Engineering and Technology Journal e-ISSN: 2456-3358
Volume 10 Issue 11 November-2025, Page No.-7777-7782
DOI: 10.47191/etj/v10i11.10, I.F. – 8.482

AI-Driven Mobile Diagnostic System for Internal Diseases with Personalized Risk Guidance

Seungkyeom Lee[1], Seunghan Park[2], Junho Jang[3], Seungjae Lee[4*]
[1,2,3,4]Department of Computer Engineering, Sun Moon University, Korea

ABSTRACT: This paper presents a personalized AI-driven diagnostic system specialized in internal medicine. The system is designed to predict potential diseases from user-generated natural language symptom descriptions. It incorporates a dual-layer symptom extraction module that combines large language models (LLMs) with a structured NLP-based symptom mapping table, overcoming the limitations of conventional fixed-choice input methods and enabling user-friendly free-form symptom entry. Unlike traditional symptom checkers, our model leverages user-specific health profiles—including gender, age, chronic conditions, and current medications—as inputs for a multi-stage disease prediction pipeline, enhancing diagnostic accuracy through personalization. The system generates top disease candidates accompanied by a quantified risk score and provides condition-specific actionable health guidelines, aiding user decision-making. Implemented as a mobile application using React Native and Node.js, the system enables real-time symptom entry, analysis, and personalized feedback. Its integrated architecture, supported by a dedicated AI server, ensures usability, responsiveness, and scalability in real-world healthcare settings. Experimental validation confirms the system's effectiveness in delivering relevant and individualized insights for internal medical conditions, laying a robust foundation for developing intelligent, user-centered self-diagnosis tools in the digital health domain.

KEYWORDS: AI diagnosis, Symptom extraction, Large Language Models, NLP Mapping Table, Disease Prediction

1. INTRODUCTION

With the growing demand for self-assessment tools in healthcare, digital symptom checker applications have become increasingly popular for preliminary diagnosis [1, 3]. However, most existing platforms rely on fixed-choice symptom inputs, lack personalization, and offer limited clinical relevance [1, 3]. This is particularly problematic in internal medicine, where symptoms are often vague, overlapping, and difficult to interpret without context. These limitations can lead to challenges for patients in obtaining accurate information, potentially resulting in unnecessary medical visits or delayed appropriate treatment.

To address these limitations, we introduce a mobile-based AI diagnostic system tailored to internal medical conditions. This system allows users to describe their symptoms freely in natural language, which are then processed through a hybrid extraction pipeline that combines large language models (LLMs) and a curated NLP symptom mapping table. These extracted symptoms, along with user-specific health information—such as gender, age, chronic diseases, and current medications—serve as input to a multi-stage prediction model designed specifically for internal medicine.

Beyond predicting the most likely disease candidates, the system computes a personalized risk score by evaluating the user's profile and symptom severity. For each predicted condition, it provides actionable health guidelines, helping users gauge the urgency and take informed next steps. The entire system is implemented as a cross-platform mobile application using React Native and Node.js, supported by a dedicated AI server for real-time inference.

The key contributions of this work are:

- Free-text Symptom Input Interface: Supported by both LLM- and rule-based extraction techniques, enabling user-friendly symptom entry.
- Personalized Disease Prediction: Leveraging structured user health data to enhance diagnostic accuracy and personal relevance.
- Risk-guided Recommendation System: Providing personalized risk scores and health guidelines for practical next steps, assisting user decision-making.

Together, these features improve diagnostic precision, enhance user engagement within the context of internal medicine, and overcome the limitations of existing symptom checker tools.

2. SYSTEM ARCHITECTURE

The proposed system employs a modular architecture composed of four primary components: the mobile client, backend server, symptom extraction module, and AI diagnosis engine. Each module operates independently but is tightly integrated to enable real-time, personalized disease prediction for internal medicine. The overall architecture follows a client-server model, where each component is built with scalable and maintainable technologies.

The mobile client, developed with React Native and Expo, allows users to register, manage their health profiles, describe symptoms in natural language, and view prediction results. All user interactions are handled through a responsive interface, with local storage that synchronizes with the backend server. Client-side validation uses Zod schemas, and state management is handled using Zustand and React Query for efficient data synchronization and caching.

