A CONTEXT-AWARE MOBILE APP TO SUPPORT EARLY DISEASE DETECTION AND EDUCATION USING GPT-4 AND VISUAL SYMPTOM SURVEYS

Qiaoman Cai [1], Qiaoqian Cai [1], Rodrigo Onate [2]

[1] Crean Lutheran High School, 12500 Sand Canyon Ave, Irvine, CA 92618
[2] California State Polytechnic University, Pomona, CA, 91768

ABSTRACT

This research project addresses the growing gap in accessible healthcare, particularly for women in underserved communities [1]. Rising healthcare costs and delayed diagnosis contribute to higher mortality, particularly in low-income areas. To address this, we developed CareBridge Health, a mobile app that uses AI (GPT-4) to simulate medical reasoning based on structured user input [2]. The app features a body diagram for symptom selection, guided surveys for pain description, and access to disease education through multimedia content. The system is divided into three components: the Ask Page for diagnosis, the Learn Page for information retrieval, and the History Page for recordkeeping. Through two experiments, we identified that both symptom clarity and AI consistency can affect diagnosis quality [3]. Despite these limitations, the app provides a scalable, low-cost way to raise health awareness and guide users toward early symptom identification. With further refinement and medical validation, CareBridge Health could become a valuable tool for community-level health empowerment.

KEYWORDS

Artificial Intelligence in Healthcare, Symptom Checker App, GPT-4 Diagnosis Assistance, Women's Health Accessibility, Mobile Health Technology

1. INTRODUCTION

The high cost of general healthcare is a major problem for women's health, especially those with high financial burdens [4].

Many women give up on diagnosis due to the lack of affordable treatment options, which causes delays in the diagnosis of early symptoms, worsening treatment outcomes, and eventually threatens their lives.

Research comparing reasons for impeding treatment concludes that women are 31% more likely to skip medical healthcare than men due to the high cost. And 21% of women refuse to go for an early diagnosis due to the unaffordability(Gebreyes et al.).

Socioeconomic inequality problems regarding health are prevalent and continuously exacerbating around the world as women, especially in low-income communities face challenges in access to healthcare and proper diagnosis. A report from the National Institutes of Health states that

David C. Wyld et al. (Eds): NLCAI, AIFU, CCSEA, BIoT, SEA, SIPRO, BDML, CLOUD – 2025
pp. 93-103, 2025. CS & IT - CSCP 2025 DOI: 10.5121/csit.2025.151007

women with income under the poverty level have a higher mortality rate due to breast cancer (1.44 times) than women with income much above the poverty level (Sprague et al.). Women's healthcare continues to increase and If the poverty problems keep hindering women from accessing proper healthcare, more women will suffer from the diseases and lead to more tragic deaths.

The first methodology, reviewed by Wallace et al., analyzed 48 digital symptom checkers and found that most had limited diagnostic and triage accuracy. Our app improves on this by using a generative AI model instead of rule-based logic. The second study, by Kusunose et al., tested GPT-4's diagnostic performance using structured clinical data and found it to outperform traditional AI in identifying final diagnoses. Our app adapts this approach for everyday users by guiding symptom input through surveys and prompts. The third methodology, from Shea et al., demonstrated GPT-4's ability to analyze complex patient histories with delayed diagnoses. While this was in a clinical setting, our app brings similar reasoning power into a consumer-friendly experience. Compared to all three, our app introduces better accessibility, body-based navigation, and survey-assisted structuring to ensure laypeople can engage with AI meaningfully, even outside of professional contexts.

This app contains tools for self-examination of body symptoms tools like OpenAI, which allows women to identify potential health issues as early as possible, thus increasing the chaxes for success of early prevention or medical treatment [5]. Unlike the situation where people need to drive to a medical clinic to see a doctor, the … app on the phone is largely accessible for most people in the U.S, regardless of income level. Compared to traditional in-person medical healthcare, this app would increase efficiency and reduce costs and time taken on the way to the clinic, thus increasing the range of its impact.

This app provides educational content including images, videos, and articles on various women's health diseases, which help raise awareness of the importance of early detection and basic knowledge of common forms of disease.

This app can also help women find affordable healthcare institutions when necessary, which addresses the significant barrier for low-income communities to access healthcare.

