electronics

MDPI

Article

Applying Large Language Model to User Experience Testing

Nien-Lin Hsueh *, Hsuen-Jen Lin and Lien-Chi Lai

Department of Information Engineering and Computer Science, Feng Chia University, Taichung 407, Taiwan; kobayasa209924@gmail.com (H.-J.L.); p1000433@mail.fcu.edu.tw (L.-C.L.)
* Correspondence: nlhsueh@fcu.edu.tw

Abstract: The maturation of internet usage environments has elevated User Experience (UX) to a critical factor in system success. However, traditional manual UX testing methods are hampered by subjectivity and lack of standardization, resulting in time-consuming and costly processes. This study explores the potential of Large Language Models (LLMs) to address these challenges by developing an automated UX testing tool. Our innovative approach integrates the Rapi web recording tool to capture user interaction data with the analytical capabilities of LLMs, utilizing Nielsen's usability heuristics as evaluation criteria. This methodology aims to significantly reduce the initial costs associated with UX testing while maintaining assessment quality. To validate the tool's efficacy, we conducted a case study featuring a tennis-themed course reservation system. The system incorporated multiple scenarios per page, allowing users to perform tasks based on predefined goals. We employed our automated UX testing tool to evaluate screenshots and interaction logs from user sessions. Concurrently, we invited participants to test the system and complete UX questionnaires based on their experiences. Comparative analysis revealed that varying prompts in the automated UX testing tool yielded different outcomes, particularly in detecting interface elements. Notably, our tool demonstrated superior capability in identifying issues aligned with Nielsen's usability principles compared to participant evaluations. This research contributes to the field of UX evaluation by leveraging advanced language models and established usability heuristics. Our findings suggest that LLM-based automated UX testing tools can offer more consistent and comprehensive assessments.

Keywords: human–computer interaction (HCI); AI in software development

check for updates

Citation: Hsueh, N.-L.; Lin, H.-J.; Lai, L.-C. Applying Large Language Model to User Experience Testing. *Electronics* **2024**, *13*, 4633. https://doi.org/10.3390/electronics13234633

Academic Editor: George Angelos Papadopoulos

Received: 24 October 2024
Revised: 14 November 2024
Accepted: 22 November 2024
Published: 24 November 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Software testing plays a critical role in the software development process, with the primary purpose of identifying defects and errors to ensure product quality and reliability. Throughout the software development lifecycle, from unit testing to user acceptance testing, each phase examines different aspects of the software's functionality, performance, security, and user experience. Through systematic testing processes, development teams can identify and fix problems early, thereby improving product stability and user satisfaction. In today's digital era, enhancing the User Experience (UX) of web pages is not just a pursuit but a necessity. The success of a web page largely depends on the quality of the user experience. However, due to the subjective and dynamic nature of user experience, which is influenced by environmental factors, traditional user experience testing and enhancement rely on manual inspection, which is not only time consuming but also costly [1].

As Nielsen and Molich observed, if a system fails to meet usability needs, it does not matter if it is easy to use [2]. Alomari et al. argued that if a system is too difficult to use, even if it meets usability needs, it does not matter because users simply cannot use the system [3]. Meanwhile, Large Language Models (LLMs) have rapidly emerged in the field of artificial intelligence and have been widely used in image analysis, language understanding, text generation, and other areas. The strength of these models lies in processing and analyzing large amounts of language data, offering potential solutions to this challenge and providing new opportunities to gain deeper insights into user behaviors and preferences during web

interactions. With the advancement of these technologies, UX researchers and designers now have the opportunity to use LLMs to capture and analyze subtle changes and complex scenarios that affect user experience.

Recent research has further explored large multimodal models beyond LLMs. Alayrac et al. introduced Flamingo, a visual language model that combines two pre-trained models: a vision model for image processing and a language model for basic reasoning [4]. After comparing the performance of 16 models, the experimental results showed that Flamingo exhibited excellent adaptability with only a few examples. As the model's training data increased, Flamingo's performance significantly improved, capable of handling up to 32 images or videos, indicating high flexibility in processing varying amounts of images or videos. Sun et al. proposed the multimodal generative pre-training model Emu, which can handle any single modality or multimodal data input without distinction and generate images and text [5]. Including image-to-image descriptions, interleaved image-to-image descriptions, video-to-video descriptions, etc., through autoregressive training, embedding images into text, and forming interleaved input sequences for end-to-end training. Emu demonstrated potential in multimodal tasks such as image description, visual question answering, video question answering, and text-to-image generation, with results showing Emu's outstanding performance in a wide range of zero-shot and few-shot tasks.

Most existing automated UX testing tools primarily focus on functional and performance testing, often neglecting users' cognitive experiences, interaction details, and emotional needs. By leveraging GPT-4's advanced semantic understanding capabilities, our approach uniquely addresses these limitations by providing a comprehensive analysis of user cognitive patterns, interaction nuances, and emotional responses. This integration of AI-powered analysis with established usability heuristics represents a significant advancement in automated UX testing, offering a more holistic evaluation framework that goes beyond traditional functional metrics. This study aims to develop a user experience detection tool based on Generative Pre-trained Transformer 4 (GPT-4) and investigate the impact of prompt engineering on its performance. The key objectives are to create an automated tool that leverages GPT-4's image understanding capabilities and Nielsen's usability heuristics to identify potential UX issues, explore how varying the content and structure of prompts influences GPT-4's ability to detect user experience issues, and assess the performance of GPT-4 in user experience detection using the developed prompts. Through these objectives, we aim to advance the application of large language models in user experience evaluation and provide insights into optimal prompt design for this task.

Summarizing the above, the research questions are as follows:

- RQ1: How does providing different information within prompts influence the accuracy of user experience detection by GPT-4?
- RQ2: What is the performance of GPT-4 in user experience detection compared to human evaluators when using optimized prompts?

This paper follows a systematic structure: Section 2 reviews the literature on user experience evaluation and AI applications; Section 3 details the methodology of the proposed automated UX assessment tool; Section 4 outlines the experimental design and data analysis methods; Section 5 discusses research findings, comparing GPT-4's performance with human evaluators; and Section 6 concludes with key insights and future research directions in automated UX evaluation.

2. Literature Review

2.1. Challenges and Complexities in User Experience Evaluation

User Experience (UX) evaluation faces significant challenges due to its multifaceted and complex nature. Vermeeren et al. [6] highlighted a critical gap in the field: the lack of systematic research on UX evaluation and measurement methodologies. Current guidelines often rely heavily on basic usability targets, which often fail to capture the full breadth of UX. The inherent characteristics of UX, as emphasized by Kashfi et al. [7], further complicate the evaluation process. These characteristics, namely subjectivity, holism, dynamism, context

dependency, and worth, have profound implications for practitioners' daily work, making it challenging to standardize assessment methodologies.

The challenge of reaching a consensual definition of UX, as revealed by Law et al.'s survey [1], directly impacts evaluation efforts. The broad range of fuzzy and dynamic concepts associated with UX, including emotional, affective, experiential, hedonic, and aesthetic variables, contributes to this definitional ambiguity, as well as the difference in emphasis between academic and industrial approaches to UX evaluation. While academia tends to focus more on hedonic aspects and emotions, industry emphasizes functionality and usability, adding another layer of complexity to evaluation practices.

Furthermore, the integration of UX evaluation methods into existing software development processes poses substantial challenges. Both Vermeeren et al. [6] and Kashfi et al. [7] noted the pressing need for evaluation methods applicable to early development phases, validated measures for UX constructs, and techniques for assessing social and collaborative experiences. The lack of practical, multi-method approaches that can effectively capture the nuanced nature of UX throughout the development lifecycle remains a considerable obstacle in the field.

2.2. User Experience Evaluation Methods and Metrics

2.2.1. Primary Evaluation Methods

In the field of User Experience (UX) research, the selection of appropriate evaluation methods and metrics is crucial for obtaining valid and reliable results. Inan et al. [8] categorize UX evaluation methods into three main types: self-report measures, observational measures, and physiological measures.

Self-report measures are the most prevalent method, involving users reporting their own experiences through questionnaires or interviews. While this approach is easily administered and requires no specialized equipment, it is susceptible to self-report bias, potentially compromising data accuracy.

