Firefly III Test Plan

1. Scope

This test plan covers the testing of **Accounts Management, Transactions, Reports, Linking, and Validations** in Firefly III, an open-source personal finance management application. The goal is to ensure that all core financial functionalities work correctly, transactions are recorded accurately, and reports generate correct insights.

2. Objective

- Validate the functionality of account creation, editing, and deletion.
- Ensure transactions are processed correctly and linked to appropriate accounts.
- Verify reports display accurate financial data.
- Test the linking of accounts and transaction impact.
- Validate all input fields and ensure error handling is implemented properly.

3. Test Strategy

- **Manual Testing**: Functional, UI, and exploratory testing.
- **Automation Testing** (if applicable in later stages): Selenium/WebDriver for UI automation, API testing for data validation.
- **Test Environment**: Firefly III Demo environment (https://demo.firefly-iii.org/)
- **Test Data**: Sample financial data including income, expenses, loans, and transfers.

4. Test Levels

- **Unit Testing**: Developers will test individual components before integration.
- **Integration Testing**: Ensuring different modules (accounts, transactions, reports) interact correctly.
- **System Testing**: End-to-end testing of the Firefly III application.

5. Risks & Mitigation Plan

Potential Risks:

- Data inconsistency due to incorrect transactions.
- Security vulnerabilities in financial data handling.
- UI responsiveness issues on different devices.
- Report generation errors.

Mitigation Plan:

- Regular database validation.
- Conduct security testing.
- Perform cross-browser and responsive UI testing.
- Validate report calculations with manual financial computations.

6. Backup Plan

- Maintain regular backups of test data.
- Use version control (GitHub) to track test artifacts.
- Document all test cases to allow easy recovery.

7. Roles & Responsibilities

- **Test Engineer**: Execute test cases, report bugs.
- **Test Lead**: Plan and monitor testing progress.
- **Developers**: Fix defects and perform unit tests.
- **Project Manager**: Ensure testing aligns with business goals.

8. Data-Driven Testing (DDT)

- Use different financial datasets for input values.
- Automate test cases with varying inputs using parameterization (if automation is implemented).

9. Schedules & Timeline

Phase	Start Date	End Date
Test Planning	TBD	TBD
Test Case Design	TBD	TBD
Test Execution	TBD	TBD
Defect Fixing	TBD	TBD
UAT	TBD	TBD

10. Entry & Exit Criteria

Entry Criteria:

- Requirements are finalized and approved.
- Test environment is set up.
- Necessary test data is available.

Exit Criteria:

- All critical test cases executed.
- No high-priority defects remain open.
- Reports and validation checks are successfully completed.

11. Templates

- **Test Case Template**: ID, Title, Steps, Expected Result, Actual Result, Status.
- **Bug Report Template**: ID, Description, Severity, Steps to Reproduce, Expected vs. Actual Result, Screenshot.
- **Test Summary Report**: Overview of test execution status.

12. Deliverables

- Test Plan Document
- Test Cases
- Test Execution Reports
- Defect Reports
- Final Test Summary Report