

Designing a Unified Marketplace for On-Demand Local Service Discovery and Booking

Dhanaya Indap¹, Sneha Murkar¹

¹Department of Computer Science, Patkar-Varde College, Mumbai, Maharashtra, India

Abstract—

Booking a local service today requires multiple apps, as each has its unique booking procedure, making it challenging to manage. This paper proposes a digital marketplace to discover, book, pay, and communicate with local service providers in one place. The application allows providers to register and showcase their services on the platform. This study's findings played a significant role in developing the application's features.

The application features were determined through survey responses, with 85.7% of participants indicating that managing multiple applications for services is stressful, and 84% desired a unified digital marketplace for booking services. These features included verified providers, invoice and payment tracking, and reviews. The application was developed using Flutter as a cross-platform framework, Firebase backend, and Razorpay for payment processing. The future scope includes personalized recommendations, location tracking, and multilingual support to make the platform helpful in discovering and managing local services.

Keywords: Unified Service Marketplace, On-Demand Services, Service Discovery, Dual-Role Mobile Platform, Flutter, Firebase, Razorpay Payment Gateway

I. Introduction

Finding someone trustworthy for household or local work used to mean asking neighbors, flipping through local directories, or going on a friend's referral. Apps have made the search easier [1], but going digital doesn't remove the trust problem—it

can sharpen it, since customers now let strangers into their homes with little to go on beyond a profile.

Trust is what determines whether people actually stick with these platforms. A study of 357 Indian users found that trust and perceived value were the strongest predictors of whether someone kept using a home-service app or recommended it onward [2]. The OECD has flagged the same pattern across peer-to-peer platforms generally: with little else to rely on, customers lean hard on other users' ratings and reviews [3]. Provider verification, transparent information, and a working feedback loop aren't nice-to-haves here — they're what these platforms are actually built on.

Fragmentation is the other recurring issue — needing a different app or contact method for every service [4]. This research's own survey confirms it. Of the 64.6% of respondents who'd booked a home service online, 46% named juggling multiple apps as a real frustration, and another 46% said finding a provider they could trust was the harder problem. Zooming out, 85.7% found switching between apps at least mildly annoying, and 84% said they'd rather have one platform do it all, with only 4% disagreeing either way.

The survey also pointed to what people want from that platform: simple booking and verified providers led at 70% each, then secure payments (68%), location tracking and support (66% each), and reviews (60%). The biggest gripes were hidden charges (52%), fake reviews (36%), late arrivals (26%), and poor support (26%). These numbers shaped the platform's feature set and priorities directly.

The proposed system responds to this by putting customers and providers on one

shared platform — browsing, comparing, booking, and paying in one place for customers, while providers register, list, and manage their services through the same system. Encryption and tokenization secure the payment layer [6][7].

A few requested features — live tracking (60%), AI-based recommendations (58%), emergency booking

II. Related Work and Literature Review

The way people find local services has been changing for a while. What used to run entirely on word of mouth, phone calls, and knowing someone nearby is increasingly happening online instead. A study of UrbanClap's aggregation model looked at how one platform can bring together many service categories under one roof — and flagged provider reliability and customer trust as recurring sticking points [1].

Trust in particular seems to decide whether people actually stay on these platforms. Research on consumer behavior toward online home-service apps found that trust and perceived value were strong predictors of whether someone kept using the app or recommended it to others [2]. A similar pattern shows up across peer-to-peer marketplaces generally — ratings and reviews do real work closing the information gap when customers are booking someone they've never met [3].

The tools for building these platforms have also matured. Flutter lets developers write one codebase that runs across platforms, cutting down the work of maintaining separate native apps [5]. Firebase handles the less visible but essential pieces — authentication, storage, notifications — without developers building that infrastructure from scratch [4]. Payment gateways like Razorpay round it out, letting platforms process transactions on established, secure infrastructure [6], [7].