Observational measures involve experts observing users as they interact with a product or system. It offers greater objectivity compared to self-report measures but can be more time-consuming and costly to conduct. Moreover, the presence of an observer may influence users' behavior, affecting the validity of the observations.

Physiological measures involve measuring users' physiological responses, such as brain activity, heart rate, or galvanic skin response. While this method is the most objective, it is also the most expensive and time-consuming to implement.

2.2.2. Types of Evaluation Metrics

In their seminal work on measuring user experience, Albert and Tullis [9] delineate four distinct categories of metrics that are essential for evaluating UX. These categories encompass a comprehensive range of measurement approaches, each offering unique insights into different aspects of user interaction and perception. Performance metrics form the foundation of objective measurement, quantifying user behavior through indicators such as task success rates, error frequencies, and time-on-task measurements. Complementing these are issue-based metrics, which focus on identifying and cataloging specific problems or challenges users encounter during their interaction with a product or system. To capture the subjective dimension of user experience, self-reported metrics play a crucial role, allowing users to articulate their opinions, satisfaction levels, and overall impressions of the product or service in question. Rounding out this framework are behavioral and physiological metrics, which delve into the physical manifestations of user interaction, measuring observable behaviors and bodily responses that may indicate cognitive load, emotional states, or levels of engagement. This multifaceted approach to UX metrics enables researchers and practitioners to construct a holistic view of the user experience, balancing objective performance data with subjective user perceptions and unconscious physiological responses.

2.2.3. Integration of Methods and Metrics

Combining these evaluation methods with appropriate metrics can provide a more comprehensive framework for UX research. Guimaraes et al. [10] utilized a mixed-method approach combining heuristic evaluation and a structured questionnaire to assess the Brazilian Immunization Information System, revealing that while the system provides easy access, it exhibits minor usability issues and struggles to fully facilitate users' interface interaction goals. Mochammad et al. [11] employed a mixed-method approach using a User Experience Questionnaire (UEQ) and Usability Testing to evaluate the Halodoc mobile health application, revealing positive UX scores across all scales but identifying specific usability issues related to pharmacy selection and medicine purchase information.

By integrating these diverse evaluation methods and metrics, researchers can select the most appropriate strategies based on their specific research questions and contexts, thereby enhancing the depth and reliability of UX studies. This integrative approach allows for a more nuanced understanding of the user experience, capturing both objective performance indicators and subjective user perceptions.

2.3. Automated User Experience Evaluation

Among the evaluation methods mentioned in Section 2.2.1, both self-report measures and observational measures require substantial human resources and time to yield results. While current automated tools have shown promise in technical evaluation aspects, they often struggle to capture the cognitive and emotional dimensions of user experience, particularly in assessing how users perceive and interact with interfaces on a psychological level. Nguyen et al. [12] also highlighted a gap in the software development lifecycle that needs bridging: the disparity between user experience designers and software, security, and IT/operations engineers. Consequently, the necessity for automated evaluation tools or methods becomes apparent. Namoun et al. [13] conducted a comprehensive study utilizing a 19-dimension usability framework to assess 10 popular web usability testing tools across 9 websites in 3 categories. Their findings revealed that while these tools effectively evaluate various technical aspects, they often neglect usability issues, produce inconsistent results, and lack integration with established usability theories. In an effort to address these shortcomings, Biringa et al. [14] developed an innovative automated user experience testing methodology. Their approach employs abstract syntax tree-based embeddings and machine learning algorithms to evaluate the impact of software updates on performance, achieving a commendable 3.7% mean absolute error rate with a random forest regressor in estimating time impacts. However, this performance-oriented approach may not fully capture the multifaceted nature of user experience, particularly in terms of usability principle integration. Further advancing the field, Whiting [15] proposed two models that combine load testing with automated UI testing. These models aim to evaluate system performance from multiple user perspectives, thereby providing a more comprehensive assessment of user experience and facilitating the creation of a derived test oracle for load testing.

These recent advancements in automated user experience evaluation methods demonstrate the field's progression towards more efficient, comprehensive, and user-centric assessment tools. However, a significant gap remains in incorporating established UX design principles into automated testing frameworks. Current automation tools primarily focus on technical performance and functionality testing, with little emphasis on evaluating user interface elements through the lens of human cognition and interaction patterns. For instance, well-known UX design guidelines, such as Jakob Nielsen's 10 heuristics for usability evaluation [2] or Donald Norman's design principles [16], have yet to be effectively integrated into existing automated testing frameworks. The emergence of advanced language models with sophisticated image understanding capabilities presents a promising opportunity to bridge this gap by combining automated efficiency with a nuanced understanding of user experience principles.

3. Research Methodology

In this study, we aim to evaluate the efficacy of GPT-4 in automated User Experience (UX) detection. To achieve this objective, we have devised a methodology that incorporates an automated web recording tool to capture user interactions, generating both operational logs and screenshots. These data are then analyzed by GPT-4 using carefully designed prompts based on heuristic principles of UX evaluation. To ground our research in a real-world context, we have selected a tennis court reservation system as our case study. This approach allows us to assess GPT-4's capabilities in identifying UX issues within a practical, applied setting, thereby providing insights into its potential as an automated UX detection tool.

3.1. User Experience Detection Tool Development

3.1.1. Design Principles and Objectives

We aim to study automated user experience detection by adopting Nielsen's usability heuristic principles. Following key principles such as visibility of system status, matching between the system and the real world, and utilizing consistency and standards can significantly enhance the overall user experience, particularly regarding the system's visibility to users—an aspect often lacking in most automated user experience tools.

By utilizing automated detection tools to record user interactions and capture screenshots, the accuracy and efficiency of user experience analysis can be significantly improved. These tools can precisely document every step of the user's interaction, whether it involves clicks, swipes, or keyboard inputs, ensuring the integrity of the process for subsequent behavioral analysis. Furthermore, through screenshot functionality, the tools can visually represent the specific interfaces and responses encountered during the user's journey, particularly when users face challenges or errors. This helps in comprehending contextual experiences more thoroughly.

The data collected by the tool reflects users' reactions to the captured screenshots. Manual analysis of such data would require substantial time and human resources. However, with automation, the system can efficiently process large volumes of screenshots and interaction records while adhering to predefined evaluation standards, such as Nielsen's usability heuristics. This approach not only accelerates the analysis process but also ensures consistency and objectivity in the results, mitigating potential biases that could arise from manual evaluation.

3.1.2. Tool Architecture and Components

The data processing flow is illustrated in Figure 1. The methodology commences with the utilization of the Rapi Recording tool (v4.0.0) to capture user interactions, which are subsequently compiled into an Operation Script File. This file undergoes a bifurcation process: the textual components are extracted to form a text script, while the visual elements are subjected to a screenshot-link conversion mechanism. The user experience evaluation phase constitutes the core of this methodology, integrating multiple inputs. It amalgamates the converted screenshot links and the text script while leveraging Nielsen's heuristics as the foundational framework for assessment. This multifaceted approach facilitates a comprehensive evaluation of the user experience. The outcomes of this evaluation process are twofold. Primarily, it generates a score explanation, providing qualitative insights into various aspects of the user experience. Secondarily, it produces GPT-4 evaluation scores for websites based on different indicators, offering a quantitative measure of diverse user experience dimensions. This systematic approach enables a rigorous and comprehensive assessment of user experience, combining both qualitative and quantitative methodologies.

