COMPLETE IMPLEMENTATION TO-DO LIST

Phase 1: Core Foundation (Weeks 1-2)

1.1 Application Architecture Setup

- [ ] Create Python project structure with modular design
- [ ] Set up main.py as application entry point
- [ ] Design GUI framework using Tkinter with ttk widgets
- [ ] Create tabbed interface using ttk.Notebook
- [ ] Establish consistent theming and styling system
- [ ] Set up error handling and logging framework

1.2 Data Persistence System

- [ ] Design JSON data structure for all application data
- [ ] Create persistence.py module for data management
- [ ] Implement auto-save functionality on data changes
- [ ] Implement auto-load functionality on application startup
- [ ] Create backup and recovery mechanisms
- [ ] Implement data validation and integrity checking

1.3 Category Management System

- [ ] Create category structure with groups (Loans, Investments, Lifestyle)
- [ ] Implement dynamic category addition/removal
- [ ] Create category validation and duplicate prevention
- [ ] Implement category keyword mapping for auto-categorization
- [ ] Create category import/export functionality

1.4 Basic GUI Framework

- [ ] Create main window with proper sizing and layout
- [ ] Implement tab navigation system
- [ ] Create reusable GUI components (tables, forms, dialogs)
- [ ] Implement consistent color scheme and styling
- [ ] Create loading indicators and progress bars

Phase 2: Core Features Implementation (Weeks 3-4)

2.1 Dashboard Tab Development

- [ ] Create monthly summary display with all required metrics
- [ ] Implement status indicators with color coding
- [ ] Create category-wise breakdown table

- [ ] Implement month navigation dropdown (24 months)
- [ ] Create quick action buttons
- [ ] Implement real-time update mechanisms
- [ ] Add graceful matplotlib fallback for visualizations

2.2 Transactions Tab Development

- [ ] Create transaction display table with sorting and filtering
- [ ] Implement manual transaction entry dialog with validation
- [ ] Create transaction editing and deletion functionality
- [ ] Implement CSV/Excel import with column mapping
- [ ] Create auto-categorization engine with keyword rules
- [ ] Implement uncategorized transaction highlighting
- [ ] Add bulk operations for transaction management
- [ ] Create transaction search and analytics features

2.3 Budget Planning Tab Development

- [ ] Create budget entry interface with month/category selection
- [ ] Implement budget validation and error handling
- [ ] Create budget template system with auto-fill functionality
- [ ] Implement configurable alert thresholds with persistence
- [ ] Create budget lifecycle management features
- [ ] Add budget analysis and comparison tools
- [ ] Implement budget copying between months

Phase 3: Advanced Features (Weeks 5-6)

3.1 Category Analysis Tab Development

- [ ] Create category overview display with performance metrics
- [ ] Implement category visualization (pie charts, bar charts)
- [ ] Create detailed category analysis with drill-down capability
- [ ] Add category comparison features
- [ ] Implement time-based analysis (trends, patterns)
- [ ] Create category performance scoring system
- [ ] Add advanced analytics and anomaly detection

3.2 What-If Simulator Tab Development

- [ ] Create scenario creation and management interface
- [ ] Implement multiple scenario types (budget changes, one-time events, investments)
- [ ] Create scenario application and calculation engine
- [ ] Implement scenario impact visualization

- [ ] Add scenario persistence and loading functionality
- [ ] Create scenario analysis and insights
- [ ] Implement scenario validation and conflict detection

3.3 Reports Tab Development

- [ ] Create multi-format export system (JSON, Excel, CSV, PDF)
- [ ] Implement report customization with date ranges and filters
- [ ] Create automated report generation system
- [ ] Design professional report templates
- [ ] Implement embedded charts in reports
- [ ] Add export process management with progress indicators

Phase 4: Visualization & Charts (Weeks 7-8)

4.1 Chart Integration

- [ ] Implement matplotlib integration with graceful fallback
- [ ] Create embedded charts in dashboard and category analysis
- [ ] Implement interactive chart features (zoom, pan, tooltips)
- [ ] Create chart export functionality
- [ ] Add responsive chart sizing and professional styling
- [ ] Implement real-time chart updates

4.2 Advanced Visualization

- [ ] Create sparklines for trend indicators
- [ ] Implement heatmaps for spending intensity
- [ ] Add multi-line charts for category trends
- [ ] Create comparison charts for scenarios
- [ ] Implement custom chart templates
- [ ] Add chart animation and smooth transitions

Phase 5: Advanced Analytics & Intelligence (Weeks 9-10)

5.1 Pattern Recognition

- [ ] Implement spending pattern identification
- [ ] Create seasonal variation detection
- [ ] Add anomaly detection for irregular spending
- [ ] Implement spending habit change tracking
- [ ] Create correlated category analysis

5.2 Predictive Analytics

- [ ] Create future spending forecasting

- [ ] Implement budget overrun risk prediction
- [ ] Add savings potential estimation
- [ ] Create long-term financial health projection
- [ ] Implement seasonal adjustment predictions

5.3 Smart Recommendations

- [ ] Create budget adjustment suggestion engine
- [ ] Implement category optimization recommendations
- [ ] Add savings opportunity identification
- [ ] Create spending efficiency improvement suggestions
- [ ] Implement investment timing recommendations

Phase 6: Configuration & Settings (Weeks 11-12)

6.1 Application Settings

- [ ] Create comprehensive settings management system
- [ ] Implement alert configuration with category-specific thresholds
- [ ] Add interface customization options
- [ ] Create import/export settings management
- [ ] Implement data management settings

6.2 User Experience Enhancements

- [ ] Add keyboard shortcuts for common actions
- [ ] Implement contextual help and tooltips
- [ ] Create user guide and documentation
- [ ] Add accessibility features
- [ ] Implement smooth animations and transitions

Phase 7: Performance & Optimization (Weeks 13-14)

7.1 Performance Optimization

- [ ] Optimize data loading and processing for large datasets
- [ ] Implement caching for frequently accessed data
- [ ] Add lazy loading for heavy operations
- [ ] Optimize chart rendering performance
- [ ] Implement memory management for 2-year datasets

7.2 Error Handling & Validation

- [ ] Create comprehensive input validation system
- [ ] Implement robust error handling throughout application
- [ ] Add data corruption recovery mechanisms

- [ ] Create user-friendly error messages and recovery options
- [ ] Implement logging and debugging tools

Phase 8: Advanced Features Implementation (Weeks 15-18)

8.1 Monthly Summary Pop-ups

- [ ] Create compact card-style monthly summary interface
- [ ] Implement automatic monthly pop-up on app launch
- [ ] Add on-demand summary access from dashboard
- [ ] Create quick export functionality from summary cards
- [ ] Implement "don't show again this month" option

8.2 Trends Tab Development

- [ ] Create dedicated Trends tab with dashboard layout
- [ ] Implement 2-year Total Planned vs Actual line charts
- [ ] Add Category Trends multi-line charts
- [ ] Create toggle controls for 6-month vs 2-year views
- [ ] Implement category filtering and export functionality

8.3 Report History Tab

- [ ] Create Report History tab with month-by-month archive
- [ ] Implement expandable summary cards for each month
- [ ] Add inline mini graphs for historical overview
- [ ] Create cross-month category analysis features
- [ ] Implement report regeneration functionality

