OLD TIMERS SAVINGS GROUP

DIGITAL RECORDS MANAGEMENT SYSTEM

SYSTEM SPECIFICATION DOCUMENT

Version: 1.0
Date: December 15, 2024
Prepared By: Luyima Alex Cedric (Secretary)
Approved By: Executive Committee
Project Status: Approved for Development

EXECUTIVE SUMMARY

This document specifies the requirements for a web-based Digital Records Management System for Old Timers Savings Group. The system will digitize and automate record-keeping, enhance transparency, and improve operational efficiency. Development is authorized per Resolution 11 of the December 14, 2024 meeting.

Key Objectives:

- Automate contribution tracking and receipt generation
- Enable member self-service access to records
- Support loans management workflow
- Facilitate welfare request and approval processes
- Provide real-time financial reporting
- Ensure data security and audit trails

Deployment: Free PythonAnywhere account with the following constraints:

- **Storage:** 512 MB disk space
- **Daily CPU:** 100 seconds/day
- **Web traffic:** One web app only
- **Database:** SQLite (included)
- **Custom domain:** Not available (uses username.pythonanywhere.com)

---

1. SYSTEM OVERVIEW

1.1 Purpose

To provide a centralized, secure, and accessible digital platform for managing all Old Timers Savings Group operations including contributions, loans, welfare support, and financial reporting.

1.2 Scope

In Scope:

- Member registration and authentication
- Contribution recording and tracking
- Automatic receipt generation
- Member statements and contribution history
- Next of kin management (Executive only)
- Meeting attendance tracking
- Meeting minutes storage
- Welfare requests (bereavement, medical support)
- Loans management (applications, approvals, tracking, repayments)
- Financial reports and dashboards
- WhatsApp notification integration
- Audit trails
- Data backup and export

Out of Scope:

- Mobile native applications (web-responsive only)
- SMS notifications (cost prohibitive)
- Email notifications (may add later if needed)
- Bank API integration
- Automated payment processing
- Biometric authentication
- Multi-language support (English only)

1.3 Users

Total Expected Users: 15 maximum (7 current, 8 potential)

User Categories:

1. **Super Admin** (1): System administrator with full access
2. **Executive Members** (3): Chairperson, Secretary, Treasurer
3. **Auditors** (2): Quarterly audit access
4. **Regular Members** (9-14): View-only access to personal records

1.4 Technology Stack

Backend:

- **Framework:** Flask 2.3+ (Python 3.10)
- **Database:** SQLite 3
- **ORM:** SQLAlchemy 2.0+
- **Authentication:** Flask-Login
- **Forms:** Flask-WTF
- **Password Hashing:** Werkzeug Security

Frontend:

- **HTML5/CSS3**
- **JavaScript:** Vanilla JS (no heavy frameworks to save space)
- **UI Framework:** Bootstrap 5.3 (CDN - saves disk space)
- **Icons:** Bootstrap Icons (CDN)
- **PDF Generation:** ReportLab (for receipts and reports)

Hosting:

- **Platform:** PythonAnywhere Free Tier
- **URL:** oldtimerssavings.pythonanywhere.com (or similar)
- **HTTPS:** Enabled by default
- **Database Location:** /home/username/mysite/database.db

External Dependencies (via CDN - no disk space used):

- Bootstrap 5.3.2
- Bootstrap Icons
- Chart.js (for financial charts)

1.5 Compliance with PythonAnywhere Free Tier Limits

Storage Management (512 MB limit):

- Flask app + dependencies: ~50 MB
- SQLite database (estimated): ~10 MB (50,000 records = ~5 MB)
- Uploaded documents (medical, death certificates): ~200 MB max
- System logs: ~10 MB (rotating logs)
- PDF receipts stored: ~50 MB
- **Total estimated:** ~320 MB (leaves 192 MB buffer)

CPU Management (100 seconds/day):

- Average page load: 0.1-0.3 seconds CPU
- Expected daily requests: ~100-200 (well within limit)
- Heavy operations (reports, PDF generation): Cached when possible
- Background tasks: Minimal (no scheduled jobs initially)

Traffic Management:

- Single web app limit: Compliant
- Expected concurrent users: 1-3 (small group)
- No high-traffic features planned

Database Size Projections:

- 15 members × 12 months × 5 years = 900 contribution records (~0.5 MB)
- 100 meetings × 5 years = 500 meeting records (~1 MB)
- 50 loan records × 5 years = 250 records (~0.5 MB)

- 50 welfare requests × 5 years = 250 records (~2 MB with documents)
- **Total 5-year projection:** ~15 MB (well within limits)

---

2. FUNCTIONAL REQUIREMENTS

2.1 User Authentication & Authorization

2.1.1 User Registration

FR-001: System shall allow Super Admin to create user accounts with:

- Full name
- Phone number (unique, serves as username)
- Email address (optional)
- Initial password (must be changed on first login)
- Role assignment (Executive, Auditor, Member)
- Member number (auto-generated: OT-001, OT-002, etc.)

FR-002: Password requirements:

- Minimum 8 characters
- At least one number
- Case-insensitive (for ease of use)

FR-003: First-time login flow:

- User enters phone number and initial password
- System forces password change
- User sets new password
- System logs successful first login

2.1.2 User Login

FR-004: Login page shall:

- Accept phone number and password
- Display "Remember Me" option (7-day session)
- Show "Forgot Password" link
- Redirect to appropriate dashboard based on role

FR-005: Session management:

- Active sessions: 30 minutes of inactivity
- "Remember Me": 7 days
- Force logout after password change
- Concurrent login detection (warn user)

2.1.3 Password Reset

FR-006: Password reset process:

- User requests reset via phone number
- Executive member verifies identity (offline)
- Executive generates temporary password
- User logs in and forced to change password

Note: No email-based reset due to unreliable email access in Uganda

2.1.4 Role-Based Access Control

Role Permissions Matrix:

Feature	Super Admin	Executive	Auditor	Member
View own records	✅	✅	✅	✅
Record contributions	✅	✅ (Treas/Sec)	❌	❌
Generate receipts	✅	✅ (Treasurer)	❌	❌
View all member records	✅	✅	✅	❌
Manage next of kin	✅	✅	❌	❌
Record attendance	✅	✅ (Secretary)	❌	❌
Upload minutes	✅	✅ (Secretary)	❌	❌
Approve welfare requests	✅	✅ (Chairman)	❌	❌
Process welfare payments	✅	✅ (Treas/Sec)	❌	❌
Manage loans	✅	✅	❌	❌
Generate reports	✅	✅	✅	✅ (limited)
View financial summary	✅	✅	✅	✅
Audit access	✅	✅	✅	❌
User management	✅	❌	❌	❌
System settings	✅	❌	❌	❌

2.2 Member Management

2.2.1 Member Profile

FR-007: Each member profile shall contain:

Personal Information:

- Full legal name
- Member number (auto-generated)
- National ID number
- Date of birth

- Gender
- Phone number (primary)
- Phone number (secondary, optional)
- Email address (optional)
- Physical address
- Occupation

Membership Information:

- Date joined
- Membership status (Active, Inactive, Suspended, Expelled)
- Membership fee paid (Yes/No, Date, Receipt No.)
- Total contributions to date
- Number of consecutive months contributed
- Qualified for benefits (Auto-calculated: Yes/No)
- Last contribution date

Next of Kin Information (Executive Only):

- Primary next of kin full name
- Relationship
- National ID
- Phone number(s)
- Physical address
- Distribution percentage
- Alternative next of kin full name
- Relationship
- Phone number
- Distribution percentage

System Information:

- Created date
- Last updated date
- Created by
- Last login date

FR-008: Member search functionality:

- Search by name, phone number, member number
- Filter by status (Active, Inactive, Suspended, Expelled)
- Filter by qualification for benefits
- Sort by join date, last contribution, total contributed

2.2.2 Member Status Management

FR-009: System shall automatically update member status based on:

Active:

- Has contributed in the last month OR
- Has valid reason for absence (recorded in system)

Inactive:

- No contribution for 1-2 months without valid reason
- Can still access system
- Not qualified for benefits until regularized

Suspended:

- No contribution for 3 consecutive months (per Resolution 8)
- Limited system access (view own records only)
- Cannot apply for loans or welfare
- Status notification sent via WhatsApp

Expelled:

- No contribution for 6 consecutive months (per Resolution 8) OR
- Dishonest conduct proven and recorded
- 80% refund calculated and flagged for payment
- System access revoked
- All data archived but not deleted (audit trail)

FR-010: Manual status override:

- Executive can manually change status with reason
- All status changes logged with timestamp and user
- Notification sent to affected member

2.3 Contributions Management

2.3.1 Recording Contributions

FR-011: Contribution entry form (Treasurer/Secretary access):

Required Fields:

- Member (dropdown selection)
- Amount (default: UGX 100,000)
- Payment date
- Payment method (Cash, Mobile Money, Bank Transfer)
- Transaction/Reference ID
- Contribution month (e.g., December 2024)
- Recorded by (auto-filled from logged-in user)

Optional Fields:

- Notes/remarks
- Proof of payment upload (image, max 2 MB)

FR-012: System validations:

- Prevent duplicate contributions for same member + month
- Warn if amount differs from standard UGX 100,000
- Warn if payment date is more than 30 days old
- Alert if member is suspended/expelled
- Require Transaction ID for mobile money/bank payments

FR-013: Auto-receipt generation:

- Generate unique receipt number (format: OT-YYYY-MM-NNNN)
- Include all contribution details
- Member name, amount, date, purpose, payment method
- Digital signature (system-generated)
- Download as PDF
- Email option (if member has email)
- Store in database linked to contribution record

FR-014: Bulk contribution entry:

- Upload CSV file with columns: Member Number, Amount, Date, Reference
- System validates each row
- Display preview before final save
- Generate bulk receipts (ZIP download)
- Error report for failed entries

2.3.2 Contribution History

FR-015: Member contribution view (accessible to member):

- List all contributions with dates, amounts, receipts
- Show total contributed to date
- Show months contributed (consecutive count)
- Show qualification status for benefits
- Download statement as PDF
- Filter by date range
- Search by receipt number

FR-016: Admin contribution view (Executive/Auditor access):

- View all members' contributions
- Filter by member, date range, payment method
- Sort by amount, date, member
- Export to Excel/CSV
- Summary statistics (total collected, average per member, etc.)

