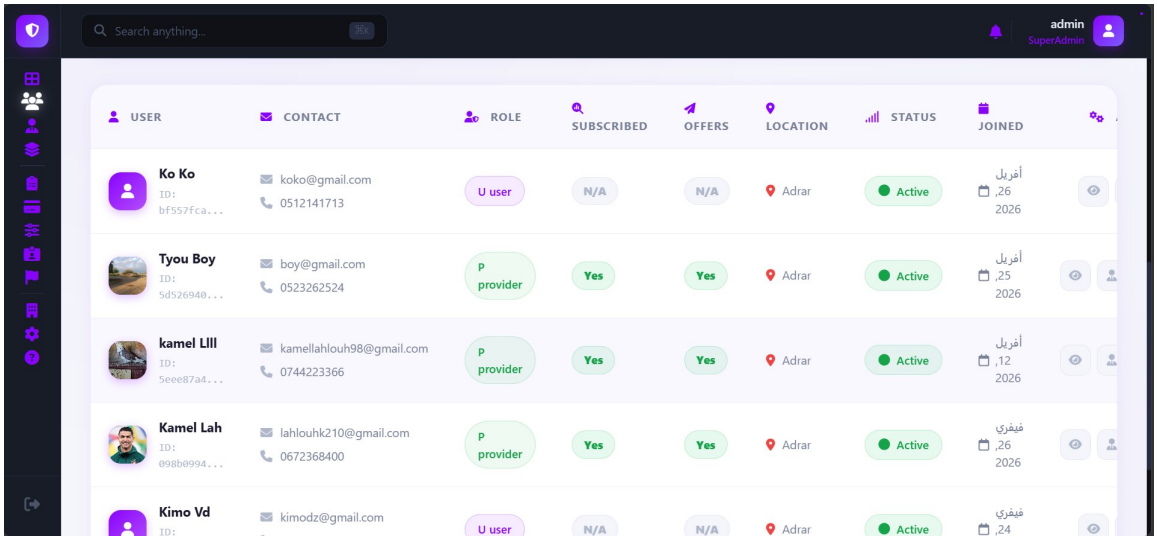
شكل 12: صفحة تسيير الأرباح والمعاملات



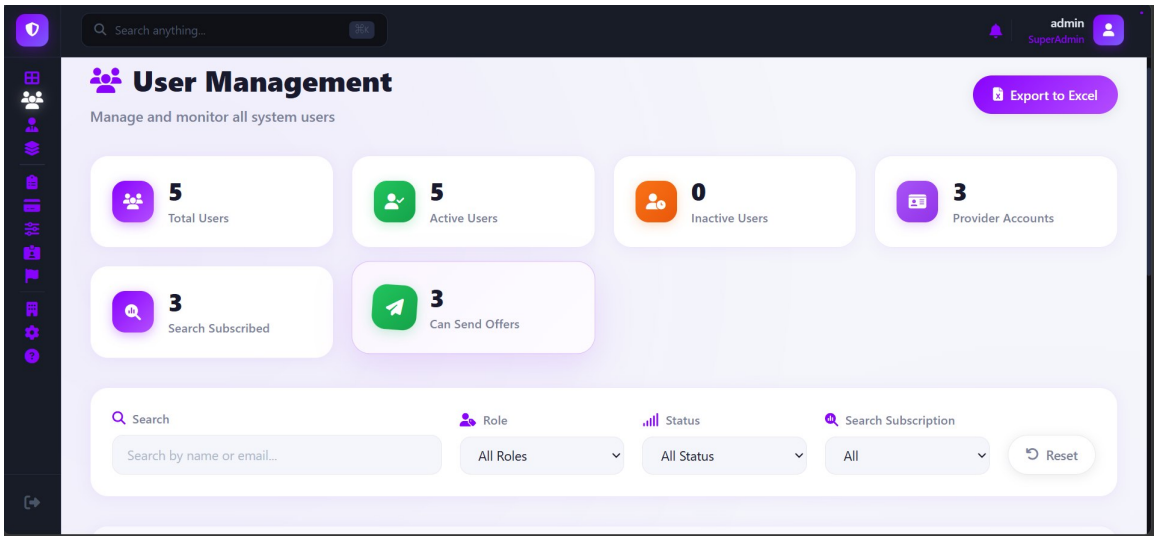
شكل 13: صفحة تسيير مزودي الخدمات



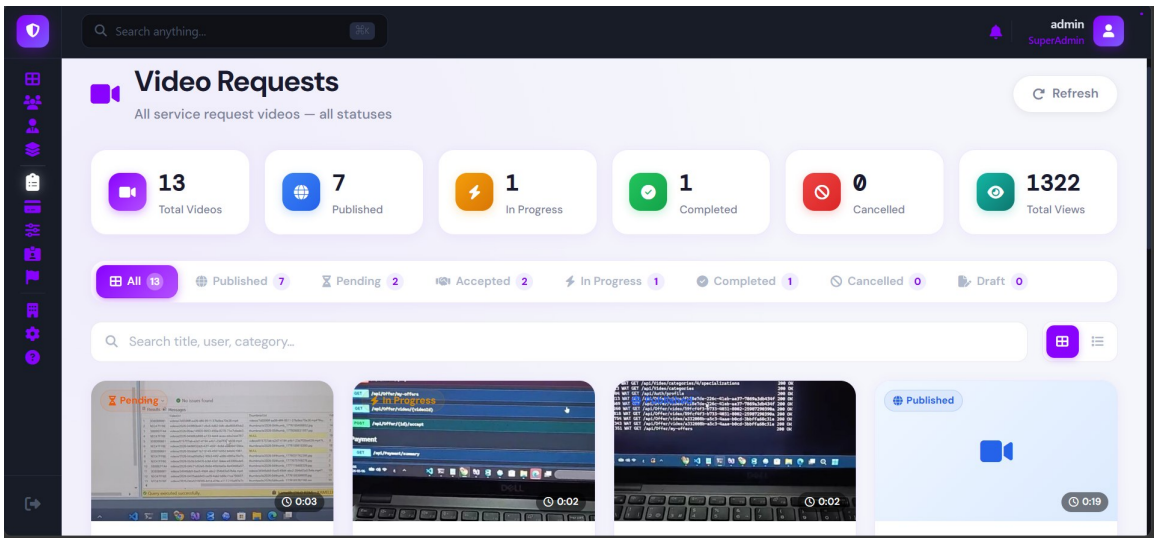
شكل 14: صفحة إدارة الشكاوى



شكل 15: صفحة تسيير المستخدمين



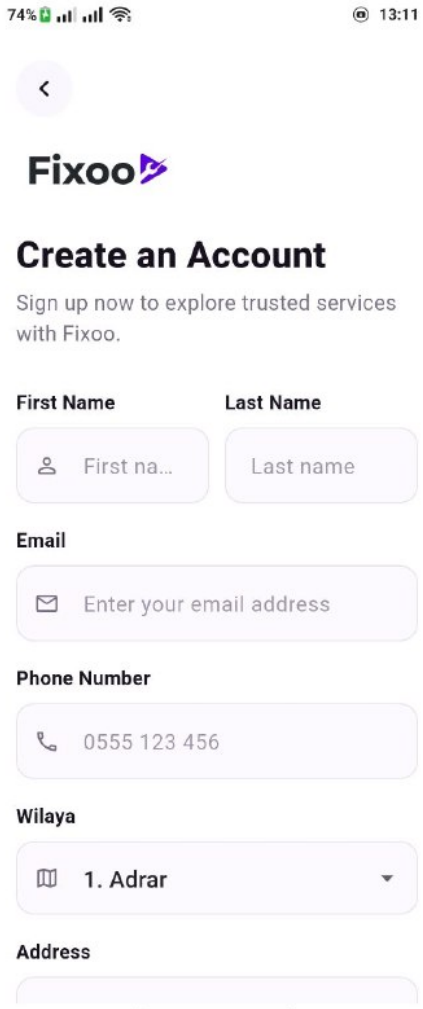
شكل 16: لوحة معلومات المستخدمين



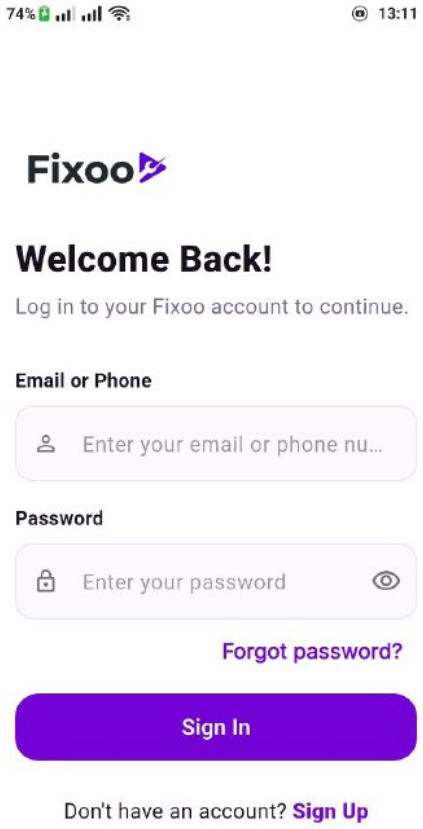
شكل 17: صفحة تسيير الفيديوهات والمنشورات