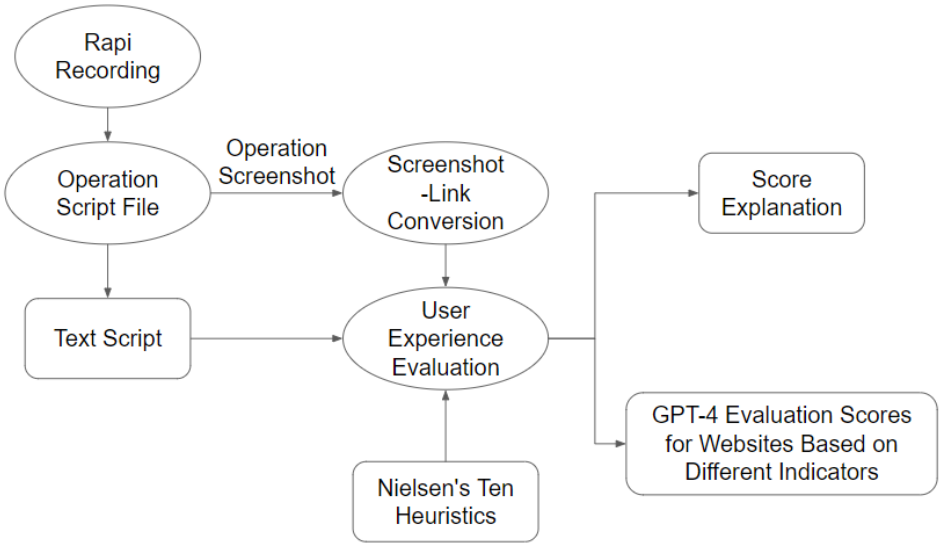


Figure 1. Framework for website user experience evaluation based on Nielsen's heuristic principles and GPT-4 scoring system.

3.1.3. Evaluation Criteria

In this study, we adopted six of Nielsen's usability heuristics as evaluation criteria for our tool, specifically: *visibility of system status*; *user control and freedom*; *help users recognize, diagnose, and recover from errors*; *consistency and standards*; *recognition rather than recall*; and *match between system and the real world*. These principles were selected due to their direct relevance to user perception and their fundamental nature in user experience design. They encompass the most prevalent aspects of user interaction, including system response status, degree of operational freedom, error handling complexity, and intuitive screen perception. These principles provide users with a clear and reliable operational environment, directly influencing their perception of the system's User Experience (UX). The selection of these specific heuristics allows for a focused evaluation of core UX aspects within the constraints of our research scope.

These principles have been sequentially numbered as NS01 through NS06 for systematic reference. Their respective explanations and illustrative examples have been incorporated into the GPT-4 prompt to ensure comprehensive understanding and application (see Appendix A for detailed elaboration). This approach facilitates a structured evaluation process and enhances the consistency of the assessment criteria across various user experience scenarios.

3.1.4. Prompt Design for GPT-4

Due to the vast amount of knowledge acquired by Large Language Models (LLMs) during pretraining on textual data, they have demonstrated the ability to solve a wide range of NLP tasks. Moreover, through effective prompt engineering, LLMs can achieve significantly improved results. This technique enables the models to perform new tasks even in the absence of large-scale, specialized training data, by carefully designing prompts that guide the model in task execution.

In user experience evaluation, different types of data capture distinct aspects of user–system interaction. Screenshots effectively document the static visual elements of the interface, while operational data records dynamic interactions such as user behaviors and system responses. This comprehensive approach is crucial as single-modal data often fails to capture the complete picture of user experience issues.

For instance, when evaluating error handling capabilities in our tennis court reservation system, screenshots alone might only reveal the final error message display. However, by incorporating operational data, we can analyze user attempts, system response times, and interaction patterns leading to the error state. Similarly, while assessing system status visibility, operational data can verify if loading times exceed acceptable thresholds, while screenshots confirm the presence of visual loading indicators.

Given GPT-4's impressive image understanding capabilities, we designed our prompt structure to effectively utilize both visual and operational data. Our approach integrates images with prompts, either containing or omitting user interaction scripts, to investigate how different data contexts affect automatic user experience detection. For example, when evaluating interface consistency, screenshots are crucial for comparing visual elements, while operational logs help verify if users encounter navigation difficulties despite visual consistency.

The structure of our prompt consists of the following components: task description, user operation data, data format specifications, a description and example of Nielsen's usability principles, and an explanation of the evaluation method. This structured approach ensures systematic evaluation of both static and dynamic aspects of user experience. A detailed example of this prompt structure can be found in Appendix B.

Through this dual-data strategy, we guide GPT-4 to consider not only what users see (interface elements) but also how they interact with these elements (operational patterns). This comprehensive evaluation approach aligns with Nielsen's holistic view of usability, representing a significant advancement in automated UX evaluation methodology.

3.2. Case Study: Tennis Course Reservation System

3.2.1. System Requirements and Functionality

The system's functional requirements encompass the following key features: course listing, course search, course reservation, and reservation cancellation. Each feature necessitates a series of user actions and system determinations to achieve the desired outcome.

Figure 2 provides an illustrative example of the user flow for reserving a training course within the context of a tennis club platform. The user journey commences at the homepage, from which the user proceeds to the course overview page. Upon selecting a specific course, the system verifies the user's login status. If the user is not logged in, the system prompts the user to do so. Following a successful login, the user is directed to the reservation confirmation page, where they have the option to either confirm or cancel the reservation. If the user confirms the reservation, the system completes the reservation process and records the reservation details. Conversely, if the user opts to cancel the reservation, the system redirects the user to the reservation record page. In the event of a failed login attempt, the system returns the user to the login page to reattempt the authentication process.

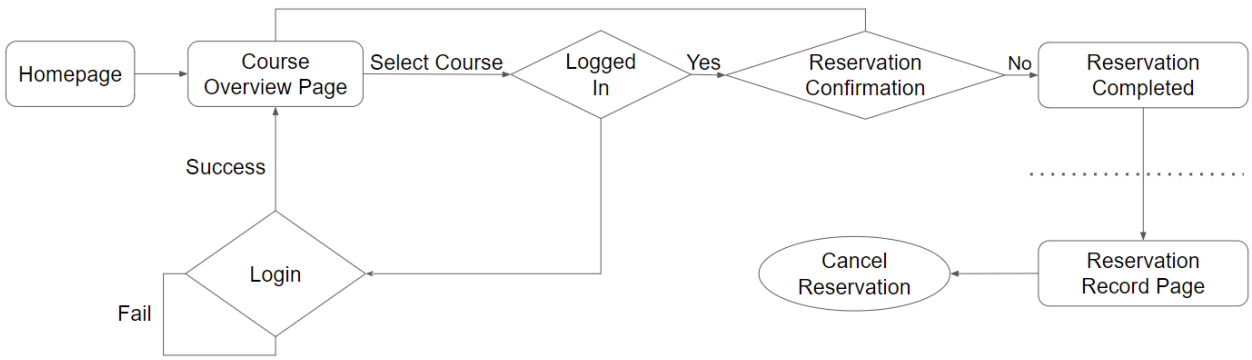


Figure 2. Flowchart of the course reservation process with login and confirmation steps.

3.2.2. Story and Task Design

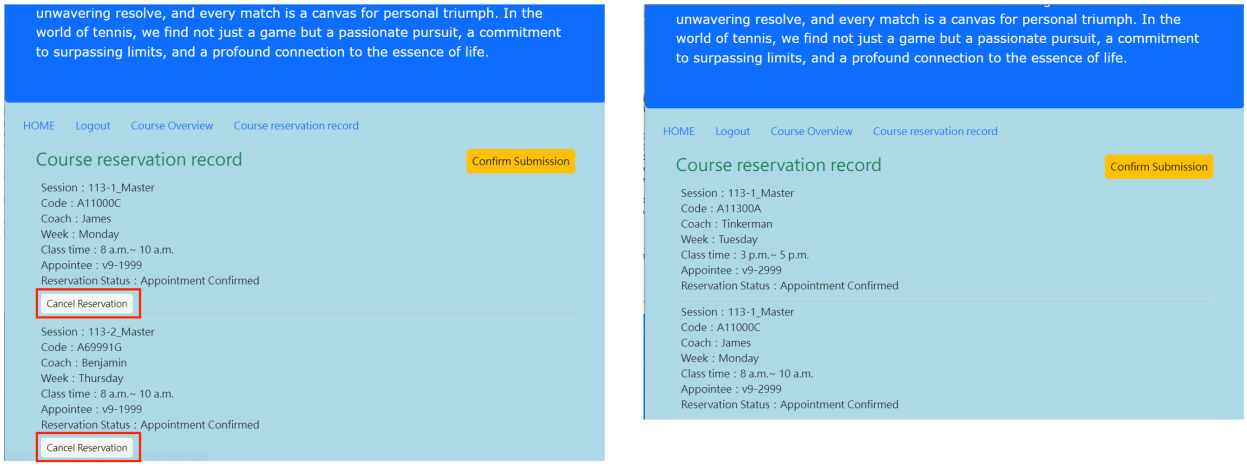
In our study, we developed a comprehensive evaluation framework based on six user perception-related principles from Nielsen's usability heuristics. This framework comprises nine distinct stories (S01–S09), each designed to assess specific aspects of user interface and system functionality in the context of a tennis court reservation platform. To facilitate a nuanced analysis, each story is bifurcated into an "Expected Scenario" and an "Unexpected Scenario", representing optimal and detrimental user experiences, respectively. Table 1 presents a detailed overview of these stories, their descriptions, and their corresponding Nielsen usability principles.

