

# Big Music — System Test Document

NIST RMF & SP 800-53 Rev. 5.1 Compliance Testing

---

## 1.1 TEST RISKS / ISSUES

### Risks associated with testing:

- External APIs (Spotify, TMDB, OMDb, TasteDive) may be unavailable during testing.
- Rate limiting or temporary API throttling could interrupt test cases.
- Network latency or outages may cause false failures.
- OAuth tokens may expire mid-test, requiring re-authentication.

### Mitigation Strategies:

- Perform tests during low-traffic hours to reduce rate-limit risk.
- Use test accounts when possible.
- Document any external API downtime as “environment failures,” not app failures.
- Refresh OAuth tokens as needed.

### Contingency Plans:

- Retry any failed tests caused by external API outages.
- Switch to backup network or hotspot if local connectivity fails.
- Perform offline validation for tests not dependent on APIs.

---

## 1.2 ITEMS TO BE TESTED / NOT TESTED

## Items to Be Tested

Item	Description	How It Will Be Tested	When	Responsibility
OAuth Authentication	Login using Spotify OAuth	Verify redirect, token handling, and session	Test Cycle 1	Developer
API Integration	TasteDive/TMDB/OMDB requests	Validate responses and sanitization	Test Cycle 1 & 2	Developer
Environment Variables	Ensure API keys secured	Confirm <code>.env</code> handling + <code>.gitignore</code>	Test Cycle 1	Developer
Error Handling	No sensitive errors leaked	Trigger malformed inputs	Test Cycle 2	Developer
Logging	Ensure no personal data logged	Review logs	Test Cycle 2	Developer

## Items Not Tested (N/A)

- Database interactions (no database exists)
- User account management (handled entirely by Spotify)
- Secure storage of user data (application stores none)
- Media handling (no uploaded files)
- Internal role-based access (single-user app)

---

## 1.3 TEST APPROACH(S)

- **Black box testing** of login flow and API interaction
- **Functional testing** for recommendation generation

- **Boundary and error testing** for malformed external API responses
  - **Security testing** via inspection of tokens and logs
  - **Configuration review** to confirm environment variable security
  - **NIST control-by-control assessment** (mapped below)
- 

## 1.4 TEST REGULATORY / MANDATE CRITERIA

This application is tested against:

- **NIST SP 800-53 Rev. 5.1 controls**
  - **NIST RMF Step 4 (Assess Security Controls)**
  - **OWASP API Top 10 (informal reference only)**
  - **Spotify OAuth 2.0 Security Requirements**
- 

## 1.5 TEST PASS / FAIL CRITERIA

**PASS if:**

- Expected output is correct
- No sensitive data is logged
- OAuth tokens are never stored
- API responses are properly handled
- Errors do not expose stack traces

**FAIL if:**

- Personal data is stored or logged
  - Tokens are written to disk
  - API keys appear in source code
  - Any unhandled exception is exposed to the user
- 

## **1.6 TEST ENTRY / EXIT CRITERIA**

### **Entry Criteria**

- Code compiles and runs locally
- External APIs reachable
- Environment variables correctly configured
- Spotify Developer App configured with redirect URI

### **Exit Criteria**

- All test cases executed
  - All critical and high-severity bugs fixed
  - All applicable NIST controls evaluated
  - Test documentation completed
- 

## **1.7 TEST DELIVERABLES**

- Completed NIST Test Document (this file)
- Test case execution logs

- Screenshots of successful OAuth login
  - API response validation logs
  - Control applicability matrix
- 

## **1.8 TEST SUSPENSION / RESUMPTION CRITERIA**

### **Suspend testing if:**

- Spotify/TMDB/OMDB/TasteDive APIs are down
- OAuth service fails
- No network available
- Application crashes without possibility to continue

### **Resume testing when:**

- External API services restored
  - Application is stable
  - Network restored
- 

