

CPS6002 Software Testing and Quality Assurance - Assessment Brief

Module Code:	CPS6002
Module Title:	Software Testing and Quality Assurance
Module Convenor:	Ruslan Posevkin
Module Level:	6

Assessment Number:	2
Assessment Title:	Software Testing Project
Assessment Weight:	60%
Assessment Individual/Group:	Group
Assessment Type:	Project
Assessment Time/Word Count Restrictions:	3000 words report 1 software artefact
Assessment Time/Word Count Limit Consequences:	It is essential that assignments keep within the time/word count limit stated above. Any work beyond the maximum time/word length permitted will be disregarded and not accounted for in the final grade.

Issue Date:	3 rd November 2025
Hand in Date:	8 th January 2026
Planned Feedback Date:	Within 3 working weeks
Mode of Submission:	Online via Moodle
Number of copies to be submitted:	1 copy of each of the following: <ul style="list-style-type: none"> • a report in pdf format • a zip file containing your software artefact

Author:	Ruslan Posevkin
Internal Moderator:	Prins Butt
Moderation Date:	15 th September 2025

CPS6002 Software Testing and Quality Assurance - Assessment Brief

Table of Contents

Introduction	3
Problem Scenario	3
Requirements.....	5
Submission	8
Assessment Criteria	9
Learning Outcomes	11
Regulations, Policies, and Guidelines	13

CPS6002 Software Testing and Quality Assurance - Assessment Brief

Introduction

This assessment aims to evaluate your ability to develop a testing plan for a software application, conduct testing using a range of testing techniques, and report on findings.

Problem Scenario

SMUCampusHub is a university web platform designed to help students and staff discover, manage, and book limited-capacity events, including talks, lab sessions, and office-hour slots. The platform supports both public-facing users and authenticated Student/Staff roles and exposes a public REST API for integration with a Single Page Application (SPA) frontend.

The university requires a robust, secure, and scalable solution capable of handling realistic high-concurrency scenarios, complex booking rules, and compliance with data protection standards.

Core Capabilities

Event Catalogue

- Browse and search events with filters (date, type, department).
- Event details page displays remaining capacity and waitlist status.

Authentication & Roles

- Students: view bookings, manage waitlists, and cancel/reschedule bookings.
- Staff: create/edit events, set capacity limits, overbook by up to 5% for critical events, export attendee data to CSV.

Booking Lifecycle

- Flow: hold → confirm → cancel.
- Automatic waitlist promotion when slots become available.
- Notifications via email and calendar integration for bookings and waitlist updates.

CPS6002 Software Testing and Quality Assurance - Assessment Brief

Non-Functional Expectations

- **Performance:** must remain responsive under up to 500 concurrent users.
- **Security:** protect sensitive personal data; enforce authentication/authorisation checks.
- **Usability & Accessibility:** accessible via common browsers and devices;
- **Resilience:** handle API or database downtime gracefully.
- **Compatibility:** consistent behaviour across major browsers and mobile viewports.

API

- Endpoints: /auth, /events, /bookings.
- Includes authentication, booking lifecycle, event management, and error handling.

Additional Considerations

Edge Cases & Complex Rules:

- Students may only book events for courses they are enrolled in.
- Staff can overbook under controlled conditions.
- Notifications must correctly reflect waitlist changes and cancellations.

Integration Points: calendar and email systems require secure handling of personal data.

High-Concurrency Scenarios: system must handle multiple simultaneous booking requests without data inconsistencies.

CPS6002 Software Testing and Quality Assurance - Assessment Brief

Your Task

You are required to plan and execute a **risk-aware, quadrant-based test campaign** for SMUCampusHub, producing evidence-based insights for stakeholders. Your campaign should cover:

- **Quadrant 1:** Unit and component testing for critical booking logic (capacity, waitlist promotion, input validation).
- **Quadrant 2:** Functional and acceptance testing of key flows (search → view → book → cancel) and staff management capabilities.
- **Quadrant 3:** Usability testing, exploratory testing with clear goals, and evaluation of accessibility features.
- **Quadrant 4:** Performance, security, and cross-browser/platform compatibility testing.

Where appropriate, use test automation tools referenced in the module to enhance coverage, reproducibility, and efficiency. Ensure your reporting captures metrics, defects, risks, and recommendations, and reflects professional, ethical practice in data handling and test execution.

Requirements

The requirements for the assessment are as follows. Complete the Basic Requirements tier first. Then you can extend to Intermediate and Advanced Requirements tiers for higher attainment, matching the tiered pattern in the sample brief.

Basic Requirements:

Planning and estimation

- Produce a concise test plan that states scope, objectives, approach by testing quadrant, environments, data needs, estimation of effort, and a high-level schedule.

Test design and execution

- Design and execute functional/acceptance tests for the key flows (*search → view event → book → cancel*) and for **Staff** role event management.
- Implement unit and component tests for critical logic (e.g. capacity checks, waitlist promotion, input validation).