The first story, S01, focuses on error handling during the login process, corresponding to the NS01 (Clear system status) and NS03 (Clear error handling) heuristics. The expected behavior involves displaying an appropriate failure message when login attempts are unsuccessful, while the unexpected scenario would be the absence of such feedback.

Story S02 evaluates the course display format, aligning with NS06 (consistent with real-world system status). The system is expected to present courses in a weekly table format, reflecting common scheduling practices. An unexpected outcome would be the use of a simple list format, which may not provide sufficient temporal context.

Stories S03 and S04 address the design consistency of navigation elements, specifically the back button, in accordance with NS04 (consistent style). The expected implementation involves a less prominent back button positioned to the left of the reservation button, maintaining a consistent visual hierarchy and layout.

User control (NS02) is assessed in stories S05 and S09, which examine the presence of essential navigation and booking management features. The expected outcomes include the provision of a back button and both course record and cancellation functions, enhancing user autonomy within the system. As illustrated in Figure 3.



(a) The interface is designed with a "Cancel Reservation" button, empowering users to proactively terminate confirmed course bookings.

(b) The interface offers no cancellation feature for reservations.

Figure 3. Comparison of user interfaces with and without course reservation cancellation options.

Stories S06 and S08 evaluate the system's ease of recognition (NS05) by assessing the availability of course search and recording functions. These features are expected to be readily accessible, supporting efficient user interaction with the platform.

Lastly, story S07 focuses on system feedback during loading processes, aligning with NS01 (Clear system status). The expected behavior involves displaying a system prompt during loading sequences, keeping users informed of ongoing processes.

Table 1. User interface stories mapped to Nielsen's usability principles with expected and unexpected scenarios.

Story	Description	NS Indicator	Expected Scenario	Unexpected Scenario
S01	Error message displayed during login failure.	NS01 Clear system status NS03 Clear error handling	A failure message is displayed when login fails.	No failure message is displayed when login fails.
S02	Courses displayed by week or in list format.	NS06 Consistent with real-world system status	Courses are displayed in weekly order using a table format.	Courses are displayed in list format.
S03	The color of the back button.	NS04 Consistent style	The back button is less prominent than the reservation button.	The back button is more prominent than the reservation button.
S04	The position of the back button.	NS04 Consistent style	The back button is positioned to the left of the reservation button.	The back button is positioned to the right of the reservation button.
S05	The presence of a back button.	NS02 User control	A back button is provided.	No back button is provided.
S06	The presence of a course search function.	NS05 Easy to recognize	A course search function is provided.	No course search function is provided.
S07	System prompt shown during loading.	NS01 Clear system status	A system prompt is displayed while loading.	No system prompt is displayed while loading.
S08	The presence of a course history recording function.	NS05 Easy to recognize	A course history record function is provided.	No course history record function is provided.
S09	The presence of a cancel booking function.	NS02 User control	Course history record and cancellation functions are provided.	The course history record function is provided, but no cancellation function is available.

4. Experimental Results and Analysis

4.1. Experimental Setup

We conducted an experiment to compare the discrepancies between human evaluators and automated user experience evaluation tools. The study involved 16 participants who had undergone three months of training in web development and software engineering.

Participants were randomly assigned a story, each comprising both an expected and an unexpected scenario operation. The time limit for this session was set at one hour. If a participant completed their initially assigned story within this timeframe, they were given an additional story to work on. Consequently, participants had the opportunity to complete multiple story operations.

Following each story operation, participants were required to complete a questionnaire designed based on Nielsen's usability principles. This questionnaire incorporated items related to principles NS01 through NS06, with responses rated on a 5-point Likert scale. The specific questions included in the questionnaire can be found in Appendix C. The questionnaire did not differentiate between the story tested by the participants. The user workflow is illustrated in Figure 4.

During the testing process, participants were instructed to use the Rapi automatic recording tool to document their operations and capture screenshots. The data were subsequently used as input for the automated UX evaluation tool.

To ensure consistency and facilitate comparison, both human evaluators and the automated tool assessed the UX based on the same set of Nielsen principles. This approach allowed for a direct comparison between human-generated and machine-generated UX evaluations, providing insights into the relative strengths and limitations of each method.

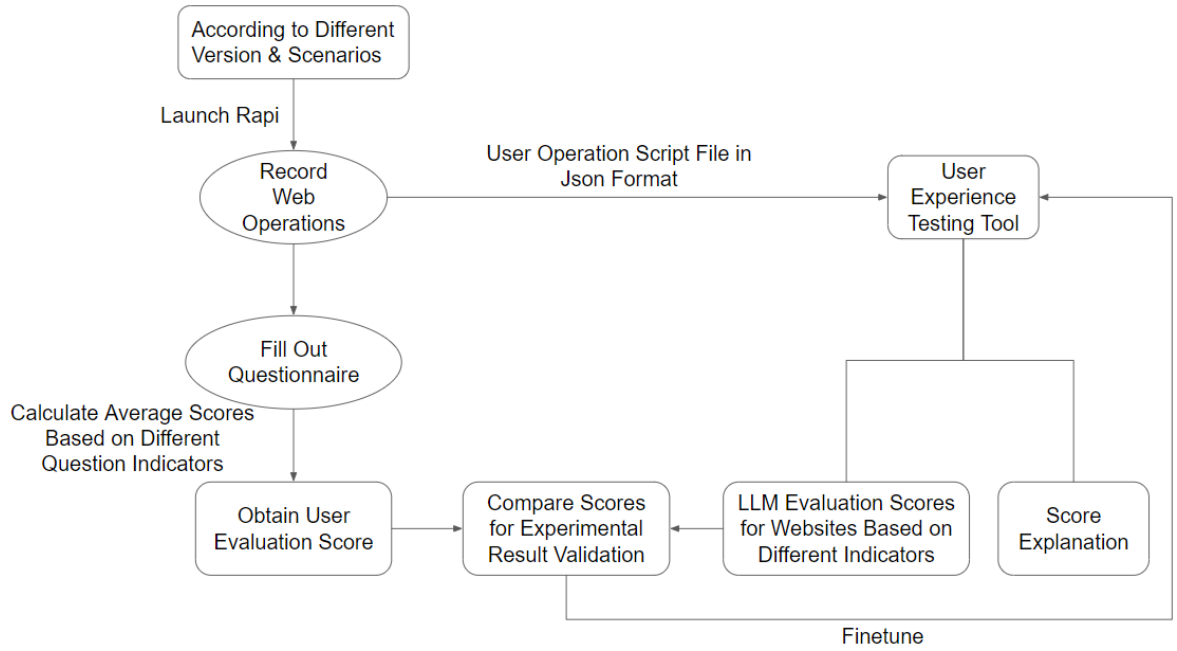


Figure 4. Workflow for evaluating user experience using web operations and LLM-based scoring.

4.2. Data Categorization and Analytical Approach

To carefully assess how well each story evaluates its assigned usability principle, we developed a focused data grouping strategy based on two distinct types of scenarios and scores. This strategy addresses the complex nature of user experience evaluation, where interface changes may have both direct and indirect effects on usability principles. Our evaluation framework comprises expected and unexpected scenarios, each generating *Target Indicator Scores (TISs)* and *Auxiliary Indicator Scores (AISs)*.

Expected scenarios represent interface implementations that strictly follow UX design principles. For instance, Story S01 displays appropriate error messages during login failures, while stories S02–S09 maintain this functionality as a standard feature. In contrast, Unexpected scenarios intentionally deviate from these standards in specific areas. In S01's case, error messages are deliberately omitted, while other stories modify their respective target elements while maintaining basic error handling.