This research included two key experiments to evaluate blind spots in the CareBridge Health app. The first experiment tested whether users could accurately report symptoms through the structured survey. Ten volunteers completed a simulated diagnosis using a guided pain input form. The results showed that while ease of use was rated highly, the alignment between dropdown options and the users' intended meanings was less consistent, suggesting a need for clearer terminology and examples. The second experiment examined the consistency of GPT-4-generated diagnoses from varying symptom descriptions that referenced the same underlying illness. Although two variations returned consistent top diagnoses, the third input generated a different but related condition, indicating that phrasing can affect AI output. These experiments demonstrate that both the user input system and prompt engineering must be refined to reduce diagnostic variability and improve the reliability of symptom interpretation in future updates of the app.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Inaccurate Diagnosis Results

The performance of this application is significantly affected by the artificial intelligence model we are using. AI lacks the clinical experience that matches professional doctors, which fails to accurately analyze the user's symptom input and thus results in inaccurate diagnosis results. Therefore, AI must be trained in a way that mimics doctors who identify and differentiate different diseases and offer reasonable diagnosis results. To address this issue, we developed explicit prompts that instruct AI to behave and think like a doctor. We would also guide AI to incorporate the user's diagnosis history to determine and rule out the possibility of some disease.

2.2. Analyze a Wide Range of Symptoms

The artificial intelligence modeling lacks training and precise and available data, this may weaken the AI model's ability to analyze a wide range of symptoms and thus decrease the credibility of the results. This model needs a large dataset containing historical patient symptoms, diagnosis knowledge, and medical lab results to function properly. However, these types of datasets often are restricted and protected by authentic institutions. In order to mitigate AI's weaknesses, we planed to establish a robust training dataset by using public medical databases, and we would refine the model by training at least 300 diverse healthcare cases for common major diseases like cancer to increase its accuracy [6].

2.3. Symptoms Description

Artificial intelligence model may fail to various analyze user's input information due to the various way user described their symptoms. Users may struggle to articulate what their symptoms look like as they may not know specific medical terminology and also ignore important details, which could interfere with AI from generating an accurate analysis. To address this, we design a survey that ensures AI to get clear and detailed information. Instead of letting user to solely describing, the app allow users to tab on a body and then fill out a survey that guide them to conduct an useful information for AI to analyze.

3. SOLUTION

The CareBridge Health application is a mobile-based health support tool designed to enhance early detection, symptom tracking, and medical education for women. The application architecture consists of three primary components that work in a seamless flow: the Ask Page (symptom diagnosis interface), the Learn Page (disease education resource), and the History Page (diagnosis logbook).

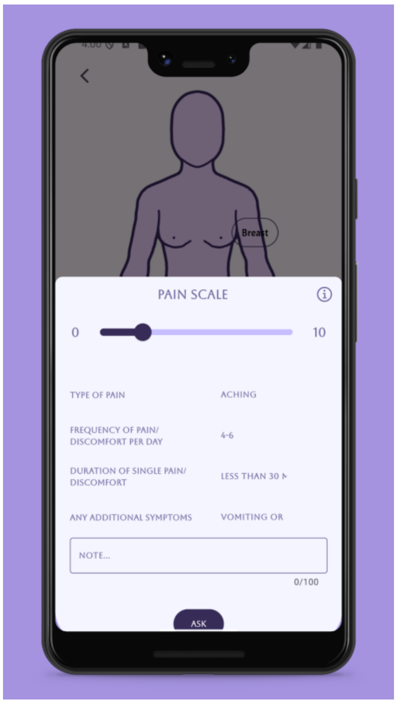
When the app launches, users arrive at the Home Page (home_screen.dart), which features navigation to the app's key functionalities. From there, users can either conduct a diagnosis, search for information, or review previous results.

The Ask Page (ask_screen.dart) is designed for guided self-diagnosis. It displays a tap-enabled body diagram, allowing users to select a specific body part they are experiencing discomfort in. When a body part is tapped, the app launches a survey popup (popup_widget.dart) to collect structured data such as pain level, type, duration, frequency, and any additional symptoms. Once submitted, the survey data is sent to OpenAI's ChatGPT API to generate three potential diagnoses. These results are displayed on the Result Page (result_screen.dart) along with a mortality risk gauge and personalized recommendations. The results are also saved locally using SharedPreferences for future reference.