CPS6002 Software Testing and Quality Assurance - Assessment Brief

- Exercise key API endpoints with appropriate requests/responses and error handling.
- Record outcomes and evidence according to test execution and reporting guidance.

Defect tracking

- Maintain a defect log with clear descriptions, steps to reproduce, and relevant artefacts (e.g. screens, logs), reflecting effective defect tracking and management.

Artefact Deliverables

- Test plan document (PDF).
- Test cases and execution evidence (screenshots/logs).
- Defect log (Excel, CSV, or integrated tool output).

Intermediate Requirements:

Test automation and tools

- Introduce test automation for suitable parts of SMUCampusHub (e.g. repeatable functional flows and/or API checks) using popular automation tools referenced in the module. Generate readable reports from automated runs.

Usability and exploratory testing

- Conduct usability testing of key screens using appropriate evaluation methods. Document findings and improvements.
- Run exploratory testing sessions (with clear goals) to uncover issues beyond scripted cases. Report notable observations and follow-ups.

Metrics, communication and ethics

- Present key test metrics (e.g. test coverage at the level you can justify, pass/fail counts, defect trends) and interpret them for key stakeholders.
- Briefly reflect on ethics and professional practice in your approach (e.g. impartiality, data handling, and the implications of automation).

Artefact Deliverables

- Updated test plan showing automation strategy.
- Automated test scripts and reports.

CPS6002 Software Testing and Quality Assurance - Assessment Brief

- Usability/accessibility evaluation summary.
- Updated defect log including exploratory testing findings.

Advanced Requirements:

Performance testing

- Apply performance testing fundamentals to representative SMUCampusHub scenarios (e.g. searching events, confirming a booking, CSV export). Report response characteristics and observations about system behaviour under load and suggest improvements.

Security testing

- Perform security testing activities aligned to the module (e.g. authentication/authorisation checks, input handling, session management). Identify potential security vulnerabilities and provide evidence and actionable recommendations.

Compatibility testing

- Execute a compatibility testing plan covering common browsers/viewports and relevant platforms. Summarise cross-browser/platform issues and proposed fixes.

Integrated Quality narrative

- Integrate results across quadrants into a succinct test report that explains risks, priorities, and next steps, using module-aligned test metrics and performance tracking to support your conclusions.

Artefact Deliverables

- Final, integrated test report (PDF) including test plan, executed cases, metrics, defects, exploratory/usability findings, automation reports, performance/security/compatibility results, and risk-based recommendations.
- Fully documented software artefact (ZIP) source code, test scripts, datasets, git logs, and instructions for running the solution and reproducing tests.

CPS6002 Software Testing and Quality Assurance - Assessment Brief

Submission

The assessment must be completed in groups (of 2 or 3). You must not share, in part or whole, your assessment with another team and party other than the module convenor and for the purpose of submission to the university. You must ensure that the University's academic misconduct guidelines are followed in their entirety.

You should use the assessment submission link on the module's Moodle page to submit the following files:

A PDF file for your report. This should not be included in the zip file but instead submitted as a separate file. Failure to do so may result in zero being awarded.

A Zip file of your software artefact. This should contain your solution files, git log, and other files required to view and run your solution.

You should ensure that you make a timely submission by the deadline stated at the start of this assessment brief.

CPS6002 Software Testing and Quality Assurance - Assessment Brief

Assessment Criteria

Your assessment will be graded according to the following criteria:

Grading criteria	Functionality	Documentation	Professional Practice
Mark band			
80-100 Pass (1st)	Expertly design and implement tests across all four quadrants. Demonstrate comprehensive unit, component, API, functional, usability, exploratory, performance, security, and compatibility testing. Automation is effectively applied, with clear justification of coverage, risk, and prioritisation.	Test plan and cases are exemplary, clearly structured, and fully executed. Metrics are insightful, evidence is complete, and defects are meticulously tracked with follow-up. Final integrated report synthesises all findings and provides actionable recommendations.	Outstanding professional and ethical practice. Reports and artefacts are highly clear, well-presented, and persuasive. Tool usage is highly effective, reflecting best practice. Ethical considerations are thoroughly addressed, including data handling, impartiality, and automation impact.
70-79 Pass (1st)	Comprehensive design and execution of quadrant-based tests. Strong coverage of unit, component, API, functional, exploratory, and usability tests. Some non-functional testing applied effectively (performance/security /compatibility). Automation is appropriate and well justified.	Well-structured test plan, cases, and reports. Defects tracked clearly with supporting evidence. Metrics are accurate and interpreted effectively.	Strong ethics and professional communication. Tool usage is competent and purposeful. Reports communicate results clearly and professionally.
60-69 Pass (2.1)	Design and execute tests for core flows and critical logic. Functional, unit, API, and some exploratory/usability testing are applied. Initial non-functional testing attempted. Partial automation implemented.	Clear test plan and cases, with adequate execution evidence. Defects are tracked and basic metrics interpreted.	Ethical practice is evident. Communication is generally clear. Tool use demonstrates basic proficiency.
50-59 Pass (2.2)	Essential functional and unit/API tests executed. Limited exploratory, usability,	Test plan and cases presented; execution evidence exists but is minimal. Defects recorded	Basic ethical awareness. Communication understandable but may lack clarity. Some use of