6.6 صور تطبيق الموبايل

1.6.6 واجهات التسجيل والدخول



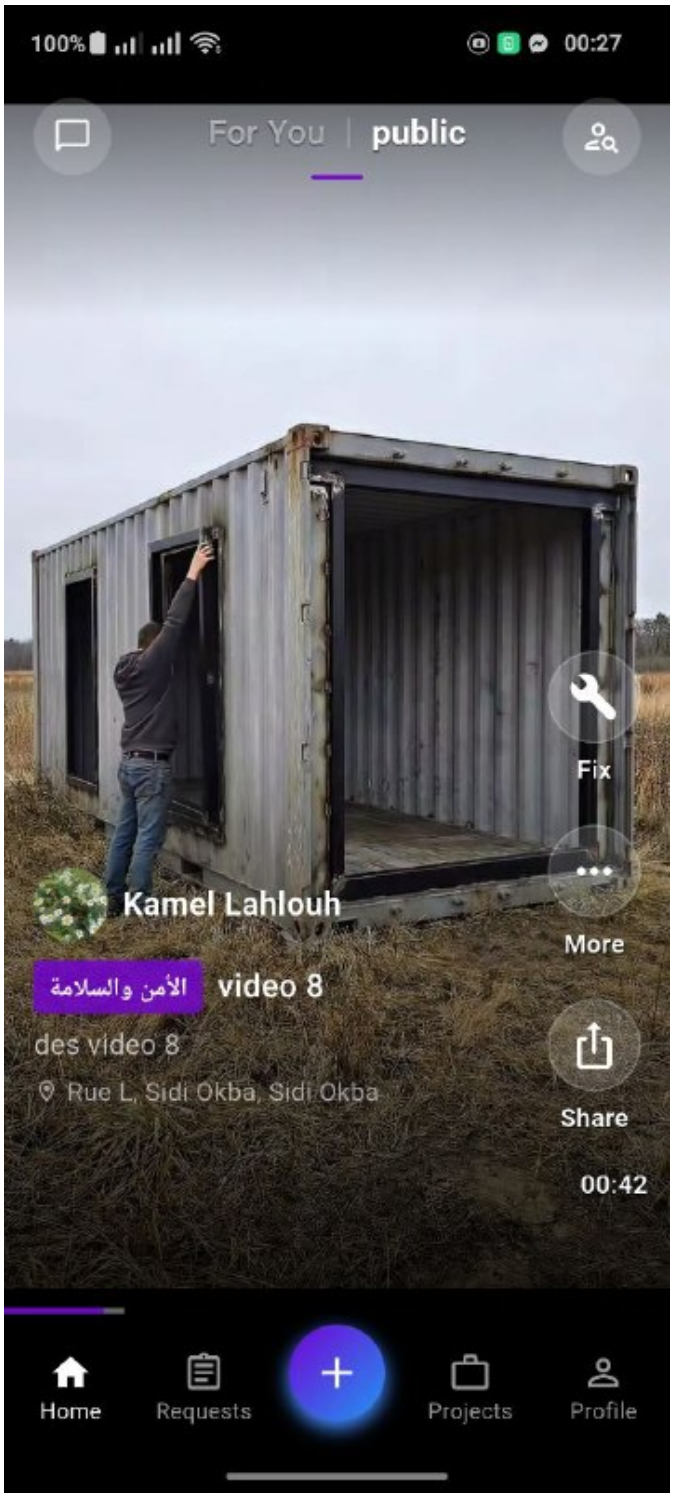
إنشاء حساب جديد



تسجيل الدخول

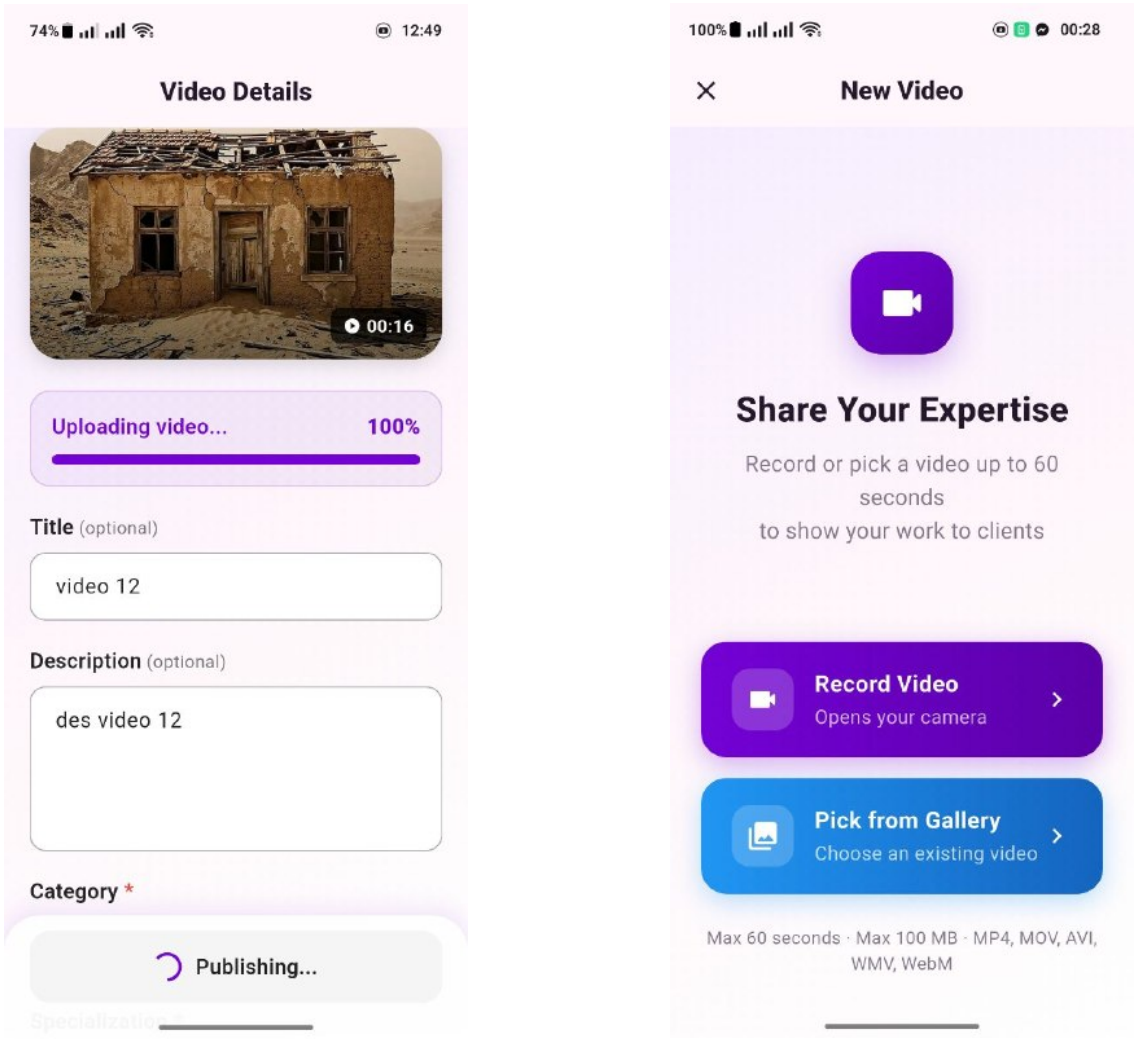
شكل 18: واجهات المصادقة والتسجيل

2.6.6 الواجهة الرئيسية



شكل 19: واجهة تصفح الفيديوهات والخدمات

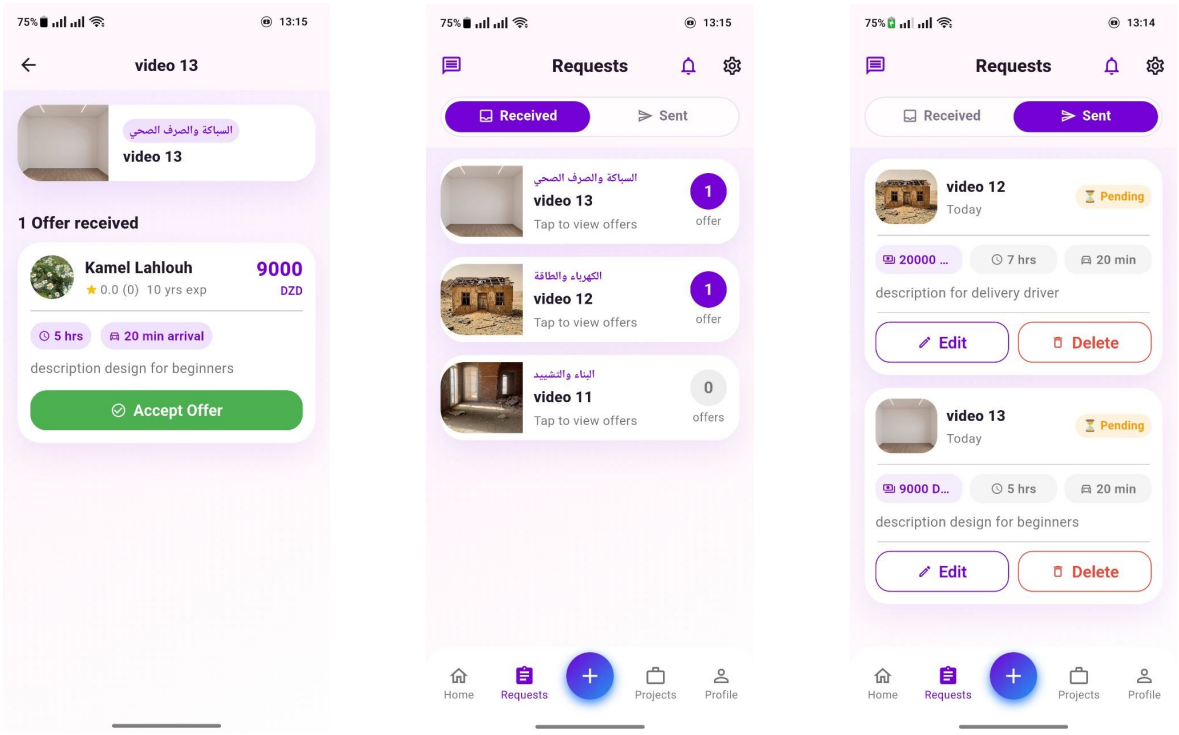
3.6.6 نشر طلب الخدمة بالفيديو



إضافة معلومات الخدمة تسجيل الفيديو

شكل 20: مراحل نشر طلب الخدمة