The Learn Page (learn_screen.dart) provides educational support by allowing users to search for disease names. The app retrieves images, videos, and news articles related to the queried illness through SerpAPI [7]. This content is intended to improve awareness and understanding of symptoms and treatments.

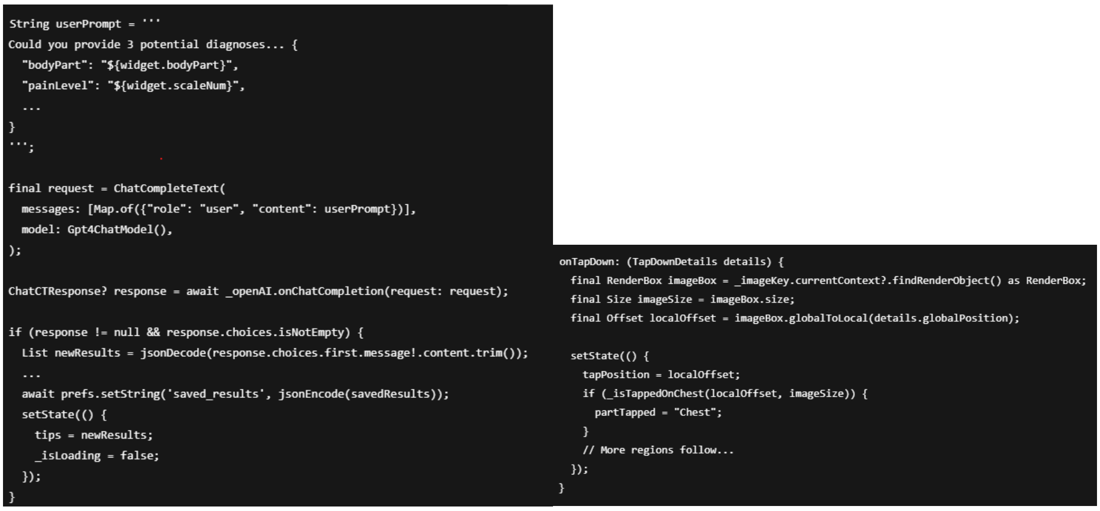
Finally, the History Page (history_page.dart) maintains a persistent log of all user diagnosis sessions. It organizes results by timestamp, and allows users to view previous symptom submissions, diagnoses, mortality scores, and AI recommendations in an expandable card layout. All components are tied together through a minimalist navigation bar and a shared color palette and typography (GoogleFonts.aboreto) to create a consistent and accessible user experience.

The Ask Page is the core of the app's diagnostic system. It enables users to tap on a body diagram to select a specific region of discomfort. This interaction triggers a symptom survey popup. The page uses Flutter's gesture detection and custom hitbox logic to identify the tapped body area. Once data is submitted, it is sent to OpenAI's GPT-4 model for analysis. This component heavily relies on Natural Language Processing (NLP) to simulate physician-like reasoning based on structured symptom data [8]. It uses chat_gpt_sdk to send a prompt and receive a list of possible diagnoses with descriptions, mortality risk, and recommendations.



TYPE OF PAIN	ACHING
FREQUENCY OF PAIN/ DISCOMFORT PER DAY	4-6
DURATION OF SINGLE PAIN/ DISCOMFORT	LESS THAN 30 M
ANY ADDITIONAL SYMPTOMS	VOMITING OR

Figure 1. Screenshot of pain scale



```
String userPrompt = '''
Could you provide 3 potential diagnoses... {
  "bodyPart": "${widget.bodyPart}",
  "painLevel": "${widget.scaleNum}",
  ...
}
''';

final request = ChatCompleteText(
  messages: [Map.of({"role": "user", "content": userPrompt})],
  model: Gpt4ChatModel(),
);

ChatCTResponse? response = await _openAI.onChatCompletion(request: request);

if (response != null && response.choices.isNotEmpty) {
  List newResults = jsonDecode(response.choices.first.message!.content.trim());
  ...
  await prefs.setString('saved_results', jsonEncode(savedResults));
  setState(() {
    tips = newResults;
    _isLoading = false;
  });
}
```

```
onTapDown: (TapDownDetails details) {
  final RenderBox imageBox = _imageKey.currentContext?.findRenderObject() as RenderBox;
  final Size imageSize = imageBox.size;
  final Offset localOffset = imageBox.globalToLocal(details.globalPosition);

  setState(() {
    tapPosition = localOffset;
    if (_isTappedOnChest(localOffset, imageSize)) {
      partTapped = "Chest";
    }
    // More regions follow...
  });
}
```

