**The Parabank Project:**

* **Website:** https://parabank.parasoft.com/
* **Tools:** Selenium (UI), Postman (API)

**The guide includes:**

* Project Overview
* User Stories
* Requirements
* Acceptance Criteria
* Needed Tools/Software
* Jira Integration
* Manual & Automation Testing Workflow

**Objective:**

Develop a complete manual + automated testing framework for a web application using Selenium, Postman, JIRA, and performance tools like JMeter, while learning test planning, bug tracking, and reporting processes.

**Project Description:**

Test the Parabank online banking website for both **functional** and **non-functional** requirements using manual testing and automation tools.

**Functional Areas to Test:**

* User Registration & Login
* Open new Account
* Transfer Funds
* Transaction History
* Bill Payment

**USER RULES:**

**Guest/User**: Can register/login, perform transactions

**User Stories and Acceptance criteria:**

**US1: User Login**

**As a registered user**, I want to log in to my account so that I can access my banking dashboard.

**Acceptance Criteria:**

* Login succeeds with valid username and password.
* Login fails with invalid credentials and shows an error.
* Login session remains active until logout or timeout.
* UI shows logged-in user’s name on the dashboard.

**US2: Transfer Funds Between Accounts**

**As a user**, I want to transfer funds between my accounts so that I can manage my money efficiently.

**Acceptance Criteria:**

* User selects source and destination accounts.
* User inputs a valid amount.
* Transfer succeeds and updates balances correctly.
* Error shown if insufficient funds.
* Transaction appears in history immediately.

**US3: View Transaction History**

**As a user**, I want to view my transaction history so that I can track my spending and verify transactions and ensure there are no unauthorized transactions.

**Acceptance Criteria:**

* Only available after successful login.
* List displays date, description, amount, and updated balance.
* Most recent transactions shown first.
* Message displayed if no transactions available.

**US4: Open a New Account**

**As a user**, I want to open a new account so that I can create separate accounts for different purposes.

**Acceptance Criteria:**

* User selects account type (e.g., checking or savings).
* New account is created and visible under “Accounts Overview”.
* Confirmation message is displayed.
* New account starts with default opening balance.
* User can start transferring funds into the new account.

**Entry & Exit Criteria**

* **Entry:**
  + Requirements are clear and approved
  + Test environment is set up
* **Exit:**
  + All test cases executed
  + No critical bugs remaining
  + Test summary and reports submitted

**Test Deliverables:**

* **Test Plan**
* **Test Cases (Manual & Automated)**
* **Bug Reports (Jira)**
* **Test Execution Reports**
* **Automation Code (GitHub repo)**

**Deliverables for Each Student Group:**

| **Category** | **Deliverable** |
| --- | --- |
| **Documentation** | Test Plan, Test Cases (Excel/Google Sheets), Bug Reports |
| **Manual** | Screenshots of test execution, Jira bug logs |
| **Automation** | Framework (code), Test Reports, GitHub repo |
| **Tools Setup** | Screenshots of Postman, Selenium runs |
| **Optional** | CI/CD setup (Jenkins)(optional), Final Demo Video |

**Risks and Mitigations:**

| **Risk** | **Mitigation** |
| --- | --- |
| **Tools setup failure** | **Provide pre-installed virtual environment** |
| **Unclear requirements** | **Communicate frequently with mentor** |

**Required Software & Tools:**

| **Purpose** | **Tool** |
| --- | --- |
| UI Automation | **Selenium WebDriver** (Java) |
| API Testing | **Postman** |
| Manual Test Management | **Jira** |
| Bug Tracking | **Jira** |
| Test Reporting | **TestNG** or **JUnit** with HTML reports |
| Performance Testing | Apache JMeter(optional) |
| Version Control | **Git + GitHub** |
| CI/CD (**optional)** | Jenkins (for automated test runs) |
| IDE | **IntelliJ / Eclipse / VS Code** |

**Test Types:**

| **Test Type** | **Tools Used** |
| --- | --- |
| Functional Testing | Manual, Selenium |
| API Testing | Postman |
| Regression Testing | Selenium |
| Performance Testing | JMeter(optional) |
| Bug Tracking | Jira |

**Test Environment:**

| **Component** | **Description** |
| --- | --- |
| Browser | Chrome / Firefox |
| OS | Windows / Ubuntu |
| Tools | Selenium, Postman, JMeter(optional), Jira, IntelliJ/VS Code |

**Manual Testing Deliverables:**

Students should manually test the web apps **before automating**, and submit:

1. **Test Plan Document**
2. **Test Scenarios and Test Cases**
3. **Test Execution Results**
4. **Bug Reports (via Jira)**
5. **Traceability Matrix (optional)**

**Automation Testing Deliverables:**

1. **Automated Test Scripts** (UI + API)
2. **Framework Setup Documentation**
3. **Execution Reports**
4. **Code Repository (GitHub)**
5. **Video of Test Execution**

**Workflow Summary (Manual + Automation):**

**1. Requirements Review**

* Read the features and prepare user stories.

**2. Manual Testing**

* Design test cases and execute them manually.
* Log defects into Jira.

**3. Test Automation**

* Build automation framework.
* Start automating high-priority test cases (e.g., login).

**4. API Testing**

* Use Postman to verify REST APIs (status code, data returned).
* Automate critical APIs with REST Assured (Java).

**5. Performance Testing**

* Simulate multiple users using JMeter (e.g., 100 users transferring funds).

**Sample Test Plan Template:**

| **Section** | **Description** |
| --- | --- |
| **Project Name** | Parabank – Online Banking App |
| **Version** | v1.0 |
| **Prepared By** | [Student Name(s)] |
| **Date** | [dd/mm/yyyy] |

**Jira Integration Guide**

**What to Track in Jira:**

* Create **Epics**: “UI Testing”, “API Testing”, “Manual Testing”
* Add **User Stories**
* Add **Tasks** : (for example)
  + “Write test cases for login”
  + “Create performance test with JMeter”
* Track **Bugs** found during manual and automated testing
* Link **Test Cases** to **Stories (optinal)** using Xray or Zephyr (if available)

**Students should update Jira Daily/weekly with:**

* Tasks in To-Do / In-Progress / Done
* Bugs with severity
* Notes on execution status

**Important notes:**

* Use the excel sheet sample that we used to make your manual tests
* Test case ID shall be like TC\_001
* Make your team roles and inform the monitor
* **The Parabank demo application, and it doesn't have a public, official API documentation like you'd find for production services so, the following are the steps to discover the endpoints:**

**Use Browser Developer Tools**:

* Open the **Parabank Demo Site**.
* Right-click on the page and select **Inspect** or press F12 to open Developer Tools.
* Go to the **Network** tab, where you'll see the requests made by the application when interacting with the UI (like login, account creation, transfers, etc.).
* Perform actions on the Parabank UI (e.g., logging in, viewing accounts, creating a transfer) and watch the network tab for any HTTP requests. These are the API calls made to the backend.