Together, the literature makes a solid case for aggregation, trust, feedback systems, and secure payments as core to these marketplaces. What's less explored is combining customer- and provider-facing functionality into a single app — most platforms still split these into separate systems. That's the gap this paper addresses. Rather than relying on literature alone, this study also grounds its design in primary survey data collected specifically to understand what users want, giving the system a more user-centered basis for bringing discovery, booking, provider management, and payments together in one place.

(52%), in-app chat (44%), and multilingual support (38%) — are earmarked for later rather than this version. Combining the existing research with the survey data, this study aims for a single platform that cuts fragmentation and makes local service discovery genuinely easier for both sides.

III. Problem Statement and Research Gap

A. Problem Statement

Online service platforms have made it easier to find on-demand services, but the actual process of finding and booking these services is still broken. Customers often need to search on different apps, websites, and messaging tools just to compare providers. This means jumping between systems, checking prices separately, and handling reservations in different ways. All of this takes time and energy.

Trust is another issue. When a customer does not know a provider well, they are unsure if the person or business is reliable. The way reviews, pricing, verification, and customer support work can vary from one platform to another. This inconsistency makes it hard to choose the service.

On the side small businesses and individuals also face problems. They want tools to list their services, set their availability, accept bookings, and communicate with customers. Most existing platforms do not give them all these features in one place.

The survey for this research supports these challenges. Eighty-five point seven percent of respondents said switching between apps was at least somewhat inconvenient. Eighty-four percent said they would prefer one app that lets them access many types of services. This shows a need for a single centralized platform.

So the main problem this research aims to solve is the lack of a platform. One that brings together various local services and allows both customers and providers to use the same application.

B. Research Gap

The literature review and survey results show gaps in current solutions:

Service fragmentation: People must use many different apps or websites to find different services. There is no place where all options are available.

Customer - Provider Separation: Customer actions and provider tasks happen on different systems. This increases confusion and makes things harder to manage.

Limited role flexibility: It is difficult for people who want to both use services and offer them through the platform. Most apps force users into one role.

Transparency: Customers need information about providers. Including verification details, ratings, and real reviews. Especially when dealing with someone unfamiliar.

Integrated service management: A single system is needed to help with service discovery, managing providers, making bookings, sending notifications, and handling payments.

User-driven requirements: There is room to design service platforms based directly on what users say they want. Features should be built using feedback.

C. Proposed Research Approach

To address the gaps surfaced by the literature review and primary survey, this study proposes a single digital marketplace that brings customers and local service providers onto one platform. Customers can search for services, check provider details, and place bookings; providers, in turn, can register, list and manage their services, update availability, and respond to booking requests.

The platform's core features — service discovery, provider verification, booking management, notifications, and digital payments — were not chosen independently but drawn directly from the pain points and expectations identified through the review and survey. In that sense, the design follows the user's stated needs rather than a predetermined feature list.

The underlying goal is to cut down the fragmentation that currently defines local service discovery by giving both sides of the market one shared platform instead of scattered channels. Bringing search, provider information, bookings, notifications, and payments together under one system is meant to make the experience more convenient, transparent, and trustworthy for customers, while giving providers a proper digital channel to run their services.

IV. Research Methodology

This research follows a user-oriented methodology combining survey-based requirement identification with the design and development of a multi-service mobile application. The process was structured across seven stages, shown in the diagram below.

The study began by identifying core difficulties in existing local-service platforms — fragmented service categories, the need to switch between apps, and unreliable providers. A literature review of hyperlocal platforms, consumer trust, and digital payments then helped establish the research gap. A user survey was conducted to gather direct evidence on booking experiences, pain points, and desired features, analyzed using percentage-based descriptive methods.

Survey and literature findings were translated into functional requirements across three user roles: customers, service providers, and administrators. These requirements shaped the system's architecture — a multi-role model where any user can act as both customer and provider within one application.

