IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511169

Parikrama Cross Platform Fees App

Gauri Joshi[1], Aaditya Bhalerao[2], Sujal Bhope[3], Prerana Ambilwade[4],Rohan Dhage[5]

Professor, Computer Department ,HSBPVT'S GOI Faculty of Engineering ,Kashti, Ahilyanagar ,India[1]

Student, Computer Department ,HSBPVT'S GOI Faculty of Engineering ,Kashti, Ahilyanagar ,India[2]

Student, Computer Department ,HSBPVT'S GOI Faculty of Engineering ,Kashti, Ahilyanagar ,India[3]

Student, Computer Department ,HSBPVT'S GOI Faculty of Engineering ,Kashti, Ahilyanagar ,India[4]

Student, Computer Department ,HSBPVT'S GOI Faculty of Engineering ,Kashti, Ahilyanagar ,India[5]

Abstract: Educational institutions often manage fee records, payment status, reminders, and receipts through manual or fragmented systems, which can create delays and errors. This paper presents the Parikrama Cross-Platform Fees App, a fee-management application built using Flutter and Firebase for administrators, staff, and students. The system supports role-based access, fee-record management, real-time synchronization, notifications, and digital receipts across supported platforms. The existing application architecture uses Flutter for the frontend and Firebase services for authentication, Firestore data storage, cloud functions, file storage, and notifications. This revised research proposes two algorithmic components: role-based access control for permission enforcement and priority-based fee reminders to organize follow-up actions. It also proposes a machine-learning extension that predicts whether a fee payment may be late using historical payment information. Logistic Regression is proposed as a baseline classifier and Random Forest as a comparison model. The predictive component is a proposed extension and requires implementation and validation using an appropriate de-identified dataset. Model performance will be evaluated using accuracy, precision, recall, F1-score, and a confusion matrix. Application performance will be evaluated through functional tests and response-time measurements. No predictive or performance results are claimed in this draft; they will be reported after experiments are completed. Initial results demonstrate the app's efficiency, with significant improvements in operational accuracy, reduced delays in data updates, and enhanced user experience. This solution exemplifies how modern technologies can simplify administrative processes, offering a scalable, secure, and cost-effective alternative to traditional methods. Future enhancements, including integration with third-party payment gateways and AI-driven fee forecasting, are proposed to further expand the app's capabilities and utility.

Keywords: Cross-Platform Development, Fee Management System, Flutter, Firebase, Machine Learning, Late-Payment Prediction, Random Forest, Logistic Regression

I. INTRODUCTION

Fee management is a fundamental administrative task in educational institutions, often involving significant manual effort and time. The traditional approaches to managing fees, such as manual record-keeping or siloed digital systems, face challenges in scalability, accessibility, and real-time updates. These limitations lead to inefficiencies, errors, and delays in fee-related operations, causing inconvenience for both administrators and students. In an era driven by digital transformation, there is a pressing need for a robust, scalable, and user-friendly solution that can streamline fee management processes across platforms.

This paper presents the Parikrama Cross-Platform Fees App, a comprehensive fee management system developed using Flutter, a modern UI toolkit by Google, and Firebase, a versatile backend-as-a-service platform. The application is designed to cater to the diverse needs of three key user groups: administrators, staff, and students. By leveraging Flutter, the app achieves true cross-platform compatibility, delivering a consistent user experience on Android, iOS, and web platforms without the need for separate codebases. Firebase serves as the backend, enabling real-time data synchronization, secure authentication, and scalable database management.

The app's core features address critical aspects of fee management, including real-time fee tracking, secure payment recording, automated notifications for payment deadlines, and downloadable receipts. Role-based access control ensures that each user type has appropriate privileges, enhancing both usability and security. Firebase Cloud Messaging is integrated to send timely reminders and updates, ensuring that users remain informed about fee-related activities.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511169

In conclusion, the Parikrama Cross-Platform Fees App offers an innovative and practical solution for modernizing fee management processes, paving the way for enhanced efficiency, accessibility, and user satisfaction.

II. PROBLEM STATEMENT

Educational institutions often face challenges in managing fee-related operations due to the limitations of traditional systems. These challenges include manual processes, lack of real-time updates, inefficient communication, and poor accessibility across platforms. Existing solutions are often fragmented, requiring separate applications for Android, iOS, and web platforms, which increases development and maintenance costs. Additionally, data security and user-specific access control are often overlooked, leading to vulnerabilities in sensitive financial information.