Within these different scenarios, we further categorized our evaluation metrics into two types of scores. *TIS* represents scores directly related to Nielsen's Usability (NS) principles being tested in each story. For example, in Story S06, which evaluates course search functionality, *TIS* specifically measures NS05 (recognition rather than recall) performance. Similarly, Story S09, focusing on booking management, uses *TIS* to assess NS02 (user control) through the presence or absence of cancellation functions. *AIS* encompasses scores for all other NS principles not specifically targeted by the story, helping us understand the potential indirect effects of interface modifications. In S06's case, *AIS* would include scores for NS01–NS04 and NS06, while in S09, it covers NS01 and NS03–NS06.

In this grouping, we mainly focus on *TIS* because our study was not specifically designed to change *AIS*. This targeted approach helps us more accurately evaluate how well each story assesses its intended usability principle. Our study design includes both expected and unexpected scenarios for each target principle, allowing us to analyze how specific usability principles are affected by controlled changes in interface design.

The comparison between expected and unexpected scenarios serves two key purposes: first, it validates that our story design effectively creates intended variations in user experience; second, it tests Nielsen's principles' sensitivity in detecting subtle interface design changes. We predict that *TIS* will show significant differences between these scenario types, reflecting our intentional design modifications while maintaining control over non-target elements.

4.3. Data Collection and Analysis Methodology for Prompt Variation Effects

To investigate the impact of varying prompt content on evaluation outcomes, we conducted an experiment utilizing two distinct input configurations: one incorporating both screenshots and operation scripts, and another employing screenshots exclusively.

Initially, we collected data by recording operations for the diverse story tasks defined in Section 3.2.2. To enhance result variability, we set the temperature parameter of the Large Language Model (LLM) API to 0.7. Subsequently, we performed 30 evaluations for each scenario in the story, encompassing both expected and unexpected scenarios, and documented the resultant data.

Given the ordinal nature of Likert scale data and our relatively small sample size (n = 27), we employed non-parametric statistical methods. The Mann–Whitney U test was chosen for comparing independent samples (expected vs. unexpected scenarios) as it does not require normality assumptions and is suitable for ordinal data. Similarly, the Wilcoxon Signed Rank test was selected for paired comparisons (AIS–TIS consistency) as it is appropriate for analyzing differences in matched pairs of observations without assuming normal distribution.

Our statistical analysis revealed several significant findings across both prompt configurations. In the configuration with operation scripts (Table 2), we observed particularly strong effects in error handling scenarios (S01–NS03, r = 0.51), where the model effectively distinguished between the presence and absence of login failure messages. Similarly, significant differences were found in feature recognition scenarios (S06–NS05, r = 0.31) related to the course search function's presence and in system status indicators (S07–NS01) concerning loading prompts.

Table 2. Statistical analysis results of tool testing with operation script in prompts.

Story	Indicator	Target Indicator Analysis		AIS-TIS Consistency	
		Mann–Whitney U U(Z), p-Value	Effect Size r	Expected W(Z), p-Value	Unexpected W(Z), p-Value
S01	NS01	4653.0 (2.61), 0.317	0.16 (S)	0.0 (−4.73), 0.317	2714.5 (−10.91), 0.196
S01	NS03	6968.0 (8.35), 0.000 *	0.51 (L)	0.0 (−4.78), 0.000 *	2342.5 (−11.25), 0.207
S02	NS06	4095.0 (1.23), 0.218	0.07 (N)	20.0 (−4.37), 0.145	2550.0 (−11.06), 0.499
S03	NS04	3780.0 (0.45), 0.083	0.03 (N)	19.5 (−4.38), 0.609	3466.0 (−10.21), 0.010 *
S04	NS04	3225.0 (−0.93), 1.000	0.06 (N)	45.5 (−3.85), 0.286	3574.5 (−10.11), 0.440
S05	NS02	4070.0 (1.17), 0.003 *	0.07 (N)	27.0 (−4.23), 0.211	5335.0 (−8.48), 0.026 *
S06	NS05	5682.5 (5.16), 0.000 *	0.31 (M)	10.5 (−4.57), 0.000 *	2157.0 (−11.43), 0.562
S07	NS01	4125.0 (1.30), 0.000 *	0.08 (N)	22.0 (−4.33), 0.100	1961.5 (−11.61), 0.000 *
S08	NS05	3575.0 (−0.06), 0.001 *	0.00 (N)	8.0 (−4.62), 0.946	2268.0 (−11.32), 0.000 *
S09	NS02	3412.0 (−0.47), 0.512	0.03 (N)	49.0 (−3.77), 0.618	5091.0 (−8.70), 0.623

Note: * indicates $p < 0.05$. Effect size (r) interpretation: N = negligible (<0.1), S = small (0.1–0.3), M = medium (0.3–0.5), L = large (>0.5). U(Z) represents Mann–Whitney U statistic with Z-score in parentheses. W(Z) represents Wilcoxon W statistic with Z-score in parentheses. Target indicator analysis examines the difference between expected and unexpected scenarios. AIS–TIS Consistency examines the relationship between auxiliary and target indicators.

The screenshot-only configuration (Table 3) showed medium effect sizes for error handling (S01–NS03, r = 0.46) and feature recognition (S06–NS05, r = 0.33), suggesting that visual cues alone were sufficient for identifying critical interface elements. Interestingly, user control scenarios (S09–NS02) showed significant differences ($p < 0.05$) with a small effect size (r = 0.14) only in the screenshot-only configuration, particularly in distinguishing between the presence and absence of booking cancellation functions.

AIS–TIS consistency analysis revealed notable patterns. In the operation script configuration, error handling scenarios (S01–NS03) and search functionality (S06–NS05) showed significant consistency in expected scenarios, while system status (S07–NS01) and feature recognition (S08–NS05) demonstrated significant consistency in unexpected scenarios. This suggests that operational context particularly enhances the evaluation of error messages and search features.

The comparative metrics (Table 4) demonstrate that including operational data leads to notably higher precision (0.84 vs. 0.71) in identifying interface elements, particularly for critical features like error messages and search functions. While the overall accuracy remained similar (0.63 vs. 0.65), the improved recall (0.55 vs. 0.50) with operational data suggests better detection of interface inconsistencies, especially in scenarios involving user control and system status indicators.

These findings indicate that while both configurations can effectively evaluate interface elements, the inclusion of operation scripts provides particular advantages in assessing dynamic interface elements (error messages, system status) and complex features (search functionality). However, static interface elements (button placement, layout consistency) can be effectively evaluated using screenshots alone, as evidenced by the comparable performance in stories S03 and S04 across both configurations.

Table 3. Statistical analysis results of tool testing without operation script in prompts.

Story	Indicator	Target Indicator Analysis		AIS-TIS Consistency	
		Mann–Whitney U U(Z), *p*-Value	Effect Size r	Expected W(Z), p-Value	Unexpected W(Z), *p*-Value
S01	NS01	3585.0 (−0.04), 0.951	0.00 (N)	24.0 (−4.29), 0.197	700.0 (−12.78), 0.500
S01	NS03	6646.0 (7.55), 0.000 *	0.46 (M)	0.0 (−4.78), 0.000 *	2271.0 (−11.32), 0.038 *
S02	NS06	3295.0 (−0.76), 0.398	0.05 (N)	6.5 (−4.65), 0.001 *	3447.0 (−10.23), 0.002 *
S03	NS04	4092.5 (1.22), 0.157	0.07 (N)	56.0 (−3.63), 0.796	2641.5 (−10.98), 0.015 *
S04	NS04	3318.0 (−0.70), 0.418	0.04 (N)	22.0 (−4.33), 0.153	2984.0 (−10.66), 0.059
S05	NS02	2994.0 (−1.50), 0.107	0.09 (N)	81.5 (−3.11), 0.125	4329.5 (−9.41), 0.882
S06	NS05	5796.0 (5.45), 0.000 *	0.33 (M)	0.0 (−4.78), 0.000 *	3462.5 (−10.21), 0.511
S07	NS01	3180.0 (−1.04), 0.074	0.06 (N)	10.0 (−4.58), 0.096	638.0 (−12.84), 0.055
S08	NS05	4108.5 (1.26), 0.153	0.08 (N)	36.0 (−4.04), 0.000 *	2951.0 (−10.69), 0.496
S09	NS02	4533.0 (2.31), 0.013 *	0.14 (S)	32.0 (−4.12), 0.039 *	4701.0 (−9.06), 0.827

Note: * indicates $p < 0.05$. Effect size (r) interpretation: N = negligible (<0.1), S = small (0.1–0.3), M = medium (0.3–0.5), L = large (>0.5). U(Z) represents Mann–Whitney U statistic with Z-score in parentheses. W(Z) represents Wilcoxon W statistic with Z-score in parentheses. Target indicator analysis examines the difference between expected and unexpected scenarios. AIS–TIS consistency examines the relationship between auxiliary and target indicators.