## **1.9 TEST ENVIRONMENTAL / STAFFING / TRAINING NEEDS**

### **Environment Needs:**

- Python 3.x
- Flask

- Installed libraries (`requests`, `spotipy`, etc.)
- Valid Spotify Developer credentials
- Stable internet connection

**Staffing:**

- Single developer/tester

**Training Needs:**

- Basic familiarity with OAuth
- NIST control awareness
- Knowledge of API security concepts

---

## 2. CONTROL-BY-CONTROL NIST TESTING RESULTS

Below is every control you listed, marked as:

- **TESTED** — application is actually responsible
- **SYSTEM LEVEL** — applies to Spotify / external APIs
- **N/A** — not applicable because app stores no data, has no users, and no roles

---

## ACCESS CONTROL (AC)

**Control**

**Status**

**Justification**

<b>AC-1 Policy and Procedures</b>	N/A	No internal access control beyond OAuth
<b>AC-2 Account Management</b>	SYSTEM LEVEL	Spotify manages accounts
<b>AC-5 Separation of Duties</b>	N/A	No roles or admins exist
<b>AC-6 Least Privilege</b>	TESTED	App only requests minimal OAuth scopes
<b>AC-20 Use of External Systems</b>	SYSTEM LEVEL	Depends on Spotify/TMDB/OMDB externally

---

## AWARENESS & TRAINING (AT)

Control	Status	Justification
<b>AT-2 Literacy Training</b>	N/A	Single-developer academic project
<b>AT-3 Role-Based Training</b>	N/A	No system roles
<b>AT-4 Training Records</b>	N/A	No training program needed

---

## AUDIT & ACCOUNTABILITY (AU)

Control	Status	Justification
<b>AU-1 Policy &amp; Procedures</b>	N/A	No audit subsystem exists
<b>AU-2 Event Logging</b>	TESTED	App logs only non-PII debug info
<b>AU-3 Content of Audit Records</b>	N/A	No user actions stored

---

## MEDIA PROTECTION (MP)

Control	Status	Justification
MP-3 Media Marking	N/A	App stores no media, files, or data

---

## RISK ASSESSMENT (RA)

Control	Status	Justification
RA-3 Risk Assessment	TESTED	External API reliance documented
RA-5 Vulnerability Monitoring	SYSTEM LEVEL	Spotify handles security patches

---

## SYSTEM & ACQUISITION (SA)

Control	Status	Justification
SA-3 SDLC	TESTED	Simple lifecycle documented
SA-10 Developer Configuration Management	TESTED	<code>.env</code> used; keys not in repo
SA-11 Testing & Evaluation	TESTED	All controls mapped here
SA-15 Development Process/Tools	TESTED	Python/Flask + APIs documented

---

## SYSTEM INTEGRITY (SI)

Control	Status	Justification
SI-2 Flaw Remediation	TESTED	Bugs fixed as discovered
SI-3 Malicious Code Protection	N/A	No file uploads or scripts



<b>SI-4 System Monitoring</b>	SYSTEM LEVEL	Spotify monitors OAuth security
<b>SI-5 Security Alerts</b>	SYSTEM LEVEL	External APIs issue notices
<b>SI-7 Integrity</b>	TESTED	Input validation for API responses
<b>SI-11 Error Handling</b>	TESTED	No stack traces shown to user

---

## SUPPLY CHAIN RISK (SR)

Control	Status	Justification
<b>SR-2 Supply Chain Risk Management</b>	SYSTEM LEVEL	Spotify/TMDB/OMDB handle infra
<b>SR-8 Notification Agreements</b>	SYSTEM LEVEL	External APIs responsible

---

## Final Conclusion

**Big Music meets all applicable NIST SP 800-53 controls for its scope.**

Most controls are either:

- **Not Applicable** because no data is stored
- **System Level** because Spotify handles authentication
- **Tested** where appropriate (OAuth, error handling, logging, etc.)