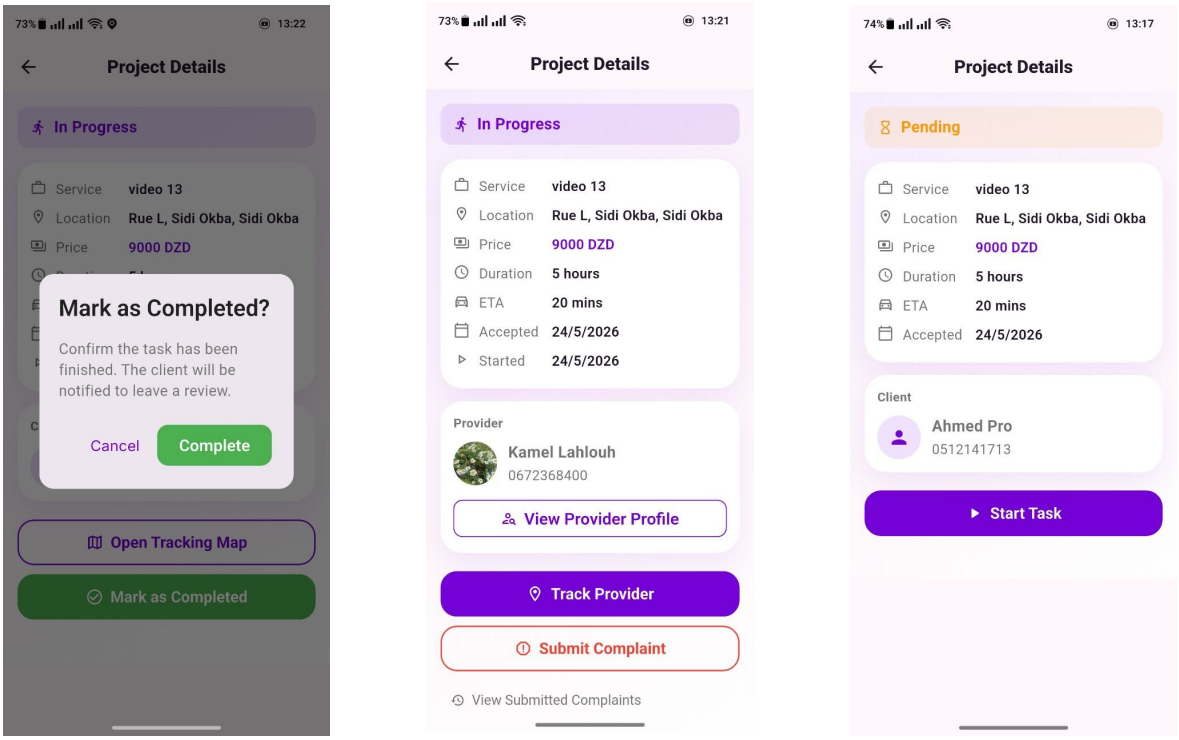
4.6.6 إدارة العروض



تفاصيل العروض استقبال العروض العروض المرسلة

شكل 21: واجهات إدارة العروض والخدمات

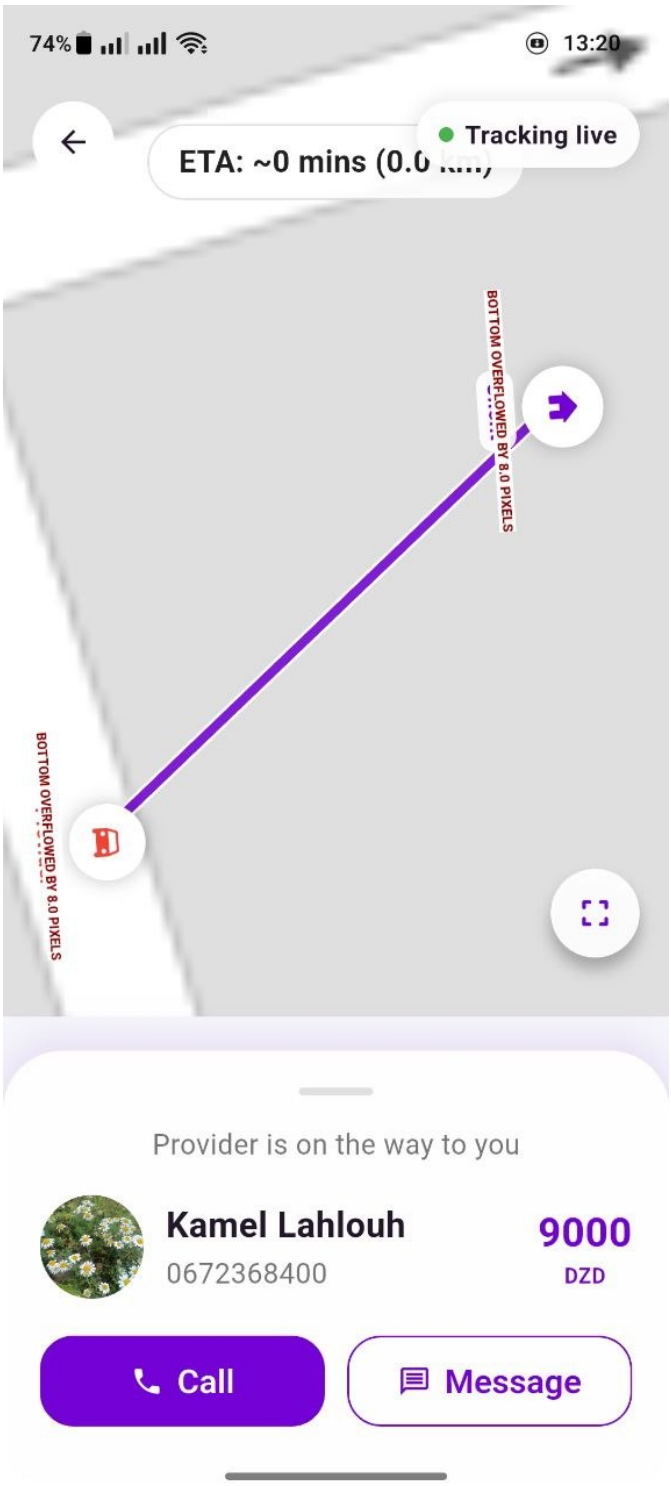
5.6.6 تنفيذ المشروع وتتبع الخدمة



بداية تنفيذ المهمة تفاصيل المشروع إنهاء المهمة

شكل 22: مراحل تنفيذ الخدمة داخل المنصة

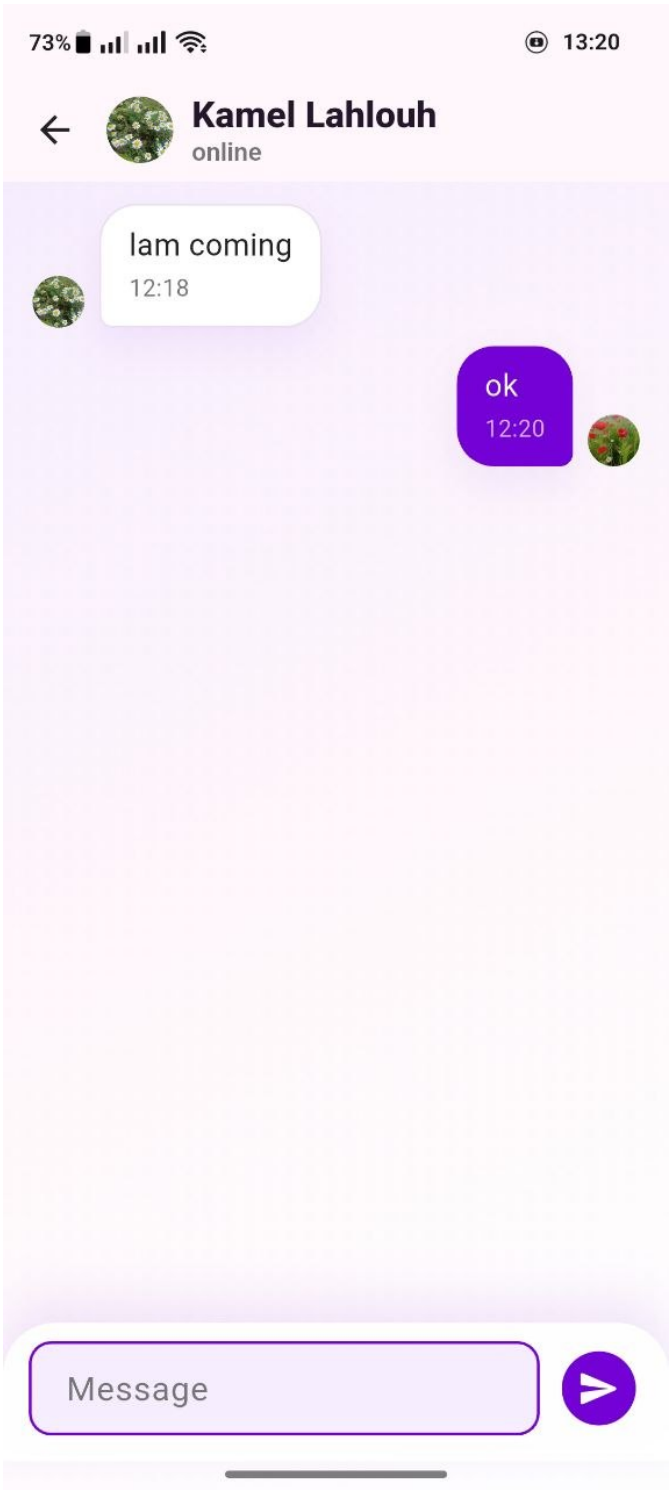
6.6.6 التتبع الجغرافي