Key issues identified in traditional fee management systems include:

1. **Manual Processes**: Dependence on physical record-keeping or standalone software leads to inefficiencies, errors, and delays.
2. **Lack of Real-Time Updates**: Changes in fee records or payment status are not immediately reflected, causing miscommunication and confusion.
3. **Poor User Accessibility**: Existing solutions are not optimized for multiple platforms, limiting accessibility for users with different devices.
4. **Ineffective Notifications**: Absence of automated reminders for deadlines results in missed payments and subsequent penalties.
5. **Data Security Concerns**: Inadequate measures to secure sensitive financial and user data lead to potential risks of breaches and unauthorized access.
6. **Limited Scalability**: Legacy systems are often unable to handle a growing number of users or integrate with modern technologies like cloud-based platforms.

These challenges create significant barriers for institutions looking to streamline their fee management processes, ensure timely payments, and provide a seamless experience for administrators, staff, and students. The absence of a unified, cross-platform solution exacerbates these inefficiencies, calling for an innovative approach to address the gaps in existing systems.

The Parikrama Cross-Platform Fees App aims to solve these problems by providing an integrated, secure, and real-time fee management solution. By leveraging Flutter for cross-platform compatibility and Firebase for backend support, the app addresses the key challenges of traditional systems and ensures scalability, security, and efficiency.

III. ACTUAL METHODOLOGY

The methodology for developing the Parikrama Cross-Platform Fees App involves a systematic approach to design, development, testing, and deployment. This ensures that the app meets the functional and performance requirements of fee management while maintaining a user-friendly experience.

IV. REQUIREMENTS GATHERING

- **Stakeholder Analysis**: Identify the key users of the app—administrators, staff, and students—and their specific needs.
- **Functional Requirements**: Define the core features:
  - User authentication and role-based access control.
  - Fee record management with CRUD (Create, Read, Update, Delete) operations.
  - Real-time synchronization of data across platforms.
  - Push notifications for fee reminders and updates.
  - Generation and download of fee receipts.
- **Non-Functional Requirements**:
  - Cross-platform compatibility (Android, iOS, and Web).
  - Scalability to support a large number of users.
  - Secure handling of user data and financial transactions.

V. SYSTEM ARCHITECTURE DESIGN

The app is designed using a modular, layered architecture to ensure flexibility and maintainability:

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511169

- **Frontend**:
  - Developed using **Flutter** to provide a single codebase for Android, iOS, and web platforms.
  - Utilizes Flutter widgets to create a responsive and consistent UI/UX.
- **Backend**:
  - Firebase serves as the backend, offering:
    - **Authentication**: Firebase Authentication for secure user login with email/password and role-based access.
    - **Database**: Firestore for real-time, cloud-based data management.
    - **Cloud Functions**: Serverless functions to handle complex logic, such as fee calculation or custom notifications.
    - **Storage**: Firebase Storage for saving and retrieving receipts or other documents.
- **Communication**: Firebase Cloud Messaging for push notifications.
- **Security**: Role-based access control and encrypted data transmission ensure compliance with security standards.

