MSc Interaction Technology

Final Project

LLM-Based Conversational Interfaces as Interpretation Layers in Digital Twin Systems: Enhancing Patient Understanding of Heart Rate Data

Biaani de Belen Cabrera Benitez

Thesis Committee:
Dr. Birna van Riemsdijk
Dr. Mariët Theune
Dr. João Rebelo Moreira

September, 2025

Faculty of Electrical Engineering,
Mathematics and Computer Science,
University of Twente

UNIVERSITY OF TWENTE.

Contents

5. Prototype Development

6. User Study

Abstract

Digital Twin (DT) systems offer promising opportunities in healthcare by enabling continuous monitoring and personalized analysis of physiological data. Yet, for patients, the outputs of these systems often remain difficult to interpret due to technical complexity, lack of contextualization, and limited transparency. This lack of interpretability can undermine patient engagement, trust, and ultimately the effectiveness of DT systems in supporting self-management and communication with care providers.

This thesis explores how Large Language Model (LLM)-based conversational interfaces can function as interpretation layers within DT architectures, translating structured physiological data into understandable, patient-facing insights. Focusing on heart rate as a representative signal, we investigate how Retrieval-Augmented Generation (RAG) techniques can enhance the clarity, contextual relevance, and trustworthiness of responses in patient-facing interactions. To do so, we developed a local prototype that combines rule-based heart rate categorization with a retrieval-enabled LLM (LLaMA 3.2), operating on simulated datasets to preserve privacy. The system generates contextual, comprehensible answers without offering clinical advice, using hourly summaries of heart rate logs as input.

A mixed-methods evaluation involving nine participants was conducted to assess both technical performance and user experience. Technical accuracy was measured using RAGAs metrics, including faithfulness, answer relevance, and context recall, while user experience was evaluated through semi-structured interviews and standardized usability questionnaires. Results indicate that the system produced relevant, understandable, and empathetic responses, helping participants identify trends in their simulated heart data. However, broader queries led to decreased response consistency, and the use of simulated rather than real patient data limited the generalizability of findings.

The main contribution of this thesis lies in embedding RAG within a DT framework to enable interpretable patient communication of heart rate data. This approach addresses key challenges in hallucination reduction and contextualization, providing a foundation for future systems that aim to translate complex physiological signals, starting with heart rate, into accessible and trustworthy explanations for end users.

Acknowledgments

I would like to sincerely thank my supervisors at the University of Twente, Mariët and Birna, for believing in this project from the very beginning and for their invaluable guidance throughout. I am also grateful to João, whose feedback and suggestions significantly strengthened this work. From imec, I wish to express my gratitude to Vigee and Jeroen for believing in me from the very first day of my internship interview and for placing their trust in me. Your support continued throughout the entire process. I am equally thankful to Ruud; together, the three of you consistently considered my ideas, supported my proposals, and encouraged me, enabling the project's outcomes to meet the desired goals.

My deepest gratitude goes to my parents, Jose Luis Cabrera Toledo and Guillermina Benitez. Without you and your unwavering support from the very beginning, this would not have been possible. Thank you for always encouraging my dreams, no matter how far they took me from home. Even in the most challenging moments, I felt your love across the distance. I only had to close my eyes to know you were near. Words will never be enough to express my gratitude. To my siblings, Jose Luis Cabrera Benitez, Aranxa Cabrera, and Ariadna Morales, thank you for consistently asking about my progress and caring for my well-being. I truly felt your support in every message, and your successes continue to inspire me to persevere and give my best.

I also wish to thank the most important person in my life, the most optimistic and intelligent person I have met, who became my daily motivation to keep moving forward through the darkest days. Thank you, Bastien Jeannelle, for believing in me even when I did not, for cheering me up every day, for being the shoulder I could lean on when I felt I could not go on, for teaching me that love is shown through actions and mutual admiration, and for taking care of me on the other side of the world. Thank you for becoming my family and my companion throughout this long and challenging journey.

Finally, to my dearest friends who supported me across screens and thousands of kilometers, thank you. Maricarmen Anguas and Lili Aguilar, thank you for celebrating my achievements as if they were your own and for showing me that our friendship is stronger than any distance. To Jimena Bermudez, my confidante and sister in this adventure abroad, thank you for reminding me what true friendship means and for bringing me the warmth of Mexico that I needed so much during this time away.

Wageningen, 29 September 2025
Biaani Cabrera

List of acronyms

AI - Artificial Intelligence

BPM - Beats Per Minute

DHT - Digital Human Twin

DT - Digital Twin

ECG - Electrocardiography

HCI - Human-Computer Interaction

HI - Health Informatics

LLM - Large Language Model

NLP - Natural Language Processing

PPG - Photoplethysmography

RAG - Retrieved-Augmented Generation

RAGAs - Retrieved-Augmented Generation Assessment

Chapter 1
Introduction

1.1 Introduction

Digital twin technology has evolved since its initial conceptualization and has numerous applications across various industries, including construction, energy, transportation, smart cities, agriculture, education, and healthcare. The term digital twin (DT) is defined as a digital copy of a physical system, ideally designed to exchange data bi-directionally in real-time between the virtual and physical components [1, 5]. Originating in the 1990s as a conceptual framework to model real-world systems, the DT paradigm gained traction with Michael Grieves' refinement in 2002 for Product Lifecycle Management [1, 2]. It was later formalized by NASA in 2010 for use in real-time simulation and monitoring of aerospace systems [3].

