# Test Strategy for Mobile App and BFF API

## 1. Objective

- Ensure the mobile app functions as intended, providing a seamless and bug-free user experience.  
 - Ensure the BFF API meets functional, performance, and security requirements and handles requests from the mobile app efficiently and securely.

## 2. Testing Scope

- Mobile Application: User interface, user experience, app performance, functionality, security, and compatibility across different devices, screen sizes, and OS versions.  
 - BFF API: Request-response handling, data processing, validation, performance, error handling, and security.

## 3. Test Plan for Mobile App

### A. Functional Testing

- Unit Testing: Verify each unit/component of the app (e.g., functions, classes, view controllers) with unit test cases.  
 - Integration Testing: Ensure the app components work together as expected, especially for areas dependent on the BFF API.  
 - End-to-End Testing: Test workflows from the user’s perspective, covering main and alternative scenarios to ensure the app functions as intended.

### B. UI/UX Testing

- Visual Testing: Verify UI elements are correctly aligned, displayed, and consistent across different devices.  
 - Usability Testing: Assess ease of use, intuitiveness, and user satisfaction. Testing with real users can be beneficial.  
 - Accessibility Testing: Verify compliance with accessibility standards (e.g., WCAG) to ensure the app is usable for people with disabilities.

### C. Performance Testing

- Load Testing: Simulate multiple users accessing the app simultaneously and evaluate response times.  
 - Battery Usage Testing: Measure battery consumption during typical user scenarios.  
 - Memory Leak Testing: Identify and resolve memory leaks to avoid crashes and poor performance over time.

### D. Device and OS Compatibility Testing

- Test on a range of devices, OS versions, and screen sizes to ensure compatibility and responsiveness.  
 - Prioritize based on analytics (if available) on user device types and OS versions.

### E. Security Testing

- Authentication & Authorization: Verify secure handling of user authentication, permissions, and access controls.  
 - Data Encryption: Ensure sensitive data is encrypted both at rest and in transit.  
 - Secure Storage: Confirm proper handling and secure storage of sensitive data on the device.

### F. Network Connectivity Testing

- Offline Mode Testing: Verify app behavior in the absence of an internet connection, including error handling.  
 - Weak Network Testing: Simulate weak/unstable network conditions to ensure smooth and graceful degradation.

### G. Regression Testing

- Ensure previously working features continue to function after updates or bug fixes. Automate critical regression tests for efficiency.

## 4. Test Plan for BFF API

### A. Functional Testing

- Endpoint Testing: Verify each API endpoint responds as expected, handling both valid and invalid inputs.  
 - Data Validation: Confirm that the API processes and validates data according to requirements, enforcing field lengths, types, and formats.  
 - Response Format Testing: Check that responses have the correct structure (e.g., JSON), status codes, and error messages.

### B. Integration Testing

- Test the integration between the BFF API and other services it interacts with (e.g., databases, third-party APIs).  
 - Verify data mapping and transformations between the app and other services via the BFF API.

### C. Performance Testing

- Load Testing: Simulate concurrent requests to evaluate response time and resource usage under load.  
 - Stress Testing: Test the API’s limits by increasing the load beyond normal operational levels to identify breaking points.  
 - Scalability Testing: Evaluate the API’s ability to scale, particularly for peak load periods.

### D. Security Testing

- Authentication and Authorization: Verify the API enforces proper authentication and authorization protocols.  
 - Data Encryption: Ensure secure data transmission using HTTPS/TLS, and confirm that sensitive data is encrypted at rest if stored.  
 - Rate Limiting: Test rate-limiting controls to prevent abuse, such as excessive request spamming from a single IP.  
 - Vulnerability Scanning: Perform vulnerability scans to detect security flaws (e.g., OWASP vulnerabilities).

### E. Error Handling and Logging

- Error Response Testing: Verify that the API returns meaningful error responses for client and server errors, including appropriate HTTP status codes.  
 - Logging: Confirm that logging captures sufficient information for troubleshooting and is securely stored (e.g., excluding sensitive data in logs).

## 5. Test Automation Strategy

### A. Mobile App Automation

- Use tools like Appium, Espresso, or XCUITest for automating unit, functional, and regression tests.  
 - Focus automation on repetitive test cases, such as regression and end-to-end testing for key user workflows.

### B. BFF API Automation

- Use tools like Postman or RestAssured to automate API testing.  
 - Focus on functional testing (endpoint testing), data validation, and regression testing for core API functionalities.

### C. CI/CD Integration

- Integrate test automation into the CI/CD pipeline to run tests continuously on code changes.  
 - Automate smoke tests for the mobile app and API after every deployment to detect issues early.

## 6. Non-Functional Testing

### A. Usability & Accessibility

- Test both the app and API for usability, ensuring error messages are user-friendly and clear.  
 - Validate accessibility across both app interface and API endpoints for inclusivity (e.g., text for screen readers, proper ARIA labels).

### B. Localization & Internationalization

- Test app localization if multiple languages are supported, verifying translations and regional settings.  
 - Ensure the BFF API supports multiple locales if relevant.

## 7. Monitoring and Logging

- Implement real-time monitoring for the BFF API to track response times, error rates, and usage patterns.  
 - Use application monitoring for the mobile app (e.g., Firebase) to capture crash logs, performance metrics, and usage analytics.