The application was built using Flutter for cross-platform development, Firebase for backend services, and Razorpay for payments, developed incrementally with continuous integration testing. Finally, the system was evaluated through functional testing against the original survey-derived requirements to confirm it addressed real user needs.

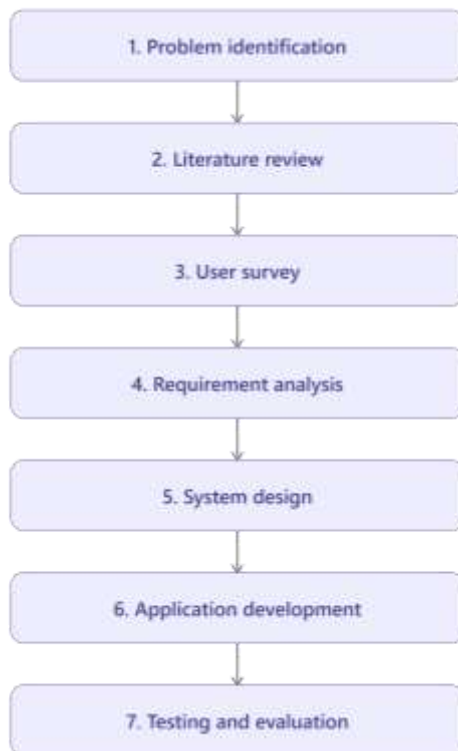


Fig. 1 . Seven-Stage Research and Development Process

V. User Survey and Requirement Analysis

To ground App design in real user needs, a survey examined how people currently book local and on-demand services, the difficulties they face, and the features they value most in a unified service platform. The findings below were used to derive the application's functional requirements.

A. Survey Design

The questionnaire covered six areas: prior experience with online service booking, problems encountered during booking, the inconvenience of using multiple apps, preference for a unified platform, desired features, and future enhancements. Responses were analyzed using percentage-based descriptive statistics; for multi-select questions, the reported figures reflect the share of respondents choosing each option and do not sum to 100%.

B. Prior Experience

Respondents' booking frequency varied widely, with most falling toward the less frequent end of the spectrum (Fig. 2). Only 10% booked home services weekly and 14% did so monthly, while the majority booked either occasionally (38%) or rarely (38%). This suggests that while online service booking is familiar to most respondents, it is not yet a routine habit for most—pointing to an opportunity for a more convenient, unified platform to encourage more frequent use.

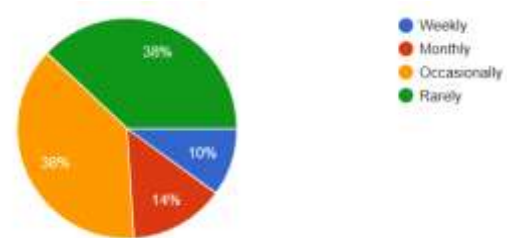


Fig. 2. Respondents' experience with online home-service booking.

C. Problems with Existing Platforms

Fig. 3 shows cost as the top complaint, with juggling multiple apps and finding trustworthy providers close behind. Fake reviews, late arrivals, and poor support came up often too, while live tracking and payment issues barely registered. Cost tops the list, but fragmentation and trust are nearly as big a problem — together they point to a need for one reliable ecosystem, not just lower prices.

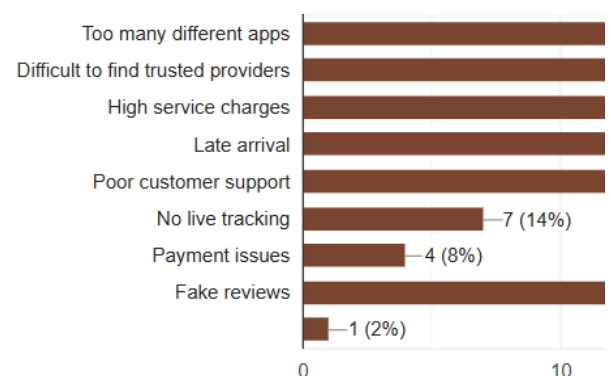


Fig. 3. Respondents' experience with existing platforms

D. App-Switching Fatigue

85.7% of respondents found switching between apps for different services at least somewhat inconvenient, with 44.9% saying "Yes" outright and a further 40.8% saying "Maybe" (Fig. 4).