Phase 9: Advanced Analytics & Intelligence (Weeks 19-22)

9.1 Enhanced Projection System

- [ ] Implement smart forecasting with seasonal adjustments
- [ ] Create lifecycle-aware projections (EMI endings, new investments)
- [ ] Add weighted moving average algorithm
- [ ] Implement seasonal pattern rules (festivals, school fees, insurance)
- [ ] Create projection accuracy tracking and validation

9.2 Advanced Bank Statement Import

- [ ] Implement intelligent column detection for various formats
- [ ] Create import preview with categorization suggestions
- [ ] Add advanced duplicate detection and resolution
- [ ] Implement import history tracking and management
- [ ] Create batch import functionality for multiple files

9.3 Enhanced Alert System

- [ ] Implement category-specific alert thresholds
- [ ] Create multiple notification types (pop-ups, status indicators)
- [ ] Add projection-based early warning system
- [ ] Implement alert history and management
- [ ] Create alert effectiveness analytics

Phase 10: Advanced Simulation & Tools (Weeks 23-26)

10.1 Enhanced What-If Simulation

- [ ] Implement complex multi-parameter scenarios
- [ ] Create scenario templates for common situations
- [ ] Add scenario comparison with side-by-side analysis
- [ ] Implement scenario optimization suggestions
- [ ] Create scenario risk assessment tools

10.2 Advanced Category Mapping

- [ ] Create intelligent bulk categorization interface
- [ ] Implement pattern learning and improvement engine
- [ ] Add custom categorization rule creation
- [ ] Create categorization confidence scoring
- [ ] Implement categorization undo/redo functionality

10.3 Professional Reporting

- [ ] Create professional PDF reports with embedded charts
- [ ] Implement executive summary report generation
- [ ] Add customizable report templates and branding
- [ ] Create automated monthly report generation
- [ ] Implement report scheduling and organization

Phase 11: Data Management & Backup (Weeks 27-28)

11.1 Enhanced Data Persistence

- [ ] Implement multiple backup strategies (daily, weekly, on-demand)
- [ ] Create data integrity verification and corruption recovery
- [ ] Add complete application state export/import
- [ ] Implement data migration tools for updates
- [ ] Create backup optimization and compression

11.2 Advanced Recovery Systems

- [ ] Implement automatic corruption detection
- [ ] Create multiple recovery options with date selection
- [ ] Add partial data recovery for specific modules
- [ ] Implement data merge capabilities for conflicts
- [ ] Create backup metadata management

Phase 12: Testing & Quality Assurance (Weeks 29-30)

12.1 Comprehensive Testing

- [ ] Create unit tests for all core functionality modules
- [ ] Implement integration tests for GUI components and workflows
- [ ] Add data validation and import/export comprehensive tests
- [ ] Create performance tests for large datasets (2-year data)
- [ ] Implement user acceptance testing scenarios
- [ ] Test all advanced features (projections, scenarios, reports)
- [ ] Validate alert systems and notification functionality

12.2 Final Polish & Documentation

- [ ] Complete comprehensive code documentation and comments
- [ ] Create detailed user manual and installation guide
- [ ] Implement final UI/UX improvements and polish
- [ ] Add sample data, templates, and example scenarios
- [ ] Create deployment and distribution package
- [ ] Finalize help system and contextual guidance

COMPREHENSIVE IMPLEMENTATION VALIDATION CHECKLIST

Core Functionality Validation (Must Pass 100%)

- [ ] All 6 main tabs (Dashboard, Transactions, Budget, Category Analysis, What-If Simulator, Reports) fully functional
- [ ] Complete transaction lifecycle (add, edit, delete, import) working correctly
- [ ] 2-year budget planning with all category types operational
- [ ] Real-time alerts and status indicators functioning properly
- [ ] What-If simulation with multiple scenario types working
- [ ] Comprehensive export functionality for all data types
- [ ] Robust data persistence and backup systems operational

Advanced Features Validation (Must Pass 100%)

- [ ] Monthly Summary Pop-ups working with all content and export options
- [ ] Trends Tab with 2-year analytics and interactive controls functional

- [ ] Report History Tab with expandable cards and cross-month analysis
- [ ] Advanced projection system with seasonal adjustments operational
- [ ] Enhanced bank import with intelligent processing working
- [ ] Multi-level alert system with category-specific thresholds functional
- [ ] Advanced What-If simulation with complex scenarios operational
- [ ] Professional reporting with embedded charts working
- [ ] Enhanced data backup and recovery systems functional

Performance Validation (Must Meet All Criteria)

- [ ] Application startup time under 3 seconds
- [ ] Data operations complete within 1 second
- [ ] Chart rendering under 2 seconds (with matplotlib)
- [ ] Import processing under 5 seconds for 1000 transactions
- [ ] Memory usage under 200MB for 2-year dataset
- [ ] Smooth UI interactions without lag or freezing
- [ ] All advanced features perform within acceptable limits

Quality & Reliability Validation (Must Pass All)

- [ ] Comprehensive error handling covers all edge cases
- [ ] Data integrity maintained across all operations and advanced features
- [ ] User interface intuitive and professional throughout
- [ ] All optional dependencies (matplotlib, pandas) handled gracefully
- [ ] Complete feature set matches all specification requirements
- [ ] Application stable and reliable under normal and heavy usage
- [ ] All backup and recovery mechanisms tested and functional
- [ ] Alert systems accurate and timely in all scenarios

User Experience Validation (Must Meet Standards)

- [ ] All features accessible within 3 clicks from main interface
- [ ] Consistent visual design and behavior across all components
- [ ] Error messages clear and actionable in all scenarios
- [ ] Help and guidance available for all complex features
- [ ] Professional appearance suitable for financial management
- [ ] Responsive interface adapting to all user actions
- [ ] Advanced features discoverable and intuitive to use# Financial Management & Simulation Tool - Complete Requirements & Specification Document

PROJECT OVERVIEW

Application Purpose

A comprehensive personal finance tracking and planning tool that will automatically capture bank transactions, track monthly budgets by category, provide real-time alerts, offer visual insights, include What-If scenario simulation capabilities, provide intelligent financial guidance with 2-year projection capabilities, and feature a configurable alert system with optional visual charts.

Core Vision

Create a desktop application that will serve as a financial command center, enabling users to:

- Track spending across multiple categories for 2 years
- Import bank transactions automatically
- Monitor budget adherence with configurable alerts
- Simulate financial scenarios for planning
- Generate comprehensive reports and visualizations
- Maintain complete data persistence and backup

COMPLETE FEATURE SPECIFICATIONS AT GRANULAR LEVEL

1. DASHBOARD TAB - FINANCIAL COMMAND CENTER

Feature: Central command center with real-time financial overview

User Stories with Detailed Descriptions:

US-1.1: Monthly Financial Status Overview

- **User Story**: As a user, I want to see my current month's financial status at a glance
- **Description**: The user needs immediate visibility into their financial health for the current month without having to navigate through multiple screens or perform calculations. This includes total budget allocated, amount already spent, remaining budget, and spending percentage. The information should be presented in a clear, visual format that allows for quick decision-making about spending for the rest of the month.
- **Business Value**: Enables quick financial health assessment and informed spending decisions
- **Acceptance Criteria**: Dashboard displays total budget, spent amount, remaining budget, and percentage in under 2 seconds of tab selection

US-1.2: Visual Spending Indicators

- **User Story**: As a user, I want visual indicators showing if I'm on track or overspending
- **Description**: Users need immediate visual feedback about their spending status through color-coded indicators and status badges. Green should indicate healthy spending (under 80% of budget), yellow should warn of approaching limits (80-100% or projection exceeding budget), and

red should alert of overspending (over 100% of budget). This visual system should be consistent across the entire application.

- **Business Value**: Provides instant spending awareness and prevents budget overruns through early warning
- **Acceptance Criteria**: Color indicators update in real-time and are visible throughout the interface with consistent meaning

US-1.3: Historical Month Navigation

- **User Story**: As a user, I want to navigate between months easily to view historical data
- **Description**: Users need the ability to review past months' financial data and plan for future months within the 2-year planning horizon (Aug-25 to Jul-27). Navigation should be intuitive with dropdown selection, previous/next buttons, and clear indicators of which months contain data versus empty months. The system should remember the last selected month for user convenience.
- **Business Value**: Enables trend analysis, historical comparison, and better financial planning
- **Acceptance Criteria**: Month navigation works smoothly with data loading under 1 second and session memory

US-1.4: Real-time Data Updates

- **User Story**: As a user, I want automatic updates when I add transactions or change budgets
- **Description**: The dashboard must reflect all changes immediately without requiring manual refresh. When users add transactions in other tabs, modify budgets, or make any data changes, the dashboard should automatically recalculate totals, update charts, refresh status indicators, and adjust projections. This ensures users always see current, accurate information.
- **Business Value**: Maintains data accuracy and provides confidence in financial information
- **Acceptance Criteria**: All dashboard elements update within 1 second of any data change across the application

US-1.5: Configurable Alert System

- **User Story**: As a user, I want color-coded alerts for different spending levels
- **Description**: Users need customizable alerts that match their personal financial management style. They should be able to set threshold percentages (default 10%, but adjustable to 20%, 50%, etc.) that trigger warnings when category spending approaches or exceeds these limits. Alerts should be visual (color changes), textual (warning messages), and contextual (showing specific amounts and recommendations).
- **Business Value**: Prevents overspending through personalized early warning system

- **Acceptance Criteria**: Alert thresholds are configurable, persistent across sessions, and trigger appropriate visual/textual warnings

US-1.6: Spending Pace Projections

- **User Story**: As a user, I want to see projections for the current month based on my spending pace
- **Description**: The system should analyze current spending patterns and project end-of-month totals based on daily spending rates. It should show if the current pace will result in budget overruns and provide specific amounts for course correction. Projections should account for weekdays vs weekends, recurring transactions, and seasonal patterns where possible.
- **Business Value**: Enables proactive budget management and spending adjustment before month-end
- **Acceptance Criteria**: Projections are calculated automatically and update daily with accuracy indicators

Granular Implementation Requirements:

1.1 Monthly Summary Display Requirements

- Must display total budget for selected month
- Must show total spent amount for selected month
- Must calculate and display remaining budget (Budget - Spent)
- Must show spending percentage (Spent / Budget * 100)
- Must display days remaining in current month
- Must calculate daily spending rate (Spent / Days Elapsed)
- Must project end-of-month spending (Daily Rate * Days in Month)

1.2 Status Indicators & Alerts Requirements

- Must implement color-coded status system:
    - Green: On track (spent < 80% of budget)
    - Yellow: Warning (spent 80-100% of budget or projection exceeds budget)
    - Red: Over budget (spent > 100% of budget)
- Must show status badge with current financial health
- Must display alert messages for overspending
- Must show projection warnings if current pace will exceed budget

1.3 Category-wise Breakdown Requirements

- Must display table with columns: Category, Planned, Spent, Remaining, Status
- Must calculate remaining amount for each category

- Must show percentage used for each category
- Must color-code each category row based on status
- Must highlight uncategorized spending in red
- Must show deviation amounts (overspent/under-spent)

1.4 Visualization Components Requirements

- Must create bar chart comparing planned vs spent by category
- Must use different colors for on-track vs overspent categories
- Must display monthly trend chart if historical data available
- Must show sparklines for category trends
- Must implement graceful fallback when matplotlib unavailable
- Must create mini pie chart showing spending distribution

1.5 Month Navigation Requirements

- Must provide dropdown selector for month/year (Aug-25 to Jul-27, total 24 months)
- Must include Previous/Next month navigation buttons
- Must auto-select current month on startup
- Must show indicators for months with data vs empty months
- Must remember last selected month in session

1.6 Real-time Updates Requirements

- Must auto-refresh dashboard when transactions added
- Must update charts when budgets modified
- Must recalculate all metrics when data changes
- Must refresh category status indicators
- Must update projections automatically

1.7 Quick Actions Requirements

- Must provide "Add Transaction" button for quick entry
- Must provide "Set Budget" button for current month
- Must provide "View Details" links for each category
- Must provide "Export Month" button for current month data

---

2. TRANSACTIONS TAB - COMPREHENSIVE TRANSACTION MANAGEMENT

Feature: Complete transaction lifecycle management with import capabilities

User Stories with Detailed Descriptions:

US-2.1: Manual Transaction Entry

- **User Story**: As a user, I want to add transactions manually with full details
- **Description**: Users need a comprehensive interface to manually enter transactions when they make purchases or payments. The entry form should include date selection (with calendar picker), category selection from their personalized categories, amount entry with validation, and optional description field. The system should provide intelligent defaults (today's date, last used category) and auto-suggestions based on historical data to speed up entry while maintaining accuracy.
- **Business Value**: Enables immediate transaction recording for accurate financial tracking
- **Acceptance Criteria**: Transaction entry completes in under 30 seconds with full validation and auto-suggestions

US-2.2: Bank Statement Import

- **User Story**: As a user, I want to import transactions from my bank statements
- **Description**: Users should be able to upload CSV or Excel files from their bank and have transactions automatically imported with intelligent processing. The system must detect column formats automatically, provide preview before import, handle various date formats, recognize amount columns (with/without currency symbols), and map description fields. Import should include duplicate detection, error reporting, and progress indication for large files.
- **Business Value**: Saves significant time and reduces manual entry errors for bulk transaction processing
- **Acceptance Criteria**: Import processes 1000+ transactions in under 5 seconds with 95%+ accuracy in column detection

US-2.3: Transaction Modification

- **User Story**: As a user, I want to edit or delete transactions when needed
- **Description**: Users need the ability to correct mistakes, update categorizations, or remove erroneous transactions. The edit functionality should preserve original entry date while tracking modification history. Deletion should require confirmation and offer undo capability. Bulk operations should be available for similar transactions, and all changes should immediately reflect in dashboard and reports.
- **Business Value**: Maintains data accuracy and allows correction of errors without data loss
- **Acceptance Criteria**: Edit/delete operations complete instantly with proper validation and confirmation dialogs

US-2.4: Intelligent Auto-categorization

- **User Story**: As a user, I want automatic categorization of imported transactions

- **Description**: The system should analyze transaction descriptions and automatically assign appropriate categories based on keywords, patterns, and learning from user corrections. It should handle various merchant names, common transaction types, and provide confidence scores for categorizations. Users should be able to create custom rules, and the system should improve accuracy over time through machine learning from user feedback.
- **Business Value**: Reduces manual categorization effort while maintaining high accuracy
- **Acceptance Criteria**: Auto-categorization achieves 85%+ accuracy rate and improves with user feedback