تتبع مزود الخدمة عبر الخريطة

شكل 23: واجهة التتبع الجغرافي

7.6.6 نظام الدردشة

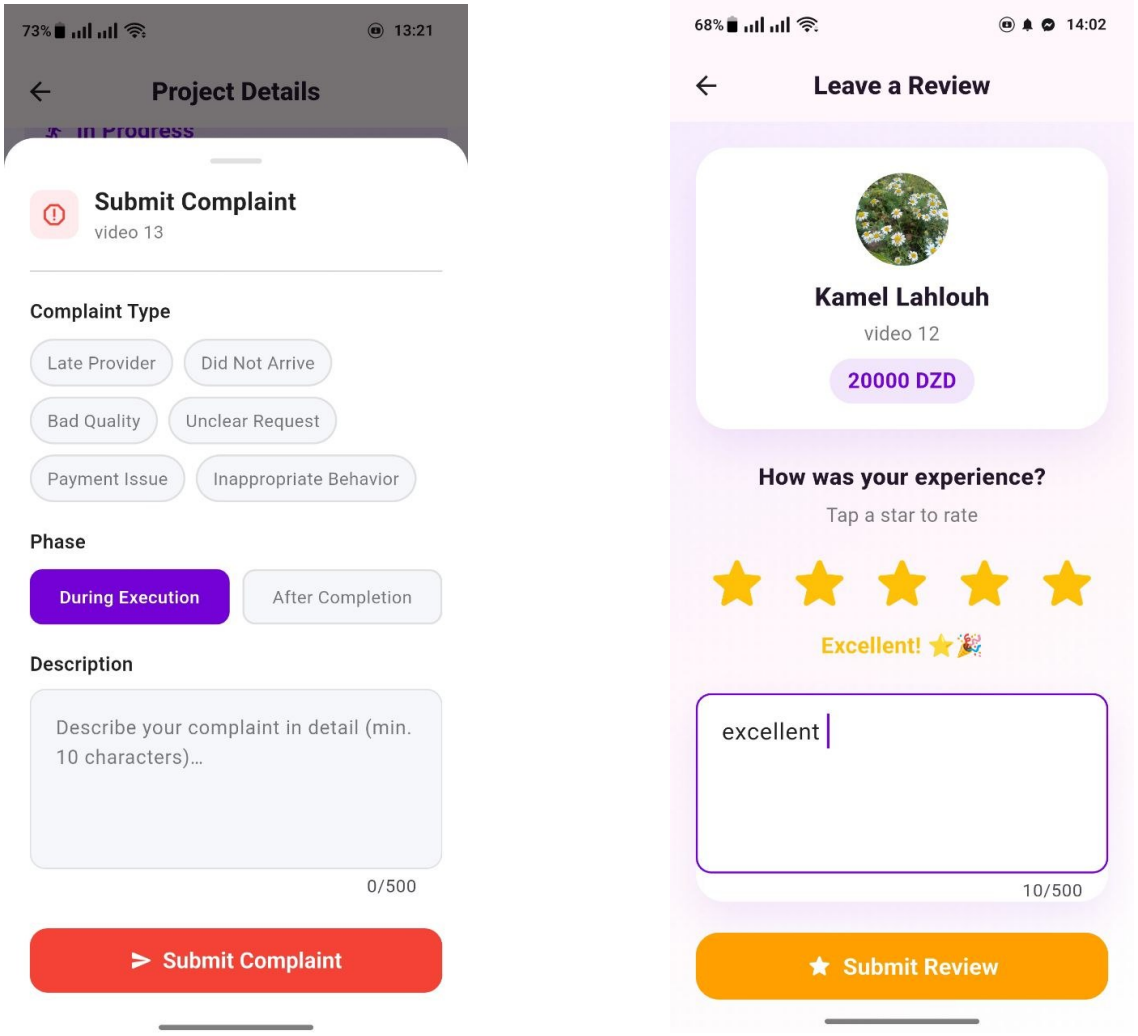


واجهة المحادثة بين العميل ومزود الخدمة

شكل 24: واجهة الدردشة

8.6.6 واجهات مزود الخدمة

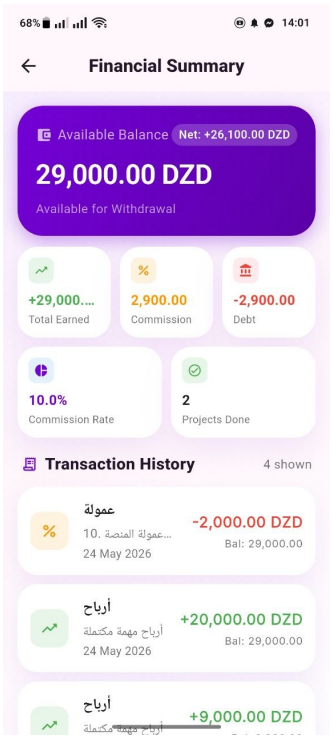
9.6.6 التقييمات والشكاوى



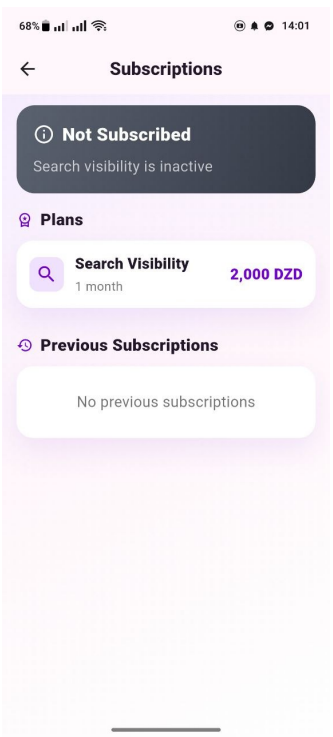
إرسال شكوى

تقييم مزود الخدمة

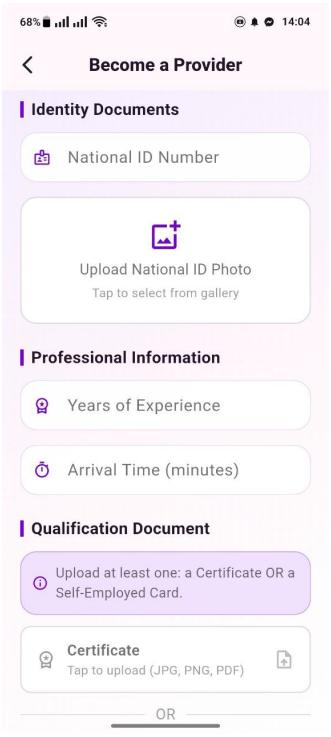