A digital twin comprises three main components: the physical system, its virtual representation, and the connection between them that ensures a dynamic, real-time data flow. An essential characteristic of a DT that makes it relevant in the industry is its capacity to collect and process data, as well as its flexibility and accuracy, since this information helps simulate the status of the physical twin. This effective integration enables the prediction of its performance, preventing unexpected issues and increasing safety and reliability by optimizing its current and future behavior [1, 4, 7].

Gradually, the concept of DT was also adopted in the healthcare domain, where it may represent either hospital infrastructure or an individual patient [5, 6]. While infrastructure-based DT aids in resource allocation and hospital management, patient-specific DTs are more complex and hold the potential to revolutionize personalized medicine. In this context, and to avoid the unique challenges that simulating a human being introduces, the concept of the Digital Human Twin (DHT) has emerged. Aiming to create a virtual counterpart of physiological processes by aggregating real-time and historical health data to simulate a patient's future health state, DTs offer insights for disease prevention, diagnosis, or personalized treatment planning [7, 8]. Depending on the application, these insights may be intended for healthcare professionals, patients, or both.

A Digital Twin in the healthcare domain can produce various outputs related to its objectives, such as prediction, simulation, monitoring, visualization, or the generation of synthetic patient data [5]. Despite these advancements, a persistent barrier remains. Patients often receive raw health data outputs with minimal contextual explanation [9, 53]. Without a clinical background, these outputs can be overwhelming. This leads to a struggle to interpret physiological data and derive meaningful personal insights, which can limit their ability to understand their health status or take appropriate action [9]. Tackling this challenge requires new interaction strategies to prioritize clarity, empathy, and usability in the communication of health data. A promising solution is the use of AI techniques through Conversational Agents, which may enhance patient communication with Digital Twin data, creating a more natural and user-friendly experience [10].

Recent research highlights a promising direction for improving healthcare communication through the integration of conversational interfaces powered by LLMs [10, 11, 13]. These models, trained on enormous amounts of text, can generate natural-sounding responses and simulate human-like dialogue. When embedded within health systems, LLMs have the potential to help patients better understand complex physiological information by presenting it in a more natural, accessible, and context-aware manner [10,11].

Several recent studies demonstrate the practical potential of LLMs in health-related applications. PhysioLLM [9], for example, is an interactive system that combines wearable physiological data with contextual information to generate personalized health insights and support users in setting actionable wellness goals. In a user study, it outperformed generic LLM chatbots in helping participants interpret their data, particularly in improving sleep quality [9]. Similarly, OpenCHA is an open-source framework designed for the analysis of physiological time series. It integrates user interaction, data sources, and analytical tools to generate accurate health insights. In a benchmark study, it successfully estimated heart rate from PPG signals and outperformed baseline models when validated against ECG data [13]. While these systems focus on real-time monitoring and analytical precision, this thesis introduces a contrasting approach: a design-oriented prototype that simulates a conversational interpretation layer (a dialogue-based system that transforms structured physiological data into natural language responses tailored for end users) for heart rate data within a Digital Twin system, using offline, static datasets strictly for research purposes. This setup does not accurately reflect how the system would operate in real clinical contexts, where continuous, real-time physiological inputs are required.

Despite the promise shown by such systems, integrating LLMs into sensitive domains, such as healthcare, also raises several critical design and ethical concerns. Key challenges include ensuring response accuracy, maintaining interpretability, protecting data privacy, minimizing bias, and preventing hallucination answers that sound credible but are incorrect [9, 12, 13]. These risks underscore the importance of careful system design, particularly in patient-facing applications where clarity, trust, and emotional sensitivity are essential.

This thesis investigates how conversational interfaces powered by Large Language Models (LLMs) can support patients in interpreting heart rate data within a Digital Twin context. The aim is to design and evaluate a prototype that serves as a bridge between complex physiological data and patient-friendly communication, translating structured outputs into clear, empathetic dialogue, which was measured in terms of emotional tone and perceived appropriateness from the users' perspective. The system employs an AI prompt-based strategy, refined through rule-based inputs and pre-evaluated using NLP metrics, to ensure clarity and prevent hallucinations. The project adopts a user-centered, design-oriented approach, combining internal testing with exploratory user evaluation to assess how such systems influence understanding, trust, and engagement, with particular attention to the clarity, usefulness, and emotional tone of the interaction in a healthcare setting.

1.2 Research questions

To guide the investigation, the following research questions are addressed:

- **RQ1:** How should LLM-based conversational interfaces be designed to improve patients' understanding and interaction with heart rate data from Digital Twin systems?

- **RQ2:** What are the risks and limitations of using LLM-based conversational interfaces in Digital Twin systems, and how do these affect patients' trust and engagement with their health data?

1.3 Report outline

Chapter 2 provides the theoretical and conceptual foundation of the study. It reviews the literature on Digital Twin systems in healthcare, Large Language Models (LLMs), and conversational interfaces, while also addressing technical, ethical, and design-related considerations in patient-facing health technologies. It introduces the core principles of Retrieval-Augmented Generation (RAG), its role and challenges in healthcare, and also reviews RAGAs as evaluation metrics to assess groundedness, relevance, and faithfulness in responses. Chapter 3 presents related work, focusing on previous implementations of LLM-based conversational assistants in healthcare and applications of Retrieval-Augmented Generation (RAG) for LLMs in healthcare. Chapter 4 describes the methodology, detailing the conceptual framework of the system, the experimental design, and the evaluation strategy adopted for both technical and user-centered perspectives.