2.3.3 Payment Tracking Dashboard

FR-017: Monthly payment tracker (Executive access):

- Grid view: Members (rows) × Months (columns)
- Color coding:
    - ✅ Green: Paid
    - ⚠ Yellow: Partial/pending
    - ❌ Red: Not paid
    - □ Gray: Excused absence
- Click cell to see payment details
- Send reminder to unpaid members (WhatsApp)
- Export monthly report

FR-018: Payment reminders:

- Automatic flagging of unpaid contributions after 5th of month
- Generate reminder list for Treasurer
- WhatsApp message template with payment details
- Track reminder sent date

2.4 Financial Management

2.4.1 Financial Dashboard

FR-019: Executive dashboard shall display:

Key Metrics (Real-time):

- Total contributions collected (all-time)
- Total contributions this month
- Total contributions this quarter
- Total contributions this year
- Current bank balance (manually updated)
- Total loans outstanding
- Total welfare payments made
- Net group savings

Charts/Graphs:

- Monthly contributions trend (line chart, last 12 months)
- Member payment status (pie chart: Paid/Unpaid this month)
- Contributions by member (bar chart, top 10)
- Loan portfolio status (paid vs outstanding)

Quick Stats:

- Total active members
- Total suspended members
- Members qualified for benefits
- Pending welfare requests
- Pending loan applications

2.4.2 Financial Reports

FR-020: System shall generate the following reports:

Monthly Contribution Summary:

- List all members with payment status for selected month
- Total collected, total expected, collection rate %
- Outstanding members list
- Export to PDF/Excel
- Per Resolution: Required report for quarterly meetings

Individual Member Statement:

- Member name and details
- All contributions with dates and amounts
- Total contributed
- Consecutive months count
- Qualification status
- Loans taken and repayment status
- Welfare received (if any)
- Current balance/equity in group
- Period covered
- Generated date
- Per Resolution: Member can download anytime

Financial Summary Report:

- Total collections (by period)
- Total payments/disbursements (by category)
    - Welfare payments
    - Loan disbursements
    - Operating expenses
    - Other
- Current balance
- Breakdown by month/quarter/year
- Per Resolution: Required for meetings and audits

Quarterly Audit Report:

- All transactions for the quarter
- Opening and closing balances
- Reconciliation with bank statements
- Discrepancies flagged
- Auditor comments section
- Per Resolution 15: Required every 3 months

Annual Report:

- Complete financial overview for the year

- Member statistics (joined, left, active)
- Total collections vs targets
- Loans performance
- Welfare disbursements
- Year-over-year comparison
- Executive summary

2.4.3 Bank Reconciliation

FR-021: Bank reconciliation module:

- Manual entry of bank statement transactions
- Match bank transactions with system records
- Flag unmatched items
- Calculate reconciled balance
- Generate reconciliation report
- Auditor approval workflow

FR-022: Cash flow tracking:

- Record all money in (contributions, loan repayments, interest)
- Record all money out (welfare, loans, expenses)
- Running balance calculation
- Monthly cash flow statement
- Variance analysis (expected vs actual)

2.5 Welfare Management

2.5.1 Welfare Request Submission

FR-023: Member welfare request form:

Request Types:

- Bereavement (Spouse, Child, Parent)
- Medical Support (Member only)
- Celebration (Wedding, Introduction, Nikah) - Information only

Required Information:

- Request type
- Affected person (relationship if bereavement)
- Date of incident/event
- Description
- Amount requested (auto-filled for bereavement: UGX 500,000)
- Supporting documents upload (max 5 files, 10 MB total)
    - Death certificate (for bereavement)
    - Medical reports/bills (for medical support)
    - Event invitation (for celebrations)

FR-024: Document upload:

- Accepted formats: PDF, JPG, PNG
- Preview uploaded documents
- Delete/replace before submission
- Timestamp and uploaded by tracking

FR-025: Submission workflow:

- Member submits request
- System validates (member qualified? Active status?)
- Notification sent to Secretary (WhatsApp)
- Request appears in Secretary's pending queue
- Submission confirmation to member

2.5.2 Welfare Request Review & Approval

FR-026: Secretary review process (per Resolution 12):

- View all pending requests
- Review uploaded documents
- Verify member qualification (5 consecutive months)
- Add review notes
- Options:
    - Forward to Chairperson (approved for review)
    - Return to member (request more info)
    - Reject (with reason)
- All actions logged with timestamp

FR-027: Chairperson approval process (per Resolution 12):

- View requests forwarded by Secretary
- Review all documents and notes
- Options:
    - Approve (specify amount if different from request)
    - Reject (with reason)
    - Request emergency meeting (for special cases)
- Approval notification sent to Treasurer & Secretary
- Member notified of decision (WhatsApp)

FR-028: Emergency meeting pathway (per Resolution 4):

- Chairperson flags request for emergency meeting
- System sends meeting notification to all members
- After meeting, Chairperson records decision
- Amount approved and payment authorized

2.5.3 Welfare Payment Processing

FR-029: Payment voucher generation (per Resolution 14):

- Auto-generate payment voucher after Chairperson approval
- Include: Beneficiary, amount, purpose, approvals
- Chairperson approval: ✅ (timestamp)
- Secretary signature field: ☐
- Treasurer signature field: ☐
- Print voucher for physical signatures

FR-030: Payment recording:

- Treasurer/Secretary records payment made
- Payment date
- Payment method (Cash, Mobile Money, Bank Transfer)
- Transaction reference
- Both Treasurer and Secretary must confirm
- Beneficiary acknowledgment (optional upload)
- Payment voucher filed digitally

FR-031: Welfare payment tracking:

- View all welfare payments (Executive/Auditor)
- Filter by type, member, date range, status
- Summary: Total paid by category
- Outstanding approved but unpaid requests
- Payment history per member

2.5.4 Special Case: Member Death

FR-032: When member dies (per Resolution 3):

- Executive records member death in system
- System automatically calculates total payout:
  - Member's total contributions: UGX X
  - Fixed bereavement: UGX 500,000
  - Individual condolences: UGX Y (manually entered)
  - **Total:** UGX (X + 500,000 + Y)
- Generate payout summary
- Payment voucher created
- Payment to registered next of kin
- Member status changed to "Deceased"
- Account archived (view-only)

2.6 Loans Management

2.6.1 Loan Application

FR-033: Member loan application form (per Resolution 5):

Loan Details:

- Amount requested
- Purpose of loan
- Preferred repayment period (max 2 months)
- Security type (Collateral OR Guarantors)

If Collateral:

- Asset type (Car, Land, Property, Other)
- Asset description
- Estimated value
- Upload documents (logbook, land title, etc.)

If Guarantors:

- Guarantor 1: Select from active members
- Guarantor 2: Select from active members
- Both must be active members (system validates)

Application Requirements:

- Member must be active
- Member must have contributed 5+ consecutive months
- Member must have no outstanding defaulted loans
- Maximum loan: To be determined by Executive

FR-034: Guarantor approval workflow:

- System sends notification to selected guarantors
- Guarantor logs in and views loan request
- Guarantor can:
  - Accept guarantee
  - Decline (with optional reason)
  - Request more information
- Both guarantors must accept before proceeding to Executive
- If declined, member can select new guarantors

2.6.2 Loan Review & Approval

FR-035: Executive Committee review (per Resolution 5):

- View all pending loan applications
- See member contribution history
- View collateral/guarantor details
- Discussion notes field
- Approval decision:
  - Approve (full amount or modified amount)
  - Approve with conditions
  - Defer (need more information)
  - Reject (with reason)
- All three Executive members must approve

- Record individual votes/comments

FR-036: Loan agreement generation:

- Auto-generate loan agreement document
- Include:
    - Borrower details
    - Loan amount and terms (5% monthly interest)
    - Repayment schedule (calculated)
    - Security details
    - Guarantor details (if applicable)
    - Terms and conditions
    - Signature blocks
- Download as PDF
- Print for physical signatures
- Upload signed copy back to system

2.6.3 Loan Disbursement

FR-037: Disbursement process (per Resolution 14):

- After loan approval, generate disbursement voucher
- Chairperson approves (digital signature)
- Treasurer and Secretary process payment
- Record disbursement:
    - Disbursement date
    - Payment method
    - Transaction reference
    - Both signatories confirm
- Loan status changes to "Active"
- Repayment schedule activated

FR-038: Loan repayment tracking:

- Display repayment schedule (principal + interest)
- Month 1 payment: Principal × 5% interest + portion of principal
- Month 2 payment: Remaining balance + 5% interest
- Record each repayment:
    - Payment date
    - Amount paid
    - Payment method
    - Transaction reference
- Calculate remaining balance
- Flag overdue payments (red alert)

FR-039: Loan portfolio dashboard:

- Total loans disbursed
- Total loans outstanding
- Total loans fully repaid

- Total interest earned
- Defaulted loans (overdue >30 days)
- Repayment rate %
- Loans by member
- Average loan size

2.6.4 Loan Guarantor Liability

FR-040: When loan defaults:

- System flags loan as defaulted (>30 days overdue)
- Notifications sent to:
  - Borrower (final demand)
  - Guarantors (liability notice)
  - Executive Committee
- Record recovery actions taken
- If recovered from guarantors, record source
- Update guarantor liability records