Figure 2. Screenshot of code 1

The symptom diagnosis component works by combining Flutter's gesture system and AI-powered inference. When a user taps on the human body image in AskPage, the tap is captured by onTapDown. The app translates the screen coordinates into local coordinates and then compares the location against predefined body part boundaries using helper functions like _isTappedOnChest.

Once a part is selected, the app displays a floating button labeled with that body part. Pressing the button opens a structured survey from PopUpWidget. This widget collects inputs such as pain level (via a slider), pain type, frequency, and duration (via dropdowns), and a freeform note field. When the user submits the survey, this data is packaged into a prompt and sent to OpenAI's GPT-4 using the chat_gpt_sdk. The ChatCompleteText call formats the prompt and requests a structured JSON response with diagnoses. Once received, the response is parsed and shown on the ResultPage. It includes:

A disease name
A diagnosis description
A mortality score (1 to 10)
A recommendation for next steps

The diagnosis results are stored locally using SharedPreferences, allowing future access via the History page.

This component blends gesture detection, form input, NLP prompt engineering, and data persistence into a seamless diagnostic flow.

The Learn Page is the educational component of the app, designed to provide users with rich, multimedia information about diseases. It uses an external API service (SerpAPI) to fetch related images, YouTube videos, and news articles based on the user's query. The system uses keyword filtering to ensure the search is health-related, and organizes the content into scrollable sections for interactive exploration. The Learn Page enhances awareness and helps users understand symptoms, treatments, and prevention through trustworthy online resources.

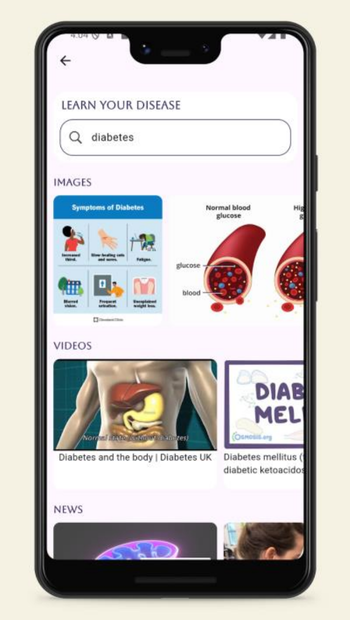
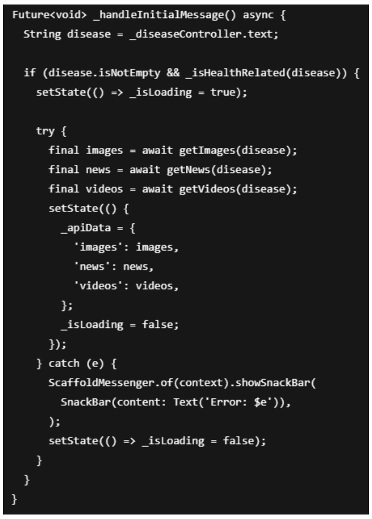


Figure 3. Screenshot of the app page



```
Future<void> _handleInitialMessage() async {
  String disease = _diseaseController.text;

  if (disease.isNotEmpty && _isHealthRelated(disease)) {
    setState(() => _isLoading = true);

    try {
      final images = await getImages(disease);
      final news = await getNews(disease);
      final videos = await getVideos(disease);
      setState(() {
        _apiData = {
          'images': images,
          'news': news,
          'videos': videos,
        };
        _isLoading = false;
      });
    } catch (e) {
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text('Error: $e')),
      );
      setState(() => _isLoading = false);
    }
  }
}
```