The backend server, built using Node.js and Express.js, functions as the central hub for communication and data processing. It handles user authentication (JWT-based), stores health records, and persists symptom records and prediction results in a PostgreSQL database using Prisma ORM. It also routes API requests to three specialized microservices: the LLM server for sentence refinement, the NLP server for symptom extraction, and the AI server for disease prediction.

The three core microservices are as follows:

- LLM server: Uses a lightweight large language model (e.g., Mistral) [9, 10] to refine user-entered free-text symptom descriptions into structured input. This transforms unstructured user input into a standardized format.
- NLP server: Applies rule-based mapping and keyword matching using a curated symptom mapping table [12, 13], extracting standardized symptom entities from both Korean and translated English texts.
- AI diagnosis engine (AI server): Integrates SBERT-based symptom embeddings [10, 11] with structured user data (age, gender, chronic diseases, medications) to perform a two-stage prediction [4, 8]. This prediction consists of:

- Coarse-level disease group classification: Classifying the input into broad disease categories (e.g., respiratory, cardiovascular, digestive).
- Fine-level disease ranking: Identifying and ranking the top three most probable diseases within the selected group.

Each output includes a risk score calculated based on the user's profile and recommended actions tailored to the user. This modular architecture supports high scalability, enables real-time inference, and ensures transparency and personalization in the diagnostic process, fostering user trust and engagement. The system workflow is illustrated in Figure1 1, showing the communication flow between the mobile client, server, and database during user authentication and profile management. Figure 2 visualizes the overall system architecture, demonstrating how user input seamlessly flows through each processing module.

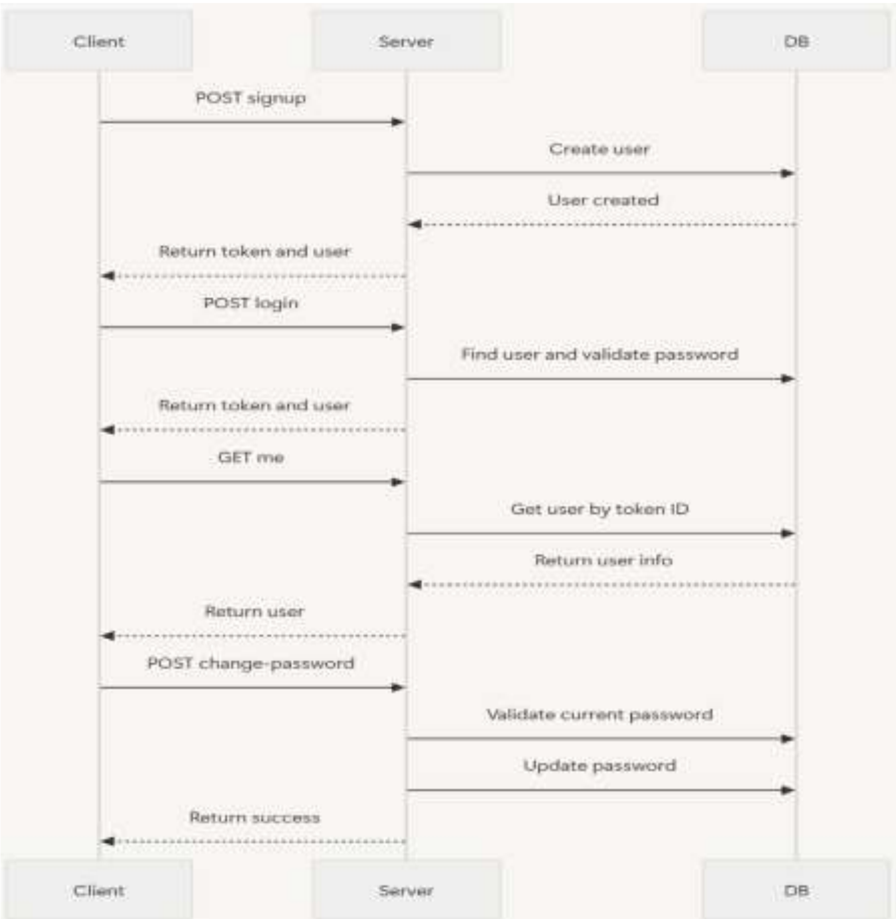

Figure 1: Sequence Diagram Illustrating Communication Among Mobile Client, Server, and Database for User Authentication and Profile Management

