Journal of Computer Science Review and Engineering | ***ISSN 1694-4488***
Published by AIR JOURNALS | *https://airjournal.org/jcsre*
16/18 Avenue des Longaniers, Quatre Bornes, Mauritius
airjournals@gmail.com; enquiry@airjournal.org

RESEARCH ARTICLE

Design and Implementation of a Web-Based Management System for Mgbakwu Cooperative Society

Nwoye, John Nnamdi[1], Mmaduakonam, Nwadiogo Eugenia[2] & Ezeanya, Ifeoma Hope PhD[3]
[1 3] Electrical Electronic Engineering Technology, Anambra State Polytechnic, Mgbakwu
[2] Computer Science Technology, Anambra State Polytechnic, Mgbakwu

Citations - APA

Nwoye, J. N., Mmaduakonam, N. E. & Ezeanya, I. H. (2025). Design and Implementation of a Web-Based Management System for Mgbakwu Cooperative Society. *Journal of Computer Science Review and Engineering, 9(3) 10-18. DOI: https://doi.org/10.5281/zenodo.17972078*

ABSTRACT

The Mgbakwu Cooperative Management System is a web-based application designed to automate and digitize the daily operations of the Mgbakwu Cooperative Society. Manual recordkeeping methods traditionally used in cooperative societies are prone to delays, data loss, and financial inaccuracies. To address these challenges, the proposed system integrates core cooperative functions—such as member registration, savings and share contributions, loan application and approval workflows, and financial accounting- into a unified digital platform. The system was designed and implemented using React for the frontend interface, Flask (a Python-based framework) as the backend framework, and MySQL as the database engine for persistent data storage. The architecture follows a client–server model with RESTful API communication between the frontend and backend layers. Role-Based Access Control (RBAC) ensures data security and accountability among users, including the President/Administrator and general members. Key features of the system include automated contribution tracking, loan amortization scheduling, real-time reporting, and document management. Testing and validation of the system confirmed its efficiency in managing member data, processing loans, and generating timely financial reports. The software significantly improves transparency, reduces human error, and enhances decision-making processes within the cooperative. The Mgbakwu Cooperative Management System thus represents a scalable and secure digital solution for modernizing cooperative operations in community-based financial organizations.

Keywords: *Plant Disease Detection; Convolutional Neutral Network (CNN); Food Security; Localized Dataset*

Introduction

Cooperative societies play a critical role in Nigeria by providing financial services to ordinary citizens, particularly those who may not have easy access to banks. They facilitate savings, provide loans at affordable interest rates, and promote community development. Despite their importance, many cooperatives still rely on manual, paper-based systems for managing members, contributions, savings, and loans. These traditional methods often lead to errors, delays, and disputes, undermining members' confidence and reducing operational efficiency (Akinbode & Adewale, 2021a; Chukwuma, 2019).

In response to these challenges, this study proposes the development of a web-based cooperative management system tailored for Nigerian cooperatives. The system aims to:

1. Automate member registration, savings, contributions, and loan processing.
2. Provide role-based access for members and administrators to ensure proper workflow control.
3. Maintain accurate, auditable transaction records to enhance accountability.
4. Deliver real-time reports and notifications for operational transparency (Akinbode & Adewale, 2020b), (Ezeanya & Umeh, 2020a).

By digitizing cooperative operations, the system seeks to reduce manual errors, improve efficiency, and provide a reliable platform for financial record-keeping that aligns with modern ICT practices in Nigeria.

Related Work

Several studies have highlighted the benefits of ICT adoption in Nigerian cooperatives. Akinbode and Adewale (2021a) and Akinbode and Adewale (2021b) demonstrated that digital tools improve access to financial services, enhance record accuracy, and encourage financial discipline among members. Ezeanya and Umeh (2020a) showed that integrating ICT in cooperative operations reduces manual errors and improves workflow efficiency. Nwachukwu and Okafor (2022a) and Nwachukwu & Okafor (2022b) emphasized that digital transformation in Nigerian cooperatives promotes accountability and transparency, particularly in loan processing and savings management.

Despite these advantages, most existing cooperative management systems in Nigeria focus either on record-keeping or reporting. They rarely provide a fully integrated platform that combines contributions, savings, loan management, notifications, and real-time reporting. This study addresses this gap by developing a comprehensive system suitable for small- and medium-sized Nigerian cooperatives.