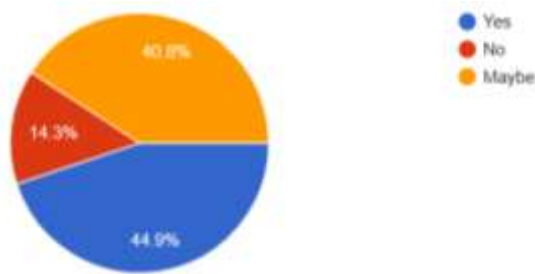


Fig. 4. Respondents' opinions on application switching

E. Demand for a Unified Platform

84% of respondents said they would prefer one app covering multiple services, against only 4% who said no (Fig. 5) — strong validation for the app's core premise of consolidating services under one platform.

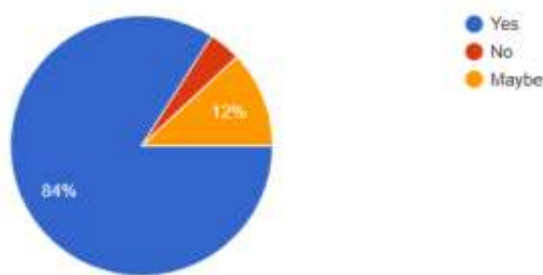


Fig. 5. Respondents' demand for unified platform

F. Feature Priorities

Fig. 6 shows what respondents want most from a booking app: easy booking and verified providers top the list, with secure payments, live tracking, and customer support close behind. Reviews, fair pricing, and quick booking also mattered to most people, while push notifications ranked lowest. Ease of use and provider trust lead the pack, with payment security and support right there too — showing users care as much about reliability and transparency as they do about convenience.

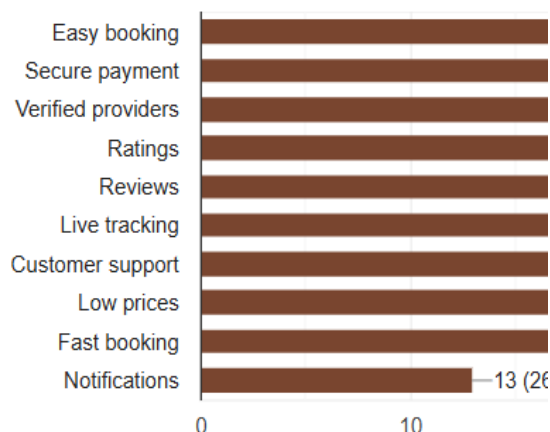


Fig. 6. Respondents' opinion about features

G. Future Feature Interest

Live tracking and AI-based recommendations drew the most interest for future updates, with emergency booking and in-app chat close behind — multilingual support, subscriptions, and voice booking mattered less. Tracking and personalization clearly top the wish list, pointing to transparency and smarter recommendations as the natural next steps.

Overall, the survey findings provided a user-centered basis for prioritizing the core requirements of the proposed platform. Service integration, convenient booking, provider verification, and secure payment processing form the primary functional requirements, while real-time tracking, intelligent recommendations, and emergency services represent potential areas for future development.

VI. Proposed Unified Platform Framework

6.1 Overview

Drawing on both the literature gaps and the survey findings, this research proposes a single mobile platform for discovering, booking, and managing local on-demand services. Customer and provider functionality live in the same app — one account can book services and, if eligible, offer them too, with no separate app required. Supported categories include cleaning, plumbing, laundry, salon services, water supply, and education-related services.