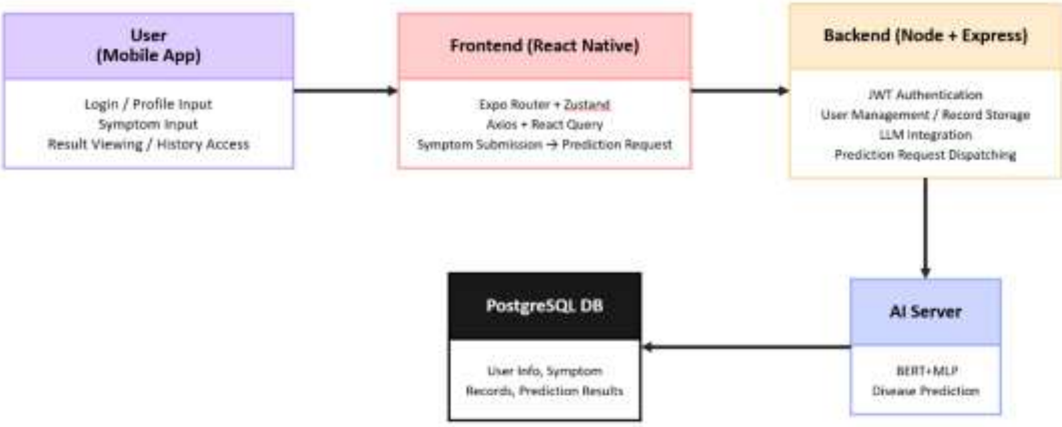

Figure 2: Overall System Architecture of the Personalized AI Diagnosis Application.

3. RESULTS IMPLEMENTATION

The proposed system has been developed as a cross-platform mobile application with a modular backend and dedicated AI processing servers. The overall architecture follows a client-server model, where each component is built with scalable and maintainable technologies suitable for real-time health analysis. To clearly represent the system's data model, we visualized the entity relationships among users, medications, and diseases, as shown in Fig. 3.

