

Salesforce based Financial Management System

Phase 1 — Problem Understanding & Industry Analysis

Executive summary

Organizations and individuals waste time, visibility and money when financial planning and expense tracking live in disconnected spreadsheets, manual forms, or siloed apps. This project proposes a compact **Salesforce-based Financial Management System** that provides a single source of truth for budgets, expenses, and simple savings planning. It is designed to be built quickly (3–4 hours for the minimum viable build) but architected so it can scale into approvals, ERP integrations and forecasting.

Key impact promises:

- Real-time visibility of planned vs actual spend.
- Reduced manual reconciliation and spreadsheet errors.
- Proactive alerts for overspend and near-limit conditions.
- Clear audit trail and role-based security for finance data.

Business problem — detailed articulation

Current situation (painful realities)

- Finance data is fragmented: budgets are created in Excel, expenses are recorded in WhatsApp/notes/receipts, and reconciliation happens at month-end in panic mode.
- Spreadsheets are error-prone and hard to govern; they become a single person's knowledge and a compliance risk. Specialist planning tools exist but many teams still rely on spreadsheets for planning and tracking.
- Small business owners and departmental leads often lack financial literacy or systems to continuously compare plan vs reality; this creates decision lag and frequent overspend.

Consequences

- Overspending in categories (travel, marketing) that could have been prevented.
- Poor forecasting — management acts reactively.
- Time wasted in data cleaning and reconciliation.
- Audit and compliance exposures due to weak trails and permissions.

Evidence & market signals

1. **Customers expect modern digital experiences in finance.** Salesforce's financial services research shows a high bar for digital UX and service organizations that fail to provide clear, proactive experiences risk losing trust.

2. **Spreadsheet reliance vs planning software is a real gap.** Industry studies show many organizations still use spreadsheets as their primary planning tool, even though specialist planning tools improve accuracy and control. That means a clear adoption pathway for a structured, low-friction solution.
3. **Small business demand for budgeting tools is growing.** Recent industry write-ups indicate that a large share of small business owners seek help with budgeting and are receptive to budgeting software. That validates the target user segment for a minimal Financial Management System.
4. **Cloud-first and SaaS adoption is mainstream.** The broad move to cloud infrastructure and SaaS spending creates favorable economics for building a cloud-hosted finance app on Salesforce — secure, manageable, and easily integrated with other cloud services.

Target users / stakeholder map

- **Primary users**
 - *Budget Owner (Manager / Department Head):* creates and owns monthly/quarterly budgets; needs visibility and alerts.
 - *Expense Submitter (Employee / Clerk):* records expenses against budgets with receipts; needs a simple UI.
 - *Finance Analyst / Accountant:* reviews expense streams, exports reports for reconciliation and taxes.
- **Secondary stakeholders**
 - *CFO / Finance Director:* cares about overall controls, compliance, and KPIs.
 - *IT / Salesforce Admin:* will manage the app, security, and integrations.
 - *Auditors / Compliance:* require traceability and role-based access records.
- **External**
 - *Banking/payment providers* (for future integrations), *ERP teams* (for future ledger syncs).

Stakeholder expectations (high level):

- Budget Owners: “Help me stop overspend and get alerts before we blow the budget.”
- Expense Submitters: “Make it faster than Excel — mobile-friendly.”
- Finance Analysts: “Let me slice, dice and export clean data for accounting.”

User personas (short, vivid)

1. Anita — Department Head (Budget Owner)

Age 34, oversees marketing budgets across campaigns. Hates late surprises. Wants a dashboard that shows “Remaining by campaign” and an alert when any campaign is >80% used.

2. Ravi — Junior Accountant (Finance Analyst)

- Age 25, reconciles expenses monthly. Spends hours cleaning spreadsheets. Wants structured data, a single place to export clean CSVs, and a reliable audit trail.

3. Leena — Field Employee (Expense Submitter)

- Age 29, travels frequently. She needs a quick way to log expenses on mobile, attach receipts, and assign the expense to the correct budget.

Core problems we will solve (succinct)

- Single source of truth for budgets & expenses.
- Automatic roll-up of actual spend per budget.
- Alerts and automation to avoid overspend.
- Clean, reportable data for finance and audit teams.

Competitive landscape :

- **Spreadsheets (Excel/Google Sheets):** ubiquitous but error-prone and hard to govern.
- **Dedicated budgeting/accounting SaaS (QuickBooks, Zoho Books, Xero):** powerful for accounting but often heavier than what a departmental budget owner needs and may not be available inside Salesforce where stakeholders already live.
- **Enterprise FP&A tools (Anaplan, Adaptive Insights):** full-featured but complex and expensive.
- **Opportunity:** a **Salesforce-native lightweight budget–expense app** sits between spreadsheets (too fragile) and heavy FP&A systems (too costly), delivering the right balance for teams and SMBs.