6.2 System Architecture

The platform is built around the mobile app, backend services, a database, cloud storage, notifications, payments, and admin controls. Firebase handles authentication, data storage, and notifications, while Razorpay manages payments. Underneath this sit eight core components — shared user/provider handling, authentication, service management, bookings, notifications, payments, admin oversight, and cloud storage — all coordinated through the backend to keep profiles, listings, bookings, and notifications in sync (Fig. 7).

6.3 Roles: Customer, Provider, and Administrator

Three roles exist: customer, provider, and administrator. Customers can browse categories, view providers, make bookings, track status, pay

online, and leave feedback. Providers can register, submit verification details, list services, set availability, and respond to bookings.

The dual-role design is central here — a single user can be a customer, a provider, or both, without needing separate accounts or apps. Administrators oversee the platform: reviewing registrations, managing content, monitoring activity, and handling verification.

6.4 Unified Model and Workflows

One account, one app, both sides of the marketplace. On the customer side: log in, browse services, pick a provider, submit a booking, get confirmation, receive the service, and pay — with status updates along the way (Fig. 7).

On the provider side: register, submit business and verification details, wait for admin review and approval, then start listing services and accepting bookings. That admin-verification step is meant to build trust and keep the marketplace transparent.

6.5 Modules and Research Contribution

The system is built from modules covering authentication, profiles, service categories, provider registration/verification, service management, bookings, notifications, location features, payments, and admin control — together covering the full service-booking lifecycle.

The main contribution here is treating customer and provider roles as two sides of one account rather than two separate systems. Paired with centralized discovery, booking, verification, notifications, and payments, this directly tackles the fragmentation the survey identified, while giving providers one integrated channel to manage their presence on the platform.

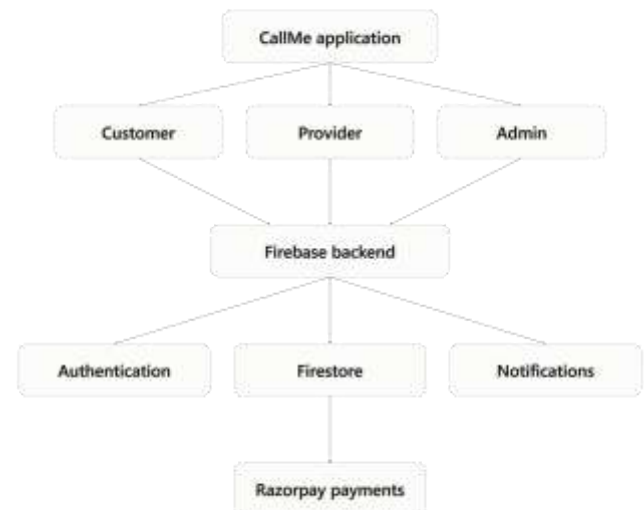


Fig. 7. Multi-Role Architecture of Unified Application

VII. System Design and Implementation

7.1 Technology Stack

The application is built on Flutter for a single cross-platform codebase, with Firebase handling authentication, data storage, and cloud functions on the backend. Razorpay is integrated for in-app payments — together, this stack runs the customer, provider, and admin sides of the platform from one codebase.

7.2 Architecture and Database

The app follows a client-cloud model: the Flutter front end talks to Firebase's cloud services rather than a self-hosted server. Firestore stores users, providers, bookings, and service data, with records linked so a booking stays tied to its customer and, once assigned, its provider. Since every role reads from the same backend, a status change on one side reflects instantly on the other (Fig. 7).

7.3 Authentication and Role Management

Access is role-based, covering customers, providers, and admins under Firebase Authentication. Rather than separate accounts, one login can hold multiple roles — someone can browse and book as a customer while also operating as a registered provider — with the interface adjusting to show only what that role needs.

7.4 Provider Registration and Verification

An existing user can apply to become a provider directly from their account, with no separate signup. The form is dynamic — steps and required fields

(tools owned, bank details, etc.) change depending on the service category chosen, rather than one generic form for everyone. Submissions stay pending until an admin reviews and approves them; only verified providers show up in customer search results (Fig. 8).