2.7 Meetings Management

2.7.1 Meeting Scheduling

FR-041: Meeting creation (Secretary access):

- Meeting type (Regular, Emergency, Annual General)
- Date and time
- Venue
- Agenda items (multiple entries)
- Expected attendees (all members by default)
- Meeting called by (auto: Secretary)
- Send notification (WhatsApp integration)

FR-042: Meeting notifications:

- Generate meeting notice with:
  - Date, time, venue
  - Agenda
  - Quorum reminder (5 members required)
  - RSVP option
- Send via WhatsApp to all members
- Track delivery status
- Reminder 24 hours before meeting

2.7.2 Meeting Attendance

FR-043: Digital attendance register:

- List all expected members
- Checkboxes: Present / Absent / Excused

- Arrival time (optional)
- Digital signature option (members sign on device)
- Quorum status (auto-calculated: need 5)
- Warning if quorum not met
- Export attendance list

FR-044: Late arrivals / early departures:

- Record time of arrival if late
- Record time of departure if early
- Notes field for reasons
- Affects attendance statistics

2.7.3 Meeting Minutes

FR-045: Minutes upload and storage (Secretary access):

- Meeting selection (from scheduled meetings)
- Upload minutes document (PDF, Word, max 5 MB)
- Or use built-in minutes editor:
  - Agenda items
  - Discussions (rich text)
  - Resolutions (numbered automatically)
  - Action items (assigned to members)
  - Next meeting date
- Chairperson approval required
- Published to all members after approval

FR-046: Minutes access:

- All members can view published minutes
- Search minutes by date, keywords
- Download as PDF
- View resolutions summary
- Track action items assigned to them

FR-047: Action items tracking:

- Extract action items from minutes
- Assign to specific members
- Set deadlines
- Track completion status
- Reminder notifications before deadline
- Report on pending action items

2.8 Audit & Compliance

2.8.1 Audit Trail

FR-048: System shall log all critical actions:

- User login/logout
- Contribution recorded/edited/deleted
- Payment approved/processed
- Loan application/approval/disbursement/repayment
- Welfare request/approval/payment
- Member status changes
- User account changes
- System settings changes

Log Entry Fields:

- Timestamp (date and time)
- User (who performed action)
- Action type
- Entity affected (member, contribution, loan, etc.)
- Old value (if edit)
- New value
- IP address
- Device type (mobile/desktop)

FR-049: Audit log access:

- Auditors and Super Admin can view all logs
- Filter by user, date range, action type
- Search by entity (e.g., specific member)
- Export to Excel/CSV
- Cannot be edited or deleted (immutable)

2.8.2 Auditor Access

FR-050: Auditor dashboard (per Resolution 10 & 15):

- Read-only access to all financial records
- View all contributions, payments, loans
- Access to bank reconciliation
- Download all financial reports
- Access audit trail
- Generate custom audit queries
- Add audit notes and findings
- Submit quarterly audit report

FR-051: Quarterly audit workflow:

- System reminds auditors 2 weeks before quarter end
- Auditors review records for the quarter
- Complete audit checklist:
  - All contributions properly receipted ✅
  - All payments have vouchers and approvals ✅
  - Bank reconciliation matches ✅

    - Loan repayments up to date ✅
    - No unauthorized transactions ✅
- Enter findings and recommendations
- Generate audit report
- Submit to Executive Committee
- Present at quarterly meeting

2.8.3 Compliance Monitoring

FR-052: Automated compliance checks:

- Monthly: Flag members with >3 months no contribution (suspension)
- Monthly: Flag members with >6 months no contribution (expulsion)
- Daily: Flag loans overdue >30 days (default)
- Weekly: Check quorum for scheduled meetings
- Monthly: Verify all welfare payments have approvals
- Quarterly: Audit report submission reminder

FR-053: Compliance dashboard:

- List all compliance issues
- Severity (High, Medium, Low)
- Status (Open, In Progress, Resolved)
- Assigned to (responsible person)
- Due date for resolution
- Action taken
- Resolution notes

2.9 Notifications & Communication

2.9.1 WhatsApp Integration

FR-054: WhatsApp notification types:

- Payment reminders (due date approaching)
- Payment confirmation (receipt generated)
- Welfare request status updates
- Loan application status updates
- Meeting notifications and reminders
- Account status changes (suspended, qualified)
- System announcements

FR-055: WhatsApp integration method:

- **Phase 1 (MVP):** Manual process
    - System generates message template
    - Executive copies and pastes to WhatsApp
    - Track "notification sent" status manually
- **Phase 2 (Future):** WhatsApp Business API

    - Automated sending (requires paid API)
    - Delivery confirmation
    - Not included in free PythonAnywhere version

FR-056: Message templates:

- Standardized messages for each notification type
- Personalization (member name, amount, dates)
- Professional tone
- Include relevant links to system
- Character limit: 1000 (WhatsApp limit)

2.9.2 In-App Notifications

FR-057: Notification center:

- Bell icon in navigation bar
- Unread count badge
- Click to view all notifications
- Mark as read/unread
- Delete notification
- Notification types:
    - System announcements
    - Personal updates (contribution recorded, loan approved)
    - Reminders (payment due, meeting scheduled)
    - Alerts (account suspended, loan overdue)

FR-058: Notification persistence:

- Store last 90 days of notifications
- Archive older notifications
- Search notifications by keyword/date
- Filter by type

2.10 Reports & Analytics

2.10.1 Standard Reports

FR-059: Pre-built reports accessible to all roles:

For Members:

- Personal contribution statement
- Personal transaction history
- Personal loan summary

For Executive/Auditors:

- Monthly contribution summary (per Resolution, required for meetings)
- Individual member statements (per Resolution)

- Financial summary (per Resolution, required for meetings)
- Quarterly audit report (per Resolution 15)
- Payment defaulters report
- Welfare disbursements report
- Loans portfolio report
- Member activity report
- Attendance summary report

FR-060: Report generation:

- Select report type
- Choose parameters (date range, members, etc.)
- Preview on screen
- Download as PDF
- Export to Excel/CSV
- Email report (if configured)
- Schedule recurring reports (future enhancement)

2.10.2 Custom Analytics

FR-061: Analytics dashboard (Executive access):

- Customizable widgets
- Drag-and-drop interface
- Chart types: Line, Bar, Pie, Area
- Data sources: Contributions, Loans, Welfare, Attendance
- Save custom dashboard layouts
- Export dashboard as image

FR-062: Key performance indicators (KPIs):

- Collection rate (actual vs expected %)
- Member retention rate
- Loan repayment rate
- Average contribution per member
- Growth rate (month-over-month)
- Welfare support ratio (paid vs collected)
- Meeting attendance rate
- Active members %

2.11 Data Management

2.11.1 Data Import

FR-063: Historical data import (one-time):

- Contributions from July 2024 to present
- Member information and next of kin
- Meeting minutes (2 meetings)
- Upload via Excel template

- Data validation before import
- Preview import results
- Confirm and execute import
- Generate import summary report

FR-064: Import templates:

- Downloadable Excel templates for:
    - Members
    - Contributions
    - Meetings
    - Next of kin
- Clear instructions and examples
- Data format validation rules
- Error checking before upload

2.11.2 Data Export

FR-065: Export functionality:

- Export entire database (Super Admin only)
- Export by module (contributions, loans, etc.)
- Export format: Excel, CSV, JSON
- Date range filters
- Member filters
- Include/exclude deleted records
- Password-protected export files (optional)

2.11.3 Data Backup

FR-066: Backup strategy (within 512 MB limit):

- Automated weekly backup (scheduled task)
- Backup location: PythonAnywhere file storage
- Backup retention: Last 4 backups (1 month)
- Backup includes:
    - Database file (complete copy)
    - Uploaded documents (incremental)
    - System logs
- Compressed ZIP format to save space
- Backup size monitoring (alert if >100 MB)

FR-067: Manual backup:

- Super Admin can trigger backup anytime
- Before major system updates
- Before bulk data operations
- Download backup to local computer
- Restore from backup functionality

2.11.4 Data Security

FR-068: Security measures:

- Database file stored outside web-accessible directory
- Encrypted password storage (Werkzeug SHA256)
- HTTPS enforced (PythonAnywhere default)
- SQL injection prevention (SQLAlchemy ORM)
- XSS protection (Flask auto-escaping)
- CSRF protection (Flask-WTF)
- Session security (secure cookies, HTTP-only)

FR-069: Sensitive data protection:

- Next of kin info: Executive access only
- National ID numbers: Masked display (show last

Continue

4:59 AM

4 digits)

- Phone numbers: Masked for non-executive users
- Financial amounts: Executive and member (own records) only
- Audit logs: Auditor and Super Admin only

FR-070: Data retention policy:

- Active member data: Retained indefinitely
- Expelled member data: Archived (read-only), retained 7 years
- Deceased member data: Archived (read-only), retained indefinitely
- Transaction data: Retained indefinitely (legal requirement)
- System logs: Retained 1 year, then archived
- Deleted records: Soft delete (marked as deleted, not purged)

2.12 System Administration

2.12.1 User Management

FR-071: Super Admin capabilities:

- Create new user accounts
- Edit user information
- Reset user passwords
- Change user roles
- Activate/deactivate accounts
- View user activity logs
- Bulk user operations

FR-072: User account lifecycle:

- New member joins: Create account
- Member becomes Executive: Upgrade role
- Member suspended: Downgrade access
- Member expelled: Deactivate account
- Member leaves voluntarily: Archive account
- Account reactivation process (if member returns)

2.12.2 System Settings

FR-073: Configurable settings (Super Admin):