Use cases & user journeys (detailed)

Use case A — Create & publish a monthly budget

1. Budget Owner creates Budget record: Category, Period (month), Planned Amount, Owner.
2. Budget status defaults to Draft → Sent (once published).
3. System adds it to the Finance Manager App and shows on the Budgets dashboard.

Use case B — Record an expense against a budget

1. Expense Submitter creates an Expense record, attaches receipt, selects Budget (Master-Detail).
2. A roll-up on Budget updates Total Expenses and Remaining automatically.
3. If Remaining ≤ 0 or $>100\%$ budget exceeded, Budget Status flips to Exceeded and an email/Slack alert is sent.

Use case C — Monitoring & Alerts

- System sends a weekly digest of budgets >80% used to owners & finance (scheduled Flow).
- Ad-hoc report filters budgets for selected periods and owners.

Use case D — Month-end reconciliation

- Finance Analyst runs a Budget vs Actual report, exports to CSV, and imports to the accounting system if needed.

Functional requirements (phase 1 focus)

MUST have

- Objects: Budget (master), Expense (child).
- Fields: Planned Amount (Budget), Total Expenses (roll-up), Remaining (formula), Status (picklist). Expense: Date, Amount, Budget (Master-detail), Receipt (file), Payment Mode.
- Automation: Roll-up summary (Expense → Budget), Record-triggered Flow to set Budget Status (Within / Exceeded), Validation rule to prevent creating negative amounts or clearly erroneous entries.
- Reports & Dashboard: Budget vs Actual report, Expense by Category report, simple Dashboard with key metrics.
- Security: Profiles/Permission sets so only Budget Owner or Finance can edit budgets.
- Documentation: 6–8 screenshots + 1 page of acceptance criteria + demo script.

SHOULD have (stretch if time allows):

- Scheduled Flow for weekly alerts when >80% used.
- A simple Income object to model inflows (optional).
- Lightweight approval for expenses > X amount (if time).

Non-functional requirements & constraints

- **Usability:** Record creation in under 60 seconds for an expense (mobile friendly).
- **Performance:** Dashboard and reports respond within a few seconds; volume in MVP will be small (hundreds of records).
- **Security & Privacy:** Use standard Salesforce sharing; restrict file downloads to authorized profiles. Consider field-level encryption for sensitive fields if needed.
- **Compliance:** Keep audit trails (CreatedBy, LastModifiedBy, attachments). If you operate in regions with specific data residency rules, plan for them in later phases.

Data model (high-level)

- **Budget__c** (Master)

- Budget_Number__c (AutoNumber)
- Category__c (Picklist)
- Period_Start__c (Date) / Period_End__c (Date)
- Planned_Amount__c (Currency)
- Total_Expenses__c (Roll-Up Summary of Expense.Amount__c)
- Remaining__c (Formula = Planned_Amount__c - Total_Expenses__c)
- Status__c (Picklist: Draft, Active, Within Budget, Near Limit, Exceeded)
- **Expense__c (Detail)**
 - Expense_Number__c (AutoNumber)
 - Budget__c (Master-Detail → Budget__c)
 - Date__c (Date)
 - Amount__c (Currency)
 - Receipt__c (File)
 - Payment_Mode__c (Picklist)
 - Notes__c (Long Text)

Key success metrics (KPIs) — what stakeholders will measure

- **Adoption:** # of budgets created and # of users who recorded expenses in first 30 days.
- **Process efficiency:** Time (hours) saved monthly in reconciliation (baseline vs post-launch).
- **Control:** % budgets flagged before being exceeded (i.e., early alerts worked).
- **Data quality:** % of expenses with receipts attached.
- **Business impact (stretch):** Reduction in overspend incidents (target: measurable drop vs previous quarter).

Suggested targets (example): 80% of active budgets have weekly monitoring; 90% of expenses include receipts; the system reduces reconciliation time by 50% for pilot teams.

Risk analysis & mitigations

1. **User resistance (people prefer spreadsheets).**
 - Mitigation: Keep UI minimal and mobile-friendly; provide a 15-minute pilot and a simple cheat-sheet. Show direct time savings in demo.
2. **Data quality (missing receipts, wrong allocations).**
 - Mitigation: Add validation rules and make Receipt recommended for expenses > threshold. Provide picklist categories.
3. **Overly complex scope for MVP.**

- Mitigation: Strictly limit to Budgets + Expenses + Roll-ups + 2 reports + 1 dashboard for Phase 1.

4. Integration creep (ERP sync, bank feeds).

- Mitigation: Treat integrations as Phase 7; for Phase 1 show how CSV export/import will handle initial needs.