Test Plan: License Management Web Application

1. Core details

Field	Detail
Document Version	1.0
Project Name	License Management Module (Version 2.1)
Prepared By	Haseeb Munawar
Date	2025-xx-xx
Purpose	To define the strategy, scope, objectives, resources, and schedule for testing the License Management and Asset Compliance features.
Target Audience	Development Team, Product Owners, Project Manager, QA Team.

2. Test Objectives

The core objective of the testing phase is to confirm that the application fully satisfies all defined functional and business requirements before deployment. This involves ensuring **Compliance Integrity** through strict enforcement of all license limits and expiration rules. We will verify **Data Accuracy** by confirming consistency between reports, the UI, and the backend database. Additionally, we must validate **Role-Based Access (RBAC)** to guarantee proper user permissions, and confirm that all key application workflows demonstrate high **Usability** and efficiency.

3. Entry Criteria

Testing will **only commence** when the following criteria are met:

- **Code Complete:** Development has confirmed that all features and fixes included in the 2.1 scope have been checked into the designated QA branch.
- **Environment Stability:** The dedicated QA/Staging environment is deployed, fully stable, and accessible to the SQA Team.
- **Test Data Ready:** All required test data (licenses, assets, user roles) has been loaded into the QA database.
- **Build Handover:** A formal, documented build (2.1.0-Beta) has been delivered by Development to QA.
- **Smoke Test Passed:** An initial successful **Smoke Test** by the QA team confirms the core application (login, dashboard access, navigation) is functional.

4. Scope of Testing

4.1 In-Scope (What We Will Test)

- **Functional Testing:** License creation, modification, deactivation, and usage tracking.
- **Compliance Logic:** Enforcement of concurrent user limits and validation of license expiration.
- **Asset Management Integration:** Linking licenses to computer assets, user accounts, and organizational groups.
- **Data Validation:** Testing all report generation (PDF/CSV) against database integrity checks.
- **Security (RBAC):** Testing user roles (Admin, Manager, Read-Only) against system features.

4.2 Out-of-Scope (What We Will NOT Test)

- Performance Testing (Load and Stress testing, planned for a separate phase).
- Security Penetration Testing (handled by a dedicated Security team).

5. Test Strategy and Types

Test Type	Objective	Approach	Criteria for Success
Smoke Testing	To ensure critical functions are stable enough for detailed testing.	After every new build, verify: Login, license creation, and basic asset display.	100% of smoke test cases must pass.
Functional Testing	To validate all business requirements and specifications.	Execution of the detailed Test Cases (TC-001 through TC-025, and others).	All High-Priority P1/P2 test cases must pass.
Regression Testing	To ensure new code changes haven't broken existing, proven functionality.	Execution of an automated regression suite (Playwright with JavaScript) focusing on core modules (Authentication, License Assignment).	Pass rate of 95% or higher on the automated suite.
API Testing	To validate the integrity of the data layer and business logic.	Direct validation of REST API endpoints using **Postman** (e.g., license usage update, asset creation).	API responses must be consistent (HTTP 200/201) and JSON payload integrity must be validated.
Database Testing	To confirm data accuracy and transaction consistency.	Executing SQL queries (**PostgreSQL**) to verify license counts, asset links, and report data integrity.	Data counts and relationships in the DB must match the UI.

6. Roles and Responsibilities

Role	Key Responsibilities
SQA Lead	Test Plan creation, test case review, daily reporting, automation framework maintenance, stakeholder communication.
SQA Engineer	Manual test case execution, detailed bug logging, regression testing.
Development Team	Bug fixing, environment setup, providing required database access/tools.
Product Owner	Final UAT (User Acceptance Testing) and sign-off on delivered features.

7. Test Environment and Tools

Category	Details
Test Environment	Staging Environment
Operating Systems	Windows 10, macOS (for browser compatibility checks).
Browsers	Chrome (Primary), Firefox (Secondary).
Test Case Management	JIRA
Defect Tracking	JIRA
API/DB Tools	Postman, DBeaver

8. Test Schedule and Deliverables

Phase	Estimated Duration	Key Activities	Deliverable
Test Design	3 days	Finalize Test Plan, create/review all Test Cases.	Approved Test Plan, Final Test Case Document.
Test Execution (Initial)	5 days	Smoke, Functional, and API testing on initial build.	Daily Status Reports, Critical Bug Reports (P1/P2).
Regression & Re-Testing	3 days	Full regression execution after major bug fixes.	Regression Test Summary Report.
UAT & Sign-Off	2 days	PO/Stakeholder execution of key scenarios.	Final Test Summary Report, Go/No-Go Decision.

9. Exit Criteria (When We Stop Testing)

Testing will be considered complete and the application ready for deployment only when the following criteria are met:

- **Test Case Completion:** 100% of all High-Priority (P1/P2) test cases are executed.
- **Defect Closure Rate:** 100% of **Critical** and **High** severity bugs are fixed, re-tested, and closed.
- **Regression Stability:** The automated regression suite achieves a passing rate of $\geq 95\%$ on the final build.
- **Stakeholder Approval:** The Product Owner has signed off on the User Acceptance Testing (UAT).