Chapter 5 introduces the prototype development, including the system architecture, datasets, data analysis, summary generation, and the implementation of the interpretation layer. It also presents the technical evaluation with NLP-based performance metrics. Chapter 6 outlines the user study, including participant characteristics, study phases, task scenarios, data handling, ethical considerations, and study procedures. It reports both qualitative and quantitative findings, integrating thematic analysis, questionnaire results, and open-ended responses to assess the clarity, usefulness, and emotional tone of the system. Quantitative user perceptions are assessed using custom questionnaires informed by Chatbot Usability Questionnaire (CUQ) [14] and the Bot Usability Scale (BUS) [15]. These instruments served as a foundation for constructing tailored questions aligned with the study's design goals.

Chapter 7 discusses the results concerning the research questions. It first interprets the system's iterative development across three prototype versions and then relates the findings to the research questions. The final Chapter 8 concludes the thesis with reflections. It summarizes the main contributions, acknowledges challenges and limitations, and outlines directions for future work on conversational health interfaces within Digital Twin systems.

Chapter 2
Research Background

2.1 Digital Twins in healthcare

2.1.1 Introduction to Digital Twins

The conceptual idea of Digital Twins (DTs) emerged around the 1990s, when David Gelernter introduced the concept of Mirror Worlds, virtual environments designed to reflect real-world systems using data from the physical world [1]. About a decade later, in 2002, Michael Grieves expanded on this by proposing the Mirrored Spaces Model within the context of Product Lifecycle Management (PLM), aiming to enhance how products are designed, tested, and managed throughout their lifecycle [2]. However, it was not until 2010 that the actual term "Digital Twin" was published and implemented for the first time by NASA's roadmap, where they used it to describe a powerful way to monitor and predict the status of an aircraft, like tracking wear and tear or forecasting failures [3]. Since then, DTs have expanded, transforming different industries, especially healthcare, where they aim to model and monitor individual patients in real time [5].

A Digital Twin (DT) is the digital representation of a physical system, object, or process, maintaining bidirectional and continuous real-time communication [1,5]. Unlike a static digital replica, a Digital Twin is a dynamic representation that evolves by continuously gathering real-time data from its physical counterpart. Its unique capability to bridge the gap between physical and digital domains is enhanced by technologies such as AI, data analytics, and sensor data. This integration allows the DT to simulate the system's current state and predict future scenarios, including potential defects or unexpected issues, enabling better decision-making in multiple fields of application. Building on this foundation, recent advancements and integration of technologies like Generative Artificial Intelligence, Cognitive Computing (CC), the Internet of Things (IoT), and sensors have paved the way for usage of DTs in diverse areas such as aerospace, energy, agriculture, education, construction, environmental and urban planning, and healthcare field [4, 6–8, 14].

2.1.2 Digital Twins in Healthcare

Throughout the evolution of digital technology, the concept of Digital Twins (DTs) has expanded in scope and application. In healthcare, this broad definition can sometimes lead to confusion if not specified, as DTs may refer to the modeling of either an entire hospital system or infrastructure, or an individual patient's system. This flexibility in application has enabled a wide range of opportunities across the healthcare sector. DTs are increasingly being explored in silico clinical trial design, medical device development, drug discovery, treatment optimization, care coordination, and surgical planning, demonstrating their potential to support both organizational and patient interventions [5, 6]. However, the broad range of use cases and definitions has led to ambiguity around what constitutes a Digital Twin, particularly in healthcare. To reduce this ambiguity, the term Digital Human Twin (DHT) has emerged as a more precise concept within the field, emphasizing the ability to replicate living physical entities. While the term suggests a complete digital replica of a human, in practice, DHTs are limited by the complexity of representing the human body as an integrated biological system, as well as by current technological constraints. Rather than attempting to simulate an entire person, DHTs typically model specific organs, physiological functions, or disease-related processes, offering predictive insights and supporting clinical decision-making in a targeted way [5].

2.1.3 Structure of Digital Twins

To build a reliable and accurate Digital Human Twin (DHT), it is essential to integrate multiple technologies and access extensive, high-quality healthcare data. These data include genetic information, electronic health records, medical imaging, multi-omics data, real-time vital signs, and lifestyle behaviors collected from various sources such as hospitals, wearable devices, mobile health applications, and innovative environments [15-17].

Figure 1 illustrates a general Digital Twin system in the healthcare context, where the physical entity corresponds to a patient's system, and the virtual entity integrates physiological and contextual data to enable personalized insights and interaction.

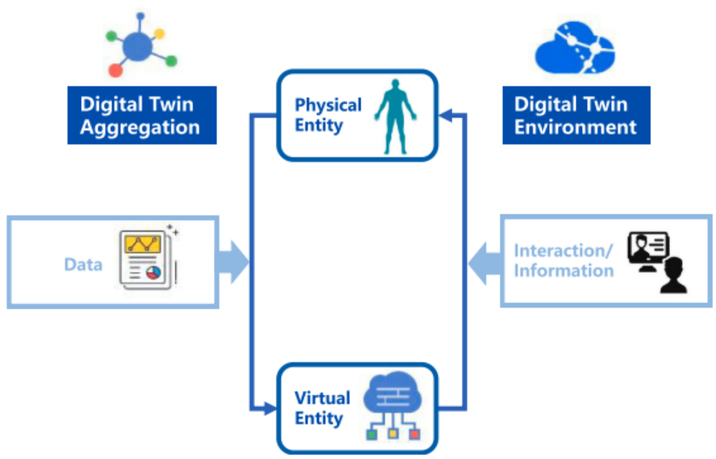


Figure 1. Structure of a Digital Twin system in healthcare, consisting of a physical entity, a virtual entity, and data flows enabling real-time communication. From K. Zhang et al. [6].