شكل 26: واجهات التقييمات والشكاوى



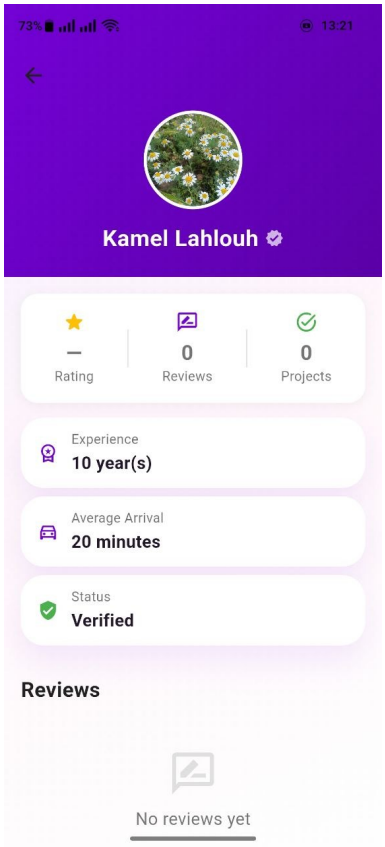
الأرباح والديون



اشتراك مزود الخدمة



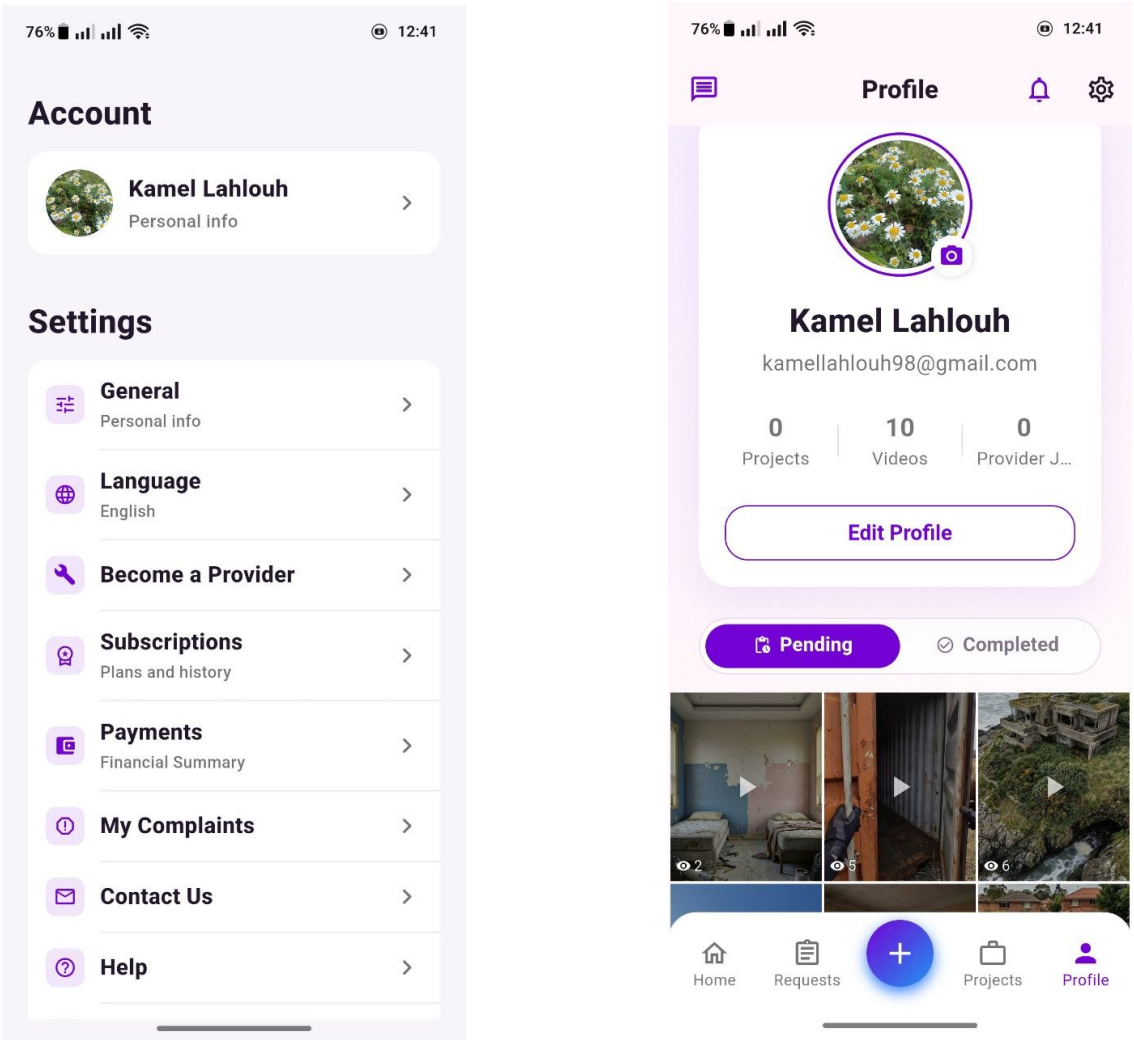
رفع الوثائق



تفاصيل مزود الخدمة

شكل 25: الواجهات الخاصة بمزودي الخدمات

10.6.6 الملف الشخصي والإعدادات

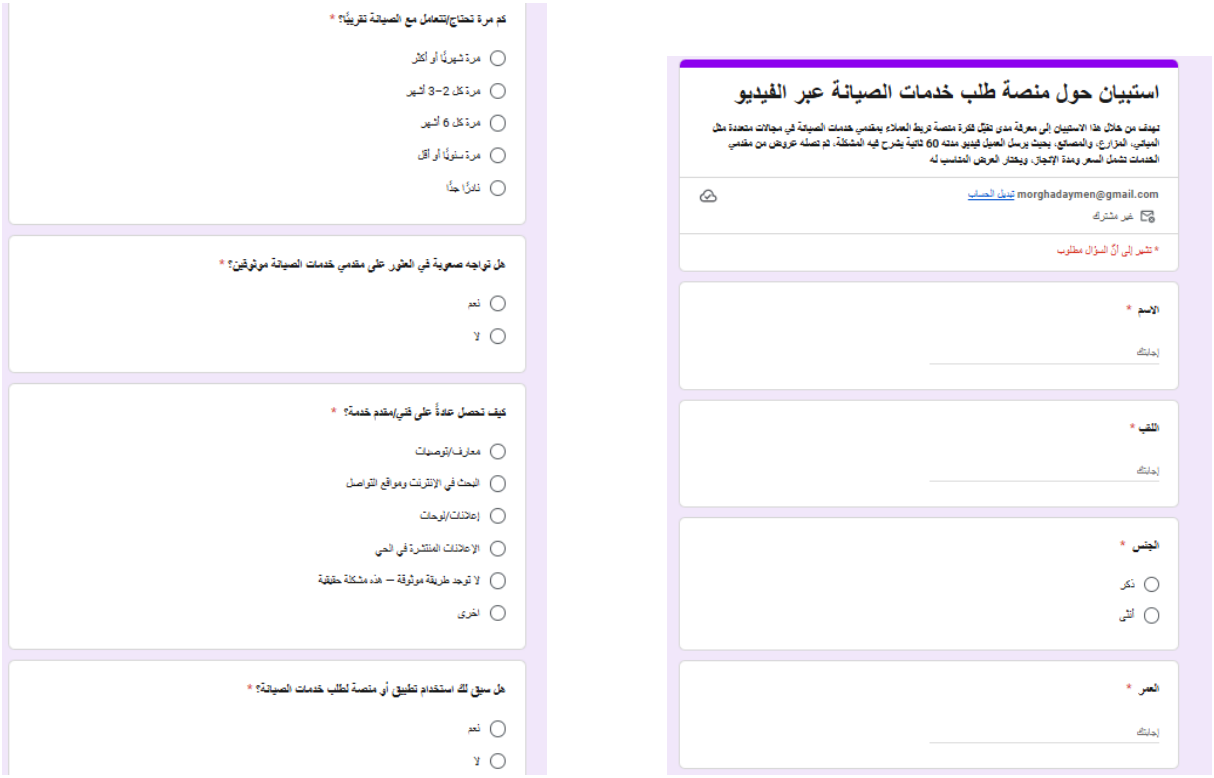


الإعدادات الملف الشخصي

شكل 27: واجهات الملف الشخصي والإعدادات

عرض الاستبيان

تم إجراء استبيان ميداني على عينة مكونة من **44 فردًا** من مختلف فئات المجتمع بهدف دراسة واقع الخدمات المنزلية والحرفية، وتحليل احتياجات المستخدمين ومدى تقبلهم لفكرة منصة Fixoo المعتمدة على الفيديو في طلب الخدمات.



شكل 28: الجزء الأول من نتائج الاستبيان


القسم الثاني — تقييم فكرة

إنشاء تطبيق يسمح للعميل بتسجيل فيديو قصير (60 ثانية) لمشكلة الصيانة، نشره، وتلقي عروض من مقدمي الخدمات (سعر + وقت الإنجاز).

