



TU856/3 SOFTWARE TESTING

ASSIGNMENT 1 - SYSTEM REQUIREMENTS



18/03/2025

PAULINA CZARNOTA C21365726

1. Introduction

This document outlines the functional and non-functional requirements for the TunePal music library and streaming service. The system enables users to:

- Create accounts and log in securely.
- Access a shared song library and manage private collections.
- Search for songs by title, artist, and release year.
- Purchase premium songs via an in-app shop.
- Manage playlists and song downloads.
- Handle song playback efficiently with pagination.
- Ensure high system performance, security, and scalability.
- Ensure development is completed before August as per project deadlines.

Each requirement is clearly defined, testable, and aligned with business needs.

2. Functional Requirements & Test Cases

Each requirement is accompanied by corresponding test cases, expected results, test objects, and test bases to ensure proper validation.

| Requirement ID | Requirement | Test Steps | Expected Result | Test Object | Test Basis |
|----------------|--|---|--|-----------------------|-----------------------------|
| FR001 | Users must be able to create an account and log in using email and password. | 1. Navigate to Sign-Up page. 2. Enter valid credentials. 3. Click "Sign Up". 4. Attempt login with the same credentials. | User successfully registered and logged in. | Authentication System | Business Requirements |
| FR002 | Users must have access to a shared library and add personal songs. | 1. Open library. 2. Search shared songs. 3. Upload a private song. 4. Verify song appears in "My Library". | Songs appear correctly in shared and private library. | Music Library | UI & Backend Specifications |
| FR003 | The library must be searchable by artist, title, and year, including special characters. | 1. Open search bar. 2. Enter artist name/title/year. 3. Include special characters (&, %, \$). 4. Click search. | Search returns relevant songs while correctly handling special characters. | Search Module | System Requirements |

| | | | | | |
|--------------|--|--|--|---------------------------|-----------------------------|
| FR004 | Users can log in on up to two devices simultaneously. | 1. Log in on Device 1. 2. Log in on Device 2. 3. Attempt login on Device 3. | Third login is denied. | Authentication System | Security Policies |
| FR005 | Songs must download within 5 seconds. | 1. Select song. 2. Click download. 3. Measure time. | Download completes ≤ 5 seconds. | Download Manager | Performance Requirements |
| FR006 | Users can access a shop to buy premium songs. | 1. Open shop. 2. Add item to basket. 3. Proceed to checkout. | Purchase completes successfully. | Payment System | Business Requirements |
| FR007 | Users can empty their shopping basket. | 1. Open basket. 2. Click "Empty Basket". 3. Verify basket is empty. | Basket is cleared successfully. | Basket Module | UI Requirements |
| FR008 | Users must receive an order confirmation email after purchase. | 1. Complete a purchase. 2. Check email inbox. | Confirmation email is received. | Email Notification System | Business Requirements |
| FR009 | Users should be able to recover a forgotten password. | 1. Click "Forgot Password". 2. Enter registered email. 3. Receive reset link. 4. Reset password successfully. | Password reset is successful. | Authentication System | Security Policies |
| FR010 | Users should be able to manage and edit their playlists. | 1. Create a playlist. 2. Add, remove, and reorder songs. 3. Save changes. | Playlist updates correctly. | Playlist Management | UI & Backend Specifications |
| FR011 | The system should support pagination for large song libraries. | 1. Load song list. 2. Navigate to next page. 3. Navigate to previous page. | Song list is paginated correctly. | Pagination System | System Requirements |
| FR012 | The system should handle invalid inputs gracefully. | 1. Enter invalid year for search. 2. Enter negative page size. | System shows appropriate error message and does not crash. | Input Validation System | Security Requirements |

| | | | | | |
|--------------|---|---|---|---------------------------|----------------------|
| FR013 | The system must be completed and ready for release by August. | 1. Verify development progress at each milestone. 2. Confirm completion by the deadline. | System is finalized and deployed before August. | Project Management System | Development Timeline |
|--------------|---|---|---|---------------------------|----------------------|

3. Non-Functional Requirements & Validation

These non-functional requirements ensure that TunePal meets performance, security, and usability standards.

| Requirement ID | Requirement | Validation Method |
|----------------|---|--------------------------------------|
| NFR001 | The app must be responsive across mobile, tablet, and desktop. | UI testing across different devices. |
| NFR002 | Login system must use password encryption. | Code review & security test. |
| NFR003 | The system must support 10,000+ concurrent users. | Load testing with JMeter. |
| NFR004 | All payments must be PCI-DSS compliant. | Compliance audit. |
| NFR005 | User data must be backed up daily. | Backup verification |
| NFR006 | Songs must stream without buffering under normal network conditions. | Performance testing. |
| NFR007 | API responses should complete within $\leq 200\text{ms}$ for standard queries. | Performance testing. |
| NFR008 | The system should be available 99.9% of the time (high availability). | Uptime monitoring. |
| NFR009 | The UI should be accessible to users with disabilities (WCAG 2.1 compliance). | Accessibility testing. |
| NFR010 | The system must log security events such as failed logins and unauthorized access attempts. | Security audit. |

4. Edge Cases & Additional Considerations

This section outlines potential system challenges and ensures proper handling.

| Scenario | Handling Approach |
|-------------------------------------|---|
| Invalid Login Attempts | System locks users out after 5 failed login attempts within 10 minutes. |
| Special Characters in Search | Ensure search correctly handles special characters (&, %, \$) . |
| Duplicate Song Uploads | System prevents duplicate songs from being uploaded to a private library. |

| | |
|---------------------------|---|
| Network Failures | Purchases & downloads resume after reconnection to avoid data loss. |
| System Scalability | Stress testing ensures that the system handles increasing user load. |
| Payment Errors | Implement retry mechanisms and clear error messaging for failed transactions. |

5. Conclusion

This document outlines the system requirements for TunePal, ensuring that both functional and non-functional requirements are testable. The corresponding test cases, test objects, and test bases are clearly identified for each requirement, making it easier to develop test scripts and ensure that all aspects of the system are validated before release.

By following this structured approach, we ensure that TunePal meets its functional, performance, and security needs, providing users with a seamless, reliable, and secure music streaming experience.