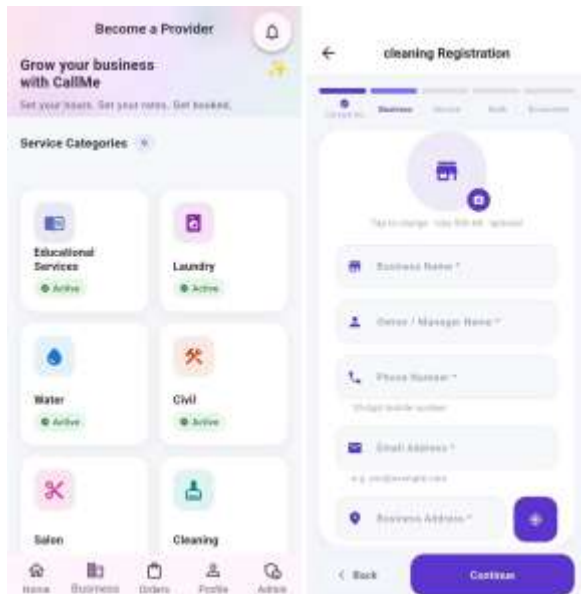


Fig. 8. Business Page

7.5 Service and Booking Management

Customers browse services by category, check provider details, and place booking requests. Requests go out to every matching, approved provider in that category rather than locking to one — whoever accepts first takes the job, so the system doesn't stall if a single provider is unavailable. Job completion is confirmed through an OTP the customer shares on-site, keeping that final step verifiable for both sides.

7.6 Digital Payment Integration

Payments run through Razorpay at checkout, with UPI, card, and cash-on-completion as options. Razorpay's own authentication layer handles transaction security, so the app itself never has to manage sensitive payment data directly.

7.7 Admin Oversight

Admins verify incoming provider applications and can decline or roll back a booking when needed. A live analytics dashboard gives them a running view of revenue, booking volume, and customer/provider growth, along with a payment-method breakdown — all pulled straight from Firestore rather than compiled after the fact (Fig. 9).



Fig. 9. Admin Dashboard Page

7.8 Provider Experience

Once approved, a provider gets a dashboard showing incoming job requests for their category in real time. They can accept or pass on each request, track jobs through to OTP-confirmed completion, and see their own booking history — inside the same app customers use, just under a different role.

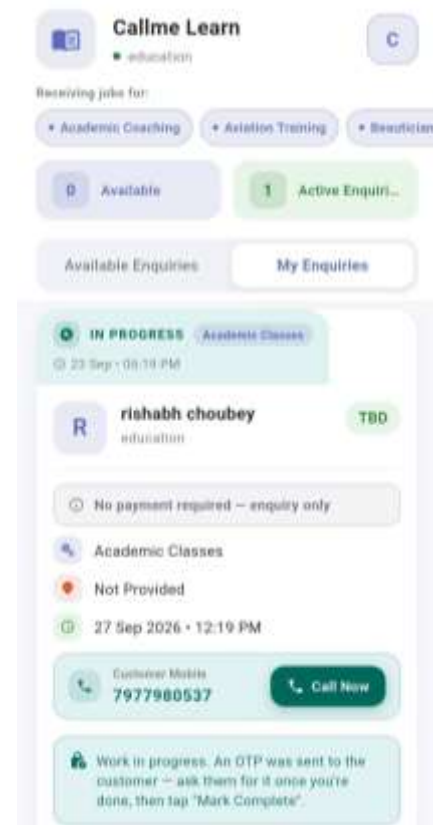


Fig. 10. Provider Dashboard Page

VIII. Evaluation and Results

This evaluation reflects where the platform stands right now, not a finished product — it currently covers select service categories and the core

customer-provider flow, with several features still ahead.