DHTs are typically constructed using a modular architecture, where different physiological or biological components are represented as separate, interoperable units. This modularity supports scalability, system flexibility, and continuous updates without disrupting the overall structure [18,19]. To ensure that the diverse datasets can be effectively interpreted and used across modules, they must be standardized and harmonized. This is often accomplished by developing structured ontologies that align formats and terminologies across various data sources, ensuring consistency and interoperability throughout the system [7].

Once collected, the data are transmitted through local networks or mobile devices to cloud-based platforms where they are securely stored and processed. The data are then analyzed using advanced

AI techniques, such as deep learning [7]. These AI-powered systems enhance the DHT's ability to simulate complex physiological states and enable predictive decision-making support, including early diagnosis, risk detection, and personalized treatment planning [20, 21]. Finally, security and privacy are critical components in the development of DHTs. Hence, ideally, multi-layered protection mechanisms are implemented to safeguard sensitive health data while maintaining system performance and regulatory compliance [19].

2.1.4 Applications of Digital Twins in Healthcare

One of the key promises of Digital Health Technologies (DHTs), including Digital Twins, lies in their ability to enable precision medicine: a healthcare strategy focused on tailoring treatments and preventive measures to individual biological, genetic, and psychosocial characteristics. Among these technologies, Digital Twins (DTs) have emerged as a compelling and representative approach. While DTs are part of the broader category of DHTs, this report will refer to them simply as Digital Twins (DTs) for clarity and consistency. By moving away from standardized care, it aims to deliver more effective and timely interventions for each patient. For instance, DHTs can generate personalized risk profiles for chronic conditions, provide lifestyle recommendations, warnings about acute health threats, and prompt timely diagnostics [6–8].

In practice, Digital Twins are being explored for a variety of patient-centered applications, including early diagnostics, chronic disease prevention, personalized therapy planning, medical device development, care coordination, and surgical optimization [5, 6]. These DT-based diagnosis and treatment systems aim to bridge existing gaps in care through AI-driven models that support more precise and individualized interventions [8]. Already, some DTs are being deployed across clinical domains to support data-driven decision-making, particularly in patient care.

- **Cardiovascular Medicine:** Modeling of the heart to enable non-invasive diagnosis and the identification of optimal treatment strategies. Cardiovascular Medicine: Modeling of the heart to allow non-invasive diagnosis and the identification of optimal treatment strategies. For example, DTs have been used to simulate heart mechanics and predict patient outcomes in cardiac resynchronization therapy [7, 8, 17, 21].

- **Surgery:** Patient-specific anatomical models support pre-operative planning and help surgical teams avoid accidental damage to critical structures. DTs in this application have been used to simulate aneurysms for implant sizing, predict post-operative complications such as portal hypertension, and assist in planning interventions to reduce surgical risks and improve recovery outcomes [7, 8, 21].

- **Pharmacy:** DTs simulate organ–drug interactions and accelerate drug development by predicting biochemical reactions and optimizing formulations. One application includes digital heart models used to assess drug safety and personalize dosing during pharmaceutical testing. Additionally, DTs have been applied to improve pharmaceutical manufacturing efficiency by integrating IoT and AI technologies for better scalability and automation. [7, 8, 17, 21].

- **Orthopaedics:** Wearable sensors and simulation systems enable real-time monitoring biomechanical properties and motion of the human body. DTs have been applied in spine modeling to support posture simulations and rehabilitation planning using kinematic analysis and real-time biomechanical modeling [8, 21].

- **Oncology:** Personalization of cancer treatment. By creating patient-specific models, DTs enable the simulation of disease progression and therapeutic responses, supporting more informed and individualized decision-making in oncology care. For instance, they have been used to simulate drug response variability in lymphoma or to support targeted cancer treatment planning. [7, 17].

- **Chronic Disease Management:** DTs integrated with IoT technologies enable continuous monitoring of vital signs to support individualized treatment strategies. By providing real-time data and dynamic feedback loops, it is possible to adjust care plans, improving the health outcomes of patients with long-term conditions. Applications include glucose level tracking and adaptive insulin pump modeling in diabetes care, demonstrating personalized control of glycemic states. [7, 17].

- **Other applications:** Technology extends beyond traditional clinical domains, offering flexible solutions across many areas of healthcare. These applications often aim to improve health outcomes, support personalized decisions, and enhance system efficiency. Whether helping with individual planning, understanding behaviors, or improving healthcare processes, Digital Twins are opening up new possibilities across the field. Some examples are improving vaccination logistics, supporting personalized nutrition and lifestyle planning, and enhancing dental care through 3D modeling for orthodontic treatment. [8, 17, 21].

2.1.5 Patients' Digital Health Data Interpretation

While Digital Twins show great promise in advancing personalized care, the data they generate is often complex for patients to interpret. Variability in biology, behavior, and environment adds complexity, and although AI and wearable technologies offer support, much of the information remains inaccessible without clinical context [6].

