SACCO Platform – Project Plan

This document outlines the comprehensive project plan for the development and deployment of the SACCO Platform. It includes project overview, objectives, features, technology stack, implementation phases, governance, risk management, and success criteria.

🎯 1. PROJECT OVERVIEW

The SACCO Platform is a digital cooperative management system designed to replace handwritten records and manual transactions with an online, transparent, and secure system for community savings and loans.
The system will:
- Enable members to save, borrow, repay, and guarantee loans digitally.
- Integrate M-Pesa payments for deposits and repayments.
- Automatically compute interest, penalties, and charges.
- Provide auditable records and reports for management and auditors.
- Foster trust and community accountability through transparency.

🧩 2. PROJECT OBJECTIVES

1. Digitize member savings & loans – Replace manual black books with a digital ledger.
2. Automate loan eligibility and guarantor flow – Simplify the 6-month savings rule and guarantor approval.
3. Enable seamless mobile payments – Support M-Pesa deposits and loan repayments directly in-app.
4. Improve transparency and trust – Provide real-time balances, audit logs, and QR-verified receipts.
5. Enhance reporting & oversight – Management dashboards and automated statements.

⚙️ 3. SYSTEM FEATURES (MVP + PHASE 2)

Phase 2 Enhancements:
- Automated interest & penalties (BullMQ)
- M-Pesa webhook integration (C2B, B2C, STK Push)
- SMS/Email notifications
- Management dashboards & reconciliations
- Dividend computations & multi-branch support.

🧱 4. TECHNOLOGY STACK

Frontend: *Next.js + React + TailwindCSS + ShadCN UI*
Backend: *NestJS + Prisma*
Database: *PostgreSQL (Supabase)*
Auth & Security: *Supabase Auth + JWT + RLS*
Payments: *M-Pesa APIs (C2B/B2C/STK)*
Storage: *Supabase Buckets*
Workers: *BullMQ + Redis (Upstash)*
Monitoring: *Sentry + OpenTelemetry*
Deployment: *Vercel (frontend) + Render (backend)*
Version Control: *GitHub*

🧠 5. PROJECT PHASES & TIMELINE (6 MONTHS)

Phase 0 – Inception & Architecture (Weeks 1–2): Environment setup.

Phase 1 – Member & Savings Modules (Weeks 3–6): Registration, deposits, receipts.

Phase 2 – Loans & Guarantorship (Weeks 7–11): Loan lifecycle and guarantor workflows.

Phase 3 – Automation & Notifications (Weeks 12–16): Background jobs and alerts.

Phase 4 – Dashboards & Go-Live (Weeks 17–20): Pilot deployment.

Phase 5 – Post-Launch Support (Weeks 21–24): Optimization & scaling.

📊 6. GOVERNANCE & RESPONSIBILITIES

- Product Lead: Define requirements and oversee build.
- Backend Engineer: API design, DB schema, RLS, background jobs.
- Frontend Engineer: Member/Teller dashboards and UX flows.
- SACCO Liaison: Bridge with church committee.
- QA Tester: Execute test cases and verify RLS.
- Treasurer / Manager: Approvals and validation.
- Auditor: Post-launch verification.

🔒 7. SECURITY & COMPLIANCE

Data Privacy: pgcrypto for PII.

Access Control: Supabase RLS.

Auditability: Database triggers for logging changes.

Authentication: JWT + 2FA.

Backups: Daily automatic backups.

Compliance: Align with SASRA SACCO standards.

⚠️ 8. RISK REGISTER

- Data import errors: Validate CSVs and test parallel runs.
- User resistance: Provide training and pilot onboarding.
- M-Pesa downtime: Implement retry queues and reconciliation.
- Cost overruns: Maintain MVP focus and scope control.
- Security vulnerabilities: Conduct audits and key rotations.

📈 9. KEY PERFORMANCE INDICATORS

1. Digital onboarding ≥ 90%
2. Transaction time ≤ 30s
3. Loan approval ≤ 48hrs
4. Uptime ≥ 99.5%
5. Audit ratio 100:0
6. M-Pesa match ≥ 98%
7. Arrears reduction ≥ 20%

💵 10. RESOURCE & BUDGET ESTIMATE

Supabase – 3,000 KES

Vercel – 3,000 KES

Render – 2,500 KES

Upstash Redis – 3,000 KES

SMS Gateway – 1,500 KES
M-Pesa API – 2,000 KES
Miscellaneous – 1,000 KES
Total ≈ 16,000 KES/month.

🧩 11. QUALITY ASSURANCE STRATEGY

Unit Tests – Jest
Integration Tests – Supertest + Prisma
E2E – Playwright
RLS Tests – Supabase policies
UAT – Pilot users
Code Quality – ESLint, Prettier

📦 12. DEPLOYMENT PIPELINE

Dev: Local Docker
Staging: Render/Vercel
Production: Supabase DB
Monitoring: Sentry + uptime checks

🪜 13. IMPLEMENTATION CHECKLIST

Before Code: finalize ERD, mockups.
During Dev: commit small features, test.
Before Go-Live: audit trail, backup test, M-Pesa integration verification.

🎯 14. SUCCESS CRITERIA

✅ All members are transacting digitally.
✅ Real-time loan tracking.
✅ Accurate statements.
✅ No manual ledgers.
✅ Auditable transparency.

🏁 15. NEXT STEPS (ACTION PLAN)

1. Setup Supabase + GitHub.
2. Implement schema + RLS.
3. Scaffold backend/frontend.
4. Build MVP.
5. Deploy staging.
6. Pilot test.
7. Train users.
8. Launch.
9. Collect feedback.
10. Scale rollout.