- Group information (name, logo, contact)
- Membership fee amount (default: UGX 20,000)
- Monthly contribution amount (default: UGX 100,000)
- Bereavement amount (default: UGX 500,000, per Resolution 2)
- Loan interest rate (default: 5%, per Resolution 5)
- Loan maximum period (default: 2 months, per Resolution 5)
- Quorum requirement (default: 5 members, per Resolution 6)
- Suspension threshold (default: 3 months, per Resolution 8)
- Expulsion threshold (default: 6 months, per Resolution 8)
- Qualification period (default: 5 months, per meeting)
- Expelled member refund % (default: 80%, per Resolution 9)

FR-074: System configuration:

- Email settings (SMTP) - optional
- WhatsApp API settings - future
- Backup schedule
- Session timeout settings
- Date/time format preferences
- Currency display format
- Fiscal year start month

2.12.3 System Monitoring

FR-075: System health dashboard (Super Admin):

- Database size (current / 512 MB limit)
- Storage used (current / 512 MB limit)
- Daily CPU usage (current / 100 seconds limit)
- Active user sessions
- Recent error logs
- System uptime
- Last backup date/time

FR-076: Error logging:

- Log all application errors

- Error details (type, message, stack trace)
- User context (who, when, what page)
- Email alert for critical errors (optional)
- Error log viewer in admin panel
- Export error logs

FR-077: Performance monitoring:

- Page load times
- Database query performance
- Slow query identification
- Resource usage alerts
- Optimization recommendations

---

3. NON-FUNCTIONAL REQUIREMENTS

3.1 Performance

NFR-001: Page load time:

- Average page load: < 2 seconds on 3G connection
- Dashboard load: < 3 seconds
- Report generation: < 5 seconds
- PDF generation: < 10 seconds

NFR-002: Database performance:

- Query response time: < 500ms for standard queries
- Complex reports: < 3 seconds
- Database backup: < 30 seconds

NFR-003: Concurrent users:

- Support minimum 5 concurrent users
- Graceful degradation if exceeded
- Session management optimized

NFR-004: CPU optimization (100 seconds/day limit):

- Efficient queries (minimize database hits)
- Caching frequently accessed data
- Lazy loading for images/documents
- Pagination for large datasets (max 50 records per page)

3.2 Usability

NFR-005: Mobile responsiveness:

- Fully functional on smartphones (320px - 768px)
- Touch-friendly buttons (minimum 44×44px)
- Readable fonts on small screens (minimum 14px)
- Simplified navigation for mobile
- Tested on Android and iOS browsers

NFR-006: User interface:

- Clean, professional design
- Consistent color scheme (group branding)
- Intuitive navigation (max 3 clicks to any feature)
- Clear labels and instructions
- Helpful error messages
- Loading indicators for slow operations

NFR-007: Accessibility:

- Readable text (good contrast ratios)
- Keyboard navigation support
- Screen reader friendly (semantic HTML)
- Alternative text for images
- No reliance on color alone for information

NFR-008: Language and localization:

- English interface
- Ugandan currency format (UGX)
- Date format: DD/MM/YYYY
- East Africa Time (EAT) timezone
- Phone number format: +256 7XX XXX XXX

3.3 Reliability

NFR-009: Uptime:

- Target: 99% uptime (excludes PythonAnywhere maintenance)
- Graceful error handling (no crashes)
- Informative error messages
- Automatic error reporting

NFR-010: Data integrity:

- Transaction consistency (ACID compliance)
- Referential integrity (foreign key constraints)
- Data validation on input
- Backup verification (restore test quarterly)

NFR-011: Fault tolerance:

- Session recovery after network interruption

- Auto-save drafts (forms)
- Retry failed operations (with user confirmation)
- Rollback on transaction failure

3.4 Security

NFR-012: Authentication security:

- Password hashing (SHA256)
- Session management (secure tokens)
- Automatic logout after inactivity (30 minutes)
- Password change on first login
- Failed login attempt tracking (lock after 5 attempts)

NFR-013: Authorization security:

- Role-based access control (RBAC)
- Least privilege principle
- Granular permissions
- Access attempt logging

NFR-014: Data transmission:

- HTTPS enforced (PythonAnywhere provides SSL)
- Secure cookies (HTTP-only, SameSite)
- No sensitive data in URLs
- CSRF token validation

NFR-015: Data storage:

- Encrypted passwords
- Sensitive data access logging
- Soft delete (retain for audit)
- Regular backup encryption

3.5 Maintainability

NFR-016: Code quality:

- Modular design (MVC pattern)
- Clear code comments
- Consistent naming conventions
- Reusable components
- Unit tests for critical functions

NFR-017: Documentation:

- User manual (PDF)
- Administrator guide
- API documentation (if applicable)

- Database schema documentation
- Change log

NFR-018: Updatability:

- Database migration scripts
- Version control (Git)
- Deployment checklist
- Rollback procedures

3.6 Compatibility

NFR-019: Browser support:

- Chrome 90+ (primary)
- Firefox 88+
- Safari 14+
- Edge 90+
- Opera 76+

NFR-020: Device support:

- Desktop (Windows, Mac, Linux)
- Android smartphones (Chrome)
- iPhone (Safari)
- Tablets (responsive layout)

NFR-021: Network compatibility:

- Works on 3G connections (optimized assets)
- Offline detection (inform user)
- Resume operations when online
- Compressed data transfer

3.7 Scalability

NFR-022: User scalability:

- Support 15 users (current limit)
- Expandable to 50 users (if group grows)
- Database design supports growth
- Query optimization for larger datasets

NFR-023: Data scalability:

- Handle 5 years of transaction data (estimated 15 MB)
- Pagination prevents memory overload
- Archive old data (>5 years)
- Database indexing for performance

NFR-024: Storage management (512 MB limit):

- Monitor storage usage
- Alert at 80% capacity (410 MB)
- Automatic compression of old documents
- Archive and download old data
- Regular cleanup of temporary files

---

4. DATABASE DESIGN

4.1 Database Schema Overview

Database: SQLite 3
ORM: SQLAlchemy
Estimated Size: 15 MB (5-year projection)
Location: /home/username/oldtimerssavings/database.db

4.2 Entity Relationship Diagram (ERD)

Core Entities:

1. Users
2. Members
3. NextOfKin
4. Contributions
5. Receipts
6. Meetings
7. Attendance
8. Minutes
9. ActionItems
10. Loans
11. LoanRepayments
12. WelfareRequests
13. WelfarePayments
14. AuditLogs
15. Notifications
16. SystemSettings

4.3 Table Definitions

4.3.1 Users Table

sql

```sql
CREATE TABLE users (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  member_id INTEGER UNIQUE NOT NULL,  -- FK to members
```

```
    username VARCHAR(20) UNIQUE NOT NULL,  -- Phone number
    password_hash VARCHAR(255) NOT NULL,
    role VARCHAR(20) NOT NULL,  -- SuperAdmin, Executive, Auditor, Member
    is_active BOOLEAN DEFAULT TRUE,
    must_change_password BOOLEAN DEFAULT TRUE,
    last_login DATETIME,
    failed_login_attempts INTEGER DEFAULT 0,
    account_locked_until DATETIME,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    updated_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (member_id) REFERENCES members(id)
);
```

4.3.2 Members Table

sql

```
CREATE TABLE members (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    member_number VARCHAR(10) UNIQUE NOT NULL,  -- OT-001
    full_name VARCHAR(100) NOT NULL,
    national_id VARCHAR(20) UNIQUE,
    date_of_birth DATE,
    gender VARCHAR(10),  -- Male, Female
    phone_primary VARCHAR(20) NOT NULL,
    phone_secondary VARCHAR(20),
    email VARCHAR(100),
    physical_address TEXT,
    occupation VARCHAR(100),
    date_joined DATE NOT NULL,
    membership_fee_paid BOOLEAN DEFAULT FALSE,
    membership_fee_date DATE,
    membership_fee_receipt VARCHAR(20),
    status VARCHAR(20) DEFAULT 'Active',  -- Active, Inactive, Suspended, Expelled, Deceased
    total_contributed DECIMAL(15,2) DEFAULT 0.00,
    consecutive_months_paid INTEGER DEFAULT 0,
    last_contribution_date DATE,
    qualified_for_benefits BOOLEAN DEFAULT FALSE,
    suspension_date DATE,
    expulsion_date DATE,
    expulsion_reason TEXT,
    refund_amount DECIMAL(15,2),
```

```sql
    refund_paid BOOLEAN DEFAULT FALSE,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    updated_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    created_by INTEGER,  -- FK to users
    FOREIGN KEY (created_by) REFERENCES users(id)
);
```

4.3.3 NextOfKin Table

sql

```sql
CREATE TABLE next_of_kin (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    member_id INTEGER NOT NULL,
    kin_type VARCHAR(20) NOT NULL,  -- Primary, Alternative
    full_name VARCHAR(100) NOT NULL,
    relationship VARCHAR(50) NOT NULL,
    national_id VARCHAR(20),
    date_of_birth DATE,
    phone_primary VARCHAR(20) NOT NULL,
    phone_secondary VARCHAR(20),
    email VARCHAR(100),
    physical_address TEXT,
    distribution_percentage INTEGER DEFAULT 100,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    updated_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (member_id) REFERENCES members(id) ON DELETE CASCADE
);
```

4.3.4 Contributions Table

sql

```sql
CREATE TABLE contributions (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    member_id INTEGER NOT NULL,
    amount DECIMAL(15,2) NOT NULL,
    payment_date DATE NOT NULL,
    contribution_month VARCHAR(7) NOT NULL,  -- YYYY-MM
    payment_method VARCHAR(20) NOT NULL,  -- Cash, MobileMoney, BankTransfer
    transaction_reference VARCHAR(50),
    notes TEXT,
    proof_of_payment_path VARCHAR(255),  -- File path
```

```sql
    receipt_number VARCHAR(20) UNIQUE,
    recorded_by INTEGER NOT NULL,  -- FK to users
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    updated_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (member_id) REFERENCES members(id),
    FOREIGN KEY (recorded_by) REFERENCES users(id),
    UNIQUE(member_id, contribution_month)
);
```

