Melany Paola Rivera Lores

NAO ID: 3332

Date: November 07th, 2025

Pathway: Technoready Bécalos

Challenge: Java and JavaScript. Programming Procedures

Backlog

This document outlines the product backlog for the BookingMx Testing and Reliability enhancement project. The primary objective of this initiative is to improve the reliability and performance of the live reservation system by implementing a comprehensive unit testing strategy. This backlog details the user stories and requirements across three Sprints, focusing on Java (JUnit) testing for the reservations module, JavaScript (Jest) testing for the graph visualization module, and the creation of thorough project documentation.

Key	User Story	Priority	Acceptance Criteria
US1	As a developer, I want to configure the Java project with JUnit dependencies to establish the testing environment.	High	- The pom.xml or build.gradle file includes JUnit 5 dependencies. - A simple test case executes without configuration errors.
US2	As a developer, I want to implement unit tests for the core functionalities of the reservations module (create, edit, cancel).	High	- Test classes exist for ReservationService (or similar). - Methods for creating, editing, and canceling reservations have tests covering success and error cases. - All tests pass successfully.
US3	As a developer (Karen), I want to achieve at least 90% code coverage with JUnit tests for the reservations module.	High	- A coverage report is generated JaCoCo. - The report shows line or branch coverage >= 90% for the reservations module.

US4	As a developer (Teresa), I want to configure the JavaScript project with Jest to create the unit testing framework.	High	- package.json includes Jest dependencies. - The npm test script runs Jest correctly. - A simple test case for a component executes successfully.
US5	As a developer (Teresa), I want to implement unit tests for the graph visualization module to verify it displays cities and distances.	High	- Tests validate the graph rendering logic. - Tests verify that input data (cities, distances) is processed and displayed correctly. - All tests for the graph module pass.
US6	As a developer (Teresa), I want to achieve at least 90% code coverage with Jest tests for the graph visualization module.	High	- A Jest coverage report is generated (--coverage). - The report shows coverage >= 90% for the graph module files.
US7	As a development team, I want to create a comprehensive README.md with a project summary, installation instructions, and test descriptions.	High	- The README.md on GitHub is updated. - It contains clear sections for "Overview," "Installation," and "Implemented Tests."
US8	As a development team, I want to add detailed code comments and create system architecture diagrams.	Medium	- Key functions and classes in Java and JavaScript have Javadoc/JSDoc comments. - At least one diagram exists explaining the architecture of the tested modules.

US9	As a project team, I want to create a final PDF presentation analyzing the testing process and the results obtained.	High	- A Test_Results.pdf document is delivered. - The document includes coverage analysis, errors found, and reliability improvements.
US10	As a project team, I want to record an MP4 video presentation to demonstrate the solution and project outcomes.	High	- A Final_Presentation.mp4 file is delivered. - The video explains the problem, the testing solution, and the results in an executive format.

Requirement	Stage (Sprint)	Time Estimate	Actual Time	Status	Deliverable	Dependency
Configure JUnit Environment (US1)	Sprint 1	1 day	Done	JUnit dependencies configured.	-	Configure JUnit Environment (US1)
Develop Reservation Tests (US2)	Sprint 1	1 day	Done	Java test classes (Test.java).	US1	Develop Reservation Tests (US2)
Validate 90% Java Coverage (US3)	Sprint 1	1 day	Done	Coverage report (JaCoCo).	US2	Validate 90% Java Coverage (US3)
Configure Jest Environment (US4)	Sprint 2	1 day	To Do	package.json & jest.config.js updated.	-	Configure Jest Environment (US4)
Develop Graph Module Tests (US5)	Sprint 2	3 days	To Do	Test files (*.test.js).	US4	Develop Graph Module Tests (US5)

Validate 90% JS Coverage (US6)	Sprint 2	1 day	To Do	Jest coverage report.	US5	Validate 90% JS Coverage (US6)
Update README.md (US7)	Sprint 3	2 days	To Do	Finalized README.md file.	US3, US6	Update README.md (US7)
Code Comments & Diagrams (US8)	Sprint 3	2 days	To Do	Commented code and diagram file.	US3, US6	Code Comments & Diagrams (US8)
Prepare PDF Presentation (US9)	Final	2 days	To Do	Test_Results.pdf	US1-US8	Prepare PDF Presentation (US9)
Record Video Presentation (US10)	Final	1 day	To Do	Final_Presentation.mp4	US9	Record Video Presentation (US10)

Upon successful completion of the deliverables outlined in this backlog, the BookingMx platform will possess a robust testing framework, ensuring a minimum of 90% code coverage for the specified modules. This initiative will significantly reduce production failures, improve code maintainability, and establish a quality assurance standard for future development. The final deliverables will provide a complete analysis of the project's impact and outcomes for stakeholder review.