

Tutorial

A Secure User Interface for Preclinical Evaluation of AI in Patient Portal Message Management: Tutorial

Kelly Gleason¹, PhD, RN; Thomas Kidu², MSE; Vignesh Babu², MSE; Brian Hasselfeld¹, MD; Jennifer Wolff³, PhD

¹School of Nursing, Johns Hopkins University, Baltimore, MD, United States

²Whiting School of Engineering, Johns Hopkins University, Baltimore, MD, United States

³Bloomberg School of Public Health, Johns Hopkins University, Baltimore, MD, United States

Corresponding Author:

Kelly Gleason, PhD, RN
School of Nursing, Johns Hopkins University
525 N. Wolfe Street
Baltimore, MD 21215
United States
Phone: 1 708334876
Email: kgleaso2@jhmi.edu

Abstract

The growing use of AI to support patient portal message management requires rigorous preclinical evaluation. Directly testing AI within electronic health record (EHR) systems poses significant safety, workflow, and data-governance risks. Here, we present a technical feasibility report on a secure user interface (UI) sandbox designed to enable clinical and technical teams to experiment with AI for portal messaging before clinical integration. In this context, a “sandbox” refers to a controlled, nonproduction environment that allows safe testing, prompt iteration, and evaluation of AI outputs without impacting live EHR systems or patient care. We developed a web UI in Python 3 with a modular backend for data handling and AI task execution that operates entirely within the institutional firewall. The system runs in a secure research environment equipped with an NVIDIA GRID T4-1Q graphics processing unit (GPU) and institutional access controls. We designed a deidentification pipeline to remove or replace personal health identifiers and assessed its precision. The platform supports single-message and batch workflows and exposes example large language model (LLM)-enabled tasks such as authorship identification, message categorization, criticality flagging, and response drafting using zero-shot, one-shot, and few-shot prompting. The system successfully executed end-to-end workflows to ingest messages, run individual or batch AI analyses, and present outputs for review. Personal health information partial masking was applied across the corpus using a deidentification pipeline validated against 110 manually adjudicated entities (sensitivity 95.1%, precision 82.1%). We ran use cases with an institutional review board–approved corpus of a dementia-relevant subset of 6941 patient portal messages categorized as “medical advice requests” from 497 unique patients. With the support of the UI, we tested which prompting strategies yielded interpretable outputs for authorship identification, categorization, and criticality flagging, and whether response drafting produced editable clinician starting points. A token-based cost readout provided transparent operating estimates for LLM-backed tasks. This framework offers a practical, secure path to test AI behavior on real messages without affecting live EHR workflows and thus supports exploratory testing, prompt iteration, and comparative analyses, including LLM prompts versus baseline models, while preserving governance boundaries. We discuss design choices, safety controls, and the limits of a sandbox approach. A secure, UI-based sandbox enables health system teams to evaluate AI for patient portal messaging before clinical integration. The goal is not to assume benefit but to generate evidence about feasibility, risks, and fit to clinical needs in a controlled setting.

JMIR Med Inform 2026;14:e83216; doi: [10.2196/83216](https://doi.org/10.2196/83216)

Keywords: patient portal; management; electronic health records; artificial intelligence; AI; large language models; tutorial

Introduction

Patient portal messaging has expanded rapidly and now encompasses clinical questions, symptom updates, and coordination tasks that can affect care quality and timeliness [1-4]. As message volumes rise, concerns about clinician workload and burnout have grown [2]. In parallel, health systems are exploring AI to help manage message flow through triage support, authorship cues, categorization, and draft responses [5]. Despite optimism that AI may help, its true impact on workload, safety, and equity remains uncertain [6-8].

The increasing use of AI to support patient portal message management requires rigorous testing and evaluation. Patient portal messages convey critical health information, from appointment reminders and laboratory results to educational content, making clarity, accuracy, and accessibility essential. Even minor ambiguities or technical errors in poorly designed AI can lead to misunderstandings and compromise care quality [9-11].