This complexity in physiological signals is exemplified by those derived from photoplethysmography (PPG) and electrocardiography (ECG). PPG estimates heart rate by detecting changes in blood volume, while ECG derives it from the electrical signals of cardiac cycles, typically by identifying R-wave peaks within the QRS complex [19, 48]. These complementary modalities provide high-resolution and clinically robust measures of heart rate, which are widely applied in the detection of cardiovascular abnormalities such as bradycardia and tachycardia [48]. Yet, despite their diagnostic value, the raw data they produce can be complex for non-clinical users to interpret without contextualization, thresholds, or personalized framing.

Nowadays, patients have access to physiological data, such as heart rate, activity levels, and sleep patterns, through digital health platforms and wearable devices. While this data can raise awareness and support healthier habits, unclear reference ranges and unfamiliar terminology often contribute to confusion, making it difficult for users to interpret the information in context or translate it into meaningful, personalized actions. Current interfaces typically offer limited interactivity, generic feedback, and minimal personalization, factors that hinder real-time understanding, combined with health literacy gaps and poor design [6, 9, 22, 23].

These challenges highlight the growing need for more intuitive, patient-centered solutions, particularly systems that can translate complex health data into clear, natural conversational guidance. In this context, AI-powered tools are emerging as promising supports to help patients interpret and manage their health. Conversational systems, in particular, can now engage in open-ended dialogue and provide relevant, contextual information more naturally and engagingly [13], offering an opportunity to overcome the limitations of current static interfaces and better align with users' needs and expectations.

2.2 Conversational Interfaces in Healthcare

Many developed AI-driven tools have helped patients understand their health data more easily. Web-based platforms now offer user-friendly features like age-specific centile curves and z-scores to improve contextualization, particularly in pediatric care [22]. Wearables and digital systems enable automated monitoring and trigger alerts for abnormal results, while machine learning models support classifying health states. However, these tools often fall short in translating clinical findings into patient-friendly, actionable insights [24-26]. In overloaded healthcare systems, where professionals may not always be available to answer patients' questions, the rise of telemedicine and remote care has accelerated the adoption of AI-driven tools to ensure consistent, high-quality communication across diverse care settings [27]. Among these emerging approaches, conversational interfaces, particularly those powered by AI, have shown unique potential to bridge this gap by providing tailored, real-time explanations through natural dialogue [9, 13, 28, 43-45].

Conversational interfaces, often called conversational agents (CAs) or chatbots, are increasingly integrated into healthcare as part of the broader adoption of AI technologies. By delivering timely, tailored responses based on individual patient input, they enhance the relevance and clarity of health communication. These designed systems simulate natural, human-like dialogue and support patients through various tasks, including accessing medical information, booking appointments, triaging symptoms, and even offering emotional support [13, 27]. Their potential lies not only in making healthcare services more personalized and accessible, but also in alleviating the administrative burden on providers by handling routine inquiries and offering real-time support [29, 36].

2.2.1 LLM-based Conversational Agents in Healthcare

Conversational interfaces powered by large language models (LLMs) are increasingly integrated with Digital Twins in healthcare through their potential to address limitations of conventional tools, enhancing patient engagement, streamlining data interaction, and supporting personalized care [28]. These interfaces enable natural and intuitive communication between users and complex digital health systems, making health data more accessible and actionable.

Large Language Models (LLMs) such as GPT-4 [46] or LlaMA [32] are trained on massive volumes of unstructured text. They can comprehend, generate, and reason with natural language, unlocking new possibilities for patient-facing applications [10, 13]. These models have reshaped how patients, professionals, and data interact, making communication more proactive and personalized. This shift supports dynamic engagement that adjusts to the needs of both patients and providers; when linked to user-specific health data, LLMs can generate context-aware insights that enhance interpretability and foster greater user trust in the information provided [29]. Their ability to aggregate large bodies of textual knowledge and provide contextual, goal-oriented responses has positioned them as key tools for interactive health support [13].

Recent research illustrates how these capabilities are being extended into physiological data interpretation. Emerging systems now connect LLM-based dialogue with wearables and clinical monitoring devices to provide personalized insights. For example, PhysioLLM enables natural-language interaction with wearable sensor data [9], the Personal Health Insights Agent (PHIA) delivers wellness feedback from sleep and activity data [43], LLM-CGM supports conversational queries on continuous glucose monitoring for diabetes care [44], and ALPHA integrates multimodal physiological signals for anomaly detection and patient guidance [45].

Ongoing efforts to improve the speed, efficiency, and safety of LLMs continue to support their broader adoption. However, the integration of these systems into clinical and personal health workflows must be carried out with careful attention to ethical, safety, and privacy considerations [10, 28].

2.2.2 Technical and Ethical Considerations in LLM-Based Health Systems

While AI conversational systems promise more accessible and personalized care, they also introduce critical ethical and trust-related challenges that must be addressed to ensure these technologies truly help and empower those who use them. Patients need reassurance that their health data is protected, that the information they receive is accurate and unbiased, and that the system responds in a human-centered, empathetic way.

Data Handling and Security

Integrating LLMs into Digital Twin systems requires special attention to data privacy and secure processing. While many healthcare systems already follow strict privacy regulations, LLM-based interfaces introduce unique challenges due to their reliance on external APIs and cloud-based processing [30, 33]. These systems must be designed with enhanced safeguards, such as local data preprocessing, encryption, on-device handling where feasible, and transparent communication of how personal data is used or stored. Projects like PhysioLLM [9] emphasize privacy-preserving designs that prevent the inclusion of identifiable information in any communication with external services. Similarly, large-scale wellness systems such as PHIA [43] highlight the risks of scaling personal data interpretation across populations, underscoring the need for transparency in how user data is processed and shared.