Methodology

A. Research Approach

The study follows a **design–science research paradigm**, emphasizing the creation of a software artifact to solve practical problems faced by Nigerian cooperatives. Data collection involved:

i. **Interviews:** Engaging cooperative executives and officers to understand existing workflow challenges (Chukwuma, 2019; Nwachukwu & Okafor, 2022a).
ii. **Observation:** Studying manual processes to identify pain points in contribution tracking, loan management, and savings.
iii. **Document Analysis:** Reviewing historical financial records to identify common errors and inefficiencies (Ezeanya & Umeh, 2020a).

These activities informed the functional and non-functional requirements of the system.

B. System Development Lifecycle

The **Waterfall model** of SDLC was adopted due to its structured, sequential phases, which support comprehensive documentation and reduce inconsistencies (Pressman & Maxim, 2022). The phases include:

1. **Requirements analysis** – Identifying the needs of members and administrators.
2. **System design** – Defining architecture, database schema, and module interfaces.
3. **Implementation** – Coding frontend, backend, and database.
4. **Testing** – Functional, performance, and security testing.
5. **Deployment** – Installing the system for real use.
6. **Maintenance** – Addressing user feedback and system improvements.

Although Waterfall is sequential, an iterative refinement approach was applied to modules such as loan approval, contributions tracking, and reporting dashboards, ensuring usability and reliability (Banker, 2022).

IV. System Architecture

The system adopts a **three-tier client–server architecture**:

A. Presentation Layer

i. Developed using **React.js**, this layer provides members with a responsive dashboard, loan application forms, savings and contribution pages, and administrative control panels (Banker, 2022).
ii. Modular design allows easy maintenance and integration with backend APIs.

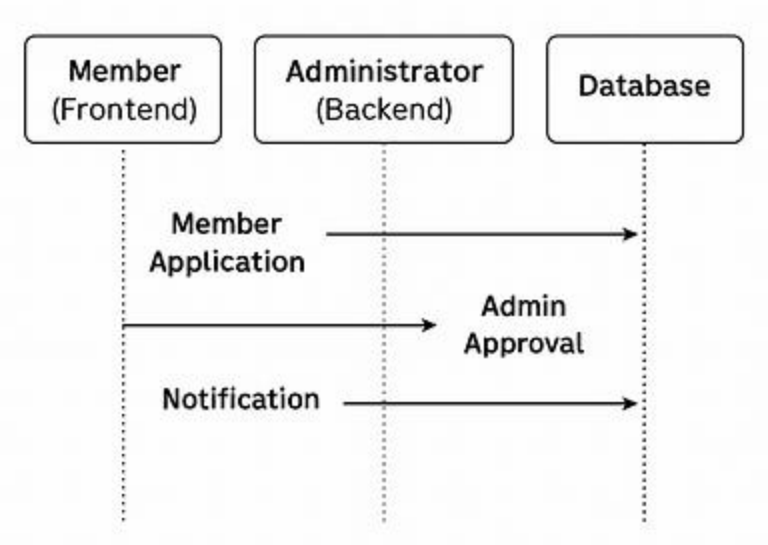
B. Application Layer

i. Implemented with **Flask**, it manages authentication with **JWT**, input validation, business logic for loan processing, contributions, and RESTful API endpoints (Ezeanya & Umeh, 2020a).
ii. Ensures secure, scalable, and efficient communication between frontend and database.

C. Database Layer

i. Uses **MySQL** to store structured data related to members, contributions, savings, loans, guarantors, notifications, and transactions (Sommerville, 2020).
ii. Normalized to **Third Normal Form (3NF)** to minimize redundancy and maintain data integrity.