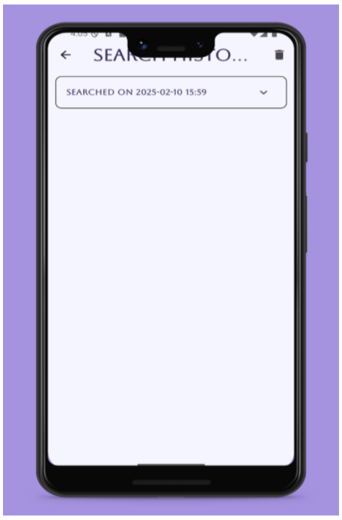
Figure 4. Screenshot of code 2

This method, _handleInitialMessage, is triggered when the user submits a disease name in the search bar. It first checks whether the input string is non-empty and passes a health-related keyword filter via _isHealthRelated. If valid, it sets the loading state to true to show a progress indicator.

Then it makes three asynchronous API calls: getImages, getNews, and getVideos, all of which use SerpAPI to retrieve online content based on the user's query. The image results come from Google Images, videos from YouTube, and articles from Google News. The responses are parsed into usable data formats: a list of image URLs, a list of news metadata (title, link, thumbnail), and a list of video information.

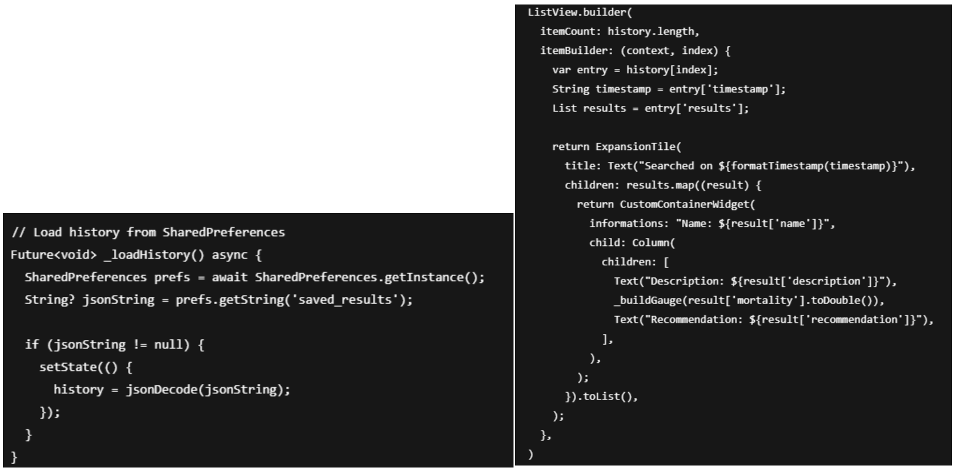
Once the data is fetched, the UI is updated with _apiData, which triggers the rendering of each content section in the Learn Page: horizontal scroll views of images, video previews, and news articles. If the query fails or the API throws an error, an error message is shown via a SnackBar. This component emphasizes API integration, async state management, and user education through curated multimedia content.

The History Page component is responsible for tracking and displaying users' past diagnosis results. It uses SharedPreferences to persist data locally on the device. Each diagnosis is stored as a JSON object containing a timestamp and a list of results [9]. This allows users to revisit their symptom history, view AI-generated diagnoses, mortality scores, and corresponding recommendations. The layout uses ExpansionTile widgets to collapse and expand entries for easy navigation. A radial gauge visualizes the severity (mortality) of each diagnosis, helping users intuitively interpret their health risk over time.



SEARCHED ON 2025-02-10 15:59

Figure 5. Screenshot of search function



```
// Load history from SharedPreferences
Future<void> _loadHistory() async {
  SharedPreferences prefs = await SharedPreferences.getInstance();
  String? jsonString = prefs.getString('saved_results');

  if (jsonString != null) {
    setState(() {
      history = jsonDecode(jsonString);
    });
  }
}
```

```
ListView.builder(
  itemCount: history.length,
  itemBuilder: (context, index) {
    var entry = history[index];
    String timestamp = entry['timestamp'];
    List results = entry['results'];

    return ExpansionTile(
      title: Text("Searched on ${formatTimestamp(timestamp)}"),
      children: results.map((result) {
        return CustomContainerWidget(
          informations: "Name: ${result['name']}",
          child: Column(
            children: [
              Text("Description: ${result['description']}"),
              _buildGauge(result['mortality'].toDouble()),
              Text("Recommendation: ${result['recommendation']}"),
            ],
          ),
        );
      }).toList(),
    );
  },
)
```

Figure 6. Screenshot of code 3

The History Page begins by calling _loadHistory() during initState(). This method fetches a stringified JSON list called saved_results from local storage using SharedPreferences. If available, the string is decoded into a list and assigned to the history state variable.