Reliability and Bias

Although LLMs show promise in knowledge summarization and natural language interaction, their application in clinical contexts is limited by concerns over data quality, accuracy, interpretability, bias, and hallucinated content [10, 31, 32]. These risks are evident in specialized domains, where models must remain closely aligned with clinical guidelines or curated datasets. For instance, LLM-CGM demonstrates the challenge of ensuring reliable interpretations in chronic disease management [44], while ALPHA illustrates how multimodal anomaly detection must balance technical precision with clinically safe recommendations [45]. These systems should therefore not be used to provide diagnoses or treatment without human oversight, but rather as support tools offering contextualized information grounded in verified sources [9].

User Experience and Trust

Conversational interfaces make digital twin systems more user-friendly, empowering patients and clinicians to interact with health data naturally and intuitively, which can improve self-care and clinical decision-making [10, 33-35]. However, for patients to trust and engage meaningfully with these systems, the experience must feel not only helpful but also safe, respectful, and transparent. This includes clearly communicating the system's capabilities and limitations, ensuring timely and fluid responses, and behaving reliably when uncertain [9]. In emotionally sensitive contexts such as elder care or mental health, thoughtful design is essential to prevent too much trust or misinterpretation. Ultimately, trust is fostered through functionality, clarity, and empathy of a real conversation [10].

2.3 Retrieval-Augmented Generation (RAG)

2.3.1 Core Principles of RAG

Retrieval-Augmented Generation (RAG) is a hybrid framework that integrates retrieval-based methods with generative models to enhance factual grounding and reliability in language model outputs. By incorporating external knowledge into the generative process, RAG significantly improves performance in knowledge-intensive tasks such as open-domain question answering and clinical data interpretation [17, 31, 36]. Unlike traditional large language models (LLMs), which are constrained by their pre-training data and cutoff dates, RAG dynamically retrieves relevant context at inference time, reducing hallucinations and producing more trustworthy responses. This ability to minimize hallucinations is especially crucial in health-related contexts, where inaccurate or invented responses could undermine patient trust or lead to serious misinterpretation of physiological data.

The importance of RAG becomes clear when considering the main limitations of LLMs. First, models are prone to hallucination, often producing confident but inaccurate answers when no reliable evidence is available. Second, they are not well-calibrated: even when wrong, models can present their outputs with undue certainty, making errors difficult for users to detect. Third, LLMs cannot directly access proprietary or domain-specific datasets, such as medical records or internal clinical notes, solely through prompting. Finally, their static training makes them poorly suited to answering questions about recent or rapidly evolving information [37].

The effectiveness of RAG lies in the complementary roles of its two components: the retriever and the generator. The retriever employs information retrieval (IR) techniques to search external databases and identify documents most likely to contain relevant evidence for a given query. These retrieved passages are then supplied to the generator, which integrates them into coherent, context-sensitive answers [36, 37]. This dual process addresses the weaknesses of simple prompting by grounding responses in curated datasets rather than relying solely on the LLM's parametric knowledge. Figure 2 illustrates this dual structure, highlighting how the retriever locates contextually relevant documents, which the generator or reader then processes to produce an evidence-based answer.

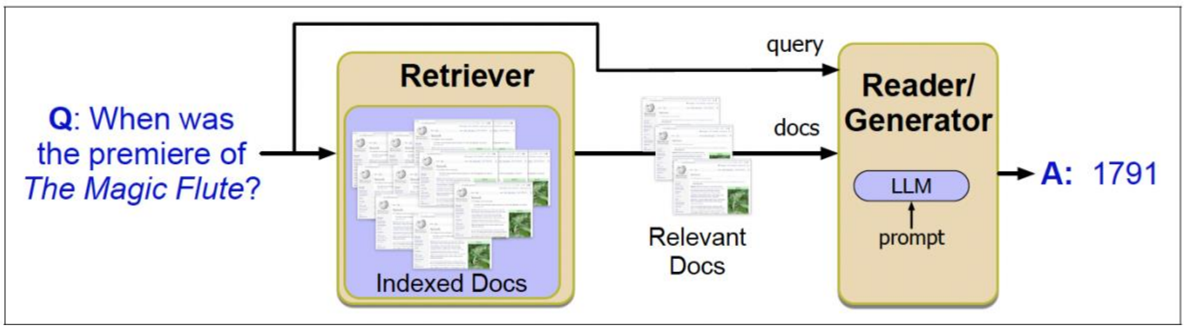


Figure 2. RAG architecture combining retrieval of relevant documents with generation of grounded responses. From Daniel Jurafsky and James H. Martin. [17].

The RAG workflow is generally described through three stages:

Indexing. Raw documents are transformed into a standardized text format, segmented into smaller "chunks" that fit within the model's context window, and converted into dense vector embeddings that preserve semantic meaning. These embeddings are stored in a vector database, enabling efficient similarity search and ensuring that LLMs can access structured data in a form they can interpret [17, 31].

Retrieval. When a user query is submitted, it is encoded into the same embedding space as the indexed chunks. The system calculates similarity scores (e.g., cosine similarity) between the query and stored vectors, and selects the top-K most relevant chunks. This mechanism ensures that the evidence provided to the model is directly aligned with the intent and context of the user's request [17, 31].

