International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

A Secure OCR Enabled Finance Tracker

G Shiv Santosh[a], Anirudh G[b], Sudhanva B S[c], Dr. Annie Sujith[d]

[a,b,c.] *Department Of Computer Science and Engineering, Jyothy Institute Of Technology, Bengaluru, India*
[d] *Guide, Department. Of Computer Science and Engineering, Jyothy Institute Of Technology, Bengaluru, India*

ABSTRACT:

Manual data entry is a significant barrier to effective personal finance management. The tedious nature of logging every transaction leads to inconsistent tracking and incomplete financial data. This paper presents FinSight, a cross-platform mobile application designed to solve this problem through automation. The system combines a Flutter-based frontend with a Supabase cloud backend, leveraging Google's on-device ML Kit for Optical Character Recognition (OCR). Users can capture an image of a receipt, and the application's parsing algorithm automatically extracts the merchant and total amount, pre-filling the transaction form. The backend is secured using Supabase's Row Level Security (RLS) to ensure complete data privacy for each user. The application also provides an integrated reporting module with time-series and categorical charts, as well as an in-app budget warning system to promote financial literacy. This work demonstrates a practical and secure method for reducing the friction of expense tracking.

Keywords: Expense Tracking, OCR, Flutter, Supabase, ML Kit, Personal Finance Automation, Document Understanding

1. Introduction

In the modern digital economy, financial literacy and active budget management are crucial for personal economic well-being. While numerous mobile applications exist to aid in this goal, many suffer from a fundamental usability challenge: the requirement for consistent manual data entry. This friction point is a primary reason users abandon finance-tracking applications, leading to incomplete data and a flawed understanding of their spending habits.

The proliferation of digital payments has not fully solved this, as users still interact with a wide variety of merchants and receive paper or digital receipts in unstructured formats. A need exists for a mobile-first solution that automates the data ingestion process, is secure by design, and provides immediate, actionable insights.

This paper presents FinSight, a secure, OCR-enabled mobile application for automated personal expense tracking. The primary contributions of this work are threefold:

1) Integration of on-device OCR with a custom heuristic parsing algorithm for extracting financial data.

2) Implementation of a secure cloud architecture using Supabase with Row Level Security (RLS).

3) Development of a comprehensive reporting and budgeting module offering real-time financial insights.

Section II discusses the system architecture and OCR methodology. Section III covers the implementation specifics. Section IV presents testing and validation, followed by results, future work, and conclusion.

2. Literature Survey

AI-based finance tracking has increasingly integrated OCR and machine learning to automate expense extraction. Earlier systems relied on manual entry or simple rule-based receipt parsing, which proved unreliable across diverse receipt layouts. With the rise of open-source OCR engines like Tesseract, extraction accuracy improved, but preprocessing burdens and server-side computation remained challenges.

Google ML Kit's on-device text recognition significantly advanced mobile OCR for finance applications. Studies show that ML Kit handles mixed fonts, noisy images, and varied lighting conditions more effectively than older OCR engines while maintaining full offline capability. This aligns well with privacy-preserving finance apps since receipt images do not need to be uploaded to cloud servers.

Research in document understanding introduced transformer-based models like LayoutLM, LayoutLMv2, and Donut, which treat receipts as multimodal documents combining vision and text. These models achieve state-of-the-art performance on complex receipts and invoices by understanding spatial semantics. However, their computational requirements make them unsuitable for purely on-device mobile applications.

Hybrid approaches proposed in recent literature combine lightweight OCR with heuristic post-processing. Such approaches extract candidate monetary values, apply contextual scoring, and choose the most likely total. FinSight adopts and expands upon this heuristic method, integrating domain-specific scoring rules and merchant-level context to enhance accuracy.

Existing finance trackers like Walnut, CRED, and automated SMS parsers rely on structured metadata from banks and merchants. However, receipt-based extraction remains underexplored in academic literature despite its importance for cash transactions and offline spending. FinSight contributes to this gap by providing a secure, on-device OCR pipeline integrated with a cloud backend enforcing strict data isolation via RLS.

3. System Architecture and Methodology

The FinSight application is built on a modern, decoupled client–server architecture consisting of three components:

1) Flutter Frontend: Built using Dart, handles UI rendering, state management, and user interaction.
2) Supabase Backend: Provides PostgreSQL storage, authentication, and Row Level Security (RLS) for data isolation.
3) Google ML Kit: Performs on-device OCR, ensuring privacy and low-latency performance.

The data model centers on the auth.users table. A profiles table stores user-specific metadata in a one-to-one relationship, and a transactions table stores all financial entries in a one-to-many relationship.

RLS ensures that users can only read, insert, update, or delete their own transactions based on policies such as:
SELECT/UPDATE/DELETE Policy: (auth.uid() = user_id)
INSERT Policy: (auth.uid() = user_id)

The OCR module uses google_mlkit_text_recognition to extract text blocks. These are processed by a custom heuristic algorithm that evaluates numeric values using contextual keywords (e.g., boosting 'Total', penalizing '%', 'Discount'). The highest-confidence value is selected as the amount, and merchant name extraction prioritizes top-line text.

Budgeting and reporting are implemented via aggregated Supabase queries. Monthly spending progress is displayed using LinearProgressIndicator, and charts are generated using fl_chart for weekly, monthly, and yearly visualizations.