Inside the widget’s build() method, a ListView.builder dynamically creates a scrollable list of past diagnoses. Each diagnosis entry is wrapped in a Card and ExpansionTile widget, which shows the search timestamp as its title. Expanding the tile reveals one or more AI-generated diagnoses.

Each result is passed to a CustomContainerWidget, which displays the disease name, description, and a radial gauge created with syncfusion_flutter_gauges to visually represent the AI-estimated mortality risk (1–10 scale). The page also includes a "Delete" icon in the AppBar that allows users to clear all stored results.

This component enables users to track past symptoms and compare changes over time. It demonstrates local storage handling, state restoration, custom widgets, and intuitive visual feedback mechanisms—enhancing the long-term usability of the app [10].

4. Experiment

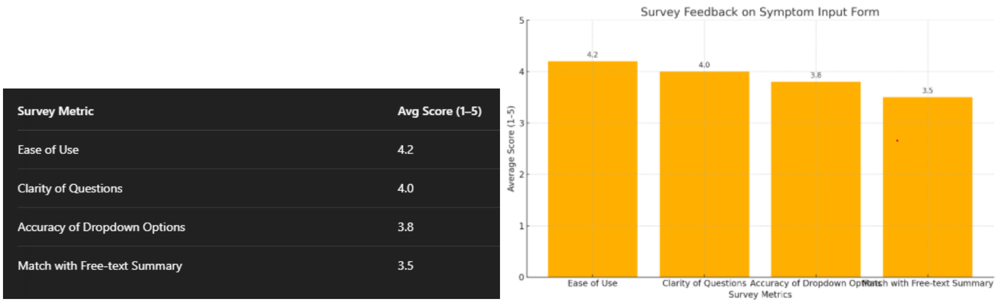
4.1. Experiment 1

One potential blind spot in the app is whether users can accurately and consistently describe their symptoms using the provided survey. The AI model relies on structured input from the user (pain level, type, frequency, etc.) to generate diagnoses. If users misunderstand the questions or provide inconsistent answers, the AI’s output may be misleading. Ensuring survey usability is essential for improving diagnostic accuracy.

To test the usability and clarity of the symptom survey, we conducted an experiment with 10 volunteer users. Each participant was asked to simulate having a common symptom (e.g., abdominal pain or headache) and use the app’s body diagram + popup survey to submit their symptoms.

We then asked participants to fill out a follow-up feedback form evaluating:
How confident they felt using the sliders and dropdowns
Whether the choices matched what they wanted to express
Whether any questions were unclear or confusing

In addition, we cross-validated their survey answers with actual symptom descriptions written in free text, to see how consistent the structured input was with their original thoughts.



Survey Metric	Avg Score (1–5)
Ease of Use	4.2
Clarity of Questions	4.0
Accuracy of Dropdown Options	3.8
Match with Free-text Summary	3.5

Figure 7. Figure of experiment 1

From the feedback and cross-validation, the average Ease of Use score was 4.2 out of 5, indicating that most users found the interface intuitive. However, the match between dropdown selections and their actual symptoms scored lower (3.5), suggesting a gap between user perception and the available structured input options.

Some participants mentioned that terms like "Cramping" or "Piercing" were confusing, and others were unsure how to rate their pain duration. These inconsistencies may cause the AI to misinterpret the input and generate incorrect diagnoses.

We observed that responses were more accurate when users had a clear understanding of what the choices meant, highlighting the importance of contextual tooltips or visual aids. This experiment shows that refining survey wording and adding brief definitions or examples could improve input accuracy and help bridge the gap between structured data and real human experience.

4.2. Experiment 2

Another critical blind spot is whether the AI-generated diagnosis varies significantly with slightly different symptom descriptions. If small input variations result in vastly different outputs, the model may lack reliability. We must ensure that the AI remains consistent and medically sound, even when users describe symptoms slightly differently.