هل تعجبك فكرة إرسال فيديو قصير يوضح المشكلة؟ *
نعم
لا

هل ستستخدم التطبيق فعلياً لتسجيل مشكلتك عبر فيديو؟ *
نعم
لا

ما هو القلق الأكبر لديك بشأن نشر فيديو لمشكلتك؟ *
الخصوصية – لا أريد كاميرا داخل ممتلكاتي
لا أعرف كيف أصور المشكلة بشكل صحيح
لا توجد لديّ مخاوف
سرعة الإنترنت قد تكون عائقًا

هل ترى أن هذه الطريقة ستوفر عليك الوقت؟ *
نعم
لا


القسم الثالث — لمقدمي الخدمات تحديداً

مقدم الخدمة يتصفح طلبات خدمات صيانة فيديوهات ويقدم عروض

هل توافق على رفع وثائقك (شهادة، رقم تعريف وطني) للحصول على توثيق الحساب؟ *
نعم
لا

كمقدم خدمة: هل سيساعدك الفيديو على تحديد تكلفة العمل بدقة أكبر قبل الذهاب للموقع؟ *
نعم
لا

هل ستستخدم التطبيق لتصفح الطلبات وتقديم عروض؟ *
نعم
لا

كمقدم خدمة: ما هي الميزة الأكثر جذبًا لك في هذا التطبيق؟ *
الوصول لعملاء جدد بدون تسويق شخصي
معرفة تفاصيل المشكلة مسبقًا عبر الفيديو
بناء سمعة وتقييمات إلكترونية
إدارة عروضي وطلباتي من مكان واحد


القسم الرابع — الثقة والأمان والخدمات المستقبلية

من ناحية الامان والخصوصية واراء واضافات المستقبلية

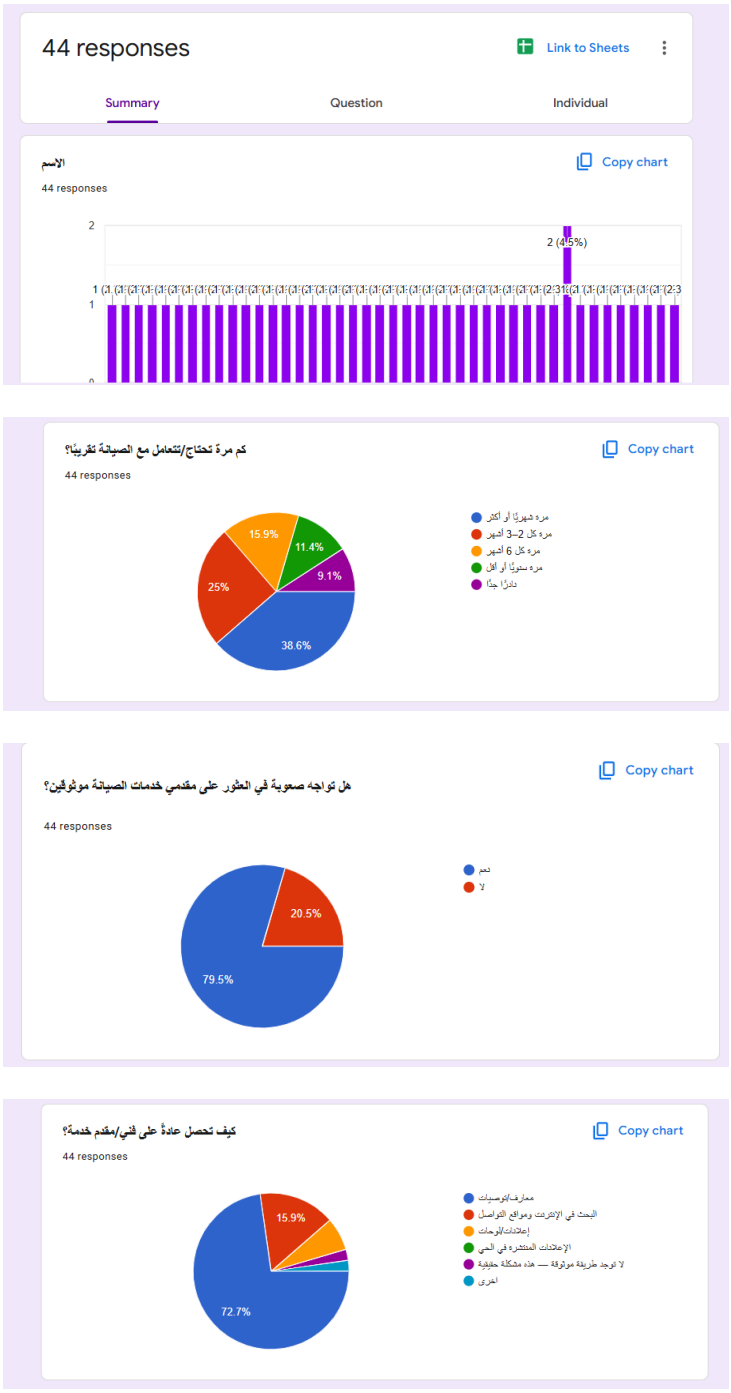
ما هو العائق الأكبر الذي قد يمنعك من استخدام هذا التطبيق؟
عدم الثقة في مقدمي الخدمات على التطبيق
القلق من الاحتيال والدفع دون تنفيذ
صعوبة استخدام التطبيق لكبار السن
ضعف الإنترنت في منطقتي
لا توجد عوائق — سأستخدمه

هل تخشى من عدم الالتزام بالموعد أو السعر بعد الاتفاق؟ *
نعم
لا

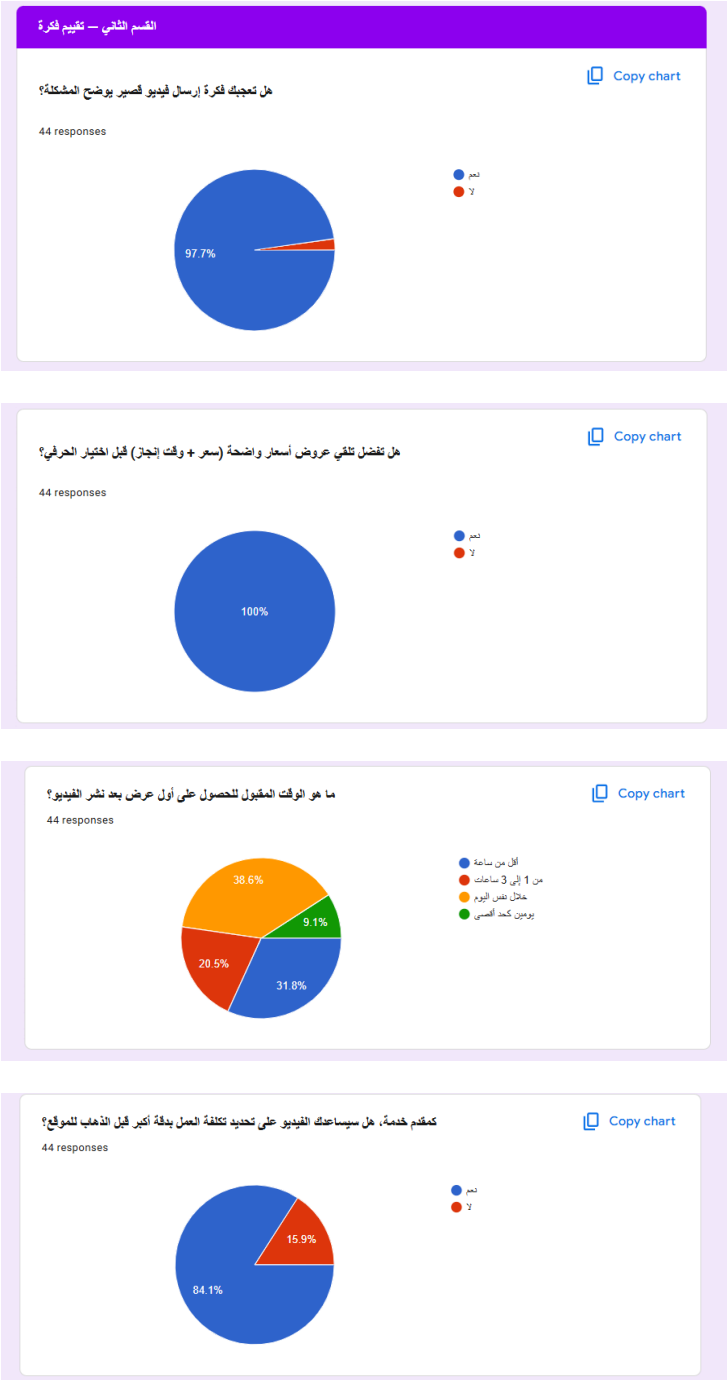
هل تثق في مقدمي الخدمات الذين يتم التحقق من هويتهم داخل التطبيق؟ *
نعم
لا