8.1 Functional Evaluation

So far, the app handles authentication, provider registration, admin verification, browsing, bookings, notifications, and payments for its current categories. Location features are partial — full real-time tracking isn't live yet — but the core loop of a customer booking a provider through to payment works end-to-end. The rest of the planned feature set hasn't been built or tested.

8.2 Survey-to-Feature Evaluation

The survey directly steered priorities. Strong demand for one app (84%) and frustration with juggling platforms (85.7%) drove the decision to consolidate everything into a single application. Easy booking and provider verification (70% each) shaped those two workflows, and demand for secure payments (68%) led straight to the payment integration. Real-time tracking came up too (60%) but was left for a future release rather than forced in now.

8.3 Current Implementation Status

As it stands, the app supports a limited category set, dual customer/provider roles, verification, booking, notifications, and payments — enough to prove the unified-marketplace concept works in practice. Tracking, AI recommendations, emergency booking, multilingual support, and subscriptions are still pending. So treat these results as an early-prototype snapshot, not a full evaluation of the finished vision.

8.4 Usability and User Acceptance

No formal usability testing has been run on the app itself yet — the survey measured what users wanted, not how well the build performs. That testing is a separate next step: real users working through the app, measured on ease of use, task completion, satisfaction, and overall usefulness, ideally with a structured usability framework.

8.5 Performance Evaluation

Performance testing hasn't happened either. Once the build is further along, it'll need testing under

realistic load — response times, booking reliability, notification delivery, database and payment performance, and scalability. Testing with multiple concurrent users would also help catch backend bottlenecks before real growth in providers and bookings arrives.

Module	Status	Evaluation
User authentication	Implemented	Tested
Service browsing	Partial (selected services)	Accessible
Provider registration	Implemented	Submittable
Provider verification	Implemented	Admin approval available
Booking management	Partial (selected services)	Tested
Notifications	Implemented	Tested
Location services	Partial	Basic functionality
Digital payment	Integration stage	Testing
Live tracking	Not implemented	Planned
Subscriptions	Not implemented	Planned
AI recommendations	Not implemented	Planned
Multilingual support	Not implemented	Planned

Fig. 11. Performance Evaluation Chart

IX. Discussion

The survey results were central to shaping what the platform needed to do. With 84% of respondents favoring a single unified app and 85.7% describing platform-switching as a hassle, the case for a centralized, multi-service design was clear.

Beyond the overall architecture, the survey also shaped individual features. Booking simplicity and provider verification each came up as priorities for 70% of respondents, which is why the app includes a structured booking flow alongside an admin-driven verification process for providers. Payment security mattered to 68% of respondents, which led directly to integrating a digital payment gateway. Notifications and location-based features were built in or planned based on similar feedback from the survey.

These choices line up with earlier research that has stressed the value of trustworthy providers, trust-building mechanisms, and transparent information-sharing in digital marketplaces [1]–[3]. What this study adds, though, is combining customer- and provider-facing functionality into one app and

grounding the feature set directly in primary survey data rather than assumptions.

A few limitations should be noted. The platform is still being built out, and only part of the planned service categories and features exist in the current version. Features such as real-time tracking, AI-based recommendations, emergency bookings, multilingual support, and subscriptions haven't been implemented or tested yet. Formal evaluation of usability, performance, scalability, and user acceptance also hasn't happened at this stage, so it's too early to draw firm conclusions about how well the complete system works. That kind of evaluation — with a bigger and more varied group of users — will be the next step once the platform is further along.

X. Limitations and Future Scope

A. Limitations

A few limitations are worth acknowledging. To start, the platform is still a work in progress — only a subset of planned service categories and features has actually been built so far, so the current version isn't representative of the full system envisioned.

The survey used in this research also has a specific scope: it captured user preferences and frustrations around existing service-booking habits, but it wasn't set up as a usability test of the app itself. That means the survey findings can point to what users want, but they can't be used on their own to judge how well the implemented system actually performs.