Generation. The retrieved chunks are concatenated with the user's query to form a context-enriched prompt. The LLM then synthesizes a response that incorporates the retrieved evidence, balancing factual accuracy with natural language fluency. In this way, the generation step shifts the system from simple retrieval toward a conversational, user-facing explanation that remains grounded in curated content [17, 31].

2.3.2 Role of RAG in Healthcare

Retrieval-Augmented Generation (RAG) is increasingly used in healthcare to enhance the accuracy, reliability, and transparency of large language model (LLM) outputs by integrating real-time, domain-specific information [17]. RAG improves clinical decision support, medical question answering, personalized patient care, healthcare education, and information extraction by grounding responses in up-to-date, evidence-based sources [21-27].

- **Clinical Decision Support.** Provides evidence-based, real-time recommendations by integrating external guidelines and literature, enhancing diagnostic accuracy, and reducing misdiagnosis rates. Use cases include personalized treatment, emergency triage, disease management, and diagnostic support [21-23, 27].

- **Medical Question Answering.** Enhances LLMs with current, domain-specific knowledge, surpassing standard models in accuracy and reliability. Applications include complex guidance interpretation, specialty-specific queries, radiology, and diabetes-related care questions [21, 22, 24, 25, 27].

- **Personalized Patient Care.** Combines patient-specific data with current research to assist personalized treatment planning. Examples include providing the latest learning resources and providing patients with accessible and personalized health information [21-23, 26].

- **Information Extraction.** Automates the extraction of relevant data from EHRs and scientific literature, streamlining clinical workflows. Key use cases include clinical trial screening, EHR summarization, and pharmacovigilance [21, 23, 24].

2.3.3 Challenges of using RAG in Healthcare

Although RAG offers a robust framework for improving the factual grounding of LLMs, it is not without limitations. At a general level, three categories of challenges have been identified: retrieval difficulties, where systems struggle with precision and may return irrelevant or incomplete chunks; generation errors, where models still risk hallucinations, irrelevance, or biased outputs; and augmentation hurdles, where integrating heterogeneous sources can result in incoherent or redundant responses [31]. These issues become even more pronounced in healthcare, where medical knowledge is complex, dynamic, and safety-critical.

The main challenges of applying RAG in healthcare can be grouped into various areas, such as:

- **Retrieval and Knowledge Limitations.** RAG depends heavily on the availability and quality of external sources. Outdated or incomplete datasets can result in knowledge deficiencies, particularly in rapidly evolving fields such as medicine. Retrieval systems may also struggle to identify the most relevant documents or collapse by repeatedly returning the same evidence, undermining contextual accuracy [21, 31].

- **Hallucinations and Factual Errors**. Even when relevant sources are retrieved, RAG models may still produce hallucinations, factual inaccuracies, or biased outputs. In healthcare, such errors carry heightened risks, as incorrect recommendations can lead directly to patient harm [18, 21, 31, 38].

- **Integration and Contextualization.** Medical knowledge is heterogeneous, spanning narrative text, structured tables, flowcharts, and imaging results. Integrating these diverse formats into coherent and contextually accurate outputs remains a challenge. Augmentation can also result in redundancy or incoherence when multiple overlapping sources are combined [21, 31].

- **Operational Challenges.** In practice, RAG also faces operational hurdles, including high computational costs, the risk of repetitive or irrelevant retrievals, and the difficulty of embedding such systems into clinical workflows without adding to clinicians' cognitive load [21, 22, 38].

- **Ethical, Legal, and Privacy Constraints.** Deploying RAG in healthcare must comply with strict privacy and safety regulations. Systems risk inadvertently exposing sensitive patient information if retrieval sources are not adequately secured. Ethical concerns also extend to fairness, as biases in medical datasets can disadvantage underrepresented populations. More broadly, the responsible adoption of RAG requires balancing innovation with accountability, transparency, and legal compliance [18, 21].

In this thesis, we focus primarily on two of these challenges: hallucinations and contextualization. Specifically, we aim to reduce hallucinations in patient-facing responses and enhance the integration of structured physiological data, such as heart rate, into coherent and trustworthy conversational outputs.

2.3.4 Retrieval-Augmented Generation Assessment (RAGAs)

Evaluating Retrieval-Augmented Generation (RAG) systems requires methods that go beyond traditional natural language processing (NLP) metrics. Standard measures such as BLEU and ROUGE [31] primarily capture surface-level word overlap. In contrast, more advanced metrics such as BERTScore [39] leverage semantic similarity but still fail to account for the interplay between retrieval and generation fully. To address this, the Retrieval-Augmented Generation Assessment (RAGAs) framework was introduced as a task-specific evaluation method designed to assess the quality of RAG pipelines in both aspects, retrieval quality and generated accuracy [17, 31, 35].

Unlike classical metrics, RAGAs does not rely solely on reference answers; it uses proxies for correctness and utility of retrieved passages, making it effective where gold-standard answers are scarce [35]. RAGAs can compute these dimensions automatically by prompting an LLM to verify claims, extract sentences, and compare embeddings between answers and queries [35]. In practice, these metrics can be combined into composite scores, balancing retrieval precision and generative fidelity [40]. The framework is typically structured around four core dimensions:

- **Faithfulness**: measures whether the generated answer remains grounded in the retrieved context. An answer is considered faithful if its claims can be directly inferred from the supporting evidence. This metric penalizes hallucinations and unverifiable statements. The LLM decomposes the answer into factual statements and verifies whether each is supported by the retrieved context [17, 31, 35, 40]. The score is computed as:

$$F = \frac{|V|}{|S|}$$

  Where $|V|$ is the number of statements in the generated answer supported by the retrieved context, and $|S|$ is the total number of statements extracted from the generated answer.