Developing AI tools for portal inbox management is particularly challenging within live electronic health record (EHR) systems. Direct integration at early stages can introduce significant risks, including threats to data integrity and barriers to agile development [12]. The sensitive nature of health information necessitates a secure, isolated environment to test messaging functions without endangering patient safety or system stability. A dedicated testing platform is therefore critical because it must realistically simulate message flows, support the creation and evaluation of dynamic content, and enable real-time interactions with patients and clinicians. Importantly, this environment should remain separate from the EHR during development and testing to allow researchers to gather rich feedback, iterate on message design, and validate functionality in a controlled setting to ensure that only vetted, effective approaches reach the live clinical environment.

A sandbox is a controlled, nonproduction environment that allows safe testing, prompt iteration, and evaluation of AI outputs without impacting live EHR systems or patient care. A preclinical sandbox can enable secure experiments with real messages, which is particularly valuable given the risks and difficulties of direct experimentation inside live EHRs [12]. A well-designed sandbox could separate front-end interactions from back-end processing, provide robust deidentification, allow plug-and-play AI components, and offer transparent telemetry estimates, including latency, token counts, and error traces. Above all, it should keep experimentation out of production until teams understand behavior, constraints, and risks [13]. Building on this need, we present a technical feasibility report describing a user interface (UI)-based sandbox developed within a secure, governed research environment.

Technical Architecture and Development

Setting and Governance

All development and experiments occurred within a secure research environment following institutional review board (IRB) approval and access controls. No production EHR systems were touched. The environment was equipped with an NVIDIA GRID T4-1Q graphics processing unit (GPU) to accelerate local natural language processing (NLP) pipelines and baseline model experiments.

Software Stack and Architecture

The system is implemented in Python (version 3.12; Python Software Foundation) and Streamlit (version 1.46.1) to design the web interface. The backend is modular, with a data layer for loading and validating input, a privacy layer for deidentification, and an AI layer for task execution. Core packages included pandas and numpy for data handling, spaCy and scispaCy for entity recognition [14,15], Faker for personal health information (PHI) replacement tokens, and the OpenAI Python software development kit (SDK) configured for Azure OpenAI (GPT-4) for large language model (LLM) tasks. The deployed model was *gpt-4.1*, version *2025-04-14*, accessed using API version *2024-12-01-preview*. All dependencies were pinned in a requirements file to support reproducibility. The UI and backend are decoupled, which makes it easy to update prompts, switch model end points, or add tasks without reworking the interface. To support reproducibility, the companion source code, pinned requirements file, and synthetic test data are available in the public repository described in the Data Availability section.

Data and Cohort Construction

We started with an IRB-approved corpus of patient portal messages over several years. For early feasibility experiments, we focused on the medical advice request message type. After standard cleaning, the working set included 6941 messages from 497 unique patient accounts in a cohort of older adults, described elsewhere [16]. This subset was chosen to capture a range of clinical questions and caregiver-authored notes, but it is not intended to represent all portal message types [16,17].

Deidentification Pipeline

To protect patient privacy, we built a deidentification pipeline that detects and transforms personal health identifiers. Deidentification was used as an additional safeguard and as a testable privacy-preserving component, not as the only privacy control. The UI stayed within the institutional firewall, with messages only accessible by the IRB-approved study team. The pipeline uses spaCy and scispaCy named entity recognition (NER) models and custom regex patterns to identify names, health care provider titles, dates, phone numbers, addresses, and medication mentions. Replacements are deterministic per document to preserve within-message coherence; for example, the same name is mapped to the same synthetic token throughout a message. Deterministic

replacement was used to preserve coherence within a single message, but synthetic placeholders were not intended to serve as persistent cross-thread pseudonyms. Conversation reconstruction was handled in the secure UI layer using internal message metadata. This design prioritizes message-level interpretability while reducing the risk that repeated placeholders could be used as linkage keys across messages. Covering medications required biomedical entity models, which we supplemented with curated lists where model coverage was limited. We sought to ensure that the output would preserve clinical semantics while removing identifiers.

