# **UiBank - Business Requirements Document**

## **1. Executive Summary**

### **1.1 Project Overview**

This document outlines the business requirements for the UiBank API Test Automation project. UiBank is a banking website developed by UiPath Academy that simulates real banking operations. The project focuses on testing the application's backend services (APIs) to ensure they work correctly and reliably.

### **1.2 Purpose**

The main goal is to automatically test UiBank's banking services to:

* Verify that all banking operations work as expected
* Catch problems before customers experience them
* Ensure the system is reliable and secure
* Save time and resources compared to manual testing

### **1.3 Scope**

This project covers testing of 7 different banking services organized into 2 main categories:

* **Banking Operations** (5 services requiring user login)
* **Loan Services** (2 services that don't require user login)

## **2. Business Context**

### **2.1 What is UiBank?**

UiBank is a simulated online banking platform that allows users to:

* Create and manage bank accounts (savings and checking)
* Transfer money between their accounts
* Apply for loans
* View their banking information

### **2.2 Why API Testing is Important**

APIs (Application Programming Interfaces) are the behind-the-scenes services that handle all banking operations. Testing these services ensures:

* Customer transactions are processed correctly
* Account information is accurate and secure
* The system can handle multiple users simultaneously
* All banking rules and regulations are followed

## **3. System Requirements**

### **3.1 Banking Operations Collection (Session-Based Services)**

These services require users to log in first and maintain their session throughout their banking activities.

#### **3.1.1 User Login Service**

**Business Purpose:** Allow customers to securely access their banking accounts

**What it does:**

* Accepts customer username and password
* Verifies the customer's identity
* Creates a secure session for the customer
* Provides a digital token (like a temporary ID card) for accessing other services

**Input Requirements:**

* Username (customer's login ID)
* Password (customer's secret password)

**Expected Results:**

* Successful login creates a session token
* Invalid credentials reject access
* Session token works for subsequent banking operations

#### **3.1.2 Get Profile Service**

**Business Purpose:** Verify that the correct customer account is being accessed

**What it does:**

* Retrieves customer's personal information
* Confirms the right person is logged in
* Displays account holder details

**Input Requirements:**

* Valid session token from login

**Expected Results:**

* Returns correct customer information
* Matches the logged-in user's details
* Protects customer privacy and data

#### **3.1.3 Create Account Service**

**Business Purpose:** Allow customers to open new bank accounts

**What it does:**

* Creates new savings or checking accounts
* Sets up initial account balance
* Links new account to customer profile

**Input Requirements:**

* Valid session token
* Account type (savings or checking)
* Initial balance amount

**Expected Results:**

* New account is created successfully
* Account number is generated
* Initial balance is set correctly
* Account is linked to the customer

#### **3.1.4 Get Account Service**

**Business Purpose:** Show customers all their bank accounts and balances

**What it does:**

* Lists all accounts belonging to the customer
* Shows current balances
* Displays account types and details

**Input Requirements:**

* Valid session token

**Expected Results:**

* Returns complete list of customer's accounts
* Shows accurate balance information
* Displays only accounts belonging to that customer

#### **3.1.5 Transfer Funds Service**

**Business Purpose:** Allow customers to move money between their accounts

**What it does:**

* Transfers money from one account to another
* Updates account balances
* Records the transaction

**Input Requirements:**

* Valid session token
* Source account (where money comes from)
* Destination account (where money goes to)
* Transfer amount

**Expected Results:**

* Money is deducted from source account
* Money is added to destination account
* Both accounts show updated balances
* Transfer only works between customer's own accounts
* Prevents transfers with insufficient funds

### **3.2 Loan Services Collection (Public Services)**

These services don't require login and can be accessed by anyone interested in loan information.

#### **3.2.1 Apply for Loan Service**

**Business Purpose:** Allow potential customers to apply for loans

**What it does:**

* Accepts loan application details
* Evaluates loan eligibility
* Approves or denies loan applications
* Provides loan decision reasoning

**Input Requirements:**

* Loan amount requested
* Loan term (how long to repay)
* Applicant's income
* Applicant's age

**Expected Results:**

* Loan application is processed
* Decision is made (approved or denied)
* Loan ID is created for approved applications
* Decision follows business rules for loan approval

#### **3.2.2 Get Loan Details Service**

**Business Purpose:** Allow customers to check their loan information

**What it does:**

* Retrieves loan information using loan ID
* Shows loan terms and conditions
* Displays repayment details

**Input Requirements:**

* Valid Loan ID

**Expected Results:**

* Returns complete loan information
* Shows accurate loan terms
* Only displays information for valid loan IDs

## **4. Security Requirements**

### **4.1 Session Management**

* All banking operations must use secure session tokens
* Session tokens must expire after a reasonable time
* Invalid or expired tokens must be rejected

### **4.2 Data Protection**

* Customer information must be protected
* Account details should only be visible to account owners
* Sensitive data should not be exposed in error messages