CPS6002 Software Testing and Quality Assurance - Assessment Brief

	and non-functional testing. Basic automation applied.	with limited analysis. Metrics are minimal or absent.	testing tools.
40-49 Pass (3rd) (Threshold)	Minimal functional, unit, or API testing. Little exploratory or usability testing. Non-functional testing largely absent. Ad-hoc automation, if any.	Test plan and cases minimally presented. Evidence of execution limited. Defect tracking is basic. Metrics absent.	Limited professional practice. Organisation and communication weak. Tool use minimal.
30-39 Fail	Very limited testing across quadrants. Major gaps in functional, unit, API, or automation testing.	Inadequate documentation. Execution evidence missing or incomplete. Defect tracking and metrics largely absent.	Weak professional practice, ethical awareness, and communication. Tool use ineffective.
0-29 Fail	Unable to design or execute basic tests. Fundamentals of test design not demonstrated.	Little or no coherent documentation. No credible artefacts or reports produced.	Professional practice and ethical awareness absent. Tool competency not demonstrated.

CPS6002 Software Testing and Quality Assurance - Assessment Brief

Learning Outcomes

This assessment will enable students to demonstrate the following learning outcomes as stated in the module outline:

[Module Learning Outcome 4]

- Apply advanced testing strategies and techniques, considering factors such as test coverage, test prioritisation, risk analysis, as well as legal, social, and ethical considerations, to validate and verify software functionality.

How is this learning outcome addressed?

Students plan and execute a risk-aware testing campaign across the testing quadrants: unit, component, API; acceptance and functional; usability and exploratory; performance, security and compatibility. They justify coverage and prioritisation choices, perform validation and verification activities, and reflect on legal, social, and ethical aspects within the test plan and report.

[Module Learning Outcome 5]

- Efficiently use industry-standard testing frameworks and tools for test automation, enhancing testing efficiency and accuracy while showcasing best practices.

How is this learning outcome addressed?

The assessment requires appropriate use of test automation and tools on suitable parts of the system (e.g. repeatable functional and/or API checks), producing readable reports and using results to inform further testing: directly reflecting the syllabus focus on test automation and appropriate tools.

CPS6002 Software Testing and Quality Assurance - Assessment Brief

[Module Learning Outcome 6]

- Demonstrate effective project management skills in software testing projects, including planning, estimation, and progress tracking, ensuring timely and successful completion of testing activities.

How is this learning outcome addressed?

Students produce a concise test plan with scope and estimation, schedule testing activities, track execution and defects. They provide regular progress evidence in their artefacts, aligning with test planning and estimation and test execution with reporting in the syllabus and the multi-week project format.

[Module Learning Outcome 7]

- Communicate and present testing results, along with well-founded recommendations for enhancing software quality, through comprehensive written reports and articulate presentations.

How is this learning outcome addressed?

The assessment culminates in a clear test report that explains outcomes, defects, risks, and recommendations, using appropriate test metrics and performance tracking. Students also present findings to stakeholders which matching the syllabus emphasis on test reporting and communicating results.

CPS6002 Software Testing and Quality Assurance - Assessment Brief

Regulations, Policies, and Guidelines

Guidance for online submissions

<https://www.stmarys.ac.uk/policies/online-submissions.aspx>

Academic Misconduct

Any submission must be students' own work and, where facts or ideas have been used from other sources, these sources must be appropriately referenced. Please find a link to the academic misconduct policy below:

<https://www.stmarys.ac.uk/policies/academic-regulations.aspx>

Ethics Policy

The work being carried out by students must be in compliance with the Ethics Policy. Where there is an ethical issue, as specified within the Ethics Policy, then students will need ethical approval prior to the start of the project. Please find a link to the ethics policy below:

<https://www.stmarys.ac.uk/research/students/ethical-review-process.aspx>

Extenuating Circumstances

The University's Extenuating Circumstances procedure helps students facing challenges in assessment submission. To request an extension or deferment, submit an EC application with evidence. Approved cases will not incur academic penalties. For longer-term issues, contact Student Services. Please find a link to the EC policy below:

<https://www.stmarys.ac.uk/policies/extenuating-circumstances.aspx>

Reassessment

If a student fails to meet the assessment criteria, they may be eligible for a reassessment. The reassessment will follow the same guidelines as the original assessment, requiring the student to reattempt the tasks outlined. Students should review any feedback provided and make improvements to meet the expected learning outcomes. In case of reassessment, it is important to consult with the module convenor for clarity on the areas of improvement and ensure compliance with submission deadlines and regulations.