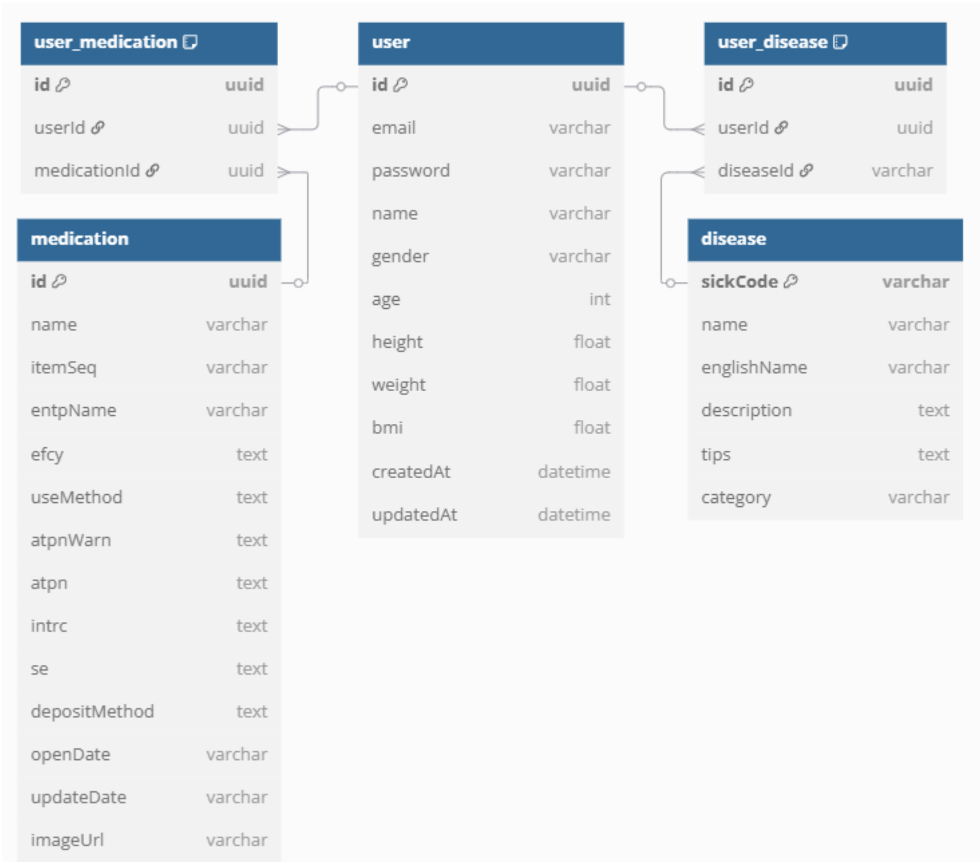

Figure 1 Entity relationship diagram showing user profile structure and its connections with chronic diseases and medications in the PostgreSQL database

The frontend is built using React Native with Expo, allowing seamless deployment on both Android and iOS. It includes user interfaces for registration, profile editing, symptom entry, and result viewing. All user input is validated on the client side using Zod schemas, and state management is handled using Zustand and React Query for efficient data synchronization and caching.

The backend server, developed with Node.js and Express.js, functions as the main API gateway. It handles

JWT-based authentication, manages user profiles, stores symptom records and prediction results in a PostgreSQL database using Prisma ORM, and routes API calls to three microservices: the LLM server for sentence refinement, the NLP server for symptom extraction, and the

AI server for disease prediction. To manage structured medical data, we designed a normalized schema using PostgreSQL and Prisma. Fig. 5 shows the full ERD structure, mapping users, symptoms, diseases, and prediction records. Fig. 6 focuses on the schema for symptom records and associated prediction outputs, designed for tracking user history and personalized feedback.

The LLM server, implemented using FastAPI, refines free-text symptom descriptions using the Mistral model hosted locally via Ollama. To support lightweight execution, the system also includes a rule-based symptom extraction module powered by a curated NLP mapping table. This hybrid approach improves performance in resource-constrained environments by ensuring consistent symptom extraction without relying solely on computationally expensive models. The AI server, also built with FastAPI, performs multi-stage predictions. It uses SBERT-based symptom embeddings along with user health profiles (age, gender, chronic diseases, medications) to (1) classify the symptom group (coarse prediction), and (2) identify the top-3 likely internal diseases (fine prediction). Each result includes a risk score and customized guideline. All module interactions are handled via RESTful APIs, and the system is designed to support local, lightweight execution during development. Figure 4 shows the full end-to-end sequence of how user requests (e.g., profile input, symptom submission, prediction retrieval) flow across the system modules

Figure 4 Sequence diagram showing the end-to-end flow of user requests across the system modules

To evaluate the effectiveness of our AI diagnosis system, we conducted a series of experiments using a curated dataset of symptom descriptions and their corresponding internal disease labels. The primary objective was to assess the system's ability to extract relevant symptoms from natural language and accurately predict internal medical conditions while incorporating user-specific health profiles. The evaluation followed a two-stage prediction pipeline.

In the first stage (coarse prediction), the system classifies the input into a broad disease category such as respiratory, cardiovascular, or digestive. In the second stage (fine prediction), a more specialized model identifies the top three most probable diseases within the selected group. These predictions are accompanied by a personalized risk score, which is computed using the user's age, gender, chronic conditions, and medications. Fig. 10 presents the full flow of prediction rendering, from symptom entry to diagnosis result visualization. It highlights how the system generates a ranked output along with the risk level and provides actionable health guidelines using the computed risk score.