- **Answer Relevance**: evaluates how well the generated response addresses the user's question. It does not measure factual accuracy, but instead penalizes incomplete or redundant answers that deviate from the original query. To estimate this, the LLM generates potential follow-up questions from the system's answer and compares them to the original query using embedding similarity [17], [31], [35], [40]. This score is computed as:

$$AR = \frac{1}{n}\sum_{i=1}^{n} sim(q, q_i)$$

  Where q is the user query, q_i are follow-up questions from the answer, and $sim(q, q_i)$ is the cosine similarity of their embeddings. The score measures how closely the answer aligns with the user's intent.

- **Context Relevance**: assesses whether the retrieved log chunks are appropriate and sufficient for answering the user's question. This metric penalizes the inclusion of irrelevant or redundant information that could dilute the answer. The LLM extracts the subset of sentences necessary to answer the question and compares them to the full retrieved context [17, 31, 35, 40]. This metric is computed as:

$$CR = \frac{|S_{ext}|}{|C|}$$

  Where $|S_{ext}|$ is the number of sentences in the retrieved context identified as relevant, and $|C|$ is the total number of sentences retrieved.

Chapter 3
Related Work

3.1 LLM-Based Conversational Interfaces in Healthcare

Conversational interfaces powered by large language models (LLMs) have gained attention as tools to help patients interpret and engage with health data. Research focuses on grounding dialogue in external evidence to ensure reliability. Wang et al. [42] introduced RAGate, an adaptive gating mechanism that decides when retrieval is needed during conversation. The system models conversational context and relevant inputs to predict whether external knowledge (via retrieval) will improve response quality. RAGate is trained using human-labeled data on when augmentation was beneficial, and it explores both large language models and attention-based neural gate models to learn this decision boundary. This work reflects a broader trend toward integrating dialogue agents with retrieval systems to enhance reliability and trustworthiness.

A second area of research has examined how LLMs can be used for interpreting physiological signals. Feli et al. [13] developed an LLM-powered agent to estimate heart rate from photoplethysmography (PPG) signals, implemented on the OpenCHA framework. By combining user interaction, analytical models, and data sources, their system increased signal-level accuracy compared to benchmark models. While this contribution primarily focuses on algorithmic accuracy, it demonstrates that applying LLMs to biosignal processing is feasible.

Building on this, Fang et al. [9] proposed PhysioLLM, an interactive system linking wearable sensor data with dialogue to provide personalized health insights. Unlike backend analytics, PhysioLLM lets users query their data in natural language, supporting reflection on lifestyle and health behaviors. Similarly, Merrill et al. [43] introduced the Personal Health Insights Agent (PHIA), leveraging LLMs to analyze wearable data like sleep and fitness metrics and generate behavioral insights. PHIA performed well on over 4,000 health queries, highlighting LLMs' potential to scale personalized wellness support.

Extending this trajectory into chronic disease management, Healey and Kohane [44] developed LLM-CGM, a benchmark for conversational querying of continuous glucose monitoring (CGM) data in diabetes care. Their work demonstrated how LLMs can assist patients in interpreting numerical glucose trends and retrospective plots by framing them in natural language, thereby reducing barriers to self-management for individuals with diabetes. Tang et al. [45] further advanced this research with ALPHA, an LLM-based system for detecting abnormal physiological health. By combining multimodal signals such as heart rate, oxygen saturation, and photoplethysmography, ALPHA offered accurate anomaly detection and health insights for users, highlighting the role of LLMs in bridging physiological data with personalized feedback.

These studies show a continuum in LLM-based health tools: from retrieval mechanisms for dialogue [42], to biosignal analysis [13], to patient systems emphasizing interpretability with wearables [9, 43], and to chronic disease management and anomaly detection [44, 45]. Our work combines these by embedding heart rate data into a conversational framework, focusing on patient interpretability within a Digital Twin system.

3.2 Retrieval-Augmented Generation (RAG) for LLMs in Healthcare

Retrieval-Augmented Generation (RAG) has emerged as a promising approach to addressing some significant limitations of large language models (LLMs) in healthcare, including outdated training data, hallucinations, and a lack of transparency in outputs. Amugongo et al. [17] provide a recent systematic review indicating that while RAG grounds responses in external evidence, there remains limited agreement on which datasets, methodologies, and evaluation frameworks are most effective in clinical applications. The review highlights a shortage of standardized benchmarks and ethical safeguards, emphasizing the need for more responsible methods of adoption.

Domain-specific innovations have shown promise in narrowing these gaps. For instance, Zhao et al. [23] integrate MedRAG knowledge graph–elicited reasoning into retrieval pipelines to enhance diagnostic accuracy and inform treatment decisions in diseases characterized by overlapping symptoms. Miao et al. [26] adapted a RAG-enabled ChatGPT system for nephrology by aligning outputs with KDIGO 2023 guidelines for chronic kidney disease. Their work illustrates RAG's potential in specialty domains, enabling outputs that remain guