1 Test Plan

1.1 Detailed Test Case Descriptions

This section provides detailed descriptions of all planned test cases that will be used to verify the ***SmartBudget*** system. Each test case includes its ID, related requirement, preconditions, steps, and expected results. These details will guide future testing activities, making it easy for developers and testers to perform and track tests systematically once it is implemented.

1.1.1 TC-001

- **Title:** Add Expense – Valid Inputs
- **Related Requirement ID(s):** FR-001; UC-1; US-1
- **Preconditions:** App is running; "Transactions" view available.
- **Test Steps:**
  1- Open Transactions; Click "Add Transaction"; Select Expense.
  2- Enter valid amount (e.g., 25.50), pick a valid date, choose category (e.g., Food), add optional note.
  3- Click "Save".
- **Expected Results:** Transaction appears in list; totals/summary update; success feedback shown; data written to local storage.
- **Actual Result:**
- **Status:**
- **Priority:** High

1.1.2 TC-002

- **Title:** Add Income – Valid Inputs
- **Related Requirement ID(s):** FR-001; UC-1; US-1
- **Preconditions:** App is running; "Transactions" view available.
- **Test Steps:**
  1- Open Transactions; Click "Add Transaction"; Select Income.
  2- Enter valid amount (e.g., 25.50), pick a valid date, choose category (e.g., Salary), add optional note.

3- Click “Save”.

- **Expected Results:** Income row added; income and balance totals increase; success feedback; persisted.
- **Actual Result:**
- **Status:**
- **Priority:** High

1.1.3 TC-003

- **Title:** Add Transaction – Reject Invalid Amount
- **Related Requirement ID(s):** FR-001; UC-1; US-1
- **Preconditions:** App is running; “Transactions” view available.
- **Test Steps:**

  1- Open Transactions; Click “Add Transaction”; Select Income/Expense.

  2- Enter non-numeric (e.g., “abc”) or negative amount.

  3- Click “Save”.
- **Expected Results:** Save blocked; clear validation message; no record created; no totals change.
- **Actual Result:**
- **Status:**
- **Priority:** High

1.1.4 TC-004

- **Title:** Add Transaction – Reject Missing Required Fields
- **Related Requirement ID(s):** FR-001; UC-1; US-1
- **Preconditions:** App running.
- **Test Steps:**

  1- Add Transaction.

  2- Leave amount or date or type empty.

  3- Click “Save”.
- **Expected Results:** Save blocked; specific error message; no changes persisted.
- **Actual Result:**
- **Status:**
- **Priority:** High

1.1.5 TC-005

- **Title:** Edit Transaction – Totals Recalculate
- **Related Requirement ID(s):** FR-001; UC-2; US-2
- **Preconditions:** At least one existing transaction.
- **Test Steps:**

  1- Open an existing transaction.

  2- Change amount (e.g., 20 → 30).

  3- Click “Save”.
- **Expected Results:** Row updates; affected totals/summary recalculates; success feedback; persisted.
- **Actual Result:**
- **Status:**
- **Priority:** High

1.1.6 TC-006

- **Title:** Delete Transaction – Confirmation and Totals Update
- **Related Requirement ID(s):** FR-001; UC-2; US-2
- **Preconditions:** At least one existing transaction.
- **Test Steps:**

  4- Open an existing transaction.

  1- Click delete.

  2- Confirm deletion.
- **Expected Results:** Row removed; totals/summary recalculates; confirmation used to prevent accidental deletion; persisted removal.
- **Actual Result:**
- **Status:**
- **Priority:** High

1.1.7 TC-007

- **Title:** Delete Transaction – Cancel Preservation
- **Related Requirement ID(s):** FR-001; UC-2; US-2
- **Preconditions:** At least one existing transaction.
- **Test Steps:**

3- Open an existing transaction.

4- Click delete.

1- In confirmation, click Cancel.

- **Expected Results:** No deletion; list and totals unchanged.
- **Actual Result:**
- **Status:**
- **Priority:** Medium

1.1.8 TC-008

- **Title:** Persistence – Transactions Survive App Restart
- **Related Requirement ID(s):** FR-001; US-8
- **Preconditions:** At least one existing transaction.
- **Test Steps:**

  2- Close app.

  3- Reopen app.

  4- Go to Transactions.
- **Expected Results:** Previously saved transactions load automatically; no manual import required.
- **Actual Result:**
- **Status:**
- **Priority:** High

1.1.9 TC-009

- **Title:** View All Transactions – Default List/Table
- **Related Requirement ID(s):** FR-002; UC-3; US-3
- **Preconditions:** At least one existing transaction.
- **Test Steps:**

  1- Open Transactions view.
- **Expected Results:** All transactions visible in a readable table/list with key columns; totals visible.
- **Actual Result:**
- **Status:**
- **Priority:** High

1.1.10 TC-010

- **Title:** Filter by Category (Single Category)
- **Related Requirement ID(s):** FR-002; UC-3; US-3
- **Preconditions:** Transactions exist across multiple categories.
- **Test Steps:**

  1- Apply category filter (e.g., Food).
- **Expected Results:** Only rows with category = Food shown; totals/summary reflect filtered subset.
- **Actual Result:**
- **Status:**
- **Priority:** High

1.1.11 TC-011

- **Title:** Filter by Date Range
- **Related Requirement ID(s):** FR-002; UC-3; US-3
- **Preconditions:** Transactions exist across multiple dates.
- **Test Steps:**

  1- Set start and end date.
- **Expected Results:** Only transactions within range shown; totals reflect range.
- **Actual Result:**
- **Status:**
- **Priority:** High

1.1.12 TC-012

- **Title:** Combined Filters – Date Range + Category
- **Related Requirement ID(s):** FR-002; UC-3; US-3
- **Preconditions:** Data spans categories and dates.
- **Test Steps:**

  2- Apply date range.

  1- Apply category.
- **Expected Results:** List narrows to intersection; totals update accordingly.
- **Actual Result:**
- **Status:**

- **Priority:** High

1.1.13 TC-013

- **Title:** Clear Filters – No Matches
- **Related Requirement ID(s):** FR-002; UC-3; US-3
- **Preconditions:** Data set present.
- **Test Steps:**

  1- Apply filters that yield zero results.
- **Expected Results:** Empty list message + zero totals displayed clearly.
- **Actual Result:**
- **Status:**
- **Priority:** Medium

1.1.14 TC-014

- **Title:** Clear Filters – Restore Full List
- **Related Requirement ID(s):** FR-002; UC-3; US-3
- **Preconditions:** Active filter(s) applied.
- **Test Steps:**

  1- Click “Clear Filters”.
- **Expected Results:** All transactions visible again; totals reflect full data set.
- **Actual Result:**
- **Status:**
- **Priority:** Medium

1.1.15 TC-015

- **Title:** Keyword Search – Narrows List
- **Related Requirement ID(s):** FR-002; UC-3; US-3
- **Preconditions:** Keyword Search – Narrows List.
- **Test Steps:**

  1- Enter keyword (e.g., “coffee”) in search.
- **Expected Results:** Only matching rows shown (by configured fields: note, payee, category); totals update; clearing search restores list.
- **Actual Result:**

- **Status:**
- **Priority:** Medium

1.1.16 TC-016

- **Title:** Set Monthly Budget – Valid Positive Amount
- **Related Requirement ID(s):** FR-003; UC-4; US-4
- **Preconditions:** Category exists (e.g., Food).
- **Test Steps:**

  1- Open Budgets.

  2- Select “Food”.

  3- Enter 200.00.

  4- Save
- **Expected Results:** Budget stored for current month; remaining allowance visible.
- **Actual Result:**
- **Status:**
- **Priority:** Medium

1.1.17 TC-017

- **Title:** Set Monthly Budget – Reject Invalid Inputs
- **Related Requirement ID(s):** FR-003; UC-4; US-4
- **Preconditions:** Budgets view open.
- **Test Steps:**

  1- Enter negative or blank budget.

  2- Save
- **Expected Results:** Save blocked; clear error message; no budget created/changed.
- **Actual Result:**
- **Status:**
- **Priority:** Medium

1.1.18 TC-018

- **Title:** Update Existing Budget – Recalculate Remaining

- **Related Requirement ID(s):** FR-003; UC-4; US-4
- **Preconditions:** Budget already set for category; some spending exists.
- **Test Steps:**

  1- Edit Food budget 200 → 250.

  2- Save
- **Expected Results:** New limit stored; remaining allowance recalculated and displayed.
- **Actual Result:**
- **Status:**
- **Priority:** Medium

1.1.19 TC-019

- **Title:** Overspending Alert – Trigger at Equal to Limit
- **Related Requirement ID(s):** FR-003; UC-6; US-5
- **Preconditions:** Budget = 100 for Food; current spending = 90.
- **Test Steps:**

  1- Add Food expense of 10.
- **Expected Results:** Alert appears indicating limit reached (category shown); remains visible until acknowledged/reset.
- **Actual Result:**
- **Status:**
- **Priority:** Medium

1.1.20 TC-020

- **Title:** Overspending Alert – Trigger when Exceeding Limit
- **Related Requirement ID(s):** FR-003; UC-6; US-5
- **Preconditions:** Budget = 100 for Food; current spending = 95.
- **Test Steps:**

  1- Add Food expense of 10 (total 105).
- **Expected Results:** Alert shows category and overage amount (e.g., +5); persists until acknowledged/reset.
- **Actual Result:**
- **Status:**

- **Priority:** Medium

1.1.21 TC-021

- **Title:** No Budget Defined – No Alert
- **Related Requirement ID(s):** FR3; UC-6; US-5
- **Preconditions:** No budgets set.
- **Test Steps:**

  1- Add multiple expenses in any category.
- **Expected Results:** No budget alert appears.
- **Actual Result:**
- **Status:**
- **Priority:** Medium

1.1.22 TC-022

- **Title:** Monthly Summary – Correct Totals
- **Related Requirement ID(s):** FR-004; UC-5; US-6
- **Preconditions:** Month has known set of incomes/expenses.
- **Test Steps:**

  1- Open Reports.

  2- Select "This Month".
- **Expected Results:** Income, expenses, and balance compute correctly and match dataset.
- **Actual Result:**
- **Status:**
- **Priority:** High

1.1.23 TC-023

- **Title:** Custom Range Summary – Correct Totals
- **Related Requirement ID(s):** FR-004; UC-5; US-6
- **Preconditions:** Data across multiple months.
- **Test Steps:**

  1- Choose custom start/end dates across months.
- **Expected Results:** Totals reflect only the selected range; displayed clearly.

- **Actual Result:**
- **Status:**
- **Priority:** Medium

1.1.24 TC-024

- **Title:** Summary – No Data Path
- **Related Requirement ID(s):** FR-004; UC-5; US-7
- **Preconditions:** Select a period with no transactions.
- **Test Steps:**

  1- Open Reports → empty range.
- **Expected Results:** Zeros shown for totals; clear "no records found" note.
- **Actual Result:**
- **Status:**
- **Priority:** Medium

1.1.25 TC-025

- **Title:** Category Chart – Accurate Visualization
- **Related Requirement ID(s):** FR-004; UC-5; US-7
- **Preconditions:** Transactions across multiple categories.
- **Test Steps:**

  1- Open Reports with chart (pie/bar).
- **Expected Results:** Chart values/percentages match underlying totals; labels/legend are clear.
- **Actual Result:**
- **Status:**
- **Priority:** Low

1.1.26 TC-026

- **Title:** Reports/Charts Auto-Update on Data/Filter Change
- **Related Requirement ID(s):** FR-004; UC-5; US-7
- **Preconditions:** Reports open with visible chart; baseline totals noted.
- **Test Steps:**

  1- Add/edit/delete a transaction OR change filters.

- **Expected Results:** Summary and chart refresh immediately to reflect the change.
- **Actual Result:**
- **Status:**
- **Priority:** Medium

NFR-Oriented Test Cases

1.1.27 TC-027

- **Title:** Usability Heuristics – Labels, Fonts, Navigation
- **Related Requirement ID(s):** NFR-Usability
- **Preconditions:** UI available (Transactions, Budgets, Reports).
- **Test Steps:**

  1- Inspect labels, font sizes, contrast.

  2- Walk through navigation paths between key views.
- **Expected Results:** Clear labels; readable fonts; predictable navigation; consistent UI components.
- **Actual Result:**
- **Status:**
- **Priority:** Medium

1.1.28 TC-028

- **Title:** Performance – Typical Dataset Responsiveness
- **Related Requirement ID(s):** NFR- Performance
- **Preconditions:** Seed transactions locally.
- **Test Steps:**

  1- Launch app and open Transactions.

  2- Apply filters/search repeatedly.
- **Expected Results:** Loads and interactions complete without noticeable lag; list/filters remain responsive.
- **Actual Result:**
- **Status:**
- **Priority:** Medium

1.1.29 TC-029

- **Title:** Reliability – Immediate Save & Graceful Invalid Handling
- **Related Requirement ID(s):** NFR- Reliability
- **Preconditions:** App running.
- **Test Steps:**

  1- Add/edit/delete; immediately kill and relaunch app.

  2- Attempt invalid inputs on add/edit.
- **Expected Results:** Most recent changes present after relaunch (immediate save); invalid inputs yield clear feedback; no data corruption.
- **Actual Result:**
- **Status:**
- **Priority:** High

1.1.30 TC-030

- **Title:** Privacy/Offline – No Network Dependency
- **Related Requirement ID(s):** Privacy/Offline
- **Preconditions:** Device offline (disable network).
- **Test Steps:**

  1- Launch app.

  2- Perform add/view/filter/report flows.
- **Expected Results:** All features function offline; no network calls attempted.
- **Actual Result:**
- **Status:**
- **Priority:** High

1.1.31 TC-031

- **Title:** Portability – Run on Windows/macOS/Linux
- **Related Requirement ID(s):** NFR-Portability
- **Preconditions:** Build packaged with supported Java runtime.
- **Test Steps:**

  1- Start app on each OS.

  2- Smoke test add/view flows.
- **Expected Results:** App launches and core flows work on all three platforms.

- **Actual Result:**
- **Status:**
- **Priority:** Medium

1.1.32 TC-032

- **Title:** Maintainability – Modular Structure & Unit Tests Present
- **Related Requirement ID(s):** NFR-Maintainability
- **Preconditions:** Repo accessible.
- **Test Steps:**

  1- Review project structure (e.g., MVC separation.

  2- Verify presence of unit tests for core model/services.
- **Expected Results:** Clear modular separation; baseline unit tests exist; naming/packaging consistent.
- **Actual Result:**
- **Status:**
- **Priority:** Medium