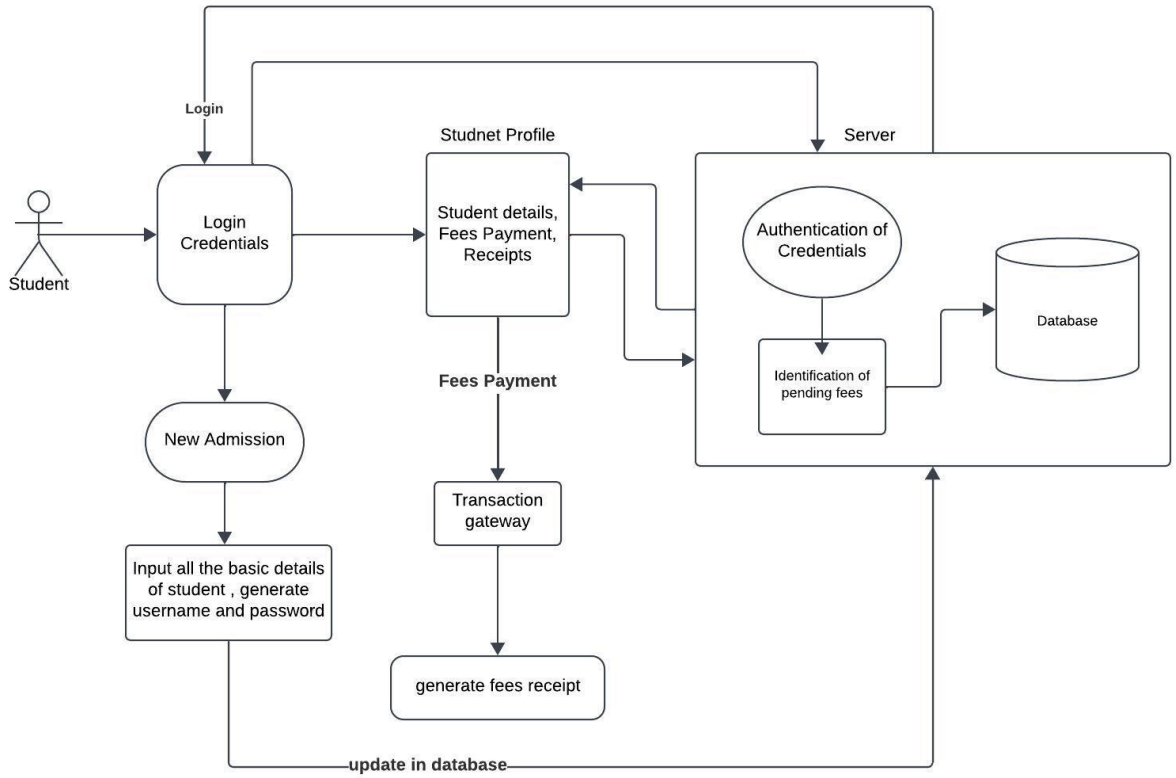


Fig. 3 1. The System Architecture of the Parikrama Cross Platform Fees App

VI. RESEARCH GAP AND OBJECTIVES

Many fee-management systems provide digital record keeping and payment notifications, but the current project draft does not describe a validated method for estimating late-payment risk. The proposed research extension combines cross-platform fee administration with transparent prediction and reminder prioritization. The research objectives are: (1) provide a unified fee-management workflow across supported platforms; (2) enforce user permissions according to roles; (3) prioritize fee reminders using documented input factors; and (4) evaluate late-payment classification models using historical, de-identified payment data.

VII. PROPOSED ALGORITHMS

A. Algorithm 1: Role-Based Access Control (RBAC)

Input: Authenticated user, requested operation, target record. Output: Access granted or denied.

1. Authenticate the user.
2. Retrieve the user role from the trusted backend.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511169

3. Identify the permission required for the requested operation.
4. Check whether the role is authorized for that permission and target record.
5. If authorized, allow the operation; otherwise, deny it.
6. Record the access event where appropriate.
7. Return the access decision.

Note: This algorithm is a design description; the implementation must enforce permissions using backend controls and Firebase Security Rules, not only client-side checks.

B. Algorithm 2: Priority-Based Fee Reminder

Input: Days overdue (D), outstanding balance (A), previous late-payment count (H). Output: Reminder priority.

1. Normalize D, A, and H to a documented range from 0 to 1.
2. Calculate PriorityScore = wD × D + wA × A + wH × H, where wD + wA + wH = 1.
3. Apply the approved thresholds: Low (0.00–0.30), Medium (0.31–0.60), and High (0.61–1.00).
4. Sort eligible records by descending priority score.
5. Generate reminder actions according to institutional policy.
6. Return priority and reminder queue.

Note: The weights and thresholds are initial design choices and must be tested and approved before deployment.

VIII. PROPOSED MACHINE-LEARNING METHODOLOGY

Prediction task: binary classification of whether a fee payment will be late. Target label: 0 = paid on or before the due date; 1 = paid after the due date. The prediction must be made using information available before the relevant due date to avoid data leakage.

Proposed input features include total fee amount, amount paid before the prediction date, outstanding balance, days remaining until the deadline, number of previous late payments, installment count or status, and payment-history indicators. Only features that are available, lawful, relevant, and consistently recorded will be included. Sensitive personal information should be excluded unless a documented ethical and institutional justification exists.

Data preparation: (1) obtain permission to use records; (2) remove direct identifiers and de-identify records; (3) inspect duplicates and missing values; (4) encode categorical variables and scale numeric features where required; (5) split the dataset into training and test sets, preferably using a time-based split when records span multiple periods; and (6) fit preprocessing only on training data.

Model 1—Logistic Regression: used as an interpretable baseline to estimate the probability of late payment. Model 2—Random Forest Classifier: used to compare performance while capturing nonlinear patterns and interactions. Hyperparameters will be selected using validation data or cross-validation, without using the final test set for tuning.