### **4.3 Authorization**

* Users can only access their own accounts
* Transfers only allowed between user's own accounts
* Loan information only accessible with valid loan ID

## **5. Business Rules**

### **5.1 Account Management Rules**

* Customers can have multiple accounts
* Each account must have a unique account number
* Account balances cannot go below zero
* Both savings and checking accounts are supported

### **5.2 Transfer Rules**

* Transfers only allowed between same customer's accounts
* Transfer amount must be positive
* Source account must have sufficient funds
* Both accounts must exist and be active

### **5.3 Loan Processing Rules**

* Loan approval based on income, age, and requested amount
* Each approved loan gets a unique loan ID
* Loan terms must be within acceptable ranges
* Age and income requirements must be met

## **6. Testing Requirements**

### **6.1 What Needs to be Tested**

#### **6.1.1 Functional Testing**

* **Login Process:** Verify users can log in with correct credentials and are rejected with wrong ones
* **Account Creation:** Confirm new accounts are created properly with correct initial balances
* **Account Retrieval:** Ensure customers see only their accounts with accurate information
* **Fund Transfers:** Validate money moves correctly between accounts with proper balance updates
* **Loan Applications:** Check loan approval/denial logic works according to business rules
* **Loan Information:** Verify loan details are retrieved accurately

#### **6.1.2 Security Testing**

* **Access Control:** Ensure users cannot access other customers' information
* **Session Security:** Verify session tokens work properly and expire when needed
* **Data Protection:** Confirm sensitive information is properly protected

#### **6.1.3 Error Handling Testing**

* **Invalid Inputs:** Test system behavior with wrong or missing information
* **Insufficient Funds:** Verify transfers are blocked when accounts don't have enough money
* **Non-existent Accounts:** Check proper error messages for invalid account numbers
* **Expired Sessions:** Ensure proper handling of expired login sessions

### **6.2 Test Scenarios for Each Service**

#### **6.2.1 Login Service Test Cases**

* Successful login with valid credentials
* Failed login with invalid username
* Failed login with invalid password
* Failed login with missing information
* Session token generation and validation

#### **6.2.2 Profile Service Test Cases**

* Retrieve profile with valid session
* Reject access with invalid session
* Return correct customer information
* Protect customer privacy

#### **6.2.3 Account Creation Test Cases**

* Create savings account successfully
* Create checking account successfully
* Create account with initial balance
* Reject creation with invalid session
* Handle duplicate account creation attempts

#### **6.2.4 Account Retrieval Test Cases**

* Get all accounts for valid customer
* Return empty list for customer with no accounts
* Reject access with invalid session
* Show accurate balance information

#### **6.2.5 Fund Transfer Test Cases**

* Successful transfer between customer's accounts
* Reject transfer with insufficient funds
* Reject transfer to non-existent account
* Reject transfer with invalid session
* Verify balance updates after transfer

#### **6.2.6 Loan Application Test Cases**

* Approve loan for qualified applicant
* Deny loan for unqualified applicant
* Handle missing application information
* Generate unique loan ID for approved loans
* Validate input data ranges

#### **6.2.7 Loan Details Test Cases**

* Retrieve loan information with valid loan ID
* Handle requests with invalid loan ID
* Return complete and accurate loan details
* Protect loan information privacy

## **7. Success Criteria**

### **7.1 Testing Success Measures**

* All test cases pass consistently
* System handles error conditions gracefully
* Security requirements are met
* Performance meets acceptable standards
* All business rules are enforced correctly

### **7.2 Quality Standards**

* Zero critical security vulnerabilities
* 100% of core banking functions work correctly
* Error messages are clear and helpful
* System responses are timely
* Data integrity is maintained

## **8. Technical Implementation Notes**

### **8.1 Testing Tools**

* **Postman:** Used for creating and running API tests
* **JavaScript:** Used for writing test logic and validations
* **JWT Tokens:** Used for session management and security

### **8.2 Test Organization**

* Tests are organized in 2 main collections
* Each collection contains related API tests
* Tests can be run individually or as complete suites
* Results are tracked and reported automatically

## **9. Risks and Mitigation**

### **9.1 Potential Risks**

* **Security Vulnerabilities:** Unauthorized access to customer data
* **Data Loss:** Incorrect balance calculations or lost transactions
* **System Downtime:** Banking services become unavailable
* **Integration Issues:** Problems when connecting different services

### **9.2 Risk Mitigation**

* Comprehensive security testing
* Thorough validation of all calculations
* Regular monitoring and testing
* Proper error handling and recovery procedures

## **10. Conclusion**

This business requirements document provides a comprehensive overview of the UiBank API Test Automation project. The testing suite ensures that all banking operations work correctly, securely, and reliably. By following these requirements, testers can create effective test cases that validate the system's functionality and protect customer interests.

The automated testing approach saves time, improves accuracy, and provides confidence that the UiBank system meets all business and technical requirements for a modern banking platform.