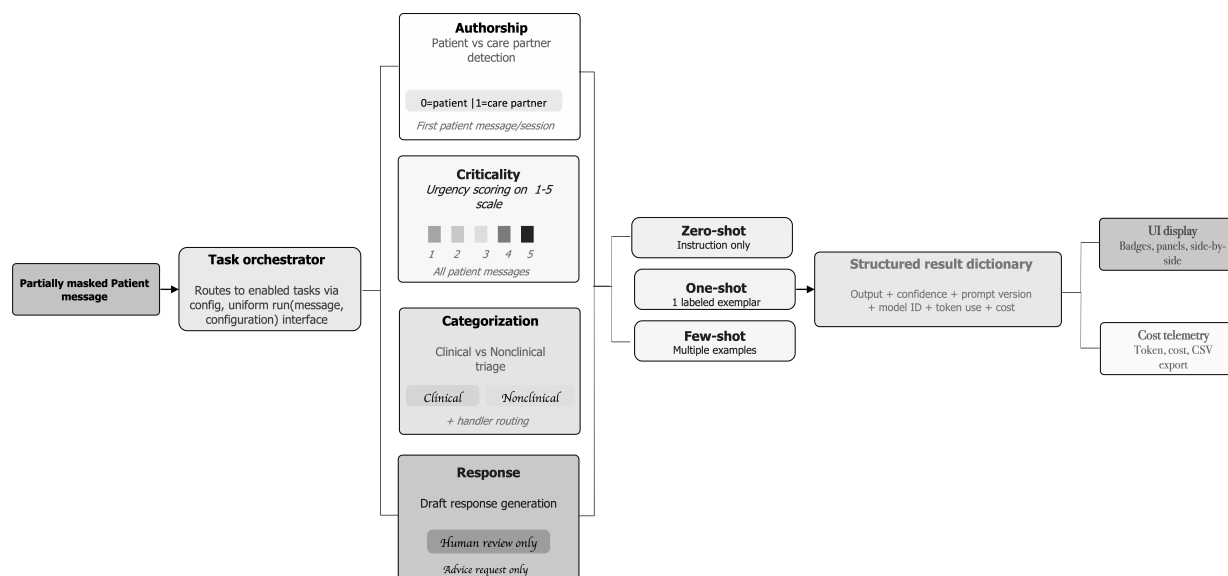
AI Tasks and Prompting Strategy

The platform was developed to enable experimentation with different AI use cases relevant to patient portal messaging. We studied four representative tasks: (1) authorship identification, to classify message authorship among 3 categories (patient, care partner, or ambiguous); (2) message categorization, to separate clinical content from

administrative requests; (3) criticality flagging, to highlight urgent versus routine communications; and (4) response generation, to create editable draft replies that clinicians can review and refine.

All of these use cases were implemented using LLMs, tested with zero-shot, one-shot, and few-shot prompting strategies (Multimedia Appendix 1). In the one-shot setting, the model is given a single example before the target message, while in the few-shot setting, multiple examples are provided to help the model generalize the intended pattern [18]. Outputs were always framed as drafts for human review, as is currently standard practice in the use of LLM-generated drafts of patient portal messages [5]. Figure 1 summarizes the AI task orchestration layer, showing how a patient message is routed through selected use cases, processed with zero-shot, one-shot, or few-shot prompting, and returned as structured outputs for UI display and cost telemetry.

Figure 1. Prompt templates. UI: user interface.



UI and Workflows

The UI was designed to have several complementary workflows within a single platform. The most basic workflow enables the creation of scenarios from the full dataset of 6941 messages. Once loaded, the deidentification engine is automatically applied. Users can then select an enterprise identifier, which links patient and clinician messages, to reconstruct conversations. This enables side-by-side comparison of a conventional scenario with its AI-augmented counterpart, in which suggested responses or flagged messages are surfaced for clinician review. This feature supports task-level testing and the evaluation of AI within the broader flow of communication.