Table 4. Comparative analysis of evaluation metrics for different prompt configurations.

Prompt Configuration	Accuracy	Precision	Recall	F1 Score
Visual Input with Operational Data	0.63	0.84	0.55	0.66
Visual Input Only	0.65	0.71	0.50	0.64

4.4. Participant Data Collection and Analysis

We collected 54 user experience evaluation questionnaires encompassing 27 distinct stories from the case system. Given the ordinal nature of questionnaire data and our relatively small sample size (n = 27), we employed non-parametric statistical methods. For independent sample comparisons, we utilized the Mann–Whitney U test to evaluate differences between expected and unexpected scenarios. For paired comparisons, we applied the Wilcoxon Signed Rank Test to analyze AIS–TIS consistency.

Statistical analysis revealed several key findings, as shown in Table 5. Regarding system status visibility (NS01), S01 demonstrated significant differences (U = 67.5, $p = 0.015$, r = 0.47), indicating that participants could clearly identify how the absence of error messages impacted user experience. Similarly, in terms of consistency and standards (NS04), S04 showed significant differences (U = 62.5, $p = 0.035$, r = 0.39), reflecting that changes in back button positioning meaningfully affected users' perception of interface consistency. These medium effect sizes suggest that these design modifications produced substantial impacts on user experience.

However, other story scenarios (S02, S03, S05–S09) failed to produce significant differences, potentially reflecting two issues: first, participants might require more training to identify subtle interface design problems; second, some design variations might not be as

pronounced as anticipated. Notably, in the AIS–TIS consistency analysis, most scenarios showed no significant differences, suggesting that our initial expectations about the impact of design changes on overall user experience might have been overestimated.

Table 5. Statistical analysis of participant questionnaire responses across testing scenarios.

Story	Indicator	Target Indicator Analysis		AIS–TIS Consistency	
		Mann–Whitney U U(Z), *p*-Value	Effect Size r	Expected W(Z), *p*-Value	Unexpected W(Z), *p*-Value
S01	NS01	67.5 (2.43), 0.015 *	0.47 (M)	0.0 (−1.60), 0.250	74.5 (−2.16), 0.924
S01	NS03	59.5 (1.81), 0.059	0.35 (M)	1.0 (−1.07), 0.500	42.0 (−3.09), 0.485
S02	NS06	34.5 (−0.12), 0.937	0.02 (N)	2.5 (−0.27), 0.750	27.0 (−3.51), 0.581
S03	NS04	37.0 (−0.61), 0.542	0.12 (S)	1.5 (−1.28), 1.000	48.0 (−2.74), 0.765
S04	NS04	62.5 (2.04), 0.035 *	0.39 (M)	0.0 (−1.60), 0.250	37.0 (−3.23), 0.523
S05	NS02	40.0 (0.31), 0.783	0.06 (N)	1.5 (−0.80), 0.500	64.5 (−2.44), 0.566
S06	NS05	11.0 (−1.93), 0.052	0.37 (M)	0.0 (−1.60), 0.317	46.5 (−2.96), 0.430
S07	NS01	39.0 (0.23), 0.845	0.04 (N)	0.0 (−1.60), 0.250	71.0 (−2.26), 0.793
S08	NS05	10.5 (−1.34), 0.183	0.26 (S)	1.5 (0.00), 1.000	42.0 (−3.24), 0.498
S09	NS02	11.0 (−1.93), 0.054	0.37 (M)	0.0 (−1.60), 0.250	56.5 (−2.67), 0.339

Note: * indicates $p < 0.05$. Effect size (r) interpretation: N = negligible (<0.1), S = small (0.1–0.3), M = medium (0.3–0.5), L = large (>0.5). U(Z) represents Mann–Whitney U statistic with Z-score in parentheses. W(Z) represents Wilcoxon W statistic with Z-score in parentheses. Target indicator analysis examines the difference between expected and unexpected scenarios. AIS–TIS consistency examines the relationship between auxiliary and target indicators.

The evaluation metrics analysis presented in Table 6 (accuracy = 0.33, precision = 0.50, recall = 0.10, F1 score = 0.16) further supports this observation. The particularly low recall rate is noteworthy, indicating that participants were conservative in identifying unexpected scenarios, possibly tending to view problematic situations as normal. This conservative tendency might stem from users' adaptability to systems, where they adjust their behavior to accommodate suboptimal designs rather than identifying them as issues.

Table 6. Evaluation metrics for participant questionnaire responses.

Accuracy	Precision	Recall	F1 Score
0.33	0.50	0.10	0.16

These findings not only highlight the complexity of user experience evaluation but also emphasize the need for more nuanced discrimination criteria in assessment method design to improve evaluation sensitivity. Simultaneously, these results provide concrete directions for future improvements in user experience evaluation methods, particularly in evaluator training and assessment criteria development. Our statistical approach, carefully chosen to match the characteristics of our data, ensures the reliability of our analysis while acknowledging the challenges inherent in conducting user experience research with limited samples.

5. Discussion

5.1. Prompt Variation Effects on UX Evaluation

To address RQ1: "How does providing different information within prompts influence the accuracy of user experience detection by GPT-4?", we synthesize our findings as follows:

The experimental results demonstrate the complex impact of incorporating diverse information within prompts on GPT-4's performance in user experience (UX) detection. Our comparative analysis reveals that prompts integrating both visual input (screenshots) and operational data (interaction scripts) yield different detection patterns compared to those utilizing visual input alone, across various UX scenarios.

The statistical analyses presented in Tables 2 and 3, along with the evaluation metrics in Table 4, support the complexity of these findings. While the overall accuracy of the

visual-only input configuration is slightly higher than the combined visual and operational data configuration (0.65 vs. 0.63), the latter demonstrates better performance in other important metrics: precision (0.84 vs. 0.71), recall (0.55 vs. 0.50), and F1 score (0.66 vs. 0.64).

This apparent contradiction highlights the multifaceted nature of UX evaluation. The higher precision indicates that GPT-4, when using prompts combining visual and operational data, more accurately identifies genuine UX issues, reducing false positives. At the same time, the improved recall suggests an enhanced ability to detect a wider range of potential issues, minimizing false negatives. The slight improvement in F1 score reflects a more balanced trade-off between precision and recall.

The complexity of the findings is further illustrated by specific scenarios. In cases evaluating visual consistency, such as S03 and S04, which focus on the color and position of the back button, the differences between the two prompt configurations were less noticeable. This observation may help explain why the overall accuracy did not significantly improve, as additional operational data may not offer extra insights for purely visual design elements. Furthermore, in scenario S09, which examines the presence of a booking cancellation function, GPT-4 showed higher sensitivity without operational scripts. This phenomenon might partially explain the slight advantage in overall accuracy for the visual-only input configuration, reminding us that simpler prompts can sometimes be more effective in certain contexts.

In conclusion, our findings show that providing richer, multi-modal information in prompts significantly influences GPT-4's performance in UX detection, albeit in a complex manner. While overall accuracy has only slightly improved, enhancements in key metrics such as precision, recall, and F1 score suggest that combining visual and operational data may offer a more comprehensive and nuanced UX evaluation.

