**Para Bank & Restful Booker Project**

**Software Requirements Specification**

**1.1**

**3/7/2025**

**Prepared for**

parabank.com

restful-booker.com

1. **Introduction**This document serves as the Software Requirements Specification (SRS) for the testing of Para Bank and Restful Booker. It defines the scope, objectives, and features to be tested.
   1. **Purpose**The purpose of this document is to outline the testing plan for

**Para Bank** and **Restful Booker** to ensure their functionality, reliability, and security.

* **Para Bank** will be tested for its customer-related features, including account management, transactions, and fund transfers.
* **Restful Booker** will be tested exclusively through its **API**, focusing on endpoints for booking creation, modification, deletion, and authentication.
  1. **Scope  
     This testing project covers:  
     1.2.1 Para Bank Functional Testing:**
* Customer login and authentication
* Account creation, balance inquiry, and fund transfers
* Transaction history validation
* Error handling and UI responsiveness

**1.2.2 Restful Booker API Testing:**

* Endpoint validation for booking management (Create, Read, Update, Delete)
* Authentication testing with API tokens
* Negative testing for invalid requests
* Performance and security checks

1. **Specific Requirements**

The Para Bank has 2 roles

1. Admin
2. Customer

**2.1** Following features / modules will be available to these 2 roles:

|  |  |
| --- | --- |
| **Admin** | **Customer** |
| Database | Open New Account |
| Data Access Mode | Accounts Overview |
| Web Service | Transfer Funds |
| Application Settings | Bill Pay |
| Accounts Overview | Find Transactions |
|  | Update Contact Info |
|  | Request Loan |
|  | Log in / Log out |

**2.2**Description of the modules

|  |  |  |
| --- | --- | --- |
| **Module Name** | **Applicable Roles** | **Description** |
| Database | Admin | **Admin:** Admin caninitialize and clear database. |
| JMS Service | Admin | **Admin:** Admin can start or stop the service. |
| Data Access Mode | Admin | **Admin:** Admin can select connection mode (SOAP, REST (XML/JSON), JDBC). |
| Web Service | Admin | **Admin:** Admin can select the website service [WSDL | WADL | OpenAPI], set the SOAP Endpoint and REST Endpoint. Also can set an Endpoint to the LoanProcessor Service. |
| Application Settings “Configure account and loan settings” | Admin | **Admin:** Admin can set the initial and minimum balance amount.  Also, admin can change:   * the Loan Provider (Web Service, JMS, Local). * Loan Processor (Available Funds, Down Payment, Combined).   And set the Threshold percentage. |
| Accounts Overview | Customer | **Customer:** Customer can see his information |
| Open New Account | Customer | **Customer:** Customer can open new account |
| Transfer Funds | Customer | **Customer:** Customer can transfer funds |
| Bill Pay | Customer | **Customer:** Customer can bill pay |
| Update Contact Info | Customer | **Customer:** Customer can update his contact information |
| Request Loan | Customer | **Customer:** Customer can request loan |
| Log in / Log out | Customer | **Customer:** Customer can log in and log out |

* 1. Restful Booker API Features

**1.** Booking Management Endpoints

* + Create a new booking
  + Retrieve an existing booking
  + Update booking details
  + Delete a booking

**2. Authentication & Security**

* Supports token-based authentication for API access

1. **Negative Testing Cases**
   * Handle invalid or missing API request parameters
   * Validate proper error handling
2. **Performance & Security Testing**
   * API response time measurement
   * System performance can be evaluated under simulated load conditions
   1. **Front End Details**

This section describes the Front end of **Para Bank**.