A second workflow allows users to enter a single message manually or upload a batch file of messages. In single-message mode, users paste a deidentified message, select one or more tasks, and view results in a side-by-side panel with the original redacted text. In batch mode, users upload a CSV

file of messages; the system runs the deidentification pipeline and then executes selected tasks, producing a downloadable results file.

Additional features include a criticality ordering tool, which prioritizes messages according to urgency, and a cost analysis panel, which aggregates token use and estimates operational expenses. Together, these workflows extend the platform beyond isolated message-level testing, making it possible to simulate how AI would function in real-world messaging streams.

Experiment Setup and Evaluation Approach

Experiments were designed to validate end-to-end functionality rather than establish clinical performance. We executed the deidentification pipeline across the 6941-message subset, then ran exemplar AI tasks in single-message and batch modes to confirm that the system produced interpretable

outputs, maintained deidentification, and recorded provenance (prompt choice and task configuration).

Informal expert walkthroughs with clinical stakeholders were conducted to identify usability issues and assess whether outputs were appropriately constrained and reviewable. Participants included 2 physicians in health system information technology leadership roles, 3 internal medicine physicians, 1 nurse who triages patient portal messages, and a coalition of care partners, patient advocates, and health information technology researchers. Each walkthrough lasted 30 to 60 minutes and included an interface review, a demonstration of core features, and time for questions and feedback. We received consistent positive feedback regarding how valuable it would be to extend access to this sandbox for various AI-driven use cases. We also received feedback that more information regarding the patient's context should be uploaded into the UI to enable users to assess the outputs. A primary usability challenge identified was that a basic level of comfort with Python was required to open the UI in the secure environment. We also incorporated insights from interviews with individuals with dementia and their care partners regarding their perceptions of AI-generated patient portal message responses [17].

We instrumented the UI to display token counts and approximate request costs. Given the volatility of LLM pricing, the cost panel was used primarily to compare prompt strategies and model choices rather than predict exact operational spending. As the focus was on the platform itself, we did not conduct head-to-head accuracy studies, which are reserved for future validation.

System Performance and Illustrative Findings

The system executed the complete pipeline, from message ingestion to output presentation, without accessing live EHR systems. Across the 6941-message feasibility corpus, the interface supported single-message testing, batch processing, prompt selection, task execution, cost telemetry, and export of results. Because this manuscript reports technical feasibility rather than formal model evaluation, we do not present task-level output distributions or performance metrics.

AI-supported task execution was averaged approximately 5 seconds end-to-end (deidentification plus all 4 AI tasks) per message. The application included retry handling for API rate-limit errors, with up to 3 retry attempts for HTTP 429 responses. Average message length was approximately 150 to 200 words, with prompt-based processing estimated at 300 to 400 input tokens and 100 to 150 output tokens per message. Using conservative pricing assumptions, this corresponded to an estimated upper-end cost of approximately US \$0.044 per message. These values are presented as operational feasibility estimates rather than production benchmarks.

We ran deidentification by replacing personal names, dates, and contact details with deterministic placeholders while preserving contextual meaning. Biomedical entity coverage for medication mentions was achieved by

combining scispaCy models with a small, curated lexicon. Inspection of deidentified text confirmed that messages remained coherent and suitable for downstream analysis, although occasional overmasking of medication names was observed. To evaluate the NER pipeline used for deidentification, we tested performance on a message set. Manual review of pipeline output was conducted to identify false positives and false negatives, yielding 110 adjudicated entities. The pipeline achieved 82.1% precision, 80.9% accuracy, 95.1% sensitivity, and 39.3% specificity, with 4 false negatives identified. Because the deidentification did not achieve 100% sensitivity, all nonsynthetic messages were treated as containing PHI, including after they underwent deidentification.