These results emphasize the importance of careful prompt engineering when using GPT-4 for automated UX assessment. Different prompt strategies may be necessary depending on the specific evaluation objectives (e.g., system status clarity, error handling, consistency, or usability). Future research should further explore how to optimize the balance between accuracy, precision, and recall across various types of UX evaluation tasks. Additionally, investigating how to tailor prompt design for specific UX domains and evaluation criteria will be crucial to fully utilize GPT-4's potential in automated UX evaluation.

5.2. Comparative Analysis of Human and Automated UX Evaluations

To address RQ2: "What is the performance of GPT-4 in user experience detection compared to human evaluators when using optimized prompts?", we synthesize our findings as follows, considering the specific tasks associated with each scenario:

The comparative analysis of GPT-4 and human evaluators in user experience detection provides significant insights into the capabilities and limitations of both approaches. Our findings indicate that GPT-4, when utilizing optimized prompts, generally outperforms human evaluators across multiple dimensions of user experience assessment.

In the domain of *TIS*, GPT-4 demonstrated markedly superior performance compared to human evaluators. As shown in Tables 2 and 6, GPT-4 achieved an accuracy of 0.63, precision of 0.84, recall of 0.55, and F1 score of 0.66, while human evaluators scored 0.33, 0.50, 0.10, and 0.16 respectively. This disparity is particularly evident in specific scenarios outlined in Table 1.

In stories S05 and S09, which focus on user control, GPT-4 performed better in recognizing the importance of the back button, reflecting its high sensitivity to user control elements. For stories S06 and S08, which evaluate easy-to-recognize functions, GPT-4 demonstrated a clear advantage in identifying the presence of course search and history records functions. This may be due to its comprehensive analysis of interface functionality completeness. In scenario S07, which assesses system status display, GPT-4 excelled in recognizing the importance of system prompts during loading, reflecting its high regard for system feedback timeliness.

In the assessment of *AIS-TIS Expected Consistency*, GPT-4 and human evaluators showed similar performance in certain stories. For instance, both identified significant consistencies in S01 (error handling) and S06 (course search function). This suggests that human intuition remains effective for some fundamental user experience principles.

However, in detecting *AIS-TIS Unexpected Consistency*, GPT-4 again demonstrated a clear advantage. GPT-4 identified significant unexpected consistencies in multiple stories (S01, S03, S05, S08), while human evaluators failed to identify any. This difference particularly highlights GPT-4's superiority in identifying potential user experience issues, especially in non-standard or unexpected situations.

These findings have important implications for advancing automated UX evaluation beyond traditional functional testing approaches. While existing automated tools excel at performance metrics and functionality testing, GPT-4 demonstrates unique capabilities in detecting nuanced aspects of user experience - from cognitive interactions to emotional responses. Our results show that GPT-4 not only consistently identifies subtle design issues and potential UX pitfalls that might be overlooked by human evaluators but also provides comprehensive analysis of user interaction patterns and emotional needs that are typically challenging for traditional automated tools to assess.

Nevertheless, the comparable performance of human evaluators in certain aspects of expected consistency assessment suggests an optimal path forward: a synergistic approach that combines GPT-4's systematic analysis with human professional judgment. This hybrid methodology could leverage GPT-4's strengths in comprehensive pattern recognition and consistent evaluation while benefiting from human experts' intuitive understanding of user needs and contextual nuances. Future research should explore this integration through broader comparative analysis across different platforms and application scenarios, ultimately advancing the field of automated UX testing while maintaining the valuable role of human expertise in the evaluation process.

5.3. Limitation

This study's findings should be interpreted in light of several limitations. The relatively small sample size may not fully represent the diverse range of user experiences, potentially limiting the generalizability of results. Environmental factors, such as testing venue and network conditions, could have introduced uncontrolled variables affecting user interactions. These constraints may have influenced the observed outcomes and their applicability to real-world scenarios.

Furthermore, the inherent subjectivity in user experience perception presents challenges in establishing a universal evaluation standard, as individual differences in background and expectations may influence assessments. Additionally, the creation of "ground truth" scenarios for evaluation, while based on established heuristics, may inadvertently incorporate researcher bias. These limitations highlight the need for future research employing larger, more diverse participant pools and refined methodologies to enhance the robustness of findings in AI-assisted UX evaluation.

6. Conclusions

This study makes significant contributions to the field of automated User Experience (UX) evaluation through the development of an innovative AI-powered tool leveraging GPT-4's capabilities. By integrating six of Nielsen's usability heuristics with user interaction data captured by the Rapi web recording tool, our approach demonstrates the potential for rapid and consistent UX assessments, potentially reducing costs for UX designers in early-stage testing.

Our research delved into the impact of different prompt configurations on GPT-4's UX evaluation performance, comparing prompts containing both visual input and operational data against those with visual input alone. While overall accuracy was similar, the combined approach showed superior precision, recall, and F1 scores, highlighting the importance of comprehensive prompt engineering. A rigorous comparison between

GPT-4 and human evaluators across multiple UX dimensions revealed GPT-4's superior performance in Task Identification Significance, suggesting its ability to identify subtle UX issues that human evaluators might overlook.

To validate our automated tool's effectiveness, we designed a comprehensive case study involving a tennis court reservation system, evaluating its performance across various scenarios and interface versions. Statistical analyses using Wilcoxon signed-rank and Mann-Whitney U tests revealed significant differences in UX scores across scenarios, confirming the tool's sensitivity to design variations.

Our research provides valuable insights into GPT-4's strengths and limitations in UX evaluation, excelling in systematic analysis of consistency, error handling, and functional completeness while also indicating areas where human expertise remains valuable. These findings contribute to the growing body of knowledge on AI-assisted UX evaluation and offer practical implications for UX practitioners, paving the way for future research to expand the tool's application to diverse web environments, explore additional forms of user interaction data, and extend its capabilities to other digital platforms. By continuing to refine and validate AI-powered UX evaluation tools, we can work towards more efficient, comprehensive, and user-centered design processes.

Building on this study's findings, future research should focus on three key areas to advance AI-powered user experience (UX) evaluation. First, investigating the complementary relationship between GPT-4 and human evaluators by specifically examining scenarios where human expertise remains crucial, such as evaluating emotional resonance in healthcare applications or cultural appropriateness in localized interfaces. Our findings showing human evaluators' comparable performance in fundamental UX principles suggest that optimal UX evaluation requires a balanced integration of AI efficiency and human insight. Second, exploring the integration of multimodal data sources while establishing rigorous ethical guidelines for AI-based UX evaluation. This includes developing transparent frameworks for handling voice and video inputs, setting clear boundaries for data privacy, and creating validation protocols to ensure GPT-4's evaluation reliability, particularly in high-stakes applications like financial or medical interfaces. Third, extending the tool's application to mobile platforms through a systematic approach that involves: optimizing prompt engineering based on our experimental findings of configuration impacts, establishing platform-specific heuristics that complement Nielsen's principles, and developing clear metrics for measuring AI-human evaluation alignment. These directions aim to develop a more holistic UX evaluation framework that not only advances technical capabilities but also ensures ethical deployment and meaningful human oversight in the evolution of automated UX assessment.

Author Contributions: Conceptualization, N.-L.H., H.-J.L. and L.-C.L.; methodology, N.-L.H., H.-J.L. and L.-C.L.; software, N.-L.H., H.-J.L. and L.-C.L.; validation, N.-L.H., H.-J.L. and L.-C.L.; writing—original draft preparation, N.-L.H., H.-J.L. and L.-C.L.; writing—review and editing, N.-L.H., H.-J.L. and L.-C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Science and Technology Council, Taiwan R.O.C., under grants NSTC112-2221-E-035-030-MY2.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to the data contains personal information of participants.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Nielsen's Heuristic Principle

NS01—Visibility of system status: the design principle that states systems should always keep users informed about what is going on, through appropriate feedback within a reasonable time. Nielsen asserts that users should never be in doubt about the system's

current state or whether their actions have been successful. Especially when the execution time in the case exceeds 3000 milliseconds, a system must prompt the current system status.

NS02—User control and freedom: User Control and Freedom refers to Nielsen's principle that users often perform actions by mistake. They need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended process. This includes all relevant buttons for undo and redo actions, providing the freedom and control to manage their interaction with the system. For example, cancel button, return button, etc.

