Design and Development of a Mobile-First Agricultural Equipment Rental Application with Offline Capability for Rural Farmers

[1]Tejas D. Fegade, [2]Sahil R. Shaikh, [3]Akash C. Kharat, [4]Prof. Khatal Kavita

[1234]*Dept. of Computer Engineering Sahyadri Valley College of Engineering and Technology Junnar, India*

---***---

Abstract - *Rural farming communities in India continue to face significant challenges in accessing digital agricultural services due to intermittent or absent internet connectivity. While web-based platforms have demonstrated effectiveness in streamlining equipment rental operations, their reliance on continuous network access renders them impractical for a large segment of rural farmers. This paper proposes the design of a mobile-first agricultural equipment rental application built on an offline-first architectural framework, enabling full Operational capability under zero-connectivity conditions. The proposed system employs Flutter as the cross-platform development framework and integrates a local SQLite database for offline data persistence, ensuring uninterrupted access to equipment listings, booking records, and user profiles without requiring active network connectivity. A background synchronization mechanism is designed to reconcile locally stored data with the remote server upon connectivity restoration, maintaining consistency across distributed data instances. The application further incorporates multilingual Interfaces for Hindi and Marathi languages to address language accessibility barriers prevalent among non-English-speaking farming communities. Firebase Cloud Messaging is adopted for push notification delivery, and role-based access control differentiates functionalities for farmers, equipment owners, and administrators. This paper presents the system Architecture, module design, database schema synchronization protocol, and a comprehensive evaluation framework proposed for validating system performance, usability, and offline reliability upon implementation.*

Key Words: Mobile Application, Offline-First Architecture, Flutter, SQLite, Agricultural Equipment Rental, Rural Connectivity, Smart Agriculture, Multilingual Support, Data Synchronization, Digital Farming

1. INTRODUCTION

Agriculture constitutes the primary source of livelihood for ap-proximately 58% of India's rural population, with small holder and marginal farmers representing the most economically vulnerable segment of the agricultural workforce [2]. Despite remarkable advancements in mobile technology and digital agricultural services over the past decade, practical adoption of these tools among rural farming communities remains disproportionately low. A fundamental barrier to this adoption is the unreliable and often absent internet infrastructure in remote agrarian regions, particularly in semi-arid and hilly ones of Maharashtra, Madhya Pradesh, and Rajasthan where farming activities are concentrated and network coverage is limited [3].Agricultural equipment rental represents a financially viable and increasingly popular strategy for farmers who cannot sustain the capital expenditure associated with purchasing and maintaining advanced machinery such as tractors, harvesters, and irrigation systems. A preceding work by the authors introduced a web-based Farming Rental Equipment Services platform that demonstrated effective digitization of the rental workflow for farmers with consistent internet access [1]. However, that platform's fundamental dependency on continuous network connectivity was identified as a critical limitation, effectively excluding the very demographic rural, low- connectivity farmers that stands to benefit most from digital rental services. This paper addresses that limitation by proposing a mobile-first application that adopts an offline-first architectural philosophy. Rather than treating network unavailability as an error condition, the proposed system treats offline operation as the default state and network connectivity as an enhancement. This inversion of the conventional online-first assumption ensures that core application functionalities remain fully accessible regardless of network status; thereby extending the practical reach of digital agricultural services to previously underserved rural communities.

The primary contributions of this paper are:

- A comprehensive offline-first mobile architecture design Using Flutter and SQLite specifically tailored for agricultural equipment rental services in rural India.
- A conflict-aware background data synchronization protocol for reconciling local and remote records upon network Restoration.
- A multilingual interface design supporting Hindi and Marathi languages to enhance accessibility for non-English-speaking farmers.
- A role-based access control framework supporting farmer, Equipment owner and administrator user classes.
- A structured evaluation framework for validating offline Reliability, synchronization performance, and multilingual Usability upon full system implementation.

The remainder of this paper is organized as follows: Section II reviews related literature, Section III presents the Proposed system architecture, Section IV describes the design methodology; Section V outlines the proposed evaluation

Framework and Section VI concludes with future directions.

II. LITERATURE REVIEW

A. Digital Platforms in Agricultural Services

The integration of digital platforms into agricultural service ecosystems has been a subject of growing scholarly interest. Wolfert et al. [4] conducted a comprehensive review of big data applications in smart farming, establishing that cloud-connected platforms substantially improve resource allocation, equipment scheduling, and operational decision-making for farm operators. Their analysis, however, also highlighted that the effectiveness of cloud-dependent systems is fundamentally constrained by the quality of network infrastructure available at the point of use a constraint that severely limits their applicability in rural environments.