In the use cases, we found that authorship prompts produced rational, auditable rationales (eg, references to third-person phrasing or caregiver self-identification) useful for reviewer feedback. We observed failure modes, such as the AI wrongly detecting care partner message authorship when in fact a patient authored the message and mentioned a spouse or family member. We also tested whether categorization and criticality prompts yielded stable labels on representative samples, and whether disagreements across prompting strategies were visible in the UI. Showing the output from the categorization and criticality prompts allowed us to collect feedback from key partners (informatics leads and physicians) that may have been otherwise hard to collect. Similarly, we were able to explore whether response drafting generated concise, editable text. We observed that the model occasionally produced overly generic and lengthy replies that did not directly address the specific clinical concern raised in the message. Batch runs were completed on the 6941-message subset and exported to a results table that combined input metadata, task outputs, and prompt provenance.

The token panel surfaced per-message and aggregate token counts, allowing users to see how different prompt templates affect inference cost. This made trade-offs between few-shot accuracy and budget more tangible during experimentation.

Discussion

A core challenge for AI in patient portal messaging is that most tools reach clinicians long before they have been studied with realistic data and users [5,19]. Our approach prioritizes a preclinical testbed that allows health system teams to probe behavior, iterate prompts, and examine costs in a controlled setting. Several design choices were important. First, we enforced a privacy boundary: all messages were partially masked before any model access and treated as containing PHI throughout the entire process. Second, we made prompts first-class, versioned objects so that results are reproducible and modifiable. Third, we exposed batch processing alongside single-message testing to support both qualitative review and early quantitative exploration without implying clinical readiness. Our initial corpus was intentionally narrow (dementia-relevant medical advice requests from older adults), which constrains claims about performance across other message types and populations. The UI

architecture, however, was built to be extensible: additional message datasets, specialized lexicons, and prompt templates can be slotted in without changing the UI.

The sandbox helps align stakeholders around evidence rather than assumptions by making experimental outputs visible and auditable. Engineers can use the environment to compare prompt templates, routing logic, and cost trade-offs; clinicians can preview which outputs reduce workload and which create new review burden; and governance teams can assess safety and equity risks before any integration decisions. The current implementation uses the OpenAI Python SDK configured for Azure OpenAI within the institutional environment. The AI layer was designed for extensibility, meaning future versions could evaluate baseline classifiers, alternative model end points, or local open-weight models as institutional policies and infrastructure evolve. However, this feasibility study did not demonstrate model substitution or formal head-to-head model comparisons. This manuscript reports a technical feasibility study rather than a formal evaluation. Future work will use this environment to conduct formal validation, comparisons with baseline classifiers and local open-weight models, turnaround times, clinician workflow impact, and staff satisfaction.

Limitations

This work has limitations. We did not conduct a formal accuracy study or randomized usability trial; our evaluation focused on feasibility, safety controls, and instrumentation. Deidentification quality, while strong in bench tests, requires formal validation against gold standards. While the sensitivity was strong, we continued to treat the messages as containing PHI after the deidentification. The UI was developed and tested in an environment within the institution's firewall, and nonsynthetic messages were only accessed by IRB-approved team members.

Deidentification was optimized for sensitivity to minimize PHI disclosure risk, accepting lower specificity as a deliberate trade-off. This level of overmasking, in which non-PHI terms, including some medication names, were redacted, may have reduced the semantic richness available to the LLM during preproduction testing and could have affected the quality of generated responses. The extent of this impact was not formally evaluated and represents a limitation of the current study. Medication and identifier coverage depends on biomedical NER supplemented by regex-based detection, curated medication lists, text normalization, and fuzzy matching. Although these steps improve detection of common variants and misspellings, broader use across clinical settings will require periodic updates for

new medications, local abbreviations, and institution-specific terminology. The pipeline was designed with HIPAA (Health Insurance Portability and Accountability Act) Safe Harbor identifier categories in mind, but it should not be interpreted as a certified Safe Harbor or Expert Determination implementation. In settings where data would leave the institutional security perimeter, formal validation of deidentification accuracy should be evaluated before using the workflow outside a similarly governed environment.