4.3.5 Receipts Table

sql

```sql
CREATE TABLE receipts (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    receipt_number VARCHAR(20) UNIQUE NOT NULL,
    contribution_id INTEGER,  -- FK to contributions (nullable for other receipts)
    receipt_type VARCHAR(20) NOT NULL,  -- Contribution, MembershipFee, LoanRepayment
    member_id INTEGER NOT NULL,
    amount DECIMAL(15,2) NOT NULL,
    payment_date DATE NOT NULL,
    payment_method VARCHAR(20) NOT NULL,
    description TEXT,
    pdf_path VARCHAR(255),  -- Generated PDF path
    generated_by INTEGER NOT NULL,  -- FK to users
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (contribution_id) REFERENCES contributions(id),
    FOREIGN KEY (member_id) REFERENCES members(id),
    FOREIGN KEY (generated_by) REFERENCES users(id)
);
```

4.3.6 Meetings Table

sql

```sql
CREATE TABLE meetings (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    meeting_type VARCHAR(20) NOT NULL,  -- Regular, Emergency, AnnualGeneral
    meeting_date DATE NOT NULL,
    meeting_time TIME NOT NULL,
    venue VARCHAR(255) NOT NULL,
    agenda TEXT NOT NULL,
    called_by INTEGER NOT NULL,  -- FK to users
```

```sql
    quorum_met BOOLEAN,
    total_attendance INTEGER,
    status VARCHAR(20) DEFAULT 'Scheduled',  -- Scheduled, Completed, Cancelled
    notification_sent BOOLEAN DEFAULT FALSE,
    notification_sent_date DATETIME,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    updated_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (called_by) REFERENCES users(id)
);
```

4.3.7 Attendance Table

sql

```sql
CREATE TABLE attendance (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    meeting_id INTEGER NOT NULL,
    member_id INTEGER NOT NULL,
    status VARCHAR(20) NOT NULL,  -- Present, Absent, Excused
    arrival_time TIME,
    departure_time TIME,
    notes TEXT,
    recorded_by INTEGER NOT NULL,  -- FK to users
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (meeting_id) REFERENCES meetings(id) ON DELETE CASCADE,
    FOREIGN KEY (member_id) REFERENCES members(id),
    FOREIGN KEY (recorded_by) REFERENCES users(id),
    UNIQUE(meeting_id, member_id)
);
```

4.3.8 Minutes Table

sql

```sql
CREATE TABLE minutes (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    meeting_id INTEGER UNIQUE NOT NULL,
    minutes_document_path VARCHAR(255),  -- Uploaded file path
    minutes_html TEXT,  -- If using built-in editor
    resolutions_summary TEXT,
    chairperson_approved BOOLEAN DEFAULT FALSE,
    approved_by INTEGER,  -- FK to users
    approved_date DATETIME,
```

```sql
    published BOOLEAN DEFAULT FALSE,
    published_date DATETIME,
    created_by INTEGER NOT NULL,  -- FK to users (Secretary)
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    updated_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (meeting_id) REFERENCES meetings(id) ON DELETE CASCADE,
    FOREIGN KEY (created_by) REFERENCES users(id),
    FOREIGN KEY (approved_by) REFERENCES users(id)
);
```

4.3.9 ActionItems Table

sql

```sql
CREATE TABLE action_items (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    meeting_id INTEGER NOT NULL,
    description TEXT NOT NULL,
    assigned_to INTEGER NOT NULL,  -- FK to members
    deadline DATE,
    status VARCHAR(20) DEFAULT 'Pending',  -- Pending, InProgress, Completed, Cancelled
    completion_date DATE,
    completion_notes TEXT,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    updated_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (meeting_id) REFERENCES meetings(id) ON DELETE CASCADE,
    FOREIGN KEY (assigned_to) REFERENCES members(id)
);
```

4.3.10 Loans Table

sql

```sql
CREATE TABLE loans (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    loan_number VARCHAR(20) UNIQUE NOT NULL,  -- LN-YYYY-NNN
    member_id INTEGER NOT NULL,
    amount_requested DECIMAL(15,2) NOT NULL,
    amount_approved DECIMAL(15,2),
    purpose TEXT NOT NULL,
    repayment_period_months INTEGER NOT NULL,  -- Max 2
    interest_rate DECIMAL(5,2) NOT NULL,  -- 5.00
    security_type VARCHAR(20) NOT NULL,  -- Collateral, Guarantors
```

```sql
    collateral_description TEXT,
    collateral_value DECIMAL(15,2),
    collateral_documents_path VARCHAR(255),
    guarantor1_id INTEGER,  -- FK to members
    guarantor2_id INTEGER,  -- FK to members
    guarantor1_approved BOOLEAN,
    guarantor2_approved BOOLEAN,
    executive_approved BOOLEAN DEFAULT FALSE,
    approved_by_chairman INTEGER,  -- FK to users
    approved_by_secretary INTEGER,
    approved_by_treasurer INTEGER,
    approval_date DATE,
    approval_notes TEXT,
    disbursed BOOLEAN DEFAULT FALSE,
    disbursement_date DATE,
    disbursement_method VARCHAR(20),
    disbursement_reference VARCHAR(50),
    total_payable DECIMAL(15,2),  -- Principal + Interest
    total_paid DECIMAL(15,2) DEFAULT 0.00,
    balance DECIMAL(15,2),
    status VARCHAR(20) DEFAULT 'Pending',  -- Pending, Approved, Disbursed, Active, Completed, Defaulted
    default_date DATE,
    recovery_notes TEXT,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    updated_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (member_id) REFERENCES members(id),
    FOREIGN KEY (guarantor1_id) REFERENCES members(id),
    FOREIGN KEY (guarantor2_id) REFERENCES members(id),
    FOREIGN KEY (approved_by_chairman) REFERENCES users(id),
    FOREIGN KEY (approved_by_secretary) REFERENCES users(id),
    FOREIGN KEY (approved_by_treasurer) REFERENCES users(id)
);
```

4.3.11 LoanRepayments Table

sql

```sql
CREATE TABLE loan_repayments (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    loan_id INTEGER NOT NULL,
    payment_date DATE NOT NULL,
    amount_paid DECIMAL(15,2) NOT NULL,
```

```
    principal_portion DECIMAL(15,2) NOT NULL,
    interest_portion DECIMAL(15,2) NOT NULL,
    payment_method VARCHAR(20) NOT NULL,
    transaction_reference VARCHAR(50),
    receipt_number VARCHAR(20),
    recorded_by INTEGER NOT NULL,  -- FK to users
    notes TEXT,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (loan_id) REFERENCES loans(id) ON DELETE CASCADE,
    FOREIGN KEY (recorded_by) REFERENCES users(id)
);
```

4.3.12 WelfareRequests Table

sql

```
CREATE TABLE welfare_requests (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    request_number VARCHAR(20) UNIQUE NOT NULL,  -- WR-YYYY-NNN
    member_id INTEGER NOT NULL,
    request_type VARCHAR(20) NOT NULL,  -- Bereavement, Medical, Celebration
    affected_person VARCHAR(100),  -- Name if bereavement
    relationship VARCHAR(50),  -- Spouse, Child, Parent
    incident_date DATE NOT NULL,
    description TEXT NOT NULL,
    amount_requested DECIMAL(15,2),
    documents_path VARCHAR(255),  -- Folder path for uploaded docs
    status VARCHAR(20) DEFAULT 'Submitted',  -- Submitted, UnderReview, Approved, Rejected, Paid
    submitted_date DATETIME DEFAULT CURRENT_TIMESTAMP,
    reviewed_by_secretary INTEGER,  -- FK to users
    secretary_review_date DATETIME,
    secretary_notes TEXT,
    approved_by_chairman INTEGER,  -- FK to users
    chairman_approval_date DATETIME,
    chairman_notes TEXT,
    amount_approved DECIMAL(15,2),
    rejection_reason TEXT,
    payment_voucher_number VARCHAR(20),
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    updated_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (member_id) REFERENCES members(id),
    FOREIGN KEY (reviewed_by_secretary) REFERENCES users(id),
```

```sql
    FOREIGN KEY (approved_by_chairman) REFERENCES users(id)
);
```

4.3.13 WelfarePayments Table

sql

```sql
CREATE TABLE welfare_payments (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    welfare_request_id INTEGER NOT NULL,
    payment_voucher_number VARCHAR(20) UNIQUE NOT NULL,
    amount_paid DECIMAL(15,2) NOT NULL,
    payment_date DATE NOT NULL,
    payment_method VARCHAR(20) NOT NULL,
    transaction_reference VARCHAR(50),
    beneficiary_name VARCHAR(100) NOT NULL,
    beneficiary_phone VARCHAR(20),
    beneficiary_acknowledgment_path VARCHAR(255),  -- Scanned signature
    paid_by_treasurer INTEGER NOT NULL,  -- FK to users
    confirmed_by_secretary INTEGER NOT NULL,  -- FK to users
    notes TEXT,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (welfare_request_id) REFERENCES welfare_requests(id),
    FOREIGN KEY (paid_by_treasurer) REFERENCES users(id),
    FOREIGN KEY (confirmed_by_secretary) REFERENCES users(id)
);
```

4.3.14 AuditLogs Table

sql

```sql
CREATE TABLE audit_logs (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    user_id INTEGER,  -- FK to users (nullable for system actions)
    action_type VARCHAR(50) NOT NULL,  -- Login, Logout, Create, Update, Delete, Approve, etc.
    entity_type VARCHAR(50),  -- Contribution, Loan, Member, etc.
    entity_id INTEGER,
    old_value TEXT,  -- JSON format
    new_value TEXT,  -- JSON format
    ip_address VARCHAR(45),
    user_agent TEXT,
    timestamp DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id)
```

```
);
```