Figure 1: Three-Tier Architecture Diagram



Database Design

A. Entity-Relationship Model

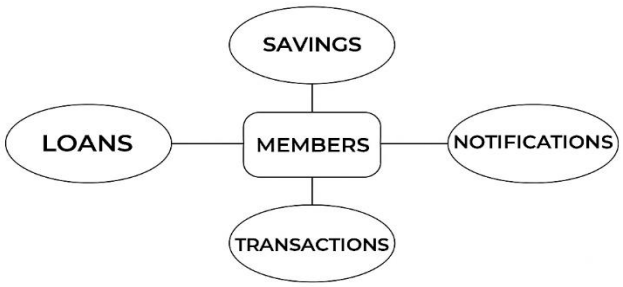
The database comprises the following key entities:

i. **Members:** Stores member details such as ID, name, email, phone, password, role, and registration date.
ii. **Contributions:** Tracks member contributions with reference to member ID and transaction date.
iii. **Savings:** Stores savings records including amount, type, and date.
iv. **Loans:** Manages loan applications, approval workflow, and guarantor relationships.
v. **Notifications:** Contains alerts sent to members and administrators.
vi. **Transactions:** Logs debit and credit entries for auditing purposes.

Table 1: Key Database Tables

Table	*Key Attributes*	*Description*
Members	id, name, email, phone, password, role, date	Stores member information
Contributions	id, member_id, amount, date	Tracks contributions
Savings	id, member_id, amount, type, date	Records savings transactions
Loans	id, loan_id, guarantor_id, relationship	Manages loan applications
Notifications	id, recipient_role, type, message, is_read, created_at	Internal alerts
Transactions	id, member_id, type, amount, description, date	Auditable transaction records

Figure 2: ER Diagram for Cooperative Database



System Modules and Workflows

A. Member Management

Handles registration, login, password management, and role-based access. Provides members with a profile dashboard for monitoring contributions, savings, and loan status (Akinbode & Adewale, 2021a; Chukwuma, 2019).

B. Savings and Contributions

Enables members to post savings and contributions and automatically logs transactions. Provides historical statements for transparency (Akinbode & Adewale, 2020b; Ezeanya & Umeh, 2020a).

C. Loan Management

Allows members to submit loan applications, links guarantors, and supports administrator approval workflows. Automatically updates loan status and sends notifications (Nwachukwu & Okafor, 2022a; Nwachukwu & Okafor, 2022b).

D. Notifications

Generates internal alerts such as contribution confirmations, loan updates, and reminders.

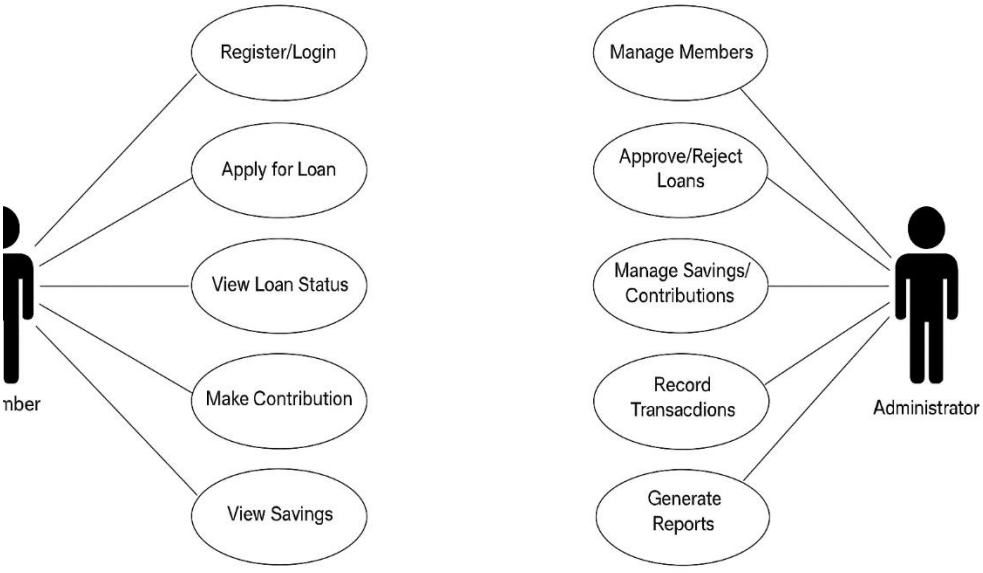
E. Financial Reporting

Generates detailed reports for members and administrators including loan summaries, cash flow analysis, and contribution history.

F. Transaction Logging

Records every debit and credit operation, creating a fully auditable financial trail to ensure accountability and prevent fraud.