It also lists a few use cases to describe the functioning of the system.  
 Following is a list of module wise fields  
  
  
**Open New Account**

* Account type
* Deposited amount

**Transfer Funds**

* Amount of money
* From account no
* To account no

**Bill Pay**

* Payee name
* Address
* City
* State
* Zip code
* Phone
* Account no
* Verify Account
* Amount
* From account no

**Find Transactions**

* Account no
* Transaction ID
* Date
* Date Range
* Between
* And
* Amount

**Update Contact Info**

* First name
* Last name
* Address
* City
* State
* Zip code
* Phone

**Request Loan**

* Loan amount
* Down payment
* From account no

**3.2 Technical Requirements:**

**Personal Information**

**First Name**:

**T1.** Numbers are not allowed.

**T2.** Special characters are not allowed.

**T3.** Must not be blank.

**T4.** First character cannot have space.

**Last Name**:

**T5.** Numbers are not allowed.

**T6.** Special characters are not allowed.

**T7.** Must not be blank.

**T8.** First character cannot have space.

**Address**:

**T9.** Must not be blank.

**T10.** First character cannot have space.

**T11.** Special characters are not allowed.

**City**:

**T12.** Special characters are not allowed.

**T13.** Must not be blank.

**State**:

**T14.** Numbers are not allowed.

**T15.** Must not be blank.

**T16.** Special characters are not allowed.

**Zip Code**:

**T17.** Must be a 5-digit number (e.g., 12345).

**T18.** Must not be blank.

**T19.** Must not contain letters or special characters.

**Phone #**:

**T20.** Characters are not allowed.

**T21.** Must not contain special characters (except hyphens).

**T22.** Must not be blank.

**SSN**:

**T23.** Must be in the format: XXX-XX-XXXX (e.g., 123-45-6789).

**T24.** Must not contain letters or special characters (except hyphens).

**T25.** Must not be blank.

**Open New Account**

**Account Type Selection:**

**T26.** The user must select an account type before proceeding.

**T27.** Two account types must be available

* Checking Account
* Saving Account

**T28.** The selected account type determines the required fields during registration.

**T29.** The system must store the selected account type in the database.

**T30.** The selection cannot be left blank.

**Transfer Funds**

**Amount of Money**:

* 1. Must be a numeric value.
  2. Must be greater than zero.

**From Account No**:

* 1. Must be selected.
  2. Must be an exist account number.

**To Account No**:

* 1. Must be selected.
  2. Must be an exist account number.