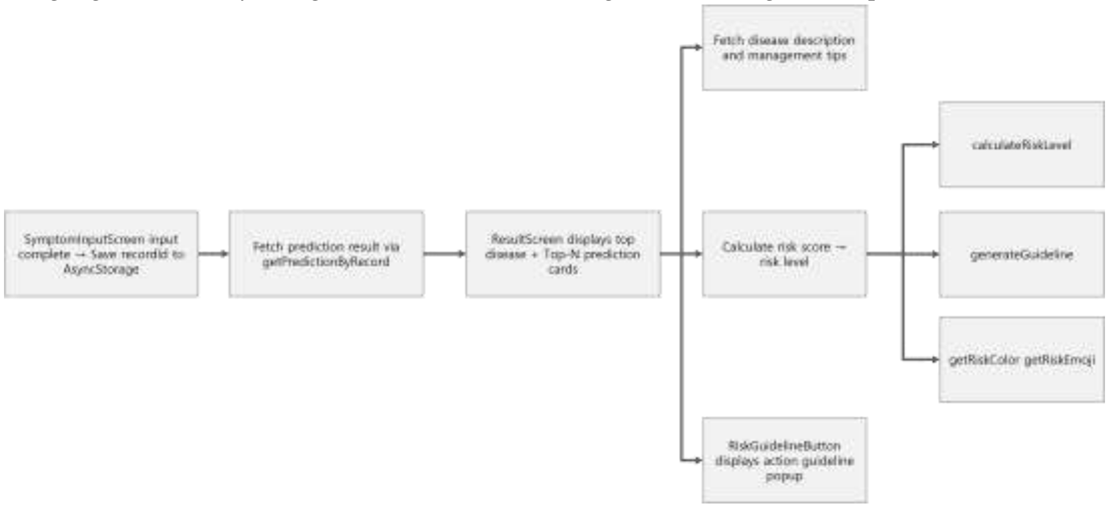

Figure 5 Flow of AI prediction result rendering: from symptom input to diagnosis result, including personalized risk score computation and guideline generation.

As a concrete example, a user entered the symptom sentence: "I feel bloated and nauseous after meals." Using the NLP symptom mapping table, the system extracted key symptoms such as nausea, indigestion, and queasiness. Given the user profile (female, 29 years old, 160 cm, 50 kg, BMI 19.53, history of acute gastritis, no medications), the system predicted Gastritis, Anemia, and Acute Pharyngitis as the top three conditions, with Gastritis ranked highest. The system computed a risk score of 2.92 and classified it as moderate, generating a recommendation to avoid irritating foods and seek medical advice if symptoms persist. This example, summarized in Table 1, demonstrates the system's ability to process natural language input, extract key symptoms, and deliver relevant, personalized medical insights.

Table 1. AI Prediction Example with User Profile and Natural Language Symptom Input.

Item	**Value**
Input Sentence	I feel bloated and nauseous after meals.
Extracted Symptoms	nausea, indigestion, queasiness
User Profile	Female, 29 years old, 160 cm, 50 kg (BMI 19.53)
Chronic Diseases	Acute Gastritis
Medications	None
Top-3 Predictions	1. Gastritis (0.885)2. Anemia (0.031)3. Acute Pharyngitis (0.030)
Final Result	Gastritis
Risk Score	2.92
Risk Level	Moderate
Recommendation	Avoid irritating food and seek medical advice if symptoms persist.

One of the key strengths of our system is its modular architecture. By decoupling the LLM server, NLP server, and AI prediction engine, the platform achieves scalability, maintainability, and the flexibility to replace or upgrade individual components independently. For example, the integration of lightweight NLP mapping logic as a fallback ensures robust symptom extraction in resource-constrained environments, which is especially relevant for mobile applications.

Another notable feature is the risk-guided feedback mechanism. Unlike conventional symptom checkers, our system interprets predicted diseases within the context of structured user profiles and assigns personalized risk levels. This enhances both interpretability and usability, aligning the system with real-world clinical decision-making practices. It also fosters greater user trust and engagement.

Nevertheless, the system has limitations. Its dependence on manually curated symptom mappings may limit recall for uncommon or novel expressions. In addition, the lack of large-scale real-world user data hinders the model's generalizability in edge cases. To address these challenges, future development should focus on implementing user feedback loops, such as incorporating confirmed diagnoses or user corrections, to continuously improve model performance and adaptability.

4. CONCLUSION

This study successfully introduced a mobile-based AI diagnosis system specifically tailored for internal medical conditions. By integrating natural language symptom input with structured personal health data, the system overcomes the limitations of traditional non-personalized, fixed-choice symptom input and enables more personalized and accurate disease prediction. Specifically, the combined use of large language model (LLM)-based and rule-based symptom extraction methods, along with a multi-stage coarse-to-fine prediction architecture, demonstrated robust performance in addressing the complexity of internal medicine.