The UI also lacked automated safeguards, such as confidence thresholds or contradiction detection. Future work will require automated safeguards and explicit strategies for long-thread management, such as truncation, summarization, or controlled context-window selection, for use cases that require reasoning across full conversation histories. Token-based cost views are approximate and will shift with pricing and prompt designs. Finally, because the system deliberately avoids live EHR integration, we did not assess downstream operational effects such as turnaround times or staff satisfaction.

The UI does not yet allow the import of demographic data; thus, we were unable to assess differential performance across demographics. With the added feature of demographic data imports in future versions, this environment could enable assessment of differential performance across demographic characteristics, such as age, race, and ethnicity, to evaluate the presence of bias and equity considerations before deployment.

Despite these limitations, a secure UI-based sandbox is a pragmatic step toward responsible AI for patient messaging. It allows organizations to study real behaviors on real messages securely, identify failure modes early, and decide which use cases, if any, merit the cost and complexity of a full integration.

Conclusions

AI for patient portal messaging is advancing quickly, but its real impact on clinicians, patients, and care partners remains uncertain [20-22]. A secure sandbox with an accessible UI provides a practical way to test and refine AI use cases before they reach clinical workflows. Our implementation demonstrates end-to-end feasibility across the full pipeline: data ingestion, privacy preservation, configurable prompting, batch analysis, and transparent cost telemetry, all within a governed environment. The goal is not to assume benefit but to create the conditions under which benefit, risk, and fit can be studied with rigor.

Acknowledgments

The authors wish to acknowledge Dr Renee Blanding for her leadership of the internship program that led to this collaboration. The authors acknowledge using Claude (Anthropic) for grammar and style editing of the paper.

Funding

This work was supported by the following grants: National Institutes of Health (NIH) National Institute on Aging (NIA) grant R35AG072310, Consumer Health Information Technology to Engage and Support ADRD Caregivers: Research Program

to Address ADRD Implementation Milestone 13; the Hopkins Artificial Intelligence and Technology Collaboratory (AITC), NIA grant P30AG073104; and the Hopkins Economics of Alzheimer's Disease and Services (HEADS) Center (NIH NIA P30AG066587).

Data Availability

The source code, pinned requirements file, and synthetic example dataset used to test the user interface are available in a public GitHub repository [23]. The 6941-message corpus contains personal health information and is not publicly available due to institutional review board (IRB) restrictions. Researchers interested in accessing the dataset may submit a formal data sharing request to the corresponding author; any access would be subject to IRB approval and a data use agreement.

Authors' Contributions

Conceptualization: KG

Methodology: KG, TK, VB, BH, JW

Supervision: JW

Validation: TK, VB

Writing—original draft: KG, TK, VB

Writing—review and editing: KG, TK, VB, BH, JW

Conflicts of Interest

None declared.

Multimedia Appendix 1

Prompt templates.

[\[DOCX File \(Microsoft Word File\), 23 KB-Multimedia Appendix 1\]](#)

References

1. Steitz BD, Unertl KM, Levy MA. An analysis of electronic health record work to manage asynchronous clinical messages among breast cancer care teams. *Appl Clin Inform.* Aug 2021;12(4):877-887. [doi: [10.1055/s-0041-1735257](#)] [Medline: [34528233](#)]
2. Lieu TA, Altschuler A, Weiner JZ, et al. Primary care physicians' experiences with and strategies for managing electronic messages. *JAMA Netw Open.* Dec 2, 2019;2(12):e1918287. [doi: [10.1001/jamanetworkopen.2019.18287](#)] [Medline: [31880798](#)]
3. Crotty BH, Mostaghimi A, O'Brien J, Bajracharya A, Safran C, Landon BE. Prevalence and risk profile of unread messages to patients in a patient web portal. *Appl Clin Inform.* 2015;6(2):375-382. [doi: [10.4338/ACI-2015-01-CR-0006](#)] [Medline: [26171082](#)]
4. Borre ED, Nicholas MW. The disproportionate burden of electronic health record messages with image attachments in dermatology. *J Am Acad Dermatol.* Feb 2022;86(2):492-494. [doi: [10.1016/j.jaad.2021.09.026](#)] [Medline: [34555485](#)]
5. Garcia P, Ma SP, Shah S, et al. Artificial intelligence-generated draft replies to patient inbox messages. *JAMA Netw Open.* Mar 4, 2024;7(3):e243201. [doi: [10.1001/jamanetworkopen.2024.3201](#)] [Medline: [38506805](#)]
6. Hristidis V, Ruggiano N, Brown EL, Ganta SR, Stewart S. ChatGPT vs Google for queries related to dementia and other cognitive decline: comparison of results. *J Med Internet Res.* Jul 25, 2023;25:e48966. [doi: [10.2196/48966](#)] [Medline: [37490317](#)]
7. Yalamanchili A, Sengupta B, Song J, et al. Quality of large language model responses to radiation oncology patient care questions. *JAMA Netw Open.* Apr 1, 2024;7(4):e244630. [doi: [10.1001/jamanetworkopen.2024.4630](#)] [Medline: [38564215](#)]
8. Chiarelli G, Stephens A, Finati M, et al. Adequacy of prostate cancer prevention and screening recommendations provided by an artificial intelligence-powered large language model. *Int Urol Nephrol.* Aug 2024;56(8):2589-2595. [doi: [10.1007/s11255-024-04009-5](#)] [Medline: [38564079](#)]
9. Zack T, Lehman E, Suzgun M, et al. Assessing the potential of GPT-4 to perpetuate racial and gender biases in health care: a model evaluation study. *Lancet Digit Health.* Jan 2024;6(1):e12-e22. [doi: [10.1016/S2589-7500\(23\)00225-X](#)] [Medline: [38123252](#)]
10. Dergaa I, Fekih-Romdhane F, Hallit S, et al. ChatGPT is not ready yet for use in providing mental health assessment and interventions. *Front Psychiatry.* 2024;14:1277756. [doi: [10.3389/fpsyt.2023.1277756](#)] [Medline: [38239905](#)]
11. Antaki F, Touma S, Milad D, El-Khoury J, Duval R. Evaluating the performance of ChatGPT in ophthalmology: an analysis of its successes and shortcomings. *Ophthalmol Sci.* 2023;3(4):100324. [doi: [10.1016/j.xops.2023.100324](#)] [Medline: [37334036](#)]
12. Sendak MP, Ratliff W, Sarro D, et al. Real-world integration of a sepsis deep learning technology into routine clinical care: implementation study. *JMIR Med Inform.* Jul 15, 2020;8(7):e15182. [doi: [10.2196/15182](#)] [Medline: [32673244](#)]