Kshetri [5] examined the role of mobile technology in transforming agricultural value chains in developing economies, concluding that smartphone penetration among rural Indian farming communities had surpassed 60% by 2022, creating a viable delivery channel for mobile-based agricultural services. Importantly, the study noted that internet connectivity quality, rather than device availability, had emerged as the primary adoption barrier.

B. Offline-First Architecture

The offline-first design paradigm has gained significant traction as a principled response to connectivity unreliability in mobile applications. Kleppmann and Beresford [6] formalized the theoretical foundations of offline-first systems through their work on Conflict-Free Replicated Data Types (CRDTs), demonstrating mathematically that distributed data stores can achieve eventual consistency without requiring continuous synchronization. Their framework provides the theoretical basis for the synchronization protocol proposed in this work.

Mehta and Gupta [7] evaluated an offline-capable mobile application for healthcare delivery in rural India, reporting a 94% reduction in data access failures relative to online-only counterparts in areas with intermittent connectivity. Their timestamp-based conflict resolution approach demonstrated practical effectiveness in resolving concurrent modification conflicts arising from multi-device offline usage a challenge directly relevant to the multi-role agricultural rental context addressed in this paper.

C. Cross-Platform Mobile Development

Flutter, Google's open-source cross-platform framework, has emerged as a leading tool for developing high-performance mobile applications from a single codebase. Biørn-Hansen et al. [8] conducted a systematic comparative evaluation of Flutter, React Native, and Xamarin, concluding that Flutter delivers superior UI rendering consistency and animation performance across Android and iOS platforms. For agricultural applications targeting the heterogeneous device landscape characteristic of rural India, Flutter's single codebase approach significantly reduces development complexity and long-term maintenance overhead.

D. Multilingual Accessibility in Rural Technology Adoption

Language represents a critical yet frequently underappreciated barrier to technology adoption among rural Indian populations. Rather and Sharma [9] investigated technology engagement patterns among Hindi and Marathi-speaking farming communities in Maharashtra, finding that language-mismatched interfaces reduced application engagement rates by up to 58% among users with limited English proficiency. Their findings strongly advocate for the embedding of regional language interfaces directly into application design rather than relying on post-hoc translation overlays.

E. Research Gap

A critical review of existing literature reveals that while offline-first design principles are well-established in general mobile development contexts, their application to agricultural equipment rental platforms in India remains largely unexplored. Current agricultural digital platforms are predominantly online-dependent, lack regional language support, and are not designed for the low-connectivity, low-literacy operational context characteristic of India's smallholder farming communities. This paper directly addresses this gap by proposing an integrated offline-first, multilingual mobile platform specifically designed for agricultural equipment rental.

III. PROPOSED SYSTEM ARCHITECTURE

A. Architectural Philosophy

The proposed system is grounded in the offline-first principle, which mandates that all user-facing operations must function independently of network availability. This is achieved by maintaining a continuously updated local data store on the device, which serves as the primary data source for all application operations. Network communication is restricted to background synchronization processes that operate asynchronously without disrupting the user experience.

B. Four-Layer Architecture

The system architecture is organized into four distinct layers as illustrated in

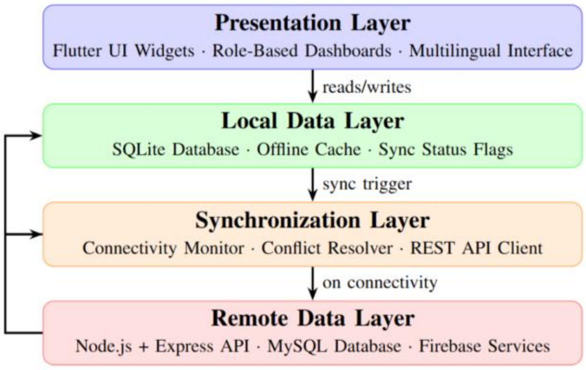

Fig. 1. Proposed Four-Layer Offline-First System Architecture

Presentation Layer: Implemented using Flutter widgets, this layer renders all user interface elements and handles user interactions. It communicates exclusively with the Local Data Layer, ensuring UI responsiveness is never degraded by network latency or unavailability.

Local Data Layer: Implemented using SQLite via the sqflite Flutter package, this layer maintains a complete device-resident replica of all data pertinent to the authenticated user, including equipment listings, booking records, payment summaries, and notification histories.

Synchronization Layer: This layer serves as the intelligent bridge between local and remote data stores. It continuously monitors network connectivity using the connectivity_plus package and activates synchronization processes upon detecting a transition from offline to online state.

Remote Data Layer: A Node.js and Express.js backend exposes RESTful APIs consumed by the synchronization layer. A MySQL database serves as the central authoritative data repository.

C. Synchronization Protocol

The proposed synchronization engine operates as a background isolate in Flutter to prevent blocking the main UI thread. The protocol executes the following sequential steps upon connectivity restoration:

1. Query local SQLite for all records flagged PENDING_SYNC.
2. Transmit pending records to the remote server via authenticated REST API endpoints.
3. Receive server acknowledgments with updated canonical timestamps.
4. Fetch remote updates generated during the offline period.
5. Merge remote updates into the local database, resolving conflicts by server-timestamp precedence.
6. Update all successfully synchronized records to SYNCED status.

D. Role-Based Access Control

Three distinct user roles are defined within the system:

• **Farmer:** Equipment browsing, booking management, payment history, and notification access.

• **Equipment Owner:** Equipment listing management, booking approval, and rental income tracking.

• **Administrator:** Full system oversight including user management, dispute resolution, and analytics.

IV. SYSTEM DESIGN AND METHODOLOGY

A. Technology Stack

Technology selections were guided by three primary criteria: offline operational capability, cross-platform compatibility, and suitability for deployment on the low-end Android devices prevalent in rural India. Table I summarizes the proposed technology stack.

TABLE I
PROPOSED TECHNOLOGY STACK

Component	Technology	Purpose
UI Framework	Flutter (Dart)	Cross-platform mobile development
Local Storage	SQLite (sqflite)	Offline data persistence
Backend	Node.js + Express	REST API and business logic
Remote DB	MySQL	Central data repository
Authentication	Firebase Auth	Secure user authentication
Notifications	Firebase FCM	Push notification delivery
State Mgmt	Riverpod	Reactive UI state handling
Connectivity	connectivity_plus	Network status detection
Localization	flutter_intl	Hindi and Marathi support

B. Database Schema Design

The local SQLite schema mirrors the remote MySQL schema with additional synchronization metadata columns. Each table incorporates sync status, local timestamp, and server timestamp fields to support the conflict resolution mechanism. The primary tables are:

• **Users-**Farmer, owner, and admin profiles with authentication tokens.

• **Equipment-Machinery** details including type, specifications, availability, price, and owner reference.

• **Bookings-Rental** requests with dates, statuses, and associated entity references.

• **Payments-Transaction** records with amounts, methods, and statuses.

• **Notifications-** Cached push notification payloads for offline display.

• **sync log-**Synchronization attempt records including outcomes and error traces.

C. Multilingual Implementation

Regional language support is implemented using Flutter's internationalization framework (flutter_intl) with Application Resource Bundle (ARB) files maintained separately for English (en), Hindi (hi), and Marathi (mr). Language selection is persisted locally to ensure consistent application of the chosen language across sessions, including fully offline ones.

Critical interface elements translated across all three languages include equipment category names, booking status descriptors, payment method labels, error messages, and all notification content strings.

D. Proposed Module Interaction

Fig. 2 illustrates the proposed user workflow for the booking process under both online and offline conditions.

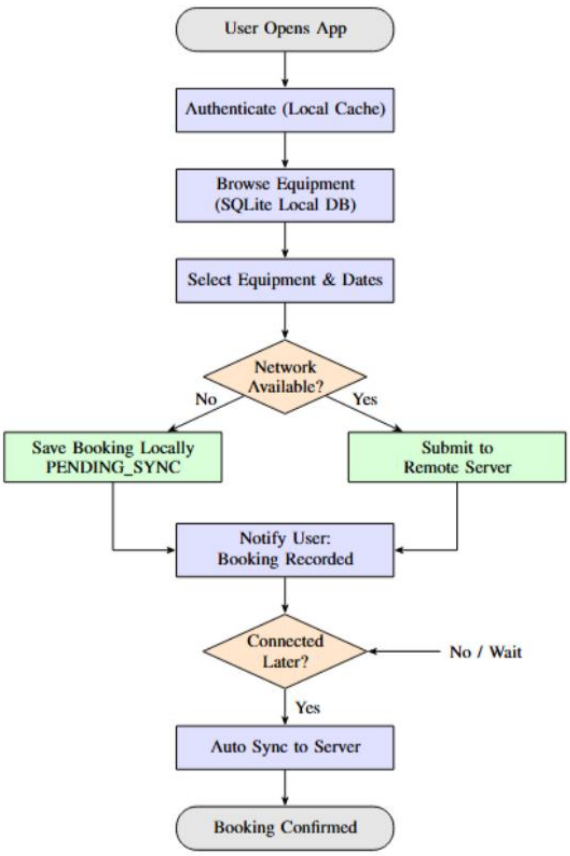

Fig. 2. Proposed Booking Workflow Under Online and Offline Conditions

E. Implementation Phases

The proposed development lifecycle is structured into five sequential phases:

Phase 1- Requirement Elicitation: Conducting structured surveys and interviews with farmers and equipment owners in the Junnar and Ambegaon talukas to capture functional requirements, connectivity constraints, and language preferences.

Phase 2- Architecture and Design: Defining the offline-first architecture, database schemas, API specifications, synchronization protocols, and UI/UX wireframes.

Phase 3-Frontend Development: Building the Flutter application including all screens, navigation flows, offline data access logic, localization ARB files, and Riverpod state management.

Phase 4- Backend Development: Constructing the Node.js REST API, MySQL database structure, Firebase Authentication integration, and FCM notification dispatch service.

Phase 5-Integration, Testing, and Deployment: Integrating all components, executing the proposed evaluation plan, and deploying for field validation.

V. PROPOSED EVALUATION FRAMEWORK

A. Evaluation Objectives

Upon full implementation, the system will be evaluated across three primary dimensions: offline functionality coverage, synchronization performance, and multilingual usability. This section presents the structured evaluation methodology proposed for each dimension.

B. Offline Functionality Assessment

The application will be tested under three simulated network conditions to assess feature availability:

• **Condition A-Full Offline:** Device in airplane mode simulating zero-coverage rural areas.

• **Condition B-Intermittent:** Network toggled at 30-second intervals to simulate signal fluctuations in fringe coverage areas.

• **Condition C-Online:** Stable 4G connectivity as a baseline performance reference.

Table II presents the expected feature availability matrix for each condition based on the proposed system design.

TABLE II
EXPECTED FEATURE AVAILABILITY BY NETWORK CONDITION

Feature	A	B	C
Equipment Browsing	✓	✓	✓
Booking Creation	✓	✓	✓
Booking History	✓	✓	✓
Profile Management	✓	✓	✓
Language Switching	✓	✓	✓
Payment Processing	×	✓*	✓
Push Notifications	×	✓*	✓
Data Synchronization	×	✓*	✓

* Available upon momentary connectivity restoration

C. Synchronization Performance Evaluation

Synchronization performance will be measured by recording the time required to sync varying volumes of pending records upon network restoration. Table III defines the proposed test parameters.

TABLE III
PROPOSED SYNCHRONIZATION PERFORMANCE TEST PARAMETERS

Test Set	Pending Record Volume	Metric Captured
S1	10 records	Sync time, conflict count
S2	50 records	Sync time, conflict count
S3	100 records	Sync time, conflict count
S4	250 records	Sync time, conflict count
S5	500 records	Sync time, conflict count

D. Multilingual Usability Study

A structured usability evaluation will be conducted with 30 participants recruited from farming communities in the Junnar taluka, Pune district 10 participants per language condition (English, Hindi, Marathi).

Each participant will complete five standardized tasks:

1. Equipment search using category filters
2. Booking initiation for a selected item
3. Booking cancellation
4. Payment record viewing
5. Notification history review

Primary metrics collected will include task completion rate, task completion time, and perceived difficulty score using a 5-point Likert scale.

E. Proposed Test Cases

Table IV defines the functional test cases that will be executed during the system validation phase.

TABLE IV
PROPOSED FUNCTIONAL TEST CASES

Test ID	Description	Expected
TC01	User registration and login	Pass
TC02	Equipment browse in offline mode	Pass
TC03	Booking creation while offline	Pass
TC04	Auto-sync on reconnection	Pass
TC05	Conflict resolution accuracy	Pass
TC06	Hindi language interface switching	Pass
TC07	Marathi language interface switching	Pass
TC08	FCM push notification delivery	Pass
TC09	Online payment processing	Pass
TC10	Role-based access enforcement	Pass
TC11	Low device storage handling	Pass
TC12	Session persistence in offline mode	Pass

F. Performance Benchmark Targets

Based on established benchmarks for mobile applications targeting low-resource devices [10], the following performance targets are defined for validation:

TABLE V
TARGET PERFORMANCE BENCHMARKS

Metric	Offline Target	Online Target
App Launch Time	< 2.0 sec	< 2.5 sec
Equipment List Load	< 0.5 sec	< 1.5 sec
Booking Submission	< 0.5 sec	< 1.0 sec
Memory Usage	< 60 MB	< 80 MB
Battery/hour	$< 3\%$	$< 5\%$
APK Size	< 25 MB	< 25 MB

VI. CONCLUSION AND FUTURE SCOPE

A. Conclusion

This paper presented the design and proposed evaluation of a mobile-first agricultural equipment rental application engineered specifically to address the connectivity constraints characteristic of rural India. By adopting an offline-first architecture grounded in local SQLite data persistence and intelligent background synchronization, the proposed system ensures that core rental functionalities equipment browsing, booking management, and user profile operations remain fully accessible regardless of network availability.

The integration of multilingual support for Hindi and Marathi directly addresses the language accessibility barrier identified as a key adoption inhibitor among non-English-speaking farming communities. The role-based access framework and Firebase-powered notification system further enhance the platform's operational completeness and user engagement potential.

The proposed system extends the authors' prior work on web-based agricultural equipment rental [1] by resolving its fundamental connectivity dependency, thereby broadening the platform's practical reach to include the rural smallholder farming communities most in need of efficient and accessible equipment rental solutions. The architectural design, module specifications, database schema, synchronization protocol, and evaluation framework presented in this paper establish a rigorous foundation for full system implementation and empirical validation in subsequent work.

B. Future Scope

• **On-Device ML Recommendations:** Integration of Tensor Flow Lite models for offline equipment recommendations based on crop type, season, and land size inputs.

• **IoT Integration:** GPS and usage sensors embedded in rented machinery for real-time location and operational monitoring.

• **Voice Interaction:** Voice command support in regional languages to reduce literacy barriers for low-education users.

• **Expanded Localization:** Extension to Telugu, Tamil, Kannada, and Bengali language interfaces.

• **Dynamic Pricing Engine:** Demand-aware rental rate optimization using seasonal booking pattern analysis.

• **iOS Deployment:** Extending Flutter application to the iOS platform for broader device coverage.

• **Peer-to-Peer Sharing:** Direct farmer-to-farmer equipment sharing arrangements within the platform ecosystem.

REFERENCES

[1] T. D. Fegade, S. R. Shaikh, A. C. Kharat, and K. Khatal, "Farming Rental Equipment Services: A Software Platform for Optimized Agricultural Machinery Usage," *Proc.* [Conference Name], [Location], [Year], pp. [xx–xx].

[2] Ministry of Agriculture and Farmers Welfare, Government of India, "Agricultural Statistics at a Glance 2022," Directorate of Economics and Statistics, New Delhi, India, 2022. [Online]. Available: https://agricoop.nic.in

[3] Telecom Regulatory Authority of India (TRAI), "Broadband India: Rural Connectivity and Quality of Service Report," New Delhi, India, Tech. Rep., 2023. [Online]. Available: https://www.trai.gov.in

[4] S. Wolfert, L. Ge, C. Verdouw, and M.-J. Bogaardt, "Big Data in Smart Farming: A Review," Agricultural Systems, vol. 153, pp. 69–80, May 2017.

[5] N. Kshetri, "The Role of Mobile Technology in Developing Countries' Agricultural Value Chains," Journal of Global Information Technology Management, vol. 23, no. 2, pp. 81–87, 2020.

[6] M. Kleppmann and A. R. Beresford, "A Conflict-Free Replicated JSON Datatype," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 10, pp. 2733–2746, Oct. 2017.

[7] P. Mehta and R. Gupta, "Offline-First Mobile Application Design for Healthcare in Low-Connectivity Rural Environments," Journal of Biomedical Informatics, vol. 98, pp. 103–112, 2019.

[8] A. Biørn-Hansen, T. A. Majchrzak, and T.-M. Grønli, "Progressive Web Apps: The Possible Web-native Unifier for Mobile Development," Proc. Int. Conf. Web Information Systems and Technologies *(WEBIST)*, pp. 344–351, 2017.

[9] S. Rathod and A. Sharma, "Language Barriers and Technology Adoption among Hindi and Marathi-Speaking Farming Communities," Int. J. Rural Development and Technology, vol. 7, no. 2, pp. 88–97, 2021.

[10] Google Developer Documentation, "Android Vitals: Performance Benchmarks for Mobile Applications," Google LLC, 2023. [Online]. Available: https://developer.android.com/topic/performance