ما هي الميزة الأكثر أهمية بالنسبة لك في هذا التطبيق؟ *
رفع فيديو المشكلة

شكل 29: الجزء الثاني من نتائج الاستبيان



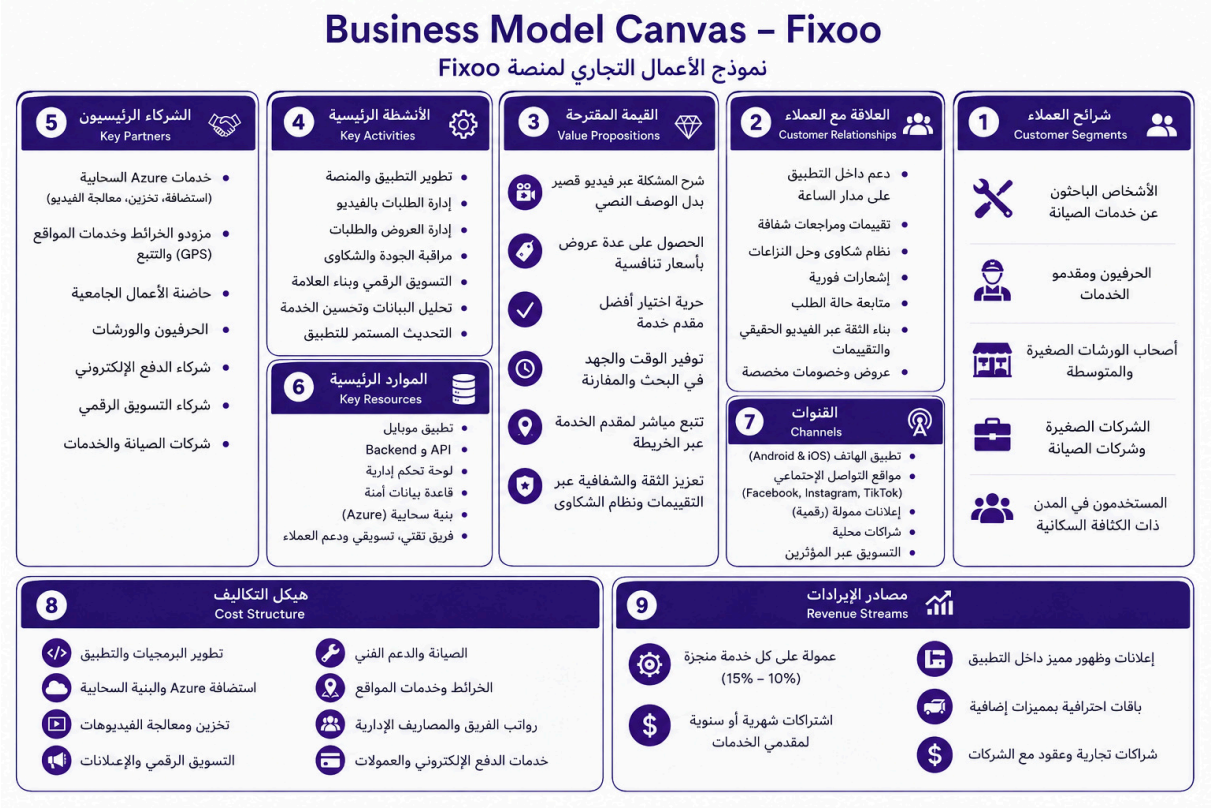
شكل 30: الجزء الأول من نتائج وإحصائيات الاستبيان



شكل 31: الجزء الثاني من نتائج وإحصائيات الاستبيان

الملحق رقم 03: نموذج العمل التجاري BMC

يوضح هذا الملحق نموذج العمل التجاري الخاص بمشروع Fixoo والذي يبين العناصر الأساسية للمشروع من حيث الشركاء، الأنشطة، القيمة المقترحة، شرائح العملاء، القنوات، الموارد، الإيرادات والتكاليف.



شكل 32: نموذج العمل التجاري الخاص بمنصة Fixoo

خاتمة

في ختام هذه الدراسة، يمكن القول إن مشروع **Fixoo** يمثل محاولة لتقديم نموذج رقمي حديث يواكب التحول المتسارع نحو الخدمات الذكية والمنصات التفاعلية المعتمدة على المحتوى المرئي. وقد جاء هذا المشروع استجابة لمجموعة من المشاكل التي يعاني منها سوق الخدمات التقليدي، خاصة ما يتعلق بصعوبة التواصل بين العميل ومقدم الخدمة، وعدم وضوح الطلبات، وضعف الثقة بين الطرفين.

يعتمد المشروع على فكرة مبتكرة تتمثل في استخدام الفيديو القصير كوسيلة أساسية لشرح الخدمة المطلوبة، مع توفير نظام عروض تنافسي يسمح للعميل باختيار العرض الأنسب من حيث السعر والمدة والكفاءة. كما يدعم النظام مجموعة من الخصائص الحديثة مثل الدردشة، التتبع الجغرافي المباشر، التقييمات، وإدارة الشكاوى، مما يساهم في تحسين تجربة المستخدم وتعزيز جودة الخدمات.

ومن خلال العمل المنجز، تم الوصول إلى نموذج أولي قابل للتجربة يضم تطبيق موبايل ومنصة ويب إدارية، مع توظيف مجموعة من التقنيات الحديثة مثل Flutter و Core ASP.NET و Server SQL والخدمات السحابية، وهو ما يعكس قابلية المشروع للتطوير والتحويل مستقبلًا إلى مؤسسة ناشئة حقيقية قادرة على المنافسة داخل السوق الجزائرية.

كما يفتح المشروع آفاقًا مستقبلية مهمة، من بينها دمج أنظمة الدفع الإلكتروني، وتوظيف تقنيات الذكاء الاصطناعي في تحليل المحتوى والتحقق من الحسابات، إضافة إلى إمكانية التوسع نحو أسواق وخدمات جديدة.

وفي الأخير، فإن مشروع Fixoo لا يهدف فقط إلى إنشاء منصة رقمية، بل يسعى إلى المساهمة في تطوير ثقافة الخدمات الرقمية في الجزائر، وتقديم تجربة أكثر سرعة وشفافية وفعالية، بما يتماشى مع متطلبات الاقتصاد الرقمي الحديث.