To test the AI's diagnostic consistency, we conducted a controlled input experiment. We created three sets of symptom inputs that described the same medical condition — urinary tract infection (UTI) — but with slight variations in phrasing or pain descriptions. Each set was submitted using the app's survey system and processed via the GPT-based diagnosis pipeline.
We then analyzed:

How consistent the AI's three suggested diagnoses were
Whether the core disease (UTI) appeared in all outputs
How mortality scores and recommendations compared

This test allowed us to evaluate the model's resilience to input variation and its sensitivity to word choices within structured symptom data.

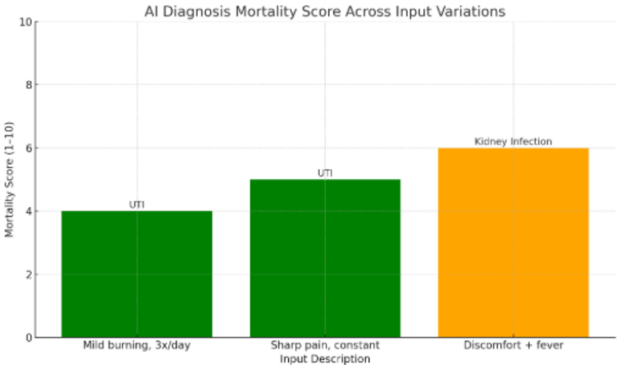


Figure 8. Figure of experiment 2

The AI correctly identified UTI as the top diagnosis in two of the three input variations, demonstrating moderate consistency. However, in the third variation (Input C), the top diagnosis was kidney infection, which is still within the same medical family but represents a more severe condition.

The mortality scores increased slightly with symptom severity (burning < sharp < fever), indicating the model's ability to adapt its recommendations based on symptom intensity. This behavior aligns with how a physician might escalate concern based on specific symptoms.

However, this experiment also highlights a subtle risk: users with overlapping symptoms may receive varying diagnoses that could cause confusion or anxiety. The AI should ideally clarify the potential relationship between diseases (e.g., UTI evolving into a kidney infection) rather than presenting them as completely distinct.

To improve reliability, we could refine the prompt to ask for overlapping or related conditions and better contextualize risk levels in future updates.

5. Related Work

Our app’s diagnostic approach differs from conventional symptom checkers by leveraging GPT-4 to interpret structured symptom inputs via prompt engineering, rather than relying solely on rule-based algorithms. In a comprehensive review by Wallace et al. (2022), 48 digital and online symptom checkers were evaluated, revealing that diagnostic accuracy for top-1 conditions averaged just 37.7%, and triage accuracy was similarly limited. These tools often lacked nuance in interpreting varied symptom presentations and failed to account for patient context [11]. In contrast, our methodology combines interactive symptom input, via a body diagram and structured survey, with generative language modeling to simulate physician-style reasoning. This approach allows for more contextual and human-like interpretation of pain characteristics and additional symptoms. While our app is not a certified diagnostic tool and is intended for educational use only, its architecture introduces a more adaptive and explainable interface that addresses usability and personalization gaps identified in existing tools.

Recent research has shown that large language models like ChatGPT-4 demonstrate strong diagnostic reasoning capabilities, particularly when given structured inputs. In a 2023 study, Kusunose et al. evaluated ChatGPT-4's ability to identify final diagnoses from clinical differential diagnosis lists and found that it correctly selected the final diagnosis 71% of the time, surpassing traditional search algorithms [12]. This supports our app's use of GPT-4 to simulate physician-level interpretation of structured symptom data. Unlike static decision-tree systems, GPT-4 dynamically weighs variables such as pain type, frequency, and associated symptoms to produce contextually accurate suggestions. While Kusunose’s study used curated medical cases, our app adapts that methodology to a consumer health context, simplifying user inputs into structured formats while still leveraging GPT-4’s reasoning power. This alignment shows how AI-based tools can bridge the gap between expert-level medical judgment and layperson accessibility when supported by proper prompt engineering and interface design.

