

## **Important Test Cases for Banking Applications**

### **1. Account Management Test Cases for Banking Apps**

An account in a banking application is the repository for all of the critical information about the customer's financial and investment status.

Some important test cases for this feature are:

1. Verify the ability to create a new account with valid information.
2. Verify the inability to create an account with incomplete or invalid information.
3. Test the process of updating account information (e.g., address, phone number).
4. Verify that account balances are updated accurately after transactions.
5. Test the account closure process and ensure all associated data is removed.
6. Check for account naming conventions and validation rules.
7. Test the ability to link or unlink accounts (e.g., joint accounts).
8. Verify that account statements are generated correctly.
9. Test account recovery options in case of forgotten credentials.
10. Check for account locking and unlocking functionality after failed login attempts.
11. Verify that account data remains confidential and is not accessible to unauthorized users.
12. Test for concurrent access to the same account from multiple devices.
13. Verify that account data is synchronized across multiple platforms (web, mobile, ATM).
14. Test the process of setting up account alerts and notifications.
15. Verify the ability to export account data in various formats (e.g., CSV, PDF).
16. Test the accuracy of interest calculations for savings accounts.
17. Confirm that accounts with different currencies are handled correctly.
18. Check for the ability to set account-specific preferences (e.g., language, notifications).
19. Verify that account histories are archived and retrievable.
20. Test the process of merging or closing duplicate accounts.

### **2. Transaction Processing Test Cases for Banking Apps**

1. Test different types of transactions (e.g., deposits, withdrawals, transfers).
2. Verify that transactions are processed in real-time or within defined timeframes.
3. Test for transaction limits and restrictions on various account types.
4. Verify that transaction fees are accurately deducted.
5. Test the ability to cancel or reverse pending transactions.
6. Check for overdraft protection and notifications.
7. Verify that international transactions are handled appropriately (currency conversion, fees).
8. Test transactions during scheduled maintenance periods.
9. Verify that transaction references and receipts are generated accurately.
10. Test for duplicate transaction detection and prevention.
11. Verify the handling of transactions involving joint accounts.
12. Test the process of setting up recurring transactions (e.g., bill payments).
13. Check for the ability to designate transaction categories (e.g., groceries, entertainment).
14. Verify that transactions are reflected correctly in account statements.
15. Test transactions with offline banking options (e.g., check deposits at ATMs).
16. Verify the ability to dispute and investigate erroneous transactions.
17. Test the integration with third-party payment gateways.
18. Verify the handling of large transactions that may require additional authorization.
19. Test the processing of transactions involving foreign exchange rates.
20. Verify that transactions are time-stamped accurately for auditing purposes.

### **3. User Authentication and Security Test Cases for Banking Applications**

1. Test the login functionality with valid credentials.
2. Verify that incorrect login attempts result in appropriate error messages.
3. Test multi-factor authentication (MFA) methods (e.g., SMS, email, token).
4. Verify the account lockout mechanism after a specified number of failed login attempts.
5. Test password reset functionality and security questions.
6. Verify the expiration and automatic logout of idle sessions.
7. Test the ability to change account passwords.
8. Verify secure password storage and encryption.
9. Test the application's response to suspicious login activity (e.g., from different locations).
10. Verify that sensitive data (e.g., passwords, PINs) is masked or hidden.
11. Test the handling of lost or stolen devices for mobile banking.
12. Verify the security of biometric authentication (e.g., fingerprint, facial recognition).
13. Test the logout functionality and session termination.
14. Verify the effectiveness of security certificates for encrypted communication.
15. Test the behavior of the application in case of a known security vulnerability.
16. Verify the implementation of role-based access control for staff members.
17. Test for SQL injection and other common security vulnerabilities.
18. Verify that user data is not stored on the client-side.
19. Test the response to brute force attacks.
20. Verify that security patches and updates are applied promptly.

### **4. Mobile Banking Test Cases**

Here are some important test cases for mobile banking:

1. Test the installation and setup of the mobile banking app.
2. Verify the compatibility with different mobile device types (iOS, Android).
3. Test the responsiveness of the app on various screen sizes and resolutions.
4. Verify that mobile check deposit functions correctly.
5. Test mobile bill payment and funds transfer.
6. Verify the accuracy of push notifications and alerts.
7. Test offline functionality, such as viewing account balances without an internet connection.
8. Verify the ability to toggle between multiple user accounts.
9. Test the mobile app's synchronization with the web-based banking platform.
10. Verify that the app is accessible to users with disabilities (ADA compliance).
11. Test the performance of the mobile app under low-network conditions.
12. Verify that biometric authentication (e.g., fingerprint, face ID) works securely.
13. Test mobile app updates and their impact on existing data.
14. Verify that the mobile app supports language preferences and localization.
15. Test mobile app-specific features like remote account deactivation.
16. Verify that mobile app data is protected against device theft or loss.
17. Test the mobile app's ability to generate and scan QR codes for transactions.
18. Verify the handling of interruptions (e.g., phone calls) during app use.
19. Test the mobile app's accessibility on both smartphones and tablets.
20. Verify the integration of mobile app analytics for performance monitoring.

### **5. Cross-platform Compatibility Test Cases for Banking Applications**

1. Test the application on different web browsers (Chrome, Firefox, Safari, Edge, etc.).
2. Verify compatibility with various mobile devices (iOS and Android) and their versions.
3. Test on different operating systems (Windows, macOS, Linux) for web-based platforms.
4. Verify the responsiveness and usability of the application on different screen sizes.
5. Test the application on various screen resolutions (e.g., HD, Full HD, 4K).
6. Verify compatibility with different input devices (mouse, touchpad, touchscreen).
7. Test on different network speeds and conditions (e.g., 3G, 4G, Wi-Fi, slow connections).
8. Verify compatibility with screen reader software for accessibility.
9. Test for compatibility with third-party plugins or extensions in web browsers.
10. Verify compatibility with different versions of Java (if applicable).
11. Test the application's performance on low-end and high-end hardware.
12. Verify compatibility with different email clients for transaction notifications.
13. Test the application's compatibility with various PDF readers for statements.
14. Verify that the application works seamlessly with different virtual private networks (VPNs).
15. Test compatibility with screen rotation on mobile devices.
16. Verify the application's compatibility with various firewall settings.
17. Test the application's behavior on devices with different default languages.
18. Verify that the application's UI elements adapt to the platform's design guidelines.
19. Test compatibility with different security software and antivirus programs.
20. Verify compatibility with various ad-blocking extensions or software.