**Bill Pay**

**Payee Name**:

**T37.** Numbers are not allowed.

**T38.** Special characters are not allowed.

**T39.** Must not be blank.

**Address**:

**T40.** Must not be blank.

**T41.** First character cannot have space.

**T42.** Special characters are not allowed.

**City**:

**T43.** Special characters are not allowed.

**T44.** Must not be blank.

**State**:

**T45.** Numbers are not allowed.

**T46.** Must not be blank.

**T47.** Special characters are not allowed.

**Zip Code**:

**T48.** Must be a 5-digit number (e.g., 12345).

**T49.** Must not contain letters or special characters.

**Phone**:

**T50.** Characters are not allowed.

**T51.** Must not contain special characters (except hyphens).

**T52.** Must not be blank.

**Account No**:

**T53.** Must not be blank.

**T54.** Must be a valid account number.

**Verify Account**:

**T55.** Must match the account number provided.

**Amount**:

**T56.** Must be a numeric value.

**T57.** Must be greater than zero.

**From Account No**:

**T58.** Must not be blank.

**T59.** Must be a valid account number.

**Find Transactions**

**Transaction ID**:

**T60.** Must not be blank.

**T61.** Must be a valid transaction ID.

**T62.** Must not contain characters or special characters.

**Date**:

**T63.** Must be in the format: MM/DD/YYYY.

**T64.** Must be a valid date.

**Date Range**:

**Between**:

**T65.** Must be in the format: MM/DD/YYYY.

**T66.** Must be a valid date.

**And**

**T67.** Must be in the format: MM/DD/YYYY.

**T68.** Must be a valid date.

**Amount**:

**T69.** Must be a numeric value.

**T70.** Must be greater than zero.

**Update Contact Info**

**First Name**:

**T71.** Numbers are not allowed.

**T72.** Special characters are not allowed.

**T73.** Must not be blank.

**T74.** First character cannot have space.

**Last Name**:

**T75.** Numbers are not allowed.

**T76.** Special characters are not allowed.

**T77.** Must not be blank.

**T78.** First character cannot have space.

**Address**:

**T79.** Must not be blank.

**T80.** First character cannot have space.

**T81.** Special characters are not allowed.

**City**:

**T82.** Special characters are not allowed.

**T83.** Must not be blank.

**T84.** First character cannot have space.

**State**:

**T85.** Numbers are not allowed.

**T86.** Must not be blank.

**T87.** Special characters are not allowed.

**T88.** First character cannot have space.

**Zip Code**:

**T89.** Must be a 5-digit number (e.g., 12345).

**T90.** Must not contain letters or special characters.

**Phone**:

**T91.** Characters are not allowed.

**T92.** Must not contain special characters (except hyphens).

**T93.** Must not be blank.

**Request Loan**

**Loan Amount**:

**T94.** Must be a numeric value.

**T95.** Must be greater than zero.

**Down Payment**:

**T96.** Must be a numeric value.

**T97.** Must be greater than or equal to zero.

**From Account No**:

**T98.** Must not be blank.

**T99.** Must be a valid account number.

**Customer Care**

**Name Field:**  
 T100. Must not be blank.

1. Numbers are not allowed.
2. Special characters are not allowed.
3. First character cannot be a space.

**Email Field:**

1. Must not be blank.
2. Must be a valid email format (e.g., user@example.com).

**Phone Field:**

1. Must not be blank.
2. Must contain only numbers.
3. Must be at least 10 digits.

**Message Field:**

**T109.** Must not be blank.  
 **T110.** Must allow at least 10 characters.  
 **T111.** Maximum length should not exceed 500 characters.