Recent studies have highlighted the diagnostic capabilities of GPT-4, demonstrating its potential in clinical settings [13]. For instance, a study published in NPJ Digital Medicine found that GPT-4 achieved a top-6 diagnostic accuracy of 61.1% on challenging internal medicine cases, surpassing the performance of internal medicine residents and faculty physicians. Similarly, research in JAMA Network Open assessed GPT-4's ability to analyze medical records of patients with delayed diagnoses, revealing that GPT-4 could identify the correct diagnosis in complex cases, suggesting its utility in supporting clinical decision-making. These findings underscore GPT-4's potential to enhance diagnostic accuracy when integrated thoughtfully into healthcare applications.

6. Conclusions

While the CareBridge Health app demonstrates promise in enhancing early disease awareness and accessibility, several limitations still exist. Most notably, the AI model is not FDA-approved and should not be used as a substitute for professional diagnosis [14]. Its responses depend heavily on user inputs, which may vary in accuracy due to differences in symptom interpretation or medical understanding. Additionally, the app currently lacks multilingual support and accessibility features for visually impaired users. Inaccurate self-reporting may also skew AI responses, especially when users are unsure of their symptoms. To improve, future versions could integrate AI-powered voice assistants for hands-free usage and symptom description, include guided illustrations for pain types, and provide clearer explanations for survey options. Further collaboration with healthcare professionals could also help validate the AI's responses and ensure medical accuracy. Expanding the dataset with diverse demographic and symptom profiles would enhance diagnostic relevance and inclusivity across socioeconomic backgrounds.

CareBridge Health represents a meaningful step toward democratizing early health assessment tools through AI and mobile technology [15]. While limitations remain, its ability to educate, guide, and record user health data in underserved populations marks a scalable solution that could improve long-term outcomes if developed and validated further with medical institutions.

REFERENCES

[1] Mouttham, Alain, et al. "Interoperable support for collaborative, mobile, and accessible health care." Information systems frontiers 14 (2012): 73-85.

[2] Lee, Peter, Carey Goldberg, and Isaac Kohane. The AI revolution in medicine: GPT-4 and beyond. Pearson, 2023.

[3] Hoang, Yen Nhi, et al. "Consistency and accuracy of artificial intelligence for providing nutritional information." JAMA network open 6.12 (2023): e2350367-e2350367.

[4] Heng, Audrey, et al. "Transgender peoples' experiences and perspectives about general healthcare: A systematic review." International Journal of Transgenderism 19.4 (2018): 359-378.

[5] Roumeliotis, Konstantinos I., and Nikolaos D. Tselikas. "Chatgpt and open-ai models: A preliminary review." Future Internet 15.6 (2023): 192.

[6] Noguerol, Teodoro Martín, et al. "Strengths, weaknesses, opportunities, and threats analysis of artificial intelligence and machine learning applications in radiology." Journal of the American College of Radiology 16.9 (2019): 1239-1247.

[7] Valavandan, Ramamurthy, et al. "Digitalization of Hindu Temples in India in Google Cloud and SerpAPI Automation in Python."

[8] Chowdhary, KR1442, and K. R. Chowdhary. "Natural language processing." Fundamentals of artificial intelligence (2020): 603-649.

[9] Bassett, Lindsay. Introduction to JavaScript object notation: a to-the-point guide to JSON. " O'Reilly Media, Inc.", 2015.

[10] Prodanovic, Dusan. "Data handling and storage." Data Requirements for Integrated Urban Water Management (2008): 127-138.

[11] Wallace, William, et al. "The diagnostic and triage accuracy of digital and online symptom checker tools: a systematic review." NPJ digital medicine 5.1 (2022): 118.

[12] Hirosawa, Takanobu, et al. "Evaluating ChatGPT-4's accuracy in identifying final diagnoses within differential diagnoses compared with those of physicians: experimental study for diagnostic cases." JMIR Formative Research 8 (2024): e59267.

[13] Shea, Yat-Fung, et al. "Use of GPT-4 to analyze medical records of patients with extensive investigations and delayed diagnosis." JAMA Network Open 6.8 (2023): e2325000-e2325000.

[14] Kinch, Michael S., et al. "An overview of FDA-approved new molecular entities: 1827–2013." Drug discovery today 19.8 (2014): 1033-1039.

[15] Otani, Tomoyuki, et al. "Application of AI to mobile network operation." ITU Journal: ICT Discoveries, Special Issue 1 (2017): 1-7.