The system's core contributions lie in its free-text symptom input, personalized disease prediction, and a risk-guided recommendation system for practical next steps. Through real-world simulation scenarios, we demonstrated the system's ability to extract meaningful symptoms from free-text inputs, generate ranked disease predictions, calculate personalized risk scores, and provide actionable health guidelines. The integrated architecture—featuring a dedicated AI server and real-time mobile interface—supports high scalability and real-time inference, while ensuring transparency and personalization in the diagnostic process, thereby fostering user trust and engagement. This lays a strong foundation for developing intelligent, user-centered self-diagnosis tools in the digital health domain.

While the current implementation shows promising performance, future work will focus on expanding the training dataset via collaborations with medical institutions, improving prediction accuracy with clinical feedback, and incorporating voice and image-based input for broader applicability.

ACKNOWLEDGMENTS

This research was supported by the MSIT(Ministry of Science ICT), Korea, under the National Program for Excellence in SW, supervised by the IITP(Institute of Information & Communications Technology Planning & Evaluation) in 2025"(No. 2024-0-00023).

REFERENCES

1. Y. You, R. Ma, and X. Gui, “User Experience of Symptom Checkers: A Systematic Review,” JMIR Hum. Factors, vol. 10, no. 1, p. e40578, 2023
2. K. Xu, W. Hou, Y. Cheng, J. Wang, and W. Li, “Medical Dialogue Generation via Dual Flow Modeling,” in Findings of the Association for Computational Linguistics: ACL 2023, 2023, pp. 6771–6784.
3. S. Semigran, J. Linder, A. Gidengil, and D. Mehrotra, “Evaluation of symptom checkers for self-diagnosis and triage: audit study,” BMJ, vol. 351, p. h3480, Jul. 2015, doi: 10.1136/bmj.h3480.
4. D. Nguyen, T. Tran, R. Wickramasinghe, and S. Venkatesh, “Deepr: A Convolutional Net for Medical Records,” IEEE J. Biomed. Health Inform., vol. 21, no. 1, pp. 22–30, Jan. 2017, doi: 10.1109/JBHI.2016.2570077.
5. Y. Pan, C. Wang, B. Hu, and M. Zhang, “A BERT-Based Generation Model to Transform Medical Texts to SQL Queries for Electronic Medical Records: Model Development and Validation,” JMIR Med. Inform., vol. 9, no. 12, p. e32698, 2021, doi: 10.2196/32698.
6. M. T. Ribeiro, S. Singh, and C. Guestrin, “Why Should I Trust You?: Explaining the Predictions of Any Classifier,” in Proc. ACM SIGKDD, San Francisco, CA, USA, 2016, pp. 1135–1144, doi: 10.1145/2939672.2939760.
7. A. N. Jagannatha and H. Yu, “Bidirectional RNN for Medical Event Detection in Electronic Health Records,” in Proc. NAACL-HLT, San Diego, CA, USA, 2016, pp. 473–482, doi: 10.18653/v1/N16-1055.
8. E. Choi, M. T. Bahadori, J. Sun, J. Kulas, A. Schuetz, and W. F. Stewart, “RETAIN: An Interpretable Predictive Model for Healthcare using Reverse Time Attention Mechanism,” in Proc. NIPS, Barcelona, Spain, 2016, pp. 3504–3512.
9. [9] T. Wolf et al., “Transformers: State-of-the-Art Natural Language Processing,” in Proc. EMNLP, Online, 2020, pp. 38–45, doi: 10.18653/v1/2020.emnlp-demos.4.
10. Y. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in Proc. NAACL-HLT, Minneapolis, MN, USA, 2019, pp. 4171–4186, doi: 10.18653/v1/N19-1423
11. E. Alsentzer, J. R. Murphy, W. Boag, et al., “Publicly Available Clinical BERT Embeddings,” arXiv preprint, arXiv:1904.03323, Apr. 2019.
12. Korean Ministry of Health and Welfare, “Disease Information Service,” Data.go.kr. Accessed: Oct. 7, 2025. [Online]. Available: https://www.data.go.kr/tcs/dss/selectApiDataDetailView.do
13. Health Insurance Review and Assessment Service (HIRA), “HIRA Disease Classification by Symptom,” Data.go.kr. Accessed: Oct. 7, 2025. [Online]. Available: https://www.data.go.kr/tcs/dss/selectApiDataDetailView.do?publicDataPk=15075057