Figure 3: Use Case Diagram



Use Case Diagram

(es)

Implementation

A. Frontend

i. React.js used for interactive dashboards.
ii. Context API and protected routes enforce role-based access.

B. Backend

i. Flask manages RESTful APIs, authentication, input validation, and business logic.
ii. Secure JWT tokens prevent unauthorized access (Ezeanya & Umeh, 2020a).

C. Database Integration

i. SQLAlchemy ORM integrates MySQL with Python backend.
ii. Supports transaction consistency, rollback on failure, and efficient queries (Sommerville, 2020).

Code Snippet: Member Model (Python/SQLAlchemy)

```
class Member(db.Model):
  id = db.Column(db.Integer, primary_key=True)
  name = db.Column(db.String(120), nullable=False)
  email = db.Column(db.String(120), unique=True, nullable=False)
  password = db.Column(db.String(255), nullable=False)
  role = db.Column(db.String(20), default='member')
  date = db.Column(db.DateTime, default=datetime.utcnow)
```

System Testing and Results

A. Functional Testing

Table 2: Functional Testing

Test Case	*Expected Result*	*Status*
Member registration	Account created	Pass
Invalid login	Reject credentials	Pass
Post contribution	Contribution saved	Pass
Loan application	Forwarded to admin	Pass
Admin approval	Loan status updated	Pass
Protected route access	Deny unauthorized access	Pass

B. Results Discussion

i. Automation eliminated manual errors.
ii. Admins can approve loans and generate reports instantly.
iii. Members monitor accounts and loan status in real-time (Akinbode & Adewale, 2021a; Nwachukwu & Okafor, 2022a; Akinbode and Adewale, 2021b; Nwachukwu & Okafor, 2022b).