US-2.5: Uncategorized Transaction Management

- **User Story**: As a user, I want to see uncategorized transactions highlighted
- **Description**: Uncategorized transactions should be immediately visible through red highlighting, special icons, and dedicated counters. Users should get summary counts of uncategorized items, bulk categorization options, and smart suggestions based on similar historical transactions. The system should provide gentle reminders and make it easy to resolve uncategorized items quickly.
- **Business Value**: Ensures complete financial tracking and prevents data gaps in analysis
- **Acceptance Criteria**: Uncategorized transactions are clearly visible with efficient bulk resolution tools

US-2.6: Advanced Search and Filtering

- **User Story**: As a user, I want to search and filter my transactions
- **Description**: Users need powerful search capabilities across all transaction fields (date, category, amount, description) with filters for date ranges, amount ranges, specific categories, and transaction sources. Search should support partial matching, date range selection, and multiple criteria combinations. Results should be sortable and exportable, with search history for frequently used queries.
- **Business Value**: Enables quick transaction location and analysis for specific needs
- **Acceptance Criteria**: Search returns results in under 1 second for any dataset size with flexible filtering options

US-2.7: Data Validation and Error Prevention

- **User Story**: As a user, I want validation to prevent data entry errors
- **Description**: The system should validate all transaction data entry with real-time feedback. This includes date format validation, amount range checking, required field verification, duplicate detection warnings, and reasonableness checks (e.g., unusually large amounts). Error messages should be clear and actionable, with suggestions for correction. The system should prevent obviously incorrect data while allowing legitimate edge cases.
- **Business Value**: Maintains data quality and prevents downstream analysis errors

- **Acceptance Criteria**: Validation catches 99%+ of data entry errors with clear, helpful error messages

Granular Implementation Requirements:

2.1 Transaction Display & Management Requirements

- Must display all transactions in sortable table/tree view
- Must include columns: Date, Category, Amount, Description, Source (Manual/Imported)
- Must sort by date (newest first by default)
- Must provide filter by category, date range, amount range
- Must include search functionality across all fields
- Must support pagination for large datasets
- Must show total count and sum of displayed transactions

2.2 Manual Transaction Entry Requirements

- Must provide pop-up dialog with fields:
  - Date (date picker with validation)
  - Category (dropdown with existing categories + "Add New")
  - Amount (numeric input with currency formatting)
  - Description (text field, optional)
- Must validate input:
  - Date must be valid format (YYYY-MM-DD)
  - Amount must be positive number
  - Category cannot be empty
  - Description length limit (500 characters)
- Must auto-suggest categories based on description
- Must default date to today
- Must remember last used category per session

2.3 Transaction Editing & Deletion Requirements

- Must allow double-click or "Edit" button to modify transaction
- Must apply same validation rules as adding
- Must show confirmation dialog for deletion
- Must provide undo functionality for recent deletions
- Must maintain audit trail for modifications (optional)

- Must support bulk selection for mass operations
- Must enable bulk categorization for similar transactions

2.4 CSV/Excel Import Functionality Requirements

- Must provide file dialog to select CSV or Excel files
- Must automatically detect and map columns:
  - Date column (multiple format recognition)
  - Amount column (handle different currency formats)
  - Description/Memo column
  - Category column (if present)
- Must show import preview screen with first 10 rows
- Must provide column mapping interface for manual adjustment
- Must detect and handle duplicate transactions
- Must validate import data with error reporting
- Must show progress bar for large files
- Must display import summary (X transactions added, Y duplicates skipped)

2.5 Auto-categorization Engine Requirements

- Must implement keyword-based categorization rules based on user's category structure:

Loan & EMI Detection:

- "EMI", "LOAN", "CREDIT CARD" → respective EMI categories
- "HOME LOAN", "HOUSING" → Home Loan EMI
- "PERSONAL LOAN" → Personal Loan EMI categories

Investment Detection:

- "MUTUAL FUND", "MF", "SIP" → Mutual Fund SIP
- "PPF", "PROVIDENT" → PPF
- "RD", "RECURRING" → RD
- "GOLD", "BULLION" → Gold Investment
- "BITCOIN", "CRYPTO" → Bitcoin Investment
- "POLICY", "INSURANCE" → respective policy categories

Lifestyle & Essentials:

- "NETFLIX", "PRIME", "HOTSTAR", "OTT" → OTT Subscriptions

- "HOSPITAL", "CLINIC", "MEDICAL", "DOCTOR" → Hospital
- "SWIGGY", "ZOMATO", "FOOD", "DELIVERY" → Swiggy/Food
- "PETROL", "FUEL", "DIESEL" → Petrol
- "SHOPPING", "MALL", "STORE" → Shopping
- "EXPENSE", "MISC" → General Expenses
- Must learn from user corrections and build personalized rules
- Must provide confidence scoring for categorizations
- Must allow manual override capability
- Must suggest categories for manual entry
- Must support user-defined keyword rules

2.6 Uncategorized Transaction Handling Requirements

- Must highlight uncategorized transactions in red
- Must show count of uncategorized transactions
- Must provide "Categorize All" bulk operation
- Must offer smart suggestions based on similar transactions
- Must implement pattern recognition for recurring transactions
- Must provide reminder notifications for uncategorized items

2.7 Transaction Validation & Error Handling Requirements

- Must check for reasonable date ranges (not future dates unless allowed)
- Must validate amount formats and ranges
- Must detect potential duplicate transactions
- Must warn for unusually large amounts
- Must check for required fields
- Must provide format correction suggestions

2.8 Transaction Analytics Requirements

- Must show transaction count per month
- Must display spending trends over time
- Must identify top spending categories
- Must show average transaction amounts
- Must detect spending pattern changes

---

3. BUDGET PLANNING TAB - 2-YEAR BUDGET MANAGEMENT

Feature: Comprehensive budget planning with lifecycle management

User Stories with Detailed Descriptions:

US-3.1: 2-Year Category Budget Planning

- **User Story**: As a user, I want to set monthly budgets for each category for 2 years
- **Description**: Users need comprehensive budget planning capability spanning 24 months (Aug-25 to Jul-27) with category-specific allocations. The interface should allow easy navigation between months, quick data entry for each category, and visual indication of completed vs pending budget entries. Users should be able to see budget totals per month and identify months without complete budget data. The system should support all user-defined category groups (Loans & EMIs, Investments, Lifestyle & Essentials).
- **Business Value**: Enables long-term financial planning and proactive budget management
- **Acceptance Criteria**: All 24 months can be budgeted with easy navigation and clear progress indication

US-3.2: Financial Lifecycle Management

- **User Story**: As a user, I want to handle changing financial situations (EMI endings, new investments)
- **Description**: The system should accommodate planned financial changes such as loan completions, new investment starts, salary changes, or life events. Users should be able to set future dates for category changes, automatically transition budgets when events occur, and model the financial impact of these changes. The system should provide reminders for upcoming changes and help users plan for the financial implications.
- **Business Value**: Enables realistic long-term planning that accounts for changing financial obligations
- **Acceptance Criteria**: Lifecycle events are easily configured and automatically applied with proper notifications

