**Case Study Addendum: Enhancing NEPTUN System Testing**

**Author:** Munkhjargal Ariunbold

**Supervisor:** Prof. László Pitlik (critical aspects, quality assurance)

**Date:** 2025.02.16

**Email:** pitlik.laszlo@kodolanyi.hu

Initial documents:

* <https://miau.my-x.hu/miau/320/moodle_cubes_logic/>
* **<https://miau.my-x.hu/miau/320/moodle_neptun_tests/>**
* https://miau.my-x.hu/miau/319/itsec\_index\_for\_home\_workers.docx

Original strategic/operative objectives:

* Strategic level:
	+ Increasing professional sensitivity concerning
		- testing,
		- quality assurance,
		- critical interpretations,
		- risk management, …
	+ Supporting final theses through
		- case studies,
		- new aspects,
		- methodologies, …
	+ Involving ChatGPT/Copilot/etc. into
		- identification processes of potential problems (for test cases)
		- solution processes of the identified problems, …
* Operative level:
	+ Identifying concrete steps concerning the Neptun-system (especially the statistical module) in order to increase the own log-databases about system behaviour patterns
	+ Realisation of concrete steps in Neptun-system (concerning the statistical module)
	+ Derivation of new risks/diagnoses (and their potential solutions/therapies), …

**Table of Contents**

1. Introduction
2. Summary of Existing Case Study Findings
3. Identified Gaps in the Original Analysis
4. Proposed Testing Enhancements
	1. AI-Driven Test Scenario Generation
	2. Security Awareness & Non-Intrusive Vulnerability Assessment
	3. Simulated Performance Analysis
	4. Usability Testing (Heuristic Evaluation)
	5. Documentation Gap Analysis
5. Methodology & Tools
6. Theoretical Test Results & Visualizations
7. Limitations
8. Recommendations
9. Conclusion
10. References
11. **Introduction**

This addendum builds upon the original NEPTUN testing documentation (c.f. https://miau.my-x.hu/miau/320/moodle\_neptun\_tests/Neptun\_testing1.pdf) by proposing advanced testing strategies for the NEPTUN Course Statistics Module. Given my limited access as a student, the proposed methods emphasize non-intrusive techniques and simulated analyses. The enhancements focus on:

* + - AI-driven test case generation using tools like ChatGPT/Copilot (see chapter 4.1.)
		- A security awareness approach that reviews potential vulnerabilities without active scanning (4.2)
		- Simulated performance analysis using hypothetical scenarios and available documentation (4.3)
		- Heuristic-based usability evaluations (4.4)
		- Documentation gap analysis to reconcile observed behaviour with published specifications (4.5)

The goal is to ensure that the module accurately reflects course fulfilment while providing a clear, user-friendly interface—all within the constraints of student access.

1. **Summary of Existing Case Study Findings**

The original (c.f. https://miau.my-x.hu/miau/320/moodle\_neptun\_tests/Neptun\_testing1.pdf) analysis of the NEPTUN system identified several key issues:

* + - **Green Checkmark Anomalies:** (c.f. https://miau.my-x.hu/miau/320/moodle\_neptun\_tests/Neptun\_testing1.pdf#page=4)

Status indicators (green check marks) appear even before any signature is stored (evidence = own experience/observation).

* + - **Documentation Gaps:**

The “Course Statistics” module lacks detailed specifications, leading to ambiguous interpretations. (c.f. chapter “Basic information units” - https://miau.my-x.hu/miau/320/moodle\_neptun\_tests/Neptun\_testing1.pdf)

* + - **User Confusion:**

Teachers may misinterpret green check marks as “100% completion” even when tasks are incomplete. (c.f. https://miau.my-x.hu/miau/320/moodle\_neptun\_tests/Neptun\_testing1.pdf#page=6)

1. **Identified Gaps in the Original Analysis**

3.1 **Strategic level**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id** | **Gap**  | **Impact**  | **Proposed Solution**  | **Operative examples (inputs and expected outputs in annexes)** |
| 1 | Lack of performance testing  | Uncertain system behaviour under load  | Simulate load scenarios based on theoretical user counts  | ???annex#1??? |
| 2 | Insufficient security evaluation  | Unverified potential data vulnerabilities  | Conduct a review of documented vulnerabilities and safe practices (no active scanning)  | ???annex#2??? |
| 3 | Limited usability evaluation  | Unclear interface clarity and navigation  | Perform heuristic analysis based on available screenshots and user feedback  | ???annex#3??? |
| 4 | Incomplete documentation  | Ambiguity in expected behaviour  | Update documentation via gap analysis using available resources  | ???annex#4??? |

3.2 **Operative level(basics for the expected annexes#1-2-3-4):**

Strategic level id(1) = potential content in Annex#1:

* Accounts:
	+ There is no theoretical user-accounts
	+ There are user-accounts for the authors of this documentation
	+ A new account can only be granted for valid persons
* Test-data:
	+ The recommended simulation does need a very concrete test-file…
	+ This test-data must describe REAL transactions – quasi as multiply repeated transactions (e.g. the same grade and/or signature should be used repeatedly)
* Risks in our/own risk-detection processes:
	+ Testing may not cause system instability
	+ (Such an “attack” should be signalized for the system administrators in advance – let alone we would need a kind of permission for this action…)

Strategic level id(2): ??? (Annex#2 = concrete URLs with cited examples based on Google-Search-transaction including searched phrases and/or ChatGPT/Copilot-conversations where concrete reference-URLs are expected for concrete citations)???

Strategic level id(3): ??? (Annex#3 = concrete demonstrations about heuristic analyses based on available screenshots and/or user feedbacks)???

Strategic level id(4): ???(Annex#4 = concrete text-recommendations)???

1. **Proposed Testing Enhancements**

**4.1 AI-Driven Test Scenario Generation**

**Objective:**

Generate diverse test cases for verifying that green check marks appear only after all required transactions are complete.

**Methodology:**

* + - Use ChatGPT/Copilot to suggest edge cases and alternative scenarios based on documented behaviour.
		- Compare AI-generated scenarios with existing documentation (e.g., oweb\_hu.pdf) to ensure consistency.

**Example:** Annex#5 – we need all the entire conversations for each robot-expertise

* + - **Prompt:** “Generate test cases to verify that NEPTUN’s green check marks appear only when all signatures and grades are complete.”
		- **Expected Outcomes:**
			* No checkmark when 0/100 signatures are recorded.
			* A checkmark appears only when 100/100 signatures are present.
			* Removal of a signature reverses the checkmark status.

**4.2 Security Awareness & Non-Intrusive Vulnerability Assessment**

**Objective:**

Identify potential vulnerabilities through a review of existing security documentation and public vulnerability reports.

**Methodology:**

* + - Review NEPTUN documentation and online resources for any reported vulnerabilities or security practices.
		- Outline potential risks (e.g., data anonymization practices) without performing active scans.

**Example Vulnerability Considerations:** Annex#6 – we need all the entire searching transactions incl. search phrases, delivered information units, examples for useless information units and for useful information units

|  |  |  |
| --- | --- | --- |
| **Area**  | **Concern**  | **Recommendation (Theoretical)**  |
| Data Anonymization  | Potential exposure of sensitive information  | Recommend encryption best practices (e.g., AES-256)  |
| Input Validation  | Risk of unsanitized inputs  | Suggest review of input handling in documentation  |

* 1. **Simulated Performance Analysis**

**Objective:**

Assess system stability under high user load using a simulated approach.

**Methodology:**

* + - Develop theoretical load scenarios based on available documentation and expected student usage patterns.
		- Use published metrics from similar systems as benchmarks.

**Mock Metrics (Hypothetical🡨we always need existing/practical expectations, benchmarks and their derivations, evidence-layers, references, URLs, etc.):**

|  |  |  |  |
| --- | --- | --- | --- |
| **Metric**  | **Threshold**  | **Simulated Value**  | **Status**  |
| Response Time  | ≤2 seconds  | 2.5 seconds  | Slightly Above |
| Error Rate  | ≤1%  | 1.5%  | Needs Review  |

*Note: Actual performance testing is not conducted due to limited system access; these values serve as benchmarks for future controlled testing.*

* 1. **Usability Testing (Heuristic Evaluation)**

**Objective:**

Evaluate the user interface for clarity and intuitive design. (Examples…)

**Methodology:**

* + - Conduct a heuristic evaluation based on established usability principles (e.g., Nielsen’s heuristics). (Examples…)
		- Analyse available screenshots and user feedback to identify areas of improvement. (Examples…)

**Focus Areas:** (Undefined abstractions like clarity and/or consistency and/or understanding in the operative levels are in general forbidden! We do always need phenomena based on the KNUTH-principle = Science/knowledge is what can we transfer/transform/transcript/translate/… into source codes!