Figure 5: Login Screenshot

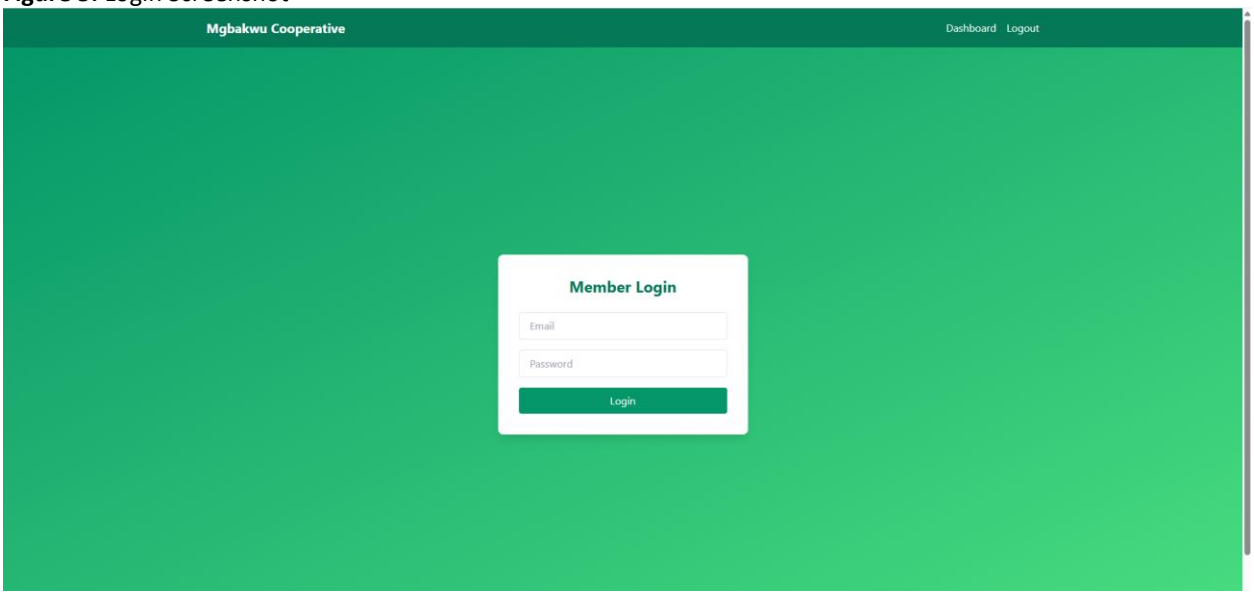


Figure 6: Admin Dashboard Screenshot

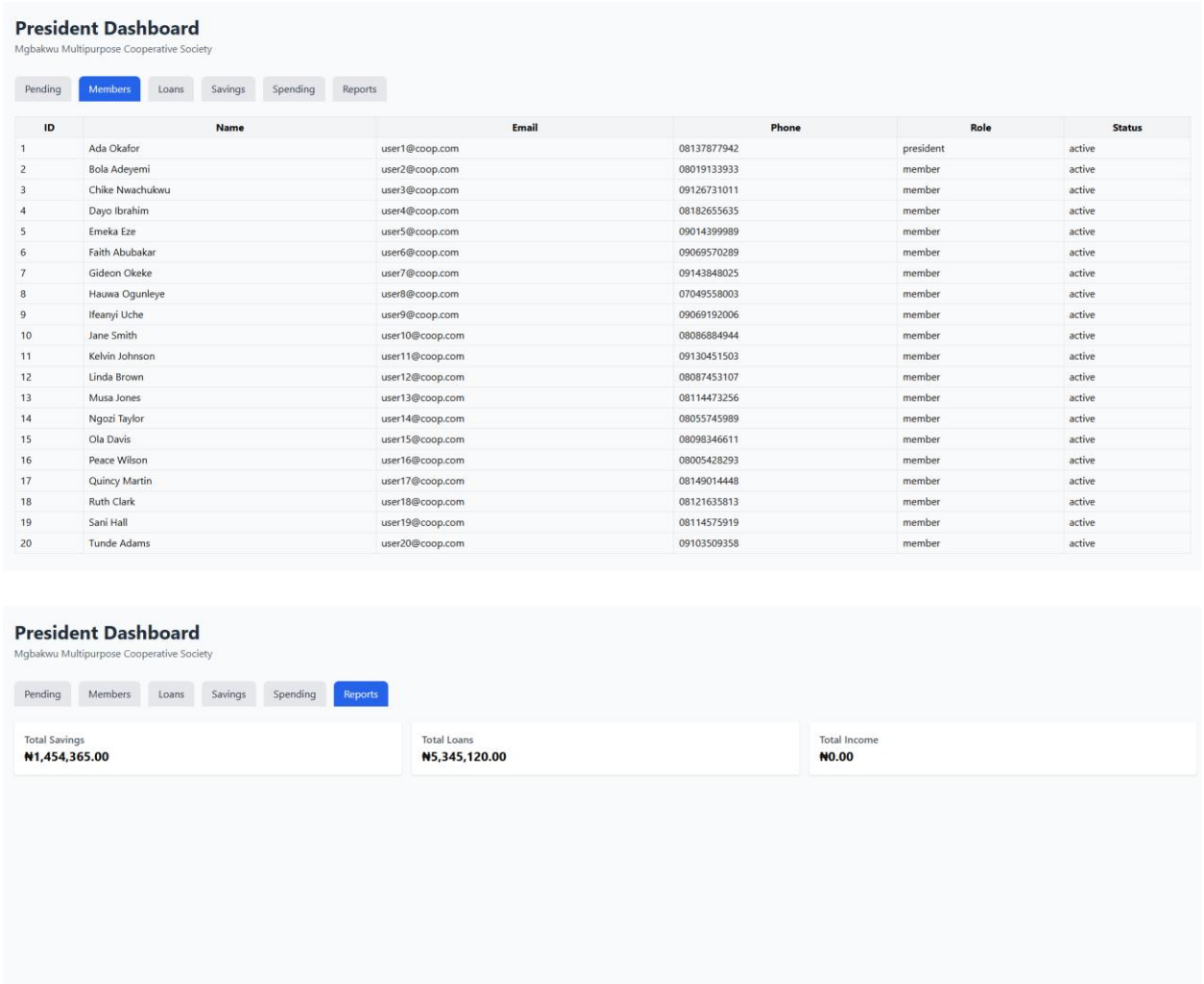
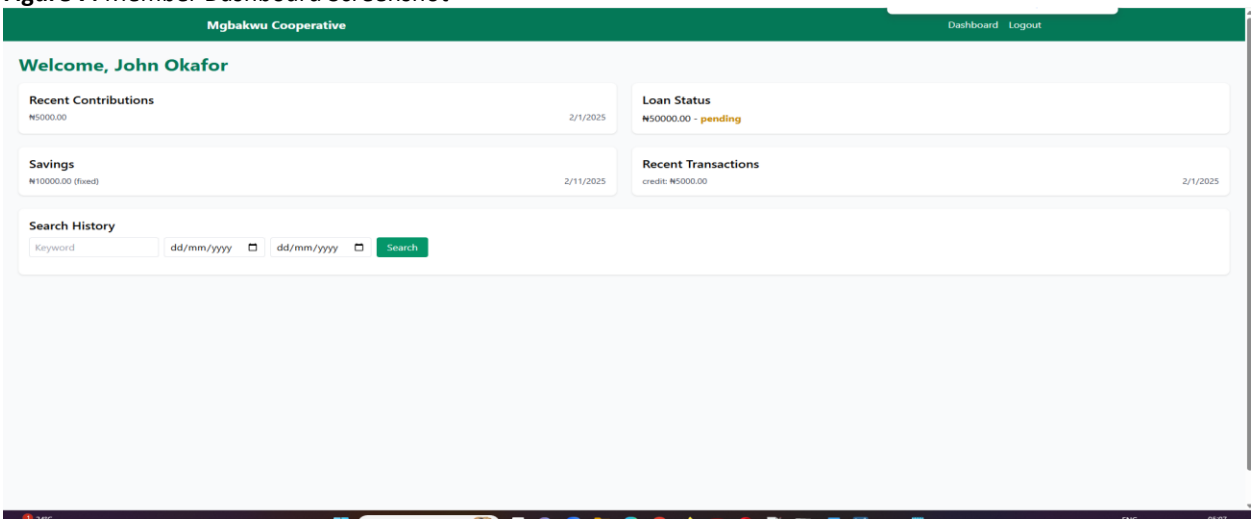