NS03—Helps users recognize, diagnose, and recover from errors: Help Users Recognise, Diagnose, and Recover from Errors is a principle that emphasizes error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution. Nielsen stresses the importance of designing systems that make it clear when an error has occurred and help users understand what has gone wrong and how they can fix it without additional frustration. For instance, during account registration, if the password format does not meet the required criteria, the system should provide a prompt specifying the particular password policy that has been violated.

NS04—Consistency and standards: Consistency and Standards refer to the principle that user interfaces should be consistent throughout a single application and also match with platform conventions and user expectations. Essentially, interfaces should follow consistent rules, and users should not have to wonder whether different words, situations, or actions mean the same thing. Jakob Nielsen emphasizes that consistency allows users to transfer knowledge and skills from one part of an application to another, and from one application to another, reducing the learning curve. For example, the return button should be to the left of the Confirm button and the Confirm button should be more conspicuous.

NS05—Recognition rather than recall: Recognition Rather Than Recall is the principle that systems should minimize the user's memory load by making objects, options, and actions visible or easily retrievable. Nielsen advocates for interfaces to reduce user's cognitive load by making elements recognizable rather than expecting users to remember information from one part of an interface to another. For example, history records and course search functions.

NS06—Match Between System and The Real World: This principle suggests that a system should speak the users' language, with words, phrases, and concepts familiar to the user, rather than system-oriented terms. It should follow real-world conventions, making information appear in a natural and logical order. The idea is to leverage users' existing knowledge of the world to make systems understandable and intuitive. For example, information can be presented in a table, or a trash bin icon is instantly recognizable as a place to discard files because it mirrors the physical object we're all familiar with.

Appendix B. Prompt Template

[Introduction and Context]
Hello, I have provided a JSON dataset that records user interactions on a website. This dataset includes each user action, associated timestamps, outcomes, snapshot, the web operations and the context of execution.

Here is a snippet of the JSON dataset:

```
{
    ''Chrome 122.0.0.6'': [
    {
        ''version'': {...},
        ''title'': ''Task2'',
        ''startTime'': ''20240327 04:07:04'',
        ''endTime'': ''20240327 04:08:13'',
        ''Logs'': [...],
        ''status'': ''completed'',
        ''snapshot'': {...},
    }
```

}

[JSON format explanation]
This JSON file contains the format of a test.

- title: The title of the test.
- startTime and endTime: The start and end times of the test.
- logs: The "message" format is: "Initializing (Inputting) or ...
- cases: Contains "title" (the title of the test case), ...
- screenshots: show the current pages while operating the ...

[Nielsen's Usability Principles]
I would like to evaluate this website using the following Nielsen's usability principles:

- NS01—Visibility of system status: Visibility of System Status is ...
- NS02—User control and freedom: User Control and Freedom refers to ...
- NS03—Help users recognize, diagnose, and recover from ...
- NS04—Consistency and standards: Consistency and Standards refer ...
- NS05—Recognition rather than recall: Recognition Rather Than Recall ...
- NS06—Match Between System and The Real World: This principle suggests ...

[Request for Evaluation]
Based on the above information, analyze and evaluate the website's performance referencing the snapshots and time from the records studies, according to Nielsen's usability principles. Rate each area on an integer scale from e to 5, adjusting the score up or down from a baseline of 3 points based on significant differences. Finally, provide justifications for each rating and summarize the overall evaluation.

Appendix C. UX Questionnaire

1. At any time, I am clear about the current state of the system.
2. During operation, the webpage responds within a reasonable time.
3. I can return to any page I want at any time.
4. It is possible to cancel and redo any operation at any time.
5. All error messages from the system are clear and understandable.
6. There are reminders for major system operations.
7. Similar color scheme is used for all interfaces.
8. Consistent layout is used for all interfaces.
9. The webpage layout design follows general website standards.
10. When using this system, I can complete tasks without having to remember too much.
11. The system's display method conforms to real-world habits.

References

1. Law, E.L.C.; Roto, V.; Hassenzahl, M.; Vermeeren, A.P.; Kort, J. Understanding, scoping and defining user experience: A survey approach. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09), New York, NY, USA, 4–9 April 2009; pp. 719–728. [CrossRef]
2. Nielsen, J.; Molich, R. Heuristic evaluation of user interfaces. In Proceedings of the SIGCHI conference on Human factors in Computing Systems, Seattle, WA, USA, 1–5 April 1990; pp. 249–256.
3. Alomari, H.W.; Ramasamy, V.; Kiper, J.D.; Potvin, G. A User Interface (UI) and User eXperience (UX) evaluation framework for cyberlearning environments in computer science and software engineering education. *Heliyon* **2020**, *6*, e03917. [CrossRef] [PubMed]
4. Alayrac, J.B.; Donahue, J.; Luc, P.; Miech, A.; Barr, I.; Hasson, Y.; Lenc, K.; Mensch, A.; Millicah, K.; Reynolds, M.; et al. Flamingo: A visual language model for few-shot learning. In Proceedings of the 36th International Conference on Neural Information Processing Systems (NIPS '22), New Orleans, LA, USA, 28 November–9 December 2022.
5. Sun, Q.; Yu, Q.; Cui, Y.; Zhang, F.; Zhang, X.; Wang, Y.; Gao, H.; Liu, J.; Huang, T.; Wang, X. Emu: Generative Pretraining in Multimodality. In Proceedings of the Twelfth International Conference on Learning Representations, Vienna, Austria, 1–7 May 2024.
6. Vermeeren, A.P.; Law, E.L.C.; Roto, V.; Obrist, M.; Hoonhout, J.; Väänänen-Vainio-Mattila, K. User experience evaluation methods: current state and development needs. In Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries, Reykjavik, Iceland, 16–20 October 2010; pp. 521–530.

7. Kashfi, P.; Nilsson, A.; Feldt, R. Integrating User eXperience practices into software development processes: implications of the UX characteristics. *PeerJ Comput. Sci.* **2017**, *3*, e130. [CrossRef]
8. Inan Nur, A.; B. Santoso, H.; O. Hadi Putra, P. The method and metric of user experience evaluation: A systematic literature review. In Proceedings of the 2021 10th International Conference on Software and Computer Applications, Kuala Lumpur, Malaysia, 23–26 February 2021; pp. 307–317.
9. Albert, B.; Tullis, T. *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics*; Newnes: Solihull, UK, 2013.
10. Guimaraes, E.A.d.A.; Morato, Y.C.; Carvalho, D.B.F.; Oliveira, V.C.d.; Pivatti, V.M.S.; Cavalcante, R.B.; Gontijo, T.L.; Dias, T.M.R. Evaluation of the usability of the immunization information system in Brazil: a mixed-method study. *Telemed. e-Health* **2021**, *27*, 551–560. [CrossRef] [PubMed]
11. Mochammad Aldi Kushendriawan, Harry Budi Santoso, P.P.O.H.M.S. Evaluating User Experience of a Mobile Health Application Halodoc using User Experience Questionnaire and Usability Testing. *J. Sist. Inf. (J. Inf. Syst.)* **2021**, *17*, 58–71. [CrossRef]
12. Nguyen, J.; Dupuis, M. Closing the feedback loop between UX design, software development, security engineering, and operations. In Proceedings of the 20th Annual SIG Conference on Information Technology Education, Tacoma, WA, USA, 3–5 October 2019; pp. 93–98.
13. Namoun, A.; Alrehaili, A.; Tufail, A. A review of automated website usability evaluation tools: Research issues and challenges. In Proceedings of the International Conference on Human-Computer Interaction, Virtual, 24–29 July 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 292–311.
14. Biringa, C.; Kul, G. Automated user experience testing through multi-dimensional performance impact analysis. In Proceedings of the 2021 IEEE/ACM International Conference on Automation of Software Test (AST), Madrid, Spain, 20–21 May 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 125–128.
15. Whiting, E. Granular Modeling of User Experience in Load Testing with Automated UI Tests. In Proceedings of the 2021 International Conference on Data and Software Engineering (ICoDSE), Bandung, Indonesia, 3–4 November 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–5.
16. Norman, D. *The Design of Everyday Things: Revised and Expanded Edition*; Basic Books: New York, NY, USA, 2013.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.