Development of an API Testing Tool

Trần Quang Vinh
Thai Nguyen University of Information and Communication Technology
Thai Nguyen University, Thai Nguyen, Vietnam
student605@ictu.edu.vn

***Abstract*—This undergraduate thesis investigates a software tool for testing application programming interfaces. Developers need a convenient way to exercise API endpoints and inspect responses during application development. The student analyses testing workflows and implements a tool for API request handling within the scope of the graduation project. The manuscript documents its requirements, interface, and implementation. The conclusion reports a completed software prototype meeting the agreed project scope. It also identifies missing functions, an unfinished interface, browser display issues, and performance and security work for later development. The source does not establish comparative defect-detection performance, coverage across API protocols, or a production security assessment. The contribution is a documented student study or prototype whose stated scope can guide further testing.**

***Index Terms*—API testing, software testing, web tool, prototype**

I. Introduction

Developers need a convenient way to exercise API endpoints and inspect responses during application development. This thesis investigates a software tool for testing application programming interfaces. It begins from a practical engineering need and explains the technologies and workflow chosen for the case. The objective is to document a student implementation or comparative investigation, together with the evidence available in the manuscript. Broader operational or economic effects require separate measurements and are not assumed from the existence of a prototype [1]. Earlier ICTU student work on related designs is included here for context [2], [3], [4]. Related published studies provide context for the subject and methods [5], [6], [7], [8], [9], [10]. These contextual works were selected for this English summary; they are not asserted to appear in the original thesis bibliography.

II. Methods

The student analyses testing workflows and implements a tool for API request handling within the scope of the graduation project. The manuscript documents its requirements, interface, and implementation. The source is summarized through its stated design, implementation material, examples, and conclusion. Features shown in the manuscript are distinguished from proposals for later development. This review does not add a new experiment, fill missing measurements with estimates, or treat a planned capability as an observed result. The source thesis remains the provenance for the reported work [1].

Supervisor: Bùi Anh Tú, Thai Nguyen University of Information and Communication Technology (batu@ictu.edu.vn).

III. Results

The conclusion reports a completed software prototype meeting the agreed project scope. It also identifies missing functions, an unfinished interface, browser display issues, and performance and security work for later development. The report's screens, configurations, examples, or measurements constitute the evidence for this summary. A capability described only as a design intention is not counted as a validated operational outcome. No additional performance, adoption, financial, or safety figure has been introduced here. The result is interpreted within the scope of the student's documented setup [1].

IV. Discussion

The source does not establish comparative defect-detection performance, coverage across API protocols, or a production security assessment. Further work could broaden test cases, refine the interface, test cross-browser display, and evaluate response handling and security systematically. These limitations affect how the evidence should be read. The current manuscript can support understanding of the design and reported demonstration, while independent tests would be needed for stronger claims. Any later deployment should document its environment and separate observed outcomes from intended benefits [1].

V. Conclusion

The thesis documents a software tool for testing application programming interfaces. The conclusion reports a completed software prototype meeting the agreed project scope. It also identifies missing functions, an unfinished interface, browser display issues, and performance and security work for later development. The source does not establish comparative defect-detection performance, coverage across API protocols, or a production security assessment. Further work could broaden test cases, refine the interface, test cross-browser display, and evaluate response handling and security systematically. Accordingly, the present contribution is a source-grounded account of the student work and its tested or illustrated scope. Claims about broader effectiveness should be assessed in later work using explicit evaluation conditions [1].

References

[1] Trần Quang Vinh, "Development of an API Testing Tool," internal undergraduate thesis (original in Vietnamese), Thai Nguyen University of Information and Communication Technology, 2024.

[2] Vũ Thị Huệ, ”Research and Application of JMeter Tool for Performance Testing of Viblo.asia Website,” internal undergraduate thesis, Thai Nguyen University of Information and Communication Technology, 2021.
[3] Nguyễn Thị Ly Ly, ”Research on Software Testing Using Automated Testing Tool Selenium WebDriver for the Luliz E-commerce Website,” internal undergraduate thesis, Thai Nguyen University of Information and Communication Technology, 2023.
[4] Nguyễn Thị Hồng Vân, ”Research on White Box Testing and Application with Unit Testing Tools in C#,” internal undergraduate thesis, Thai Nguyen University of Information and Communication Technology, 2021.
[5] Olga Melkozerova et al., ”Circuit description of API interface for software testing,” Computer Science and Cybersecurity, 2019. doi:10.26565/2519-2310-2019-4-05.
[6] Pratik Prakash Kore et al., ”API Testing Using Postman Tool,” International Journal for Research in Applied Science and Engineering Technology, 2022. doi:10.22214/ijraset.2022.48030.
[7] Marc Roper, ”Software testing: A selected annotated bibliography,” Software Testing, Verification and Reliability, 1992. doi:10.1002/stvr.4370020303.
[8] S. Venkatraj et al., ”Development of Test Automation Framework for REST API Testing,” Journal of Computational and Theoretical Nanoscience, 2019. doi:10.1166/jctn.2019.7749.
[9] Maheswara Reddy Basireddy, ”Performing Security API Testing using Postman AI,” Journal of Artificial Intelligence & Cloud Computing, 2022. doi:10.47363/jaicc/2022(1)304.
[10] A.N. Alpatov, ”SIMULATION SERVER REST API BASED ON THE SPECIFICATION FOR TESTING WEB APPLICATIONS,” Современные наукоемкие технологии (Modern High Technologies), 2020. doi:10.17513/snt.38371.