4.3.15 Notifications Table

sql

```sql
CREATE TABLE notifications (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    user_id INTEGER NOT NULL,  -- FK to users
    notification_type VARCHAR(50) NOT NULL,  -- PaymentReminder, LoanApproved, etc.
    title VARCHAR(255) NOT NULL,
    message TEXT NOT NULL,
    related_entity_type VARCHAR(50),  -- Contribution, Loan, etc.
    related_entity_id INTEGER,
    is_read BOOLEAN DEFAULT FALSE,
    read_at DATETIME,
    whatsapp_sent BOOLEAN DEFAULT FALSE,
    whatsapp_sent_at DATETIME,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
);
```

4.3.16 SystemSettings Table

sql

```sql
CREATE TABLE system_settings (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    setting_key VARCHAR(100) UNIQUE NOT NULL,
    setting_value TEXT NOT NULL,
    data_type VARCHAR(20) NOT NULL,  -- String, Integer, Decimal, Boolean, Date
    description TEXT,
    updated_by INTEGER,  -- FK to users
    updated_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (updated_by) REFERENCES users(id)
);
```

4.4 Database Indexes

Performance Optimization:

sql

```sql
-- Users table
CREATE INDEX idx_users_username ON users(username);
CREATE INDEX idx_users_member_id ON users(member_id);
```

```sql
-- Members table
CREATE INDEX idx_members_member_number ON members(member_number);
CREATE INDEX idx_members_status ON members(status);
CREATE INDEX idx_members_phone ON members(phone_primary);

-- Contributions table
CREATE INDEX idx_contributions_member_id ON contributions(member_id);
CREATE INDEX idx_contributions_month ON contributions(contribution_month);
CREATE INDEX idx_contributions_date ON contributions(payment_date);

-- Loans table
CREATE INDEX idx_loans_member_id ON loans(member_id);
CREATE INDEX idx_loans_status ON loans(status);
CREATE INDEX idx_loans_number ON loans(loan_number);

-- WelfareRequests table
CREATE INDEX idx_welfare_member_id ON welfare_requests(member_id);
CREATE INDEX idx_welfare_status ON welfare_requests(status);
CREATE INDEX idx_welfare_type ON welfare_requests(request_type);

-- Meetings table
CREATE INDEX idx_meetings_date ON meetings(meeting_date);
CREATE INDEX idx_meetings_type ON meetings(meeting_type);

-- AuditLogs table
CREATE INDEX idx_audit_user_id ON audit_logs(user_id);
CREATE INDEX idx_audit_timestamp ON audit_logs(timestamp);
CREATE INDEX idx_audit_entity ON audit_logs(entity_type, entity_id);

-- Notifications table
CREATE INDEX idx_notifications_user_id ON notifications(user_id);
CREATE INDEX idx_notifications_is_read ON notifications(is_read);
```

4.5 Initial Data (Seed Data)

System Settings:

sql

```sql
INSERT INTO system_settings (setting_key, setting_value, data_type, description) VALUES
('group_name', 'Old Timers Savings Group', 'String', 'Official group name'),
```

```
('membership_fee', '20000', 'Decimal', 'One-time membership fee in UGX'),
('monthly_contribution', '100000', 'Decimal', 'Standard monthly contribution in UGX'),
('bereavement_amount', '500000', 'Decimal', 'Fixed bereavement support amount in UGX'),
('loan_interest_rate', '5.00', 'Decimal', 'Monthly loan interest rate percentage'),
('loan_max_period', '2', 'Integer', 'Maximum loan repayment period in months'),
('quorum_requirement', '5', 'Integer', 'Minimum members required for quorum'),
('suspension_threshold', '3', 'Integer', 'Months of non-payment before suspension'),
('expulsion_threshold', '6', 'Integer', 'Months of non-payment before expulsion'),
('qualification_period', '5', 'Integer', 'Consecutive months required to qualify for benefits'),
('expelled_refund_percentage', '80', 'Integer', 'Percentage of contributions refunded to expelled members'),
('fiscal_year_start', '01', 'Integer', 'Month when fiscal year starts (1-12)'),
('currency_code', 'UGX', 'String', 'Currency code'),
('date_format', 'DD/MM/YYYY', 'String', 'Display format for dates'),
('timezone', 'Africa/Kampala', 'String', 'System timezone');
```

---

5. USER INTERFACE DESIGN

5.1 Design Principles

Mobile-First Approach:
- Design for 375px width first (iPhone SE)
- Scale up to tablet (768px) and desktop (1200px+)
- Touch-friendly elements (minimum 44×44px)
- Simplified navigation for small screens

Simplicity:
- Clean, uncluttered layouts
- Focus on one primary action per page
- Minimal text, clear instructions
- Consistent design patterns

Accessibility:
- High contrast (WCAG AA minimum)
- Readable fonts (minimum 14px on mobile)
- Keyboard navigation support
- Clear focus indicators

Performance:
- Lightweight Bootstrap 5 (via CDN)
- Minimal custom CSS
- No heavy JavaScript frameworks
- Lazy loading images
- Pagination for long lists

5.2 Color Scheme

Primary Colors:
- Primary: #2C3E50 (Dark blue-gray) - Headers, primary buttons
- Secondary: #27AE60 (Green) - Success states, positive actions
- Accent: #E67E22 (Orange) - Alerts, important information

Status Colors:
- Success: #27AE60 (Green)
- Warning: #F39C12 (Amber)
- Danger: #E74C3C (Red)
- Info: #3498DB (Blue)

Neutral Colors:
- Background: #ECF0F1 (Light gray)
- Card/Panel: #FFFFFF (White)
- Text: #2C3E50 (Dark gray)
- Muted text: #7F8C8D (Medium gray)
- Border: #BDC3C7 (Light gray)

5.3 Typography

Font Family: System default fonts (no custom fonts to save bandwidth)
- `-apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, "Helvetica Neue", Arial, sans-serif`

Font Sizes:
- H1: 28px (mobile), 36px (desktop)
- H2: 24px (mobile), 30px (desktop)
- H3: 20px (mobile), 24px (desktop)
- Body: 14px (mobile), 16px (desktop)
- Small: 12px (mobile), 14px (desktop)

Font Weights:

- Regular: 400
- Medium: 500
- Bold: 700

5.4 Layout Structure

Page Template:
```
┌─────────────────────────────────────┐
│      Top Navigation Bar             │ ← Logo, User menu, Notifications
├─────────────────────────────────────┤
│ Breadcrumb (if needed)              │
├─────────────────────────────────────┤
│                                     │
│      Page Header                    │ ← Title, primary action button
│                                     │
├─────────────────────────────────────┤
│                                     │
│                                     │
│      Main Content Area              │ ← Forms, tables, cards
│                                     │
│                                     │
├─────────────────────────────────────┤
│      Footer                         │ ← Copyright, version
└─────────────────────────────────────┘
```

Mobile Navigation:
- Hamburger menu (toggles sidebar)
- Bottom navigation bar for key actions (optional)
- Sticky header

Desktop Navigation:
- Left sidebar (collapsible)
- Top bar with user profile
- Breadcrumbs for deep navigation

5.5 Key Screens/Pages

5.5.1 Login Page

Elements:
- Group logo/name (centered)
- Phone number input field
- Password input field (with show/hide toggle)
- "Remember Me" checkbox
- Login button (full width on mobile)
- "Forgot Password?" link
- System version footer

Mobile Wireframe:
```
┌─────────────────────────────┐
│                 │
│  [Group Logo]    │
│                 │
│  Old Timers SG    │
│                 │
│ ┌───────────────────────┐ │
│ │ Phone Number   │ │
│ └───────────────────────┘ │
│ ┌───────────────────────┐ │
│ │ Password  [👁] │ │
│ └───────────────────────┘ │
│ □ Remember Me      │
│ ┌───────────────────────┐ │
│ │  LOGIN       │ │
│ └───────────────────────┘ │
│ Forgot Password?   │
│                 │
│  v1.0 Dec 2024    │
└─────────────────────────────┘
```

5.5.2 Member Dashboard

Widgets (cards):

- Welcome message with name
- Quick stats:
  - Total contributed
  - Consecutive months
  - Qualified for benefits? (Yes/No badge)
  - Last contribution date

- Recent contributions (last 3)
- Outstanding payments (if any - highlighted)
- Quick actions:
  - View my statement
  - Submit welfare request
  - Apply for loan
- Group announcements (latest 2)
- Upcoming meetings

Mobile Wireframe:

```
┌─────────────────────────┐
│ ☰ Dashboard  🔔3 │
├─────────────────────────┤
│ Welcome, Alex!    │
├─────────────────────────┤
│ ┌─────────────────────┐ │
│ │ Total: 500,000 │ │
│ │ Months: 5 ✓    │ │
│ │ Qualified: YES │ │
│ └─────────────────────┘ │
│ ┌─────────────────────┐ │
│ │ Recent Contrib │ │
│ │ Dec: 100,000 ✓ │ │
│ │ Nov: 100,000 ✓ │ │
│ │ Oct: 100,000 ✓ │ │
│ └─────────────────────┘ │
│ ┌─────────────────────┐ │
│ │ Quick Actions  │ │
│ │ • My Statement │ │
│ │ • Request Help │ │
│ │ • Apply Loan   │ │
│ └─────────────────────┘ │
└─────────────────────────┘
```

5.5.3 Executive Dashboard

Widgets:

- Key financial metrics (4 cards):
  - Total contributions (this month)
  - Total savings (all-time)
  - Pending payments (count + amount)
  - Loans outstanding