13. Shortliffe EH, Sepúlveda MJ. Clinical decision support in the era of artificial intelligence. JAMA. Dec 4, 2018;320(21):2199-2200. [doi: [10.1001/jama.2018.17163](https://doi.org/10.1001/jama.2018.17163)] [Medline: [30398550](https://pubmed.ncbi.nlm.nih.gov/30398550/)]
14. Honnibal M, Montani I, Van Landeghem S, Boyd A. spaCy: industrial-strength natural language processing in Python. Zenodo. 2020. URL: <https://zenodo.org/records/10009823> [Accessed 2026-07-08]
15. Neumann M, King D, Beltagy I, Ammar W. ScispaCy: fast and robust models for biomedical natural language processing. In: Proceedings of the 18th BioNLP Workshop and Shared Task. Association for Computational Linguistics; 2019. [doi: [10.18653/v1/W19-5034](https://doi.org/10.18653/v1/W19-5034)]
16. Gleason KT, Wu MM, Wec A, et al. Use of the patient portal among older adults with diagnosed dementia and their care partners. Alzheimers Dement. Dec 2023;19(12):5663-5671. [doi: [10.1002/alz.13354](https://doi.org/10.1002/alz.13354)] [Medline: [37354066](https://pubmed.ncbi.nlm.nih.gov/37354066/)]
17. Gleason K, DeGennaro A. Evaluating ChatGPT in portal messaging for dementia care: identifying care partners and exploring acceptance. Innov Aging. Dec 1, 2025;9(Supplement_2):igaf122.380. [doi: [10.1093/geroni/igaf122.380](https://doi.org/10.1093/geroni/igaf122.380)] [Medline: [40255279](https://pubmed.ncbi.nlm.nih.gov/40255279/)]
18. Brown TB, Mann B, Ryder N, et al. Language models are few-shot learners. In: NIPS '20: Proceedings of the 34th International Conference on Neural Information Processing Systems. Curran Associates Inc; 2020:1877-1901. [doi: [10.5555/3495724.3495883](https://doi.org/10.5555/3495724.3495883)]
19. Liu S, McCoy AB, Wright AP, et al. Leveraging large language models for generating responses to patient messages-a subjective analysis. J Am Med Inform Assoc. May 20, 2024;31(6):1367-1379. [doi: [10.1093/jamia/ocae052](https://doi.org/10.1093/jamia/ocae052)] [Medline: [38497958](https://pubmed.ncbi.nlm.nih.gov/38497958/)]
20. Shimada SL, Petrakis BA, Rothendler JA, et al. An analysis of patient-provider secure messaging at two Veterans Health Administration medical centers: message content and resolution through secure messaging. J Am Med Inform Assoc. Sep 1, 2017;24(5):942-949. [doi: [10.1093/jamia/ocx021](https://doi.org/10.1093/jamia/ocx021)] [Medline: [28371896](https://pubmed.ncbi.nlm.nih.gov/28371896/)]
21. Semere W, Crossley S, Karter AJ, et al. Secure messaging with physicians by proxies for patients with diabetes: findings from the ECLIPSE study. J Gen Intern Med. Nov 2019;34(11):2490-2496. [doi: [10.1007/s11606-019-05259-1](https://doi.org/10.1007/s11606-019-05259-1)] [Medline: [31428986](https://pubmed.ncbi.nlm.nih.gov/31428986/)]
22. Brown W 3rd, Balyan R, Karter AJ, et al. Challenges and solutions to employing natural language processing and machine learning to measure patients' health literacy and physician writing complexity: the ECLIPSE study. J Biomed Inform. Jan 2021;113:103658. [doi: [10.1016/j.jbi.2020.103658](https://doi.org/10.1016/j.jbi.2020.103658)] [Medline: [33316421](https://pubmed.ncbi.nlm.nih.gov/33316421/)]
23. thom22/Patient-Portal-AI-Sandbox. GitHub. URL: <https://github.com/thom22/Patient-Portal-AI-Sandbox> [Accessed 2026-07-10]

Abbreviations

EHR: electronic health record
GPU: graphics processing unit
HIPAA: Health Insurance Portability and Accountability Act
IRB: institutional review board
LLM: large language model
NER: named entity recognition
NLP: natural language processing
PHI: personal health information
SDK: software development kit
UI: user interface

Edited by Andrew Coristine; peer-reviewed by Fabien Hagenimana, Joshua Odeniya, Katrina M Romagnoli, ABI ROLAND; submitted 02.Sep.2025; final revised version received 29.Jun.2026; accepted 30.Jun.2026; published 20.Jul.2026

Please cite as:

Gleason K, Kidu T, Babu V, Hasselfeld B, Wolff J
A Secure User Interface for Preclinical Evaluation of AI in Patient Portal Message Management: Tutorial
JMIR Med Inform 2026;14:e83216
URL: <https://medinform.jmir.org/2026/1/e83216>
doi: [10.2196/83216](https://doi.org/10.2196/83216)

© Kelly Gleason, Thomas Kidu, Vignesh Babu, Brian Hasselfeld, Jennifer Wolff. Originally published in JMIR Medical Informatics (<https://medinform.jmir.org>), 20.Jul.2026. This is an open-access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and

reproduction in any medium, provided the original work, first published in JMIR Medical Informatics, is properly cited. The complete bibliographic information, a link to the original publication on <https://medinform.jmir.org/>, as well as this copyright and license information must be included.