Along the same lines, formal testing of performance, scalability, reliability, and user acceptance hasn't been carried out yet. A number of features that survey respondents flagged as desirable — real-time tracking, AI recommendations, emergency bookings, multilingual support, subscriptions, and voice-based booking — are also not yet part of the working build.

Lastly, a platform like this only works as well as the providers on it. Its real-world performance will naturally depend on how many providers sign up, how reliable and well-verified they are, and how consistent service quality and coverage turn out to be across different locations.

B. Future Scope

Going forward, the priority is to broaden the range of service categories and refine the booking and provider-management flows already in place. Real-time provider tracking is one of the first planned additions, giving customers better visibility into where a provider is and when they'll arrive.

AI-driven recommendations are another target — using signals like past bookings, stated needs, and location to help users find the right service or provider faster. Beyond that, the roadmap includes emergency booking, multilingual support, built-in messaging, subscription tiers, and voice-assisted booking.

Longer-term, there's room for deeper analytics, more personalized discovery, automated provider matching, sharper location features, and better tools for administrators to monitor the platform. These can be rolled out gradually and refined as the system matures.

Once the platform reaches a more stable state, the next step is a full evaluation with a larger, more varied group of users — looking at usability, task completion, satisfaction, response times, reliability, and scalability. That kind of testing will give real evidence of how well the unified marketplace model holds up in practice.

XI. Conclusion

This work set out to tackle a common problem: local service discovery and booking are scattered across too many separate apps. The solution proposed here is a single mobile platform that brings multiple service categories together, serving both customers looking to book services and providers looking to offer them — without needing separate apps for either side.

Survey data guided the core design decisions. A large majority of respondents (84%) said they'd prefer one unified app over several separate ones, and even more (85.7%) called switching between platforms inconvenient. These results justified building a centralized, multi-service system rather than a single-purpose one. Beyond that, the survey also pointed to specific needs — straightforward booking, verified providers, secure payments, a review system, responsive customer support, and location-based features — all of which shaped the requirements.

On the technical side, the app was built in Flutter for cross-platform support, with Firebase handling authentication, data storage, cloud functions, and notifications, and Razorpay integrated for payments. What's been built so far covers a core set of service categories, dual customer/provider functionality, provider verification, booking management, notifications, and payment processing.

It's worth being clear that this is a working first version, not a finished product. Features like real-time tracking, AI-driven recommendations, emergency bookings, multilingual support, and subscription plans are still on the roadmap. The system also still needs thorough usability, performance, scalability, and user-acceptance testing before its real-world effectiveness can be confirmed.

In short, this project shows what a requirements-driven approach to building a unified service marketplace can look like — combining discovery, provider onboarding, bookings, verification, and payments in one place lays useful groundwork for cutting down the fragmentation users currently deal with.

References

- [1] N. Jyani and H. Bansal, "UrbanClap: India's Largest Home Service Provider," *South Asian Journal of Business and Management Cases*, vol. 10, no. 2, 2021, doi: 10.1177/09728201211018978.
- [2] "Examining Consumers' Behavioral Intentions Towards Online Home Services Applications," *Journal of Global Information Management*, vol. 31, no. 1, 2023, doi: 10.4018/JGIM.326547.
- [3] Organisation for Economic Co-operation and Development (OECD), "Trust in Peer Platform Markets: Consumer Survey Findings," *OECD Digital Economy Papers*, no. 263, 2017, doi: 10.1787/1a893b58-en.
- [4] Google, "Firebase for Flutter," *Firebase Documentation*. [Online]. Available: [Firebase Documentation](#).
- [5] Google, "Flutter Documentation," *Flutter Documentation*. [Online]. Available: [Flutter Documentation](#).
- [6] Razorpay, "Payment Gateway Documentation," *Razorpay Documentation*. [Online]. Available: [Razorpay Documentation](#).
- [7] Razorpay, "Razorpay Security," *Razorpay Documentation*. [Online]. Available: [Razorpay Documentation](#).