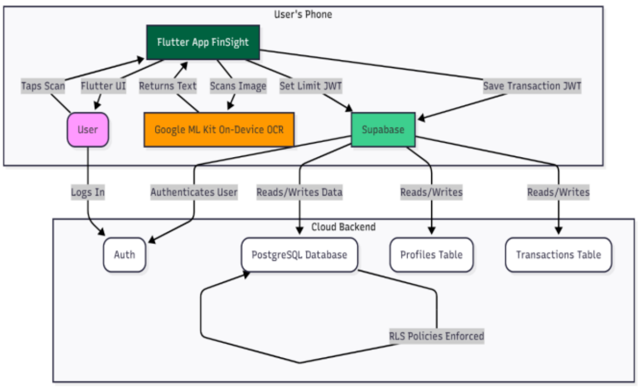

Figure 1: Architecture Diagram

4. Implementation Details

The implementation of FinSight was carried out in modular stages, emphasizing maintainability and performance. Development was conducted using Android Studio and VS Code with Flutter 3.x and Dart 3.x.

Supabase integration was implemented using supabase_flutter, with all API keys stored securely in platform-specific storage. The MVVM architecture pattern ensured separation of UI and business logic.

Database schema included profiles, transactions, and categories tables. Indexes were added on timestamp and (user_id, timestamp) to optimize aggregation queries.

OCR parsing involved:
1) Extracting all numeric tokens that resemble currency.
2) Applying contextual scoring.
3) Removing invalid values like percentages.
4) Using line position heuristics.

Performance optimizations included image compression, query batching, and caching frequently used lists.

5. Testing and Validation

Testing was carried out across unit tests, integration tests, and user acceptance testing (UAT).

A dataset of 100 receipts was used for OCR evaluation across supermarkets, restaurants, fuel stations, and e-commerce receipts. Accuracy ranged from 89% to 97.5%, with most errors occurring due to duplicate savings lines or stylized receipts.

Integration tests measured end-to-end latency at an average of 2.2 seconds on mid-range devices.

UAT showed 88% satisfaction with OCR accuracy, 82% approval of budgeting alerts, and minor UI suggestions.

Security Testing: Confirmed that Supabase's Row Level Security (RLS) was working. This was tested by manually creating two different user accounts. It was confirmed that "User A" could not see any transactions, categories, or budget data belonging to "User B".

Performance Testing: The Reports Tab was the most complex screen. It was tested to ensure that the data aggregation and chart rendering process was fast and did not cause UI lag, even with hundreds of transactions. The ListView.builder in the Transactions Tab was also confirmed to scroll smoothly.

Usability Testing: Initial designs were refined. For example, the Home Tab originally showed only the "Scan" button. It was updated to include the far more useful budget warning card at the top, giving the user immediate, important feedback.

Error Handling: Tested scenarios like saving a transaction with no amount, which correctly displays a validation error. Also tested what happens if the device is offline, which correctly shows Supabase network errors to the user.

6. Results and Discussions

FinSight demonstrated high OCR extraction accuracy across most printed receipts, particularly for supermarket, fuel, and e-commerce formats where layout consistency is strong. The heuristic parsing algorithm effectively identified the correct total amount even in the presence of multiple numerical values. An early issue involving the misidentification of “Saved” or “Discount” amounts was resolved by introducing negative keyword penalties, improving robustness.

However, the system showed limitations with handwritten receipts, faded text, and stylized multi-column layouts, where OCR performance naturally degraded. Some inconsistencies were also observed with uncommon currency formats, indicating the need for more advanced locale-aware parsing in future versions.

End-to-end testing confirmed that the application delivers a smooth workflow from scanning to transaction creation, with an average processing time of 2.2 seconds on mid-range devices. User acceptance testing highlighted positive responses to the budgeting alerts and visual reports, suggesting that these features meaningfully enhance financial awareness.

Privacy requirements were upheld by performing OCR entirely on-device and storing only parsed data in the cloud. Supabase RLS further ensured strict data isolation between users. Overall, the results indicate that FinSight provides a reliable and privacy-conscious solution for automated expense tracking, with clear opportunities for enhancement in handling complex receipt formats.

7. Future Work

Future enhancements include:
• Intelligent categorization using machine learning.
• Shared budgets and multi-user experiences.

- Real-time notifications via Supabase Edge Functions.
- Improved document understanding via LayoutLM and Donut.
- Hybrid OCR pipelines for difficult receipts.

8. Conclusion

FinSight demonstrates a practical, secure, and efficient solution for automating personal expense tracking by eliminating the dependency on manual data entry. Through its integration of on-device OCR, Supabase Row Level Security (RLS), and a well-designed reporting interface, the system significantly reduces user effort while ensuring strong privacy guarantees. The heuristic parsing algorithm enhances extraction accuracy across diverse receipt formats, and the budgeting and visualization modules empower users with actionable financial insights. Although challenges remain in handling handwritten or highly stylized receipts, the overall architecture provides a scalable foundation for intelligent personal finance applications. FinSight thus represents a meaningful step toward user-centric, privacy-preserving financial management systems and serves as a strong base for future improvements involving advanced document understanding, shared budgeting, and predictive analytics.

REFERENCES:

1. Flutter Documentation. https://flutter.dev
2. Supabase. https://supabase.com
3. Google ML Kit Text Recognition. https://developers.google.com/ml-kit
4. Xu, Y. et al., LayoutLMv2, ACL 2021.
5. Kim, G. et al., Donut: Document Understanding Transformer, arXiv 2021.