* + - Clarity of visual indicators (e.g., green check marks)
		- Consistency of layout and navigation
		- Overall ease of understanding system status

**4.5 Documentation Gap Analysis**

**Objective:**

Ensure that the system behaviour aligns with published documentation. (Examples…)

**Methodology:**

* + - Compare observed behaviour with documented expectations. (Examples…)
		- Identify and propose updates to documentation for any discrepancies (e.g., premature appearance of green check marks). (Examples…)
1. **Methodology & Tools**

|  |  |  |  |
| --- | --- | --- | --- |
| **Activity**  | **Tools/Methods**  | **Expected Outcome**  | **Examples** |
| AI-Driven Test Case Generation  | ChatGPT/Copilot  | A set of refined test cases covering diverse scenarios  | Annex#i |
| Security Review  | Literature review, online reports  | A summary report of potential security concerns  | Annex#j |
| Simulated Performance Analysis  | Theoretical modelling, comparative benchmarks  | Estimated performance metrics based on available data  | Annex#l |
| Usability Evaluation  | Heuristic Analysis, online research  | Identification of interface issues and improvement areas  | Annex#m |
| Documentation Analysis  | Manual review  | An updated gap analysis report  | Annex#n |

1. **Theoretical Test Results & Visualizations**

**Figure A:**

*Green Checkmark Logic Flowchart*

*(Insert a flowchart image outlining the theoretical logic for checkmark visibility based on task completion.)*

We do really need the flowchart itself!

**Figure B:**

*Simulated Performance Test Summary*

|  |  |  |
| --- | --- | --- |
| **Users (Simulated)**  | **Avg. Response Time**  | **Error Rate** |
| 100  | 1.8 seconds  | 0.8%  |
| 500  | 2.5 seconds  | 1.5%  |

*(These results are based on simulated scenarios and serve as a baseline for future testing in a controlled environment.)*

We do really need real testing results!

1. **Limitations**
	* + **Restricted System Access:**

Testing is based solely on student/teacher-level access, available screenshots, and published documentation.

* + - **Theoretical Approaches:**

Several analyses (e.g., performance and security testing) are simulated or based on literature reviews due to the inability to conduct intrusive tests on the live system. (We do need testing concepts being realisable…)

1. **Recommendations**
	* 1. **Enhance AI-Powered Test Case Generation:**

Utilize AI tools to continuously refine and expand test scenarios as more system behaviour becomes observable. (Examples in Annexes)

* + 1. **Adopt a Non-Intrusive Security Review:**

Regularly review published vulnerability reports and documentation to stay informed of potential risks. (Examples in Annexes)

* + 1. **Plan for Future Controlled Testing:**

If possible, request access to a sandbox environment for live performance and security testing. (Examples in Annexes)

* + 1. **Improve Documentation:**

Update the system’s documentation to clearly define expected behaviours and edge cases. (Examples in Annexes)

* + 1. **Implement a Formal Usability Evaluation:**

Use heuristic analysis combined with user feedback to enhance interface clarity. (Examples in Annexes)

1. **Conclusions / Expectations**

This addendum extends the original NEPTUN testing analysis by proposing a set of advanced yet nonintrusive testing strategies that are feasible within the constraints of student-level access. The black-and-white texts can be seen as results of a useful multi-layered thinking experiment (plan, draft, concept). This empty-bubble-concept-level will however/never be accepted in final theses, where we do need the real/realistic examples overall.

Through AI-driven test case generation, simulated performance analysis, heuristic usability evaluation, and documentation gap analysis, the proposed methods aim to provide a strong foundation for continuous system improvement. All these parallel layers do need real examples (partially: automated, partially: manually realized).

While some tests remain theoretical due to access restrictions (theoretical aspects must be handled in the chapter “introduction: about the structure-parameter of the final thesis” and/or in the chapter “Future”), these recommendations serve as a practical roadmap (draft, concept, etc. = empty bubbles being never proved until matured examples are not delivered) for future, more in-depth testing efforts should access to a sandbox environment become available.

1. **References**
	* 1. NEPTUN System Documentation: [oweb\_hu.pdf](https://neptun.kodolanyi.hu/oktato/help/oweb_hu.pdf)
		2. OWASP ZAP (Review-based): [https://www.zaproxy.org](https://www.zaproxy.org/)
		3. Apache JMeter (Simulated Analysis): [https://jmeter.apache.org](https://jmeter.apache.org/)
		4. ChatGPT/Copilot internal test case generation

**Submission Message:**

Dear Professor Pitlik,

Please find attached my case study addendum for the software testing assignment, which focuses on enhancing the NEPTUN Course Statistics Module. This document builds on the original NEPTUN testing analysis and proposes advanced yet non-intrusive testing strategies—including AI-driven test case generation, simulated performance analysis, heuristic usability evaluation, and documentation gap analysis—tailored to my current access level. I look forward to your feedback and any suggestions for further improvement.

Best regards,

Munkhjargal Ariunbold

Dear Students!

You see an excellent initiative! Thanks for the author! Please try to complete this documentation with the needed annexes! Each Student should deliver at least one matured annex-version OR the own software behind the own final thesis should be monitored, checked, tested with the same detail-oriented engagement as here and now!