Figure 7: Member Dashboard Screenshot



Conclusion and Recommendations

The web-based cooperative management system successfully automates core operations, improves accuracy, and enhances transparency.

Limitations:

i. Web-based only; no mobile support.
ii. SMS/email notifications not integrated.
iii. Loan approval workflow is basic.
iv. Payment gateway integration missing.

Recommendations

i. Develop Android/iOS applications for easy access.
ii. Integrate SMS/email notification system.
iii. Add payment gateway for online contributions and loan repayments.
iv. Introduce machine learning for intelligent loan scoring.
v. Enhance reporting with graphs, charts, and offline functionality (Ezeanya & Umeh, 2020a; Sommerville, 2020; Ezeanya & Umeh, 2020b).

This system offers Nigerian cooperative societies a reliable, transparent, and efficient platform for daily operations.

References

Akinbode, A. O., & Adewale, O. J. (2021a). The role of cooperative societies in promoting financial inclusion in Nigeria. *Journal of Cooperative Development Studies, 9*(2), 45–53.

Akinbode, A. O., & Adewale, O. J. (2021b). Digitization and cooperative financial services in Nigeria. *Journal of Cooperative Studies, 54*(2), 33–45.

Banker, J. (2022). *Mastering React: Building modern web applications with reusable components*. O'Reilly Media.

Chukwuma, I. C. (2019). Challenges of record management in Nigerian cooperative societies. *Nigerian Journal of Management and Social Sciences, 7*(1), 112–123.

Ezeanya, K., & Umeh, F. (2020b). Manual recordkeeping and financial inefficiencies in cooperative societies. *International Journal of Social Finance, 6*(3), 22–31.

Ezeanya, S. C., & Umeh, P. O. (2020a). ICT adoption and its impact on cooperative society management in developing economies. *International Journal of Information Systems and Technology, 14*(3), 201–210.

Nwachukwu, L. A., & Okafor, C. I. (2022a). Digital transformation of cooperative societies: A case study of community-based finance in Anambra State. *Journal of Applied Management Research, 18*(4), 89–102.

Nwachukwu, P., & Okafor, J. (2022b). Administrative challenges in Nigerian cooperative societies. *Journal of Development Administration, 14*(4), 55–68.

Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill Education.

Sommerville, I. (2020). *Software engineering* (10th ed.). Pearson Education.