Proposed workflow: collect and de-identify records → define prediction point and target → prepare features → split data → train Logistic Regression and Random Forest → evaluate on held-out data → compare metrics → document limitations. The model output may be presented as a risk estimate for administrative follow-up; it should not automatically impose penalties or make consequential decisions about students.

IX. EXPERIMENTAL EVALUATION PLAN

Predictive evaluation will report accuracy, precision, recall, F1-score, and a confusion matrix for each model. Because late-payment records may be imbalanced, recall, precision, and F1-score will be considered alongside accuracy. Where feasible, probability calibration and ROC-AUC or PR-AUC may also be reported. The evaluation will state dataset size, class distribution, split method, feature list, and model settings so the experiment can be reproduced.

Application evaluation will include functional testing of login and role permissions, fee-record creation and updates, receipt generation, synchronization, and notification workflows. Response-time measurements will be taken over documented test runs and summarized using averages and observed ranges. These tests will assess the application independently from the ML model.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511169

X. RESULTS (TO BE COMPLETED AFTER IMPLEMENTATION AND TESTING)

No experimental ML or system-performance results are available in this draft. After implementation, report actual measurements in the tables below; do not substitute estimated or fabricated values.

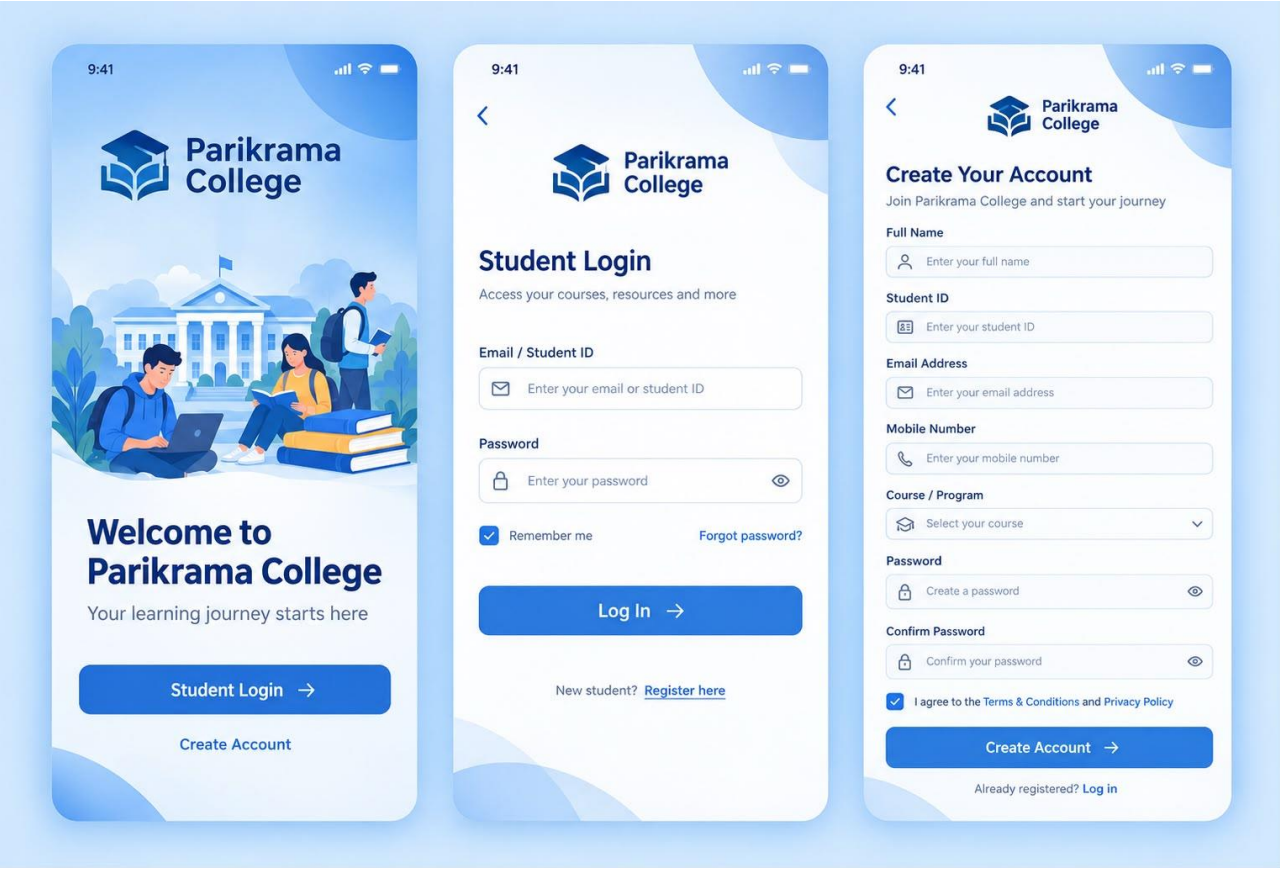
Table 1—Model comparison: Model | Accuracy | Precision | Recall | F1-score | Confusion matrix. Rows: Logistic Regression; Random Forest.

Table 2—Application performance: Test operation | Number of runs | Average response time | Observed range. Include login, fee-record update, receipt generation, synchronization, notification, and prediction response if implemented.

XI. LIMITATIONS AND RESPONSIBLE USE

The proposed prediction model depends on the quality, size, and representativeness of historical records. Small or imbalanced datasets may result in unreliable estimates. Payment patterns may change over time, so model performance should be monitored and periodically re-evaluated. Predictions should support staff review rather than automatically determine penalties, restrict access, or make other consequential decisions. Data use must follow institutional approval, privacy safeguards, and applicable policies.

XII. RESULT



XIII. CONCLUSION

The Parikrama Cross-Platform Fees App is designed to address fee-management challenges through a cross-platform interface and Firebase-based backend. The existing design covers user authentication, fee records, synchronization, notifications, and digital receipts. This revised paper also proposes role-based access-control and reminder-prioritization algorithms, together with a machine-learning extension for late-payment risk prediction. The predictive component remains subject to implementation and experimental validation.



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511169

The cross-platform approach is intended to provide a consistent experience across Android, iOS, and web while Firebase services support data synchronization and backend operations. The proposed Logistic Regression and Random Forest models will be compared using a held-out dataset and standard classification metrics. Application functionality and response times will also be tested. These evaluations are necessary before claiming measurable improvements or predictive effectiveness.

In conclusion, the project provides a foundation for digital fee administration and proposes predictive analytics as a future enhancement. Further work includes implementing and validating the algorithms and ML models, reporting measured results, verifying references, and assessing privacy, fairness, and reliability before practical deployment.

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to all those who have supported and guided us throughout the development of the Parikrama Cross-Platform Fees App.

First and foremost, we are deeply thankful to our institution, Parikrama Institute, for providing us with the opportunity and resources to work on this project. The invaluable support from our faculty members and mentors has been instrumental in shaping the direction and execution of this work.

We extend our heartfelt thanks to Prof. Monali Bansode, whose expert guidance, encouragement, and constructive feedback have helped us overcome challenges and refine our solution. Their technical insights and critical reviews have greatly enhanced the quality of this app and its underlying methodologies.

Finally, we are grateful to our peers, family, and friends for their unwavering support and motivation throughout this journey. Their encouragement inspired us to strive for excellence and complete this project successfully.

This project would not have been possible without the collective efforts and collaboration of everyone involved. We are proud to present the Parikrama Cross-Platform Fees App as a testament to the transformative power of technology in solving real-world challenges.

REFERENCES

[1]. Breiman, L. (2001). Random Forests. Machine Learning, 45, 5–32. https://doi.org/10.1023/A:1010933404324
[2]. Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830. https://jmlr.org/papers/v12/pedregosa11a.html
[3]. Cox, D. R. (1958). The Regression Analysis of Binary Sequences. Journal of the Royal Statistical Society: Series B, 20(2), 215–242.
[4]. Flutter documentation. https://docs.flutter.dev/ (accessed 19 November 2026).
[5]. Firebase Authentication documentation. https://firebase.google.com/docs/auth (accessed 19 November 2026).
[6]. Cloud Firestore documentation. https://firebase.google.com/docs/firestore (accessed 19 November 2026).
[7]. Firebase Cloud Messaging documentation. https://firebase.google.com/docs/cloud-messaging (accessed 19 November 2026).
[8]. Firebase Security Rules documentation. https://firebase.google.com/docs/rules (accessed 19 November 2026).
[9]. Scikit-learn documentation. https://scikit-learn.org/stable/ (accessed 19 November 2026).
[10]. REFERENCE VERIFICATION NOTE: The original draft included several references with incomplete bibliographic details (including Raj & Patil; Sharma & Gupta; Pal; Singh & Mehta). These entries have been omitted from this revised bibliography pending verification of exact titles, authors, publication venues, volume/issue/pages, and DOI or publisher links. Add verified literature on student payment behavior and educational fee-management systems before final submission.

 This work is licensed under a Creative Commons Attribution 4.0 International License