US-3.3: Budget Template Automation

- **User Story**: As a user, I want template-based budget creation for similar months
- **Description**: Users should be able to create budget templates for recurring patterns (e.g., "normal month," "high expense month") and apply them quickly to multiple months. The system should provide default templates based on user's category structure, allow template customization, and support template sharing/export. Auto-fill functionality should populate all 24 months with base template while allowing individual month adjustments.
- **Business Value**: Saves time in budget setup and ensures consistency across planning periods

- **Acceptance Criteria**: Templates can be created, modified, and applied to multiple months in under 2 minutes

US-3.4: Threshold-based Budget Alerts

- **User Story**: As a user, I want alerts when my spending approaches budget limits
- **Description**: The system should monitor spending against budgets and provide early warnings when approaching limits. Users should be able to configure threshold percentages (10%, 20%, 50%) both globally and per category. Alerts should be visual (color changes), textual (specific messages), and actionable (showing exact amounts over/under budget). The alert system should consider spending pace and provide projections for end-of-month status.
- **Business Value**: Prevents budget overruns through proactive monitoring and early intervention
- **Acceptance Criteria**: Alerts are configurable, accurate, and provide actionable information for budget management

US-3.5: Cross-month Budget Operations

- **User Story**: As a user, I want to copy budgets between months
- **Description**: Users need efficient ways to replicate successful budget allocations across multiple months. The copy function should allow selective category copying, percentage-based adjustments (e.g., increase all by 5%), and bulk operations across date ranges. Users should be able to copy from any source month to any target month(s) with preview before applying changes.
- **Business Value**: Speeds up budget planning and maintains consistency across time periods
- **Acceptance Criteria**: Budget copying operations complete quickly with preview and selective category options

US-3.6: Budget vs Actual Analysis

- **User Story**: As a user, I want to see budget vs actual comparisons
- **Description**: The system should provide comprehensive analysis comparing planned budgets with actual spending across all categories and time periods. This includes variance analysis (over/under spending), trend identification, pattern recognition, and performance scoring. Users should see visual representations of budget adherence and receive insights about their budgeting accuracy and spending patterns.
- **Business Value**: Improves budget accuracy and spending discipline through historical analysis
- **Acceptance Criteria**: Comparison analysis is comprehensive, visual, and provides actionable insights

US-3.7: Personalized Alert Configuration

- **User Story**: As a user, I want configurable alert thresholds

- **Description**: Users should have complete control over when and how they receive budget alerts. This includes setting different threshold percentages for different categories (e.g., strict 10% for entertainment, relaxed 20% for medical), choosing alert types (visual, popup, status indicators), and customizing alert messages. The system should remember these preferences and apply them consistently across all budget monitoring functions.
- **Business Value**: Provides personalized budget monitoring that matches individual financial management styles
- **Acceptance Criteria**: Alert configuration is comprehensive, persistent, and immediately effective across the application

Granular Tasks:

3.1 Budget Entry & Management

- Display budget table with columns: Month, Category, Planned, Actual, Remaining, Status
- Month/Year selector (Aug-25 to Jul-27)
- Category-wise budget entry for selected month
- Support for user-specific category structure:

Loans & EMIs:

- Credit Card EMI 1
- Credit Card EMI 2
- Personal Loan EMI 1
- Personal Loan EMI 2
- Home Loan EMI

Investment Categories:

- Mutual Fund SIP
- PPF (Public Provident Fund)
- RD (Recurring Deposit)
- Ponmagan Policy
- Gold Investment
- Bitcoin Investment
- Baby Health Policy
- Baby Education Policy

Lifestyle & Essentials:

- OTT Subscriptions

- Hospital (Medical Expenses)
- Swiggy/Food Delivery
- Petrol
- General Expenses
- Shopping

Dynamic Category Management:

- Support for adding new categories during runtime
- Category grouping by type (Loans, Investments, Lifestyle)
- Custom category creation with validation
- Category modification and deletion capabilities
- Import/export of category configurations

3.2 Budget Templates & Automation

- Default budget template with predefined amounts
- "Auto-fill 24 months" functionality
- Budget lifecycle management:
  - CC EMI removal from Dec-25
  - New investments starting Apr-26
  - Loan reduction from Dec-26
- Template customization and saving
- Copy budget from previous month
- Apply template to multiple months

3.3 Budget Validation & Error Handling

- Positive number validation for all amounts
- Reasonable amount range checking
- Total budget calculation and display
- Warning for significant budget changes
- Confirmation for budget deletions
- Duplicate category prevention

3.4 Alert System & Thresholds

- Configurable alert threshold (default 10%)
- User input for threshold percentage (1-100%)

- Color-coded alerts:
  - Green: Below threshold
  - Yellow: At threshold
  - Red: Over budget
- Threshold persistence in settings
- Real-time threshold updates across interface

3.5 Budget Analysis & Reporting

- Calculate total monthly budget
- Show budget utilization percentages
- Compare month-to-month budget changes
- Identify categories with frequent overruns
- Show budget trend analysis
- Export budget data to Excel/CSV

3.6 Budget Lifecycle Management

- Handle EMI completion dates
- Manage investment start/end dates
- Track recurring vs one-time expenses
- Support for seasonal budget adjustments
- Automatic budget transitions

3.7 Advanced Budget Features

- Budget rollover for unused amounts
- Category priority ranking
- Emergency fund allocation
- Savings goal tracking
- Investment allocation planning

---

4. CATEGORY ANALYSIS TAB - DEEP SPENDING INSIGHTS

Feature: Comprehensive category-wise analysis with advanced visualization

User Stories with Detailed Descriptions:

US-4.1: Long-term Category Spending Patterns

- **User Story**: As a user, I want to see spending patterns for each category over 2 years

- **Description**: Users need comprehensive visibility into how their spending in each category has evolved over the 24-month planning period. This includes trend analysis showing increasing/decreasing patterns, seasonal variations, and anomaly detection for unusual spending months. The analysis should provide insights into spending consistency, volatility, and long-term trajectories for each category with visual trend lines and statistical summaries.
- **Business Value**: Enables data-driven decisions about category budget adjustments and spending optimization
- **Acceptance Criteria**: 2-year trends are clearly visualized with statistical insights and pattern recognition

US-4.2: Interactive Category Comparisons

- **User Story**: As a user, I want visual comparisons between categories
- **Description**: The system should provide side-by-side category analysis through various visualization types including pie charts (spending distribution), bar charts (category comparisons), and stacked charts (monthly breakdowns). Users should be able to select multiple categories for comparison, adjust time periods, and see relative performance metrics. Visualizations should be interactive with drill-down capabilities and export options.
- **Business Value**: Facilitates informed budget reallocation and spending prioritization decisions
- **Acceptance Criteria**: Category comparisons are visually clear with interactive features and multiple chart types

US-4.3: Overspending Pattern Identification

- **User Story**: As a user, I want to identify which categories I consistently overspend on
- **Description**: The system should analyze historical data to identify categories where actual spending regularly exceeds budgeted amounts. This includes frequency analysis (how often overspending occurs), magnitude analysis (by how much), and trend analysis (is overspending getting worse or better). Users should receive specific recommendations for budget adjustments and spending behavior modifications based on these patterns.
- **Business Value**: Improves budget realism and helps focus spending discipline efforts on problematic categories
- **Acceptance Criteria**: Overspending patterns are clearly identified with specific recommendations and trend analysis

US-4.4: Seasonal Spending Analysis

- **User Story**: As a user, I want seasonal spending pattern analysis
- **Description**: The system should detect and display seasonal variations in spending across categories, such as increased food expenses during festivals, higher medical costs during certain months, or seasonal shopping patterns. Analysis should identify recurring seasonal patterns,

predict future seasonal needs, and suggest budget adjustments to accommodate these patterns. Visual indicators should highlight seasonal peaks and valleys.
- **Business Value**: Enables better budget planning that accounts for predictable seasonal variations
- **Acceptance Criteria**: Seasonal patterns are automatically detected and clearly presented with predictive insights

US-4.5: Detailed Category Deep-dive

- **User Story**: As a user, I want to drill down into category details
- **Description**: Users should be able to click on any category to access detailed analysis including all transactions within that category, monthly spending progression, budget vs actual history, and performance metrics. The deep-dive should show transaction-level details, identify largest expenses, detect unusual transactions, and provide category-specific insights and recommendations.
- **Business Value**: Provides detailed understanding of spending drivers and opportunities for optimization
- **Acceptance Criteria**: Category drill-down provides comprehensive details with transaction-level visibility

US-4.6: Category Export and Reporting

- **User Story**: As a user, I want to export category analysis reports
- **Description**: Users should be able to generate and export detailed reports for specific categories or category groups. Reports should include charts, trend analysis, statistical summaries, and recommendations in professional formats (PDF, Excel). Export should allow customization of time periods, chart types, and level of detail. Reports should be suitable for sharing with financial advisors or family members.
- **Business Value**: Enables external sharing and professional financial planning discussions
- **Acceptance Criteria**: Category reports are comprehensive, professional, and exportable in multiple formats

Granular Tasks:

4.1 Category Overview Display

- Summary table with columns: Category, Total Spent, Average Monthly, Budget Allocated, Variance
- Calculate spending totals for each category across all months
- Show average monthly spending per category
- Display budget vs actual for each category
- Color-code categories based on performance

4.2 Category Visualization

- Pie chart showing spending distribution by category
- Bar chart comparing categories side-by-side
- Stacked bar chart showing monthly breakdown by category
- Line chart showing category trends over 2 years
- Sparklines for quick trend visualization
- Heatmap showing spending intensity by month/category

4.3 Detailed Category Analysis

- Click on category to view detailed breakdown
- Show monthly spending for selected category
- Display budget vs actual trend for category
- Identify overspend months for category
- Calculate category spending volatility
- Show spending pattern consistency

4.4 Category Comparison Features

- Multi-select categories for comparison
- Side-by-side spending comparison
- Relative performance analysis
- Category ranking by various metrics
- Correlation analysis between categories

4.5 Time-based Analysis

- Monthly trend analysis for each category
- Seasonal pattern identification
- Year-over-year comparison (when applicable)
- Quarter-wise spending analysis
- Week-wise spending patterns

4.6 Category Performance Metrics

- Budget adherence percentage per category
- Spending consistency scoring
- Overspend frequency calculation
- Average overspend amount when exceeded

- Best/worst performing categories

4.7 Advanced Analytics

- Identify spending anomalies
- Detect spending pattern changes
- Forecast future category spending
- Calculate category-wise inflation impact
- Suggest budget adjustments based on patterns

4.8 Category Management

- Add new custom categories
- Merge similar categories
- Split categories for better granularity
- Set category-specific alert thresholds
- Category description and notes

---

5. WHAT-IF SIMULATOR TAB - SCENARIO PLANNING ENGINE

Feature: Advanced financial scenario simulation and planning

User Stories:

- As a user, I want to simulate multiple budget changes simultaneously
- As a user, I want to see the impact of increasing/decreasing specific categories
- As a user, I want to model one-time expenses and their impact
- As a user, I want to simulate investment pausing/resuming
- As a user, I want to save scenarios for future reference
- As a user, I want visual comparisons of different scenarios
- As a user, I want to export scenario analysis

Granular Tasks:

5.1 Scenario Creation & Management

- Add scenario dialog with fields:
  - Scenario name
  - Scenario type (Budget Change, One-time Event, Investment Adjustment)
  - Category affected
  - Amount change (+/-)

    - Start month
    - End month
    - Description/notes
- Scenario list management (add, edit, delete, duplicate)
- Scenario categorization and tagging
- Scenario templates for common changes

5.2 Scenario Types Implementation

- **Budget Changes**: Increase/decrease category budgets
    - MF increase by ₹2,000 from Jan-26
    - Swiggy reduction by ₹1,500 for 6 months
- **One-time Events**: Large expenses or income
    - Hospital expense of ₹50,000 in Mar-26
    - Bonus income of ₹1,00,000 in Dec-25
- **Investment Adjustments**: Pause/resume investments
    - Pause Gold investment for 3 months
    - Double PPF contribution for tax saving

5.3 Scenario Application & Calculation

- Apply multiple scenarios simultaneously
- Calculate combined impact on budgets
- Show month-by-month effect of scenarios
- Handle overlapping scenarios intelligently
- Maintain original data integrity (temporary changes only)

5.4 Scenario Impact Visualization

- Before/after comparison charts
- Side-by-side budget comparison tables
- Impact timeline showing changes over months
- Category-wise impact analysis
- Total impact summary with key metrics

5.5 Scenario Analysis & Insights

- Calculate net savings impact
- Show budget sufficiency analysis

- Identify months with budget shortfalls
- Calculate required adjustments for balance
- Risk assessment for scenarios

5.6 Scenario Persistence & Sharing

- Save scenarios to JSON file
- Load saved scenarios on startup
- Export scenario results to Excel/PDF
- Scenario comparison reports
- Scenario documentation and notes

5.7 Advanced Scenario Features

- Scenario probability weighting
- Monte Carlo simulation for uncertainty
- Sensitivity analysis for key variables
- Scenario optimization suggestions
- Automated scenario recommendations

5.8 Scenario Validation

- Check scenario feasibility
- Warn about unrealistic scenarios
- Validate date ranges and amounts
- Conflict detection between scenarios
- Impact magnitude warnings

---

6. REPORTS TAB - COMPREHENSIVE REPORTING SYSTEM

Feature: Multi-format export and professional reporting

User Stories:

- As a user, I want to export my financial data in multiple formats
- As a user, I want reports with embedded charts and visualizations
- As a user, I want to customize report date ranges and categories
- As a user, I want professional-looking reports for sharing
- As a user, I want automated monthly report generation
- As a user, I want backup and restore capabilities

Granular Tasks:

6.1 Export Format Support

- **JSON Export**: Complete data backup
  - All budgets, transactions, scenarios, settings
  - Structured format for easy parsing
  - Compressed option for large datasets
- **Excel Export**: Professional spreadsheets
  - Multiple sheets: Summary, Transactions, Budgets, Categories, Scenarios
  - Formatted tables with headers and styling
  - Embedded charts and graphs
  - Conditional formatting for alerts
- **CSV Export**: Simple data format
  - Transactions export with all fields
  - Budget export by month/category
  - Category summary export
- **PDF Export**: Professional reports
  - Executive summary with key metrics
  - Charts and visualizations embedded
  - Professional formatting and layout

6.2 Report Customization

- Date range selection (specific months or full 2-year range)
- Category filtering (select specific categories)
- Report template selection
- Custom report title and description
- Logo and branding options
- Currency and number formatting options

6.3 Automated Report Generation

- Monthly report auto-generation at month-end
- Scheduled report generation
- Email delivery options (future enhancement)
- Report naming conventions

- Archive management for old reports

6.4 Report Content & Structure

- **Executive Summary**:
  - Total spending vs budget
  - Key performance indicators
  - Month-over-month changes
  - Top spending categories
- **Detailed Analysis**:
  - Category-wise breakdown
  - Monthly trends
  - Budget adherence rates
  - Spending pattern analysis
- **Visual Elements**:
  - Spending distribution pie charts
  - Trend line graphs
  - Budget vs actual bar charts
  - Alert summary tables

6.5 Export Process Management

- File dialog for save location selection
- Progress indicators for large exports
- Export validation and error handling
- Success confirmation with file location
- Export history tracking

6.6 Data Backup & Restore

- Complete data backup functionality
- Restore from backup with validation
- Backup scheduling options
- Multiple backup retention
- Data integrity verification

---

7. DATA PERSISTENCE & STORAGE SYSTEM

Feature: Robust data management with backup and recovery

User Stories:

- As a user, I want my data automatically saved without manual intervention
- As a user, I want data recovery if files become corrupted
- As a user, I want efficient storage that doesn't slow down the app
- As a user, I want data migration support for future app updates
- As a user, I want multiple backup options
- As a user, I want data validation and integrity checking

Granular Tasks:

7.1 JSON Data Structure Design

```json
```

```
{
  "categories": {
    "Loans & EMIs": [
      "Credit Card EMI 1", "Credit Card EMI 2",
      "Personal Loan EMI 1", "Personal Loan EMI 2", "Home Loan EMI"
    ],
    "Investments": [
      "Mutual Fund SIP", "PPF", "RD", "Ponmagan Policy",
      "Gold Investment", "Bitcoin Investment",
      "Baby Health Policy", "Baby Education Policy"
    ],
    "Lifestyle & Essentials": [
      "OTT Subscriptions", "Hospital", "Swiggy/Food",
      "Petrol", "General Expenses", "Shopping"
    ],
    "Custom": []
  },
  "budgets": {
    "Aug-25": {
      "Credit Card EMI 1": 15000,
      "Mutual Fund SIP": 5000,
      "Hospital": 10000,
      "Swiggy/Food": 5000,
      ...
    },
    "Sep-25": {...}
  },
  "transactions": {
    "Aug-25": [
      {
        "id": "uuid-string",
        "date": "2025-08-15",
        "category": "Credit Card EMI 1",
        "amount": 15000,
        "description": "Monthly Credit Card EMI",
        "source": "manual"
      }
    ]
  },
  "scenarios": {
    "scenario_1": {
      "name": "Increase SIP Investment",
      "type": "budget_change",
      "changes": [
        {
          "category": "Mutual Fund SIP",
```

```json
          "amount_change": 2000,
          "start_month": "Jan-26",
          "end_month": "Dec-26"
        }
      ]
    }
  },
  "settings": {
    "alert_threshold": 10,
    "last_selected_month": "Aug-25",
    "ui_preferences": {...},
    "category_keywords": {
      "Credit Card EMI 1": ["CC1", "CREDIT CARD", "EMI"],
      "Mutual Fund SIP": ["MF", "SIP", "MUTUAL FUND"],
      "Hospital": ["HOSPITAL", "CLINIC", "MEDICAL"],
      ...
    }
  }
}
```

7.2 Auto-save Implementation

- Save after every transaction addition/modification
- Save after budget changes
- Save after scenario modifications
- Save settings changes immediately
- Debounced saving to prevent excessive file writes
- Atomic file operations to prevent corruption

7.3 Data Loading & Validation

- Load data.json on application startup
- Validate JSON structure and data types
- Handle missing or corrupted files gracefully
- Default data creation for new installations
- Data migration for schema updates
- Error reporting for loading issues

7.4 Data Integrity & Backup

- Daily backup creation (last 7 days retained)
- Manual backup on demand

- Backup validation and verification
- Restore from backup functionality
- Data corruption detection and recovery
- Checksum validation for critical data

7.5 Performance Optimization

- Lazy loading for large datasets
- In-memory caching for frequently accessed data
- Efficient data indexing for quick searches
- Memory management for 2-year datasets
- Optimized JSON serialization

7.6 Data Migration & Updates

- Version tracking in data files
- Automatic migration for data structure changes
- Backward compatibility maintenance
- Migration rollback capability
- Update validation and testing

---

8. USER INTERFACE & EXPERIENCE SYSTEM

Feature: Professional, intuitive interface with real-time updates

User Stories:

- As a user, I want a clean, professional interface that's easy to navigate
- As a user, I want consistent visual design across all tabs
- As a user, I want real-time updates when data changes
- As a user, I want keyboard shortcuts for common actions
- As a user, I want helpful error messages and validation
- As a user, I want the app to work smoothly even with large datasets

Granular Tasks:

8.1 GUI Framework & Architecture

- Tkinter with ttk widgets for professional appearance
- Tabbed interface using ttk.Notebook
- Responsive layout design for different screen sizes

- Consistent color scheme and theming
- Professional fonts and spacing
- Icon integration for better UX

8.2 Real-time Updates & Synchronization

- Auto-refresh all tabs when data changes
- Event-driven updates between components
- Efficient re-rendering to avoid flickering
- State synchronization across tabs
- Real-time chart updates
- Live calculation updates

8.3 Input Validation & Error Handling

- Real-time input validation with visual feedback
- User-friendly error messages
- Input format guidance and hints
- Field validation before submission
- Graceful error recovery
- Validation summary for complex forms

8.4 Chart Integration & Visualization

- Optional matplotlib integration with fallback
- Embedded charts in tab frames
- Interactive chart features (zoom, pan)
- Chart export functionality
- Responsive chart sizing
- Professional chart styling

8.5 Navigation & Usability

- Intuitive tab organization
- Breadcrumb navigation where applicable
- Quick action buttons and shortcuts
- Context menus for advanced operations
- Drag-and-drop functionality where useful
- Accessibility features

8.6 Performance & Responsiveness

- Efficient rendering for large datasets
- Progress indicators for long operations
- Background processing for heavy tasks
- Memory management and optimization
- Smooth animations and transitions
- Fast startup and shutdown

8.7 Keyboard Shortcuts & Quick Actions

- Ctrl+N: New transaction
- Ctrl+I: Import transactions
- Ctrl+S: Manual save
- Ctrl+E: Export current view
- F5: Refresh current tab
- Ctrl+Tab: Switch between tabs

8.8 Help & Documentation

- Tooltips for all interactive elements
- Contextual help text
- Quick start guide
- Feature documentation
- Error code explanations
- Troubleshooting guides

---

9. ADVANCED ANALYTICS & INSIGHTS SYSTEM

Feature: Intelligent financial guidance and pattern recognition

User Stories:

- As a user, I want smart recommendations based on my spending patterns
- As a user, I want to compare my spending against historical averages
- As a user, I want seasonal pattern recognition and adjustments
- As a user, I want early warnings for potential budget overruns
- As a user, I want trend analysis and forecasting

Granular Tasks:

9.1 Pattern Recognition Engine

- Identify recurring spending patterns
- Detect seasonal variations in spending
- Recognize irregular spending anomalies
- Track spending habit changes over time
- Identify correlated spending categories

9.2 Smart Recommendations

- Budget adjustment suggestions based on patterns
- Category optimization recommendations
- Savings opportunity identification
- Spending efficiency improvements
- Investment timing suggestions

9.3 Predictive Analytics

- Forecast future spending based on trends
- Predict budget overrun risks
- Estimate savings potential
- Project long-term financial health
- Seasonal adjustment predictions

9.4 Comparative Analysis

- Rolling average comparisons (3/6/12 months)
- Year-over-year comparisons
- Category performance benchmarking
- Spending volatility analysis
- Improvement tracking over time

9.5 Alert Intelligence

- Proactive overspending warnings
- Trend-based risk alerts
- Unusual spending pattern notifications
- Budget rebalancing suggestions
- Goal achievement tracking

---

10. CONFIGURATION & SETTINGS SYSTEM

Feature: Comprehensive application configuration management

User Stories:

- As a user, I want to customize alert thresholds for my preferences
- As a user, I want to configure categories for my spending patterns
- As a user, I want to set up automatic backup schedules
- As a user, I want to customize the interface to my preferences
- As a user, I want to manage import/export settings

Granular Tasks:

10.1 Alert Configuration

- Global alert threshold setting (default 10%)
- Category-specific thresholds
- Alert notification preferences
- Warning level customization
- Alert history and management

10.2 Category Management

- Add/remove custom categories within existing groups
- Category color coding and visual identification
- Category grouping and hierarchy (Loans & EMIs, Investments, Lifestyle & Essentials)
- Default category assignments and keyword mapping
- Category templates and presets for quick setup
- Support for the following category structure:

Loans & EMIs Group:

- Credit Card EMI 1, Credit Card EMI 2
- Personal Loan EMI 1, Personal Loan EMI 2
- Home Loan EMI
- Future loan categories as needed

Investments Group:

- Mutual Fund SIP, PPF, RD
- Ponmagan Policy, Gold Investment, Bitcoin Investment

- Baby Health Policy, Baby Education Policy
- Future investment categories as needed

Lifestyle & Essentials Group:

- OTT Subscriptions, Hospital, Swiggy/Food
- Petrol, General Expenses, Shopping
- Future lifestyle categories as needed

Custom Categories:

- User-defined categories for specific needs
- Temporary or project-specific categories
- Categories for special events or goals

10.3 Import/Export Settings

- Default export formats and locations
- Import column mapping preferences
- Auto-categorization rules management
- File handling preferences
- Data validation settings

10.4 Interface Customization

- Theme and color scheme selection
- Layout preferences and window sizing
- Default tab and view settings
- Chart preferences and styling
- Language and localization options

10.5 Data Management Settings

- Backup frequency and retention
- Data archival policies
- Performance optimization settings
- Memory usage configuration
- Cache management options

TECHNICAL IMPLEMENTATION REQUIREMENTS

Technology Stack

- **Language**: Python 3.8+
- **GUI Framework**: Tkinter with ttk widgets
- **Data Storage**: JSON files for lightweight persistence
- **Visualization**: Matplotlib (optional dependency)
- **Data Processing**: Pandas for Excel operations (optional)
- **File Handling**: Built-in Python libraries

Default Category Configuration

```python
DEFAULT_CATEGORIES = {
    "Loans & EMIs": [
        "Credit Card EMI 1",
        "Credit Card EMI 2",
        "Personal Loan EMI 1",
        "Personal Loan EMI 2",
        "Home Loan EMI"
    ],
    "Investments": [
        "Mutual Fund SIP",
        "PPF",
        "RD",
        "Ponmagan Policy",
        "Gold Investment",
        "Bitcoin Investment",
        "Baby Health Policy",
        "Baby Education Policy"
    ],
    "Lifestyle & Essentials": [
        "OTT Subscriptions",
        "Hospital",
        "Swiggy/Food",
        "Petrol",
        "General Expenses",
        "Shopping"
    ],
    "Custom": [
        # User-defined categories added here
    ]
}
```

Category Management Features

- **Dynamic Category Addition**: Users can add new categories at runtime
- **Category Grouping**: Categories organized by type for better management
- **Category Validation**: Prevent duplicates and ensure proper naming
- **Category Migration**: Handle category changes in existing data
- **Category Templates**: Pre-defined sets for quick setup
- **Category Export/Import**: Share category configurations between users

---

IMPLEMENTATION PRIORITY & PHASES

Phase 1: Core Foundation (Week 1-2)

1. Complete GUI framework with all tabs
2. Basic transaction management (add, view, edit)
3. Simple budget planning functionality
4. JSON persistence implementation
5. Basic dashboard with summary display

Phase 2: Advanced Features (Week 3-4)

1. CSV/Excel import functionality
2. Category analysis with basic charts
3. What-If simulator basic functionality
4. Alert system implementation
5. Export functionality (JSON, Excel)

Phase 3: Visualization & Analytics (Week 5-6)

1. Complete chart integration
2. Advanced category analysis
3. Complete What-If simulation
4. Trend analysis and forecasting
5. Advanced reporting features

Phase 4: Polish & Enhancement (Week 7-8)

1. UI/UX improvements and polish
2. Performance optimization
3. Advanced error handling

4. Documentation and help system
5. Testing and quality assurance

SUCCESS CRITERIA & VALIDATION

Functional Requirements Validation

- [ ] All 6 tabs fully functional with intended features
- [ ] Complete transaction lifecycle (add, edit, delete, import)
- [ ] 2-year budget planning with lifecycle changes
- [ ] Real-time alerts and status indicators
- [ ] What-If simulation with multiple scenario types
- [ ] Comprehensive export functionality
- [ ] Robust data persistence and backup

Performance Requirements

- [ ] Application startup < 3 seconds
- [ ] Data operations complete < 1 second
- [ ] Chart rendering < 2 seconds
- [ ] Import processing < 5 seconds for 1000 transactions
- [ ] Memory usage < 200MB for 2-year dataset

Usability Requirements

- [ ] Intuitive navigation requiring minimal learning
- [ ] Consistent visual design across all components
- [ ] Clear error messages and validation feedback
- [ ] Responsive interface adapting to user actions
- [ ] Professional appearance suitable for personal finance management

Reliability Requirements

- [ ] Graceful handling of missing optional dependencies
- [ ] Data corruption recovery mechanisms
- [ ] Consistent data integrity across all operations
- [ ] Reliable backup and restore functionality
- [ ] Error recovery without data loss

FINAL DELIVERABLES

Application Components

1. **Complete Desktop Application** with all specified features

2. **Source Code** with modular architecture and documentation
3. **User Guide** with installation and usage instructions
4. **Technical Documentation** for maintenance and enhancement
5. **Test Suite** for validation and regression testing

Data Management

1. **Data Import Templates** for common bank statement formats
2. **Budget Templates** for quick setup and recurring patterns
3. **Export Templates** for professional reporting
4. **Backup Procedures** for data safety and recovery

This comprehensive requirements document ensures that no detail is missed and provides a complete roadmap for implementing the Financial Management & Simulation Tool with all requested features and functionality.