**Register Page:**

**First Name**:

**T112.** Numbers are not allowed.

**T113.** Special characters are not allowed.

**T114.** Must not be blank.

**T115.** First character cannot have space.

**Last Name**:

**T116.** Numbers are not allowed.

**T117.** Special characters are not allowed.

**T118.** Must not be blank.

**T119.** First character cannot have space.

**Address**:

**T120.** Must not be blank.

**T121.** First character cannot have space.

**T122.** Special characters are not allowed.

**City**:

**T123.** Special characters are not allowed.

**T124.** Must not be blank.

**State**:

**T125.** Numbers are not allowed.

**T126.** Must not be blank.

**T127.** Special characters are not allowed.

**Zip Code**:

**T128.** Must be a 5-digit number (e.g., 12345).

**T129.** Must not be blank.

**T130.** Must not contain letters or special characters.

**Phone #**:

**T131.** Characters are not allowed.

**T132.** Must not contain special characters (except hyphens).

**T133.** Must not be blank.

**4.1 Functional Validation:**

**Balance Enquiry**

**F1 (Manager):** Enter a valid account number → Balance is **displayed.**

**F2 (Manager):** Enter an invalid account number → Errormessage is displayed.

**F3 (Customer):** Enter a valid account number associated withthe customer → Balance is displayed.

**F4 (Customer**): Enter an account number that does not belong to the customer → Error message is displayed**.**

**Fund Transfer**

**F5 (Manager):** Enter a valid source and destination account → Transfer is successful.

**F6 (Manager):** Enter an invalid source or destination account → Error message is displayed.

**F7 (Manager):** Enter a valid source account with insufficient funds → Error message is displayed.

**F8 (Manager**): Enter a source account that does not belong to the manager → Error message is displayed.

**F9 (Customer):** Enter a valid source and destination account → Transfer is successful.

**F10 (Customer):** Enter the same source and destination account → Error message is displayed.

**F11 (Customer):** Enter a valid source account with insufficient funds → Error message is displayed**.**

**F12 (Customer):** Enter a source account that does not belongto the customer → Error message is displayed.

**Withdrawal**

**F13 (Manager):** Enter a valid source account with sufficient funds → Withdrawal is successful.

**F14 (Manager):** Enter an invalid source account → Error message is displayed.

**F15 (Manager):** Enter a source account with insufficient funds → Error message is displayed.

**F16 (Customer):** Enter a valid source account with sufficient funds → Withdrawal is successful**.**

**F17 (Customer):** Enter an invalid source account → Error message is displayed.

**F18 (Customer):** Enter a source account with insufficient funds → Error message is displayed.

**Deposit**

**F19 (Manager):** Enter a valid destination account → Deposit is successful.

**F20 (Manager):** Enter an invalid destination account → Error message is displayed.

**F21 (Customer):** Enter a valid destination account → Deposit is successful.

**F22 (Customer):** Enter an invalid destination account → Error message is displayed.

**Edit Account**

**F29 (Manager):** Enter a valid account number → Account details are updated.

**F30 (Manager):** Enter an invalid account number → Error message is displayed.

**New Account**

**F31 (Manager):** Enter a valid customer ID with sufficient initial deposit → New account is created.

**F32 (Manager):** Enter an invalid customer ID → Error message is displayed.

**F33 (Manager):** Enter an initial deposit lower than the required minimum → Error message is displayed.

**New Customer**

**F34 (Manager):** Enter a unique email and valid details → Customer is registered.

**F35 (Manager):** Enter an email that already exists → Error message is displayed.

**Edit Customer**

**F36 (Manager):** Enter a valid customer ID → Customer details are updated.

**F37 (Manager):** Enter an invalid customer ID → Error message is displayed.

**F38 (Manager):** Enter an email that already exists → Error message is displayed.

**Transaction History**

**F43 (Manager):** Enter a valid account number → Transaction history is displayed.

**F44 (Manager):** Enter an invalid account number → Error message is displayed.

**F45 (Manager):** Enter an account number with no transactions → Error message is displayed.

**F46 (Customer):** Enter a valid account number → Transaction history is displayed.

**F47 (Customer):** Enter an invalid account number → Error message is displayed.

**F48 (Customer):** Enter an account with no transactions → Error message is displayed.

**4.2 Non-Functional Requirements**

1. **Performance**
   * API should respond within 200ms for valid requests
2. **Usability**
   * System should be accessible from multiple devices
   * API documentation should be clear and easy to follow
3. **Security  
   ParaBank:**
   * System should properly handle invalid inputs and prevent unauthorized access through basic validation.

**Restful Booker API:**

* + Token-based authentication must be implemented to restrict access.
  + API should return proper authorization error codes (401 for unauthorized, 403 for forbidden).
  + User sessions should be securely managed.

**4.3 User Stories & Use Cases**

**4.3.1 User Stories**

**As a Customer:**

* I want to log into my account so that I can manage my finances.
* I want to transfer funds between my accounts so that I can manage my money efficiently.
* I want to view my transaction history so that I can track my expenses.
* I want to update my contact information so that my details remain current.
* I want to request a loan so that I can meet my financial needs.

**As an Admin:**

* I want to manage the database so that I can maintain system integrity.
* I want to configure loan and account settings so that I can define banking rules.

**4.3.2 Use Cases**

**Use Case: Customer Login**

* **Actors**: Customer
* **Preconditions**: Customer is registered in the system.
* **Steps**:
  1. Customer enters username and password.
  2. System validates credentials.
  3. If valid, system grants access.
  4. If invalid, system displays an error message.

**Use Case: Transfer Funds**

* **Actors**: Customer
* **Preconditions**: Customer has an active account.
* **Steps**:
  1. Customer selects the transfer funds option.
  2. Customer enters the amount, source, and destination accounts.
  3. System validates the request.
  4. If valid, funds are transferred.
  5. Confirmation message is displayed.

**Use Case: API Booking Creation**

* **Actors**: External API consumer
* **Preconditions**: Valid API authentication.
* **Steps**:
  1. API consumer sends a request to create a booking.
  2. System validates request parameters.
  3. If valid, booking is created and ID is returned.
  4. If invalid, system returns an error message.

**4.3.3 Expected Behavior**

* The system should prevent invalid user actions with proper error messages.
* The API should return appropriate status codes (200 for success, 400 for invalid requests, 401 for unauthorized access, etc.).
* All user inputs should be validated based on defined constraints.
* The UI should be responsive and load within 3 seconds.

**4.4 Inverse Requirements**

Nil.

**4.5 Design Constraints**Some users of Para Bank and Restful Booker may not have advanced technical knowledge. Therefore, the system must be user-friendly, intuitive, and easy to navigate to ensure a smooth experience for all users.

**4.6 Logical Database Requirements**

Nil.

**4.7 Other Requirements**

Nil.

**5. Analysis Models**

Nil

**6. Conclusion**

This document outlines the requirements and validation rules for testing Para Bank and Restful Booker. It ensures robust test coverage to verify functionality, security, and reliability for both platforms.