- Monthly payment status grid (members × paid/unpaid)
- Pending approvals (count badges):
  - Welfare requests
  - Loan applications
  - Minutes pending approval
- Recent transactions (last 10)
- Charts:
  - Monthly contributions trend
  - Member payment rate
- Quick actions:
  - Record contribution
  - Generate report
  - Send reminders

5.5.4 Record Contribution Page

Form Fields:

- Member (searchable dropdown)
- Amount (default: 100,000, editable)
- Payment date (date picker)
- Contribution month (month picker)
- Payment method (radio buttons: Cash, Mobile Money, Bank)
- Transaction reference (text input, required if not cash)
- Notes (textarea, optional)
- Proof of payment (file upload, optional)

Buttons:

- Save & Generate Receipt (primary)
- Save Draft (secondary)
- Cancel

Validation:

- Required fields highlighted
- Duplicate check (member + month)
- Amount warning if not 100,000
- Past date warning (>30 days)

5.5.5 Member Statement Page

Header:

- Member name and number
- Period selector (date range)
- Download PDF button
- Print button

Summary Section:

- Total contributed
- Number of contributions
- Consecutive months
- Qualification status
- Average monthly contribution

Transactions Table:

- Date | Month | Amount | Method | Receipt No. | Actions
- Pagination (20 per page)
- Sort by date (newest first)
- Download receipt icon per row

Footer:

- Generated date/time
- Generated by (system user)

5.5.6 Welfare Request Form

Step 1: Request Type

- Radio buttons with icons:
  - 💐 Bereavement
  - 🛡️ Medical Support
  - 🎉 Celebration

Step 2: Details (conditional based on type)

If Bereavement:

- Affected person name
- Relationship (dropdown: Spouse, Child, Parent)
- Date of death
- Description
- Upload death certificate

If Medical:

- Description of illness/condition
- Hospital name
- Date of treatment
- Amount requested (max 1,000,000)
- Upload medical documents (bills, reports)

If Celebration:

- Event type (dropdown: Wedding, Introduction, Nikah)
- Event date

- Description
- Upload invitation (optional)

Step 3: Review & Submit

- Summary of entered information
- Document preview
- Confirmation checkbox
- Submit button

5.5.7 Loan Application Form

Step 1: Loan Details

- Amount requested
- Purpose (textarea)
- Repayment period (dropdown: 1 month, 2 months)
- Auto-calculated fields (read-only):
  - Interest rate: 5%
  - Total interest
  - Total repayable
  - Monthly payment breakdown

Step 2: Security

- Security type (radio: Collateral, Guarantors)

If Collateral:

- Asset type (dropdown)
- Description
- Estimated value
- Upload documents

If Guarantors:

- Guarantor 1 (dropdown: active members, exclude self)
- Guarantor 2 (dropdown: active members, exclude self and G1)
- Both must accept before proceeding

Step 3: Review & Submit

- Loan summary
- Repayment schedule table
- Terms & conditions checkbox
- Submit application button

5.5.8 Reports Page

Report Categories (tabs):

- Financial Reports
- Member Reports
- Loan Reports
- Welfare Reports
- Audit Reports

Report Generator:

- Report type (dropdown)
- Parameters (varies by report):
  - Date range
  - Member selection
  - Status filters
- Preview button
- Generate PDF button
- Export to Excel button

Recent Reports:

- List of previously generated reports
- Download/view again
- Delete old reports

---

6. IMPLEMENTATION PLAN

6.1 Development Phases

Phase 1: Foundation (Weeks 1-2)

- Development environment setup
- Database design and creation
- User authentication system
- Basic layout and navigation
- User management (admin only)

Deliverable: Working login system, admin can create users

Phase 2: Core Features (Weeks 3-5)

- Member management (CRUD)
- Next of kin management
- Contribution recording
- Receipt generation (PDF)
- Member dashboard
- Executive dashboard
- Basic reporting

Deliverable: Can record and track contributions, generate receipts

Phase 3: Welfare & Loans (Weeks 6-7)

- Welfare request submission
- Welfare approval workflow
- Welfare payment tracking
- Loan application
- Loan approval workflow
- Loan disbursement
- Loan repayment tracking

Deliverable: Complete welfare and loans management

Phase 4: Meetings & Compliance (Week 8)

- Meeting scheduling
- Attendance recording
- Minutes upload
- Action items tracking
- Audit trail implementation
- Audit reports

Deliverable: Complete meetings and compliance features

Phase 5: Reports & Analytics (Week 9)

- All standard reports
- Financial dashboards
- Analytics charts
- Export functionality
- Data visualization

Deliverable: Comprehensive reporting system

Phase 6: Notifications & Polish (Week 10)

- In-app notifications
- WhatsApp integration (manual templates)
- Payment reminders
- UI/UX refinements
- Mobile responsiveness testing
- Performance optimization

Deliverable: Fully functional system with notifications

Phase 7: Testing & Deployment (Weeks 11-12)

- User acceptance testing
- Bug fixes

- Data migration (historical data)
- User training materials
- PythonAnywhere deployment
- Production launch

Deliverable: Live system with historical data

Phase 8: Training & Handover (Week 13)

- User training sessions (Executive)
- User training sessions (Members)
- Administrator training
- Documentation handover
- Support plan
- First month monitoring

Deliverable: Trained users, system in production

6.2 Resource Requirements

Human Resources:

- 1 Developer (Alex): Full development, testing, deployment
- 1 Tester (volunteer from group): User acceptance testing
- Executive Committee: Requirements validation, UAT, training coordination

Technical Resources:

- Development laptop/computer
- Internet connection
- PythonAnywhere free account
- Git repository (GitHub free)
- Testing devices (1 Android phone, 1 desktop browser minimum)

Tools & Software:

- Python 3.10+
- Visual Studio Code (or preferred IDE)
- DB Browser for SQLite (database management)
- Postman (API testing, if needed)
- Git for version control

Budget:

- UGX 0 (free tier hosting)
- Optional: UGX 50,000 for domain name (if desired later)

6.3 Deployment Strategy

Pre-Deployment:

1. Complete all testing in local development environment
2. Fix all critical and high-priority bugs
3. Prepare deployment checklist
4. Backup all code and database
5. Create deployment documentation

Deployment Steps:

1. Create PythonAnywhere free account
2. Upload code via Git or file upload
3. Install Python dependencies (requirements.txt)
4. Configure environment variables
5. Create database and run migrations
6. Import seed data (system settings)
7. Create super admin account
8. Test all critical features
9. Import historical data (contributions, members)
10. Final smoke tests
11. Go live

Post-Deployment:

1. Monitor error logs (first 48 hours closely)
2. Performance monitoring
3. User feedback collection
4. Quick hotfixes if needed
5. Weekly check-ins (first month)

URL Structure:

- Production: `oldtimerssavings.pythonanywhere.com`
- Staging (if needed): Use separate PythonAnywhere account

6.4 Data Migration Plan

Historical Data to Import:

1. **7 Members:**
   - Towongo John (Chairman)
   - Senoga William (Treasurer)
   - Luyima Alex Cedric (Secretary)
   - Kule Bagonza
   - Luyonga Ronald
   - Kakande Sadat
   - Namanya [Full Name]
2. **Contributions (July 2024 - December 2024):**
   - From physical notebook
   - Fields: Name, Date, Amount, Transaction Ref
   - Total: ~UGX 3,420,000
3. **Meetings:**

- 2 meetings with minutes

Migration Process:

1. Create Excel template for each entity
2. Manually enter data from physical records
3. Validate data (check totals, dates, duplicates)
4. Import via system's bulk import feature
5. Verify imported data
6. Generate reconciliation report
7. Executive approval before finalizing

Data Validation:

- Total contributions must match UGX 3,420,000
- All receipt numbers sequential
- All dates valid
- No duplicate contributions (member + month)

6.5 Training Plan

Training Audiences:

1. Super Admin (Alex)
2. Executive Members (3)
3. Auditors (2)
4. Regular Members (remaining)

Training Methods:

- In-person demonstrations (preferred)
- Screen sharing sessions (WhatsApp video call)
- Video tutorials (recorded, shared via YouTube unlisted)
- User manual (PDF)
- Quick reference cards

Training Schedule:

Week 1: Executive Training (3 hours)

- System overview and objectives
- Login and navigation
- Recording contributions
- Generating receipts
- Approving welfare requests
- Managing loans
- Viewing reports
- Hands-on practice with test data

Week 2: Member Training (2 hours)

- Login and first-time setup
- Viewing personal dashboard
- Checking contribution history
- Downloading statements
- Submitting welfare requests
- Applying for loans
- Q&A session

Week 3: Auditor Training (1.5 hours)

- Audit dashboard
- Accessing financial records
- Generating audit reports
- Using audit trail
- Best practices

Ongoing Support:

- WhatsApp group for questions
- Weekly office hours (first month)
- Email support
- System updates communication

6.6 Success Criteria

Technical Success:

- ✅ System loads in < 3 seconds on 3G
- ✅ Zero critical bugs in first month
- ✅ 99% uptime (excluding PythonAnywhere maintenance)
- ✅ All data migrated accurately
- ✅ Mobile-responsive on all tested devices

User Adoption Success:

- ✅ 100% of members trained within 3 weeks
- ✅ 80% of members log in at least once in first month
- ✅ Executive uses system for all contribution recording
- ✅ Zero paper receipts after go-live
- ✅ All meeting minutes uploaded digitally

Business Success:

- ✅ Contribution tracking 100% digital
- ✅ Receipt generation automated
- ✅ Monthly reports generated in < 5 minutes
- ✅ Reduced administrative time by 50%

- ✅ Improved transparency (member feedback)
- ✅ Accurate audit trails

6.7 Risk Management

Risk 1: PythonAnywhere Free Tier Limitations

- **Mitigation:** Monitor usage closely, optimize queries, implement caching
- **Contingency:** Upgrade to paid tier (UGX 20,000/month) if needed

Risk 2: Poor Internet Connectivity

- **Mitigation:** Optimize for 3G, lightweight pages, caching
- **Contingency:** Offline mode (future enhancement)

Risk 3: Low User Adoption

- **Mitigation:** Comprehensive training, user-friendly design, executive buy-in
- **Contingency:** Extended training period, one-on-one support

Risk 4: Data Loss

- **Mitigation:** Weekly automated backups, manual backups before major operations
- **Contingency:** Restore from backup, maintain paper records in parallel (first 3 months)

Risk 5: Security Breach

- **Mitigation:** Strong passwords, HTTPS, regular security updates, access logs
- **Contingency:** Immediate password reset, audit trail review, incident report

Risk 6: Developer Unavailability

- **Mitigation:** Comprehensive documentation, code comments, backup developer identified
- **Contingency:** External developer engagement (if budget allows)

---

7. MAINTENANCE & SUPPORT

7.1 Ongoing Maintenance

Daily:

- Monitor system health (storage, CPU, errors)
- Check for critical errors in logs
- Respond to urgent support requests

Weekly:

- Review user activity
- Check backup completion
- Review and address non-critical bugs
- Update documentation if needed

Monthly:

- Generate system usage report
- Review performance metrics
- Plan enhancements based on feedback
- Security updates (if any)

Quarterly:

- Comprehensive system review
- Database optimization
- Cleanup temporary files
- Audit data integrity
- Review and update training materials

Annually:

- Major version update (if needed)
- Technology stack review
- Security audit
- Disaster recovery drill

7.2 Support Model

Tier 1: Self-Service

- User manual (PDF)
- Video tutorials
- FAQ page in system
- Quick reference cards

Tier 2: Peer Support

- WhatsApp group for questions
- Executive members help regular members
- Knowledge sharing

Tier 3: Administrator Support

- Alex (Secretary) provides direct support
- Response time: 24 hours for non-critical
- Response time: 4 hours for critical
- Support hours: Monday-Friday, 9 AM - 5 PM EAT

Tier 4: Emergency Support

- System down or critical bug
- Response time: 2 hours
- Available 24/7 via phone

7.3 Enhancement Process

Feature Request Workflow:

1. Member submits feature request (via system or WhatsApp)
2. Secretary logs request
3. Executive Committee reviews and prioritizes
4. General Meeting approval (if significant change)
5. Development and testing
6. User notification before deployment
7. Training (if needed)
8. Deployment

Change Management:

- No changes during month-end (1st-5th)
- All changes tested in staging first
- Users notified 7 days before major changes
- Rollback plan for every deployment

---

8. DOCUMENTATION DELIVERABLES

8.1 Technical Documentation

1. System Architecture Document

- Technology stack
- Database schema
- API documentation (if any)
- Security architecture
- Deployment configuration

2. Developer Guide

- Setup instructions
- Code structure
- Naming conventions
- Development workflow
- Testing procedures

3. Database Documentation

- ERD diagram
- Table descriptions
- Relationships
- Indexes
- Sample queries

4. Deployment Guide

- Environment setup
- Deployment steps
- Configuration
- Troubleshooting

8.2 User Documentation

1. User Manual (PDF, ~30 pages)

- Introduction
- Getting started
- Feature-by-feature guide with screenshots
- Common tasks
- Troubleshooting
- FAQs
- Glossary

2. Quick Reference Cards (1-page each)

- Member quick reference
- Executive quick reference
- Auditor quick reference

3. Video Tutorials (~10-15 minutes each)

- System introduction
- Recording contributions
- Submitting welfare requests
- Applying for loans
- Generating reports

4. Administrator Guide (PDF, ~20 pages)

- User management
- System settings
- Backup and restore
- Security management
- Troubleshooting
- Maintenance tasks

8.3 Process Documentation

1. Standard Operating Procedures (SOPs)

- Monthly contribution recording process
- Welfare request approval process
- Loan approval and disbursement process
- Meeting minutes publication process
- Quarterly audit process
- Data backup and recovery process

2. Policies

- Data privacy policy
- Password policy
- Access control policy
- Data retention policy
- Incident response policy

---

9. APPENDICES

9.1 Appendix A: Glossary

Active Member: Member who has contributed in the current month or has a valid excuse for absence

Bereavement: Death of member's spouse, child (5+ days old), or parent, triggering UGX 500,000 support (Resolution 2)

Consecutive Months: Continuous monthly contributions without gaps; resets if contribution missed

Expelled Member: Member removed after 6 months of non-payment or for dishonesty; receives 80% refund (Resolution 9)

Executive Committee: Chairperson, Secretary, and Treasurer with administrative access

Qualified for Benefits: Member who has contributed for 5 consecutive months (meeting resolution)

Quorum: Minimum 5 members required for meeting to proceed (Resolution 6)

Suspended Member: Member who missed 3 consecutive contributions; limited access and benefits (Resolution 8)

Welfare: Support provided for bereavement, medical emergencies, or celebrations

9.2 Appendix B: Meeting Resolutions Reference

This system implements the following resolutions from the December 14, 2024 meeting:

Resolution	System Implementation
Resolution 1: Group objectives	System objectives and purpose statement
Resolution 2: Bereavement UGX 500,000	System setting, welfare module
Resolution 3: Member death payout	Automated calculation in welfare module
Resolution 4: Emergency meetings for illness/celebrations	Meeting scheduling module
Resolution 5: Loans at 5%, 2-month max	Loan management module with automated calculations
Resolution 6: Quorum 5 members	Meeting attendance validation
Resolution 7: By-law amendments 50%+1	System settings change approval workflow
Resolution 8: Suspension/expulsion thresholds	Automated member status management
Resolution 9: 80% refund for expelled	Refund calculation in member module
Resolution 10: Auditors appointed	Auditor role and access
Resolution 11: Digital system approved	This entire system
Resolution 12: Namanya deadline Dec 31	Member status tracking
Resolution 13: Dispute resolution committee	Future enhancement (manual process for now)
Resolution 14: Three-step withdrawal approval	Welfare and loan payment voucher workflow
Resolution 15: Quarterly audits	Audit module and reporting

9.3 Appendix C: System Requirements Traceability Matrix

Maps functional requirements to meeting resolutions and user needs:

Requirement ID	Feature	Resolution	User Need
FR-001 to FR-006	User Authentication	Resolution 11	Secure access
FR-007 to FR-010	Member Management	Resolutions 8, 9	Track member status
FR-011 to FR-018	Contributions	Resolutions 1, 8	Record payments
FR-019 to FR-022	Financial Management	Resolutions 2, 3, 5	Track money
FR-023 to FR-032	Welfare Management	Resolutions 2, 3, 4, 14	Support members
FR-033 to FR-040	Loans Management	Resolution 5	Provide loans
FR-041 to FR-047	Meetings	Resolutions 4, 6, 7	Document meetings
FR-048 to FR-053	Audit & Compliance	Resolutions 10, 15	Accountability
FR-054 to FR-058	Notifications	Resolution 11	Inform members
FR-059 to FR-062	Reports	Meeting requirement	Transparency

9.4 Appendix D: PythonAnywhere Free Tier Specifications

Account Limits:

- Web apps: 1

- Disk space: 512 MB
- Daily CPU: 100 seconds
- Python version: 3.10
- MySQL databases: Not available on free tier
- SQLite: Included
- Custom domains: Not available
- HTTPS: Included (subdomain.pythonanywhere.com)
- Always-on tasks: Not available
- Scheduled tasks: Not available on free tier

Bandwidth:

- Unlimited page views
- Reasonable use policy applies

Support:

- Community forums
- Email support (paid tiers only)

Upgrade Path:

- Hacker plan: $5/month (UGX ~20,000)
- Additional disk space, CPU, scheduled tasks
- Custom domain support

Monitoring:

- Built-in error logs
- Web app status page
- CPU usage dashboard

---

10. CONCLUSION

This specification document provides a comprehensive blueprint for the Old Timers Savings Group Digital Records Management System. The system is designed to:

✅ **Automate** manual record-keeping processes
✅ **Enhance** transparency and accountability
✅ **Improve** operational efficiency
✅ **Support** all resolutions from the December 14, 2024 meeting
✅ **Operate** within PythonAnywhere free tier constraints
✅ **Scale** to accommodate group growth to 15 members
✅ **Provide** secure, role-based access
✅ **Generate** comprehensive reports and analytics

Next Steps:

1. ✅ Review and approve this specification (Executive Committee)
2. ✅ Begin Phase 1 development (Foundation)
3. ✅ Regular progress updates at Executive meetings
4. ✅ User acceptance testing (UAT) in Week 11
5. ✅ Production deployment by end of January 2025
6. ✅ User training and handover in February 2025

Success Factors:

- Executive Committee support and engagement
- Member willingness to adopt digital system
- Consistent internet connectivity
- Regular system usage and data entry
- Ongoing feedback and improvements

Long-Term Vision:

- Foundation for group growth beyond 15 members
- Platform for additional features (SMS, email, advanced analytics)
- Model for other savings groups
- Potential for SACCO registration and formal recognition

Document Prepared By:
Luyima Alex Cedric
Secretary, Old Timers Savings Group
December 15, 2024

Document Approved By:
[Pending Executive Committee Review]

Version History:

- v1.0 - December 15, 2024 - Initial specification

END OF SPECIFICATION DOCUMENT

This specification is now complete and ready for your review! Would you like me to:

1. Create a project timeline/Gantt chart?
2. Generate the requirements.txt file for Python dependencies?
3. Start building the actual Flask application?

4. Create user training materials?
5. Design the database migration templates for your historical data?