Product Requirements Document: Session-Based Test Management Tool

Introduction/Overview

The Session-Based Test Management (SBTM) Tool is a web application designed to support the creation, editing, and management of test session reports. The tool addresses current pain points in testing workflows including manual tracking, scattered information, and lack of visibility into testing activities.

SBTM organizes testing into "sessions" - blocks of uninterrupted time during which a tester performs testing work. These sessions are documented in session reports that capture the testing story while providing management with necessary visibility without micromanaging the testing process.

Goal: Create a comprehensive web-based tool that minimizes overhead for tracking test sessions, provides flexibility for different SBTM variants, and serves the needs of testers, test leads, and middle management.

Goals

1. **Reduce Administrative Overhead:** Minimize time spent on session tracking and reporting
2. **Improve Visibility:** Provide clear insights into testing activities for management
3. **Maintain Testing Flexibility:** Preserve analytical and creative freedom for testers
4. **Centralize Information:** Eliminate scattered testing documentation
5. **Enable Data-Driven Decisions:** Provide normalized TBS metrics and analytics
6. **Support Multiple Projects:** Allow complete control over settings and templates per project

User Stories

As a Tester:

- I want to quickly create a session from a charter template so that I can start testing without administrative delays
- I want to edit my session notes in a distraction-free environment so that I can focus on documenting findings
- I want to track my time allocation (TBS metrics) so that I can analyze my testing efficiency
- I want auto-save functionality so that I don't lose my work if something unexpected happens

As a Test Lead:

- I want to view all team sessions and their status so that I can track testing progress
- I want to create and organize charter templates so that testers have clear mission statements
- I want to filter and search sessions by various criteria so that I can quickly find specific information
- I want to export session data so that I can create reports for stakeholders

As a Manager:

- I want to view normalized TBS statistics across the team so that I can understand testing productivity
- I want to see testing growth over time so that I can track team performance trends
- I want to export statistics to CSV so that I can incorporate data into executive reports
- I want read-only access to ensure I can monitor without interfering with testing activities

Functional Requirements

Core Session Management

1. The system must allow users to create new test sessions
2. The system must allow users to select from existing charter templates when creating sessions
3. The system must allow users to enter local charters (up to 2K characters) for individual sessions
4. The system must allow users to associate sessions with multiple charters from the charter repository
5. The system must require association with one or more testers for each session
6. The system must require association with one or more products for each session
7. The system must allow optional association with test environments, documents, issues, bugs, files, and tags
8. The system must capture TBS metrics (Testing, Bug investigation, Setup/admin) that sum to 100%
9. The system must record session start time and duration
10. The system must support session status tracking (Not Started/In Progress/Completed/Accepted)
11. The system must provide a large, distraction-free notes editor supporting up to 32K characters of markdown text and images
12. The system must allow the notes editor to be displayed as a floating window
13. The system must provide an option to keep the notes editor window on top of other windows
14. The system must auto-save sessions in progress at regular intervals
15. The system must maintain complete version history of all session changes with timestamps and user attribution

Session Views and Navigation

16. The system must provide a session list view showing all completed sessions and templates
17. The system must display normalized TBS statistics in the session list view
18. The system must allow users to search sessions by any field
19. The system must provide filtering and sorting capabilities for the session list
20. The system must allow sessions to be grouped and viewed by tags
21. The system must support exporting sessions to JSON format
22. The system must support exporting sessions to mind map format (FreeMind and XMind)

Charter Management

23. The system must provide a charter template repository for reusable charter statements
24. The system must allow charter templates to support up to 32K characters of rich text
25. The system must allow charter templates to be organized into nested groups
26. The system must provide search, filter, and sort capabilities for charter templates
27. The system must allow charter templates to be grouped and viewed by tags
28. The system must support exporting charter templates to JSON format
29. The system must allow editing of charter templates
30. The system must support printing charter templates
31. The system must provide one-click creation of charter templates from existing charters

Repository Management

32. The system must maintain a tester repository that does not require system accounts for all testers
33. The system must maintain a product repository with name, description, and version tracking
34. The system must maintain a test environment repository with versioning and change history
35. The system must maintain a tag repository with hierarchical organization
36. The system must maintain a document repository with version control
37. The system must allow search, filter, sort, and export capabilities for all repositories
38. The system must support JSON and XML export formats for tester, product, and tag data

Tag System

39. The system must allow tags to be organized into nested hierarchical groups
40. The system must allow tags to belong to multiple groups simultaneously
41. The system must support mind map visualization of tag hierarchies

42. The system must allow importing tags from JSON, XML, or mind map formats
43. The system must provide tag remapping functionality (search and replace tags)

Analytics Dashboard

44. The system must display normalized session growth over time in a plot format
45. The system must show pie charts breaking down TBS metrics and opportunity time
46. The system must plot normalized TBS statistics against collapsible tag hierarchy
47. The system must allow users to set time scales and tag-based filters for all graphs
48. The system must support exporting statistics to CSV format

Project Management

49. The system must support multiple projects with complete isolation of settings and data
50. The system must allow copying items between projects
51. The system must provide project-level control over all templates and configurations

Session Templates

52. The system must allow creation of session templates from partially completed sessions
53. The system must allow session templates to include notes, tags, charters, and associated documents
54. The system must allow session templates to be organized into nested groups

User Access Control

55. The system must implement role-based permissions (Tester, Test Lead, Manager)
56. The system must allow testers to edit only their own sessions (unless specifically granted additional permissions)
57. The system must allow test leads to view and edit all sessions within their projects
58. The system must provide managers with read-only access to dashboards and reports

Non-Goals (Out of Scope)

- **Issue/Bug Tracking System:** This tool will reference external issue and bug reports but will not replace dedicated bug tracking systems
- **Test Case Management:** This tool focuses on session-based testing, not traditional test case execution
- **Test Automation Integration:** Initial version will not integrate with automated testing frameworks

- **Advanced Project Management:** This tool will not replace project management software for broader project tracking
- **User Account Management:** The system will not include comprehensive user onboarding, password reset, or profile management beyond basic authentication
- **Real-time Collaboration:** Initial version will not support simultaneous editing by multiple users
- **Mobile Application:** Initial version will be web-only, not a dedicated mobile app
- **Advanced Reporting:** Complex custom report builders are out of scope for MVP

Design Considerations

Technology Stack:

- Frontend: Angular with TypeScript
- Backend: Python
- Database: PostgreSQL
- Styling: Tailwind CSS with a custom design system

UI/UX Requirements:

- Responsive web design that works on desktop and tablet devices
- Clean, minimalist interface that reduces distractions during testing
- Floating/modal windows for notes editing to maximize screen real estate
- Consistent design system across all views
- Fast loading times and smooth interactions
- Accessible design following WCAG guidelines

Key Interface Views:

- Session View: Primary interface for creating and editing individual sessions
- Session List View: Overview of all sessions with filtering and search
- Charter View: Management of charter templates and organization
- Statistics Dashboard: Visual analytics and reporting interface
- Repository Views: Management interfaces for testers, products, environments, tags, and documents

Technical Considerations

Database Design:

- Implement proper foreign key relationships between all repositories
- Use PostgreSQL JSON columns for flexible metadata storage
- Design for efficient querying of hierarchical tag structures
- Implement soft deletes to maintain data integrity

Performance Requirements:

- Page load times should be under 2 seconds
- Search results should return within 1 second
- Auto-save should occur every 30 seconds without blocking user input
- Support concurrent users up to 100 simultaneous sessions

Security:

- Implement role-based access control
- Secure API endpoints with proper authentication
- Validate all user inputs to prevent XSS and SQL injection
- Encrypt sensitive data at rest

Integration Points:

- Export APIs for JSON, XML, CSV, and mind map formats
- Import APIs for tags and templates
- Consider future integration hooks for external bug tracking systems

Success Metrics

Primary Metrics:

- **Time Savings:** Reduce session report creation time by 50% compared to manual methods
- **Adoption Rate:** Achieve 80% active usage by testers within 3 months of deployment
- **Data Completeness:** Increase complete session reports (all required fields filled) to 90%

Secondary Metrics:

- **Management Visibility:** Enable 100% of test leads to access real-time testing status
- **Search Efficiency:** Users can find specific sessions within 30 seconds
- **Export Usage:** 70% of test leads use export functionality monthly for reporting

User Satisfaction Metrics:

- User satisfaction score of 4.0+ out of 5.0
- Less than 5% of sessions require manual correction after initial entry
- Zero critical bugs in production after first month

Open Questions

1. **Concurrent Editing:** How should the system handle conflicts when multiple users attempt to edit related data (e.g., charter templates being used in active sessions)?
2. **Data Retention:** What are the requirements for archiving old projects and sessions? How long should revision history be maintained?
3. **Performance Scale:** What is the expected maximum number of sessions per project and total number of projects the system should support?
4. **Integration Priority:** Which external systems (if any) should be prioritized for future integration phases?
5. **Offline Capability:** Should future versions support offline editing with synchronization when connectivity returns?
6. **Advanced Analytics:** What additional statistical analyses or visualizations would be most valuable for management reporting?
7. **Custom Fields:** Should the system support project-specific custom fields for sessions, or is the defined schema sufficient?
8. **Backup and Recovery:** What are the specific requirements for data backup, disaster recovery, and business continuity?

MVP Feature Prioritization

To be determined collaboratively with stakeholders. Suggested approach: prioritize core session creation, editing, and basic reporting functionality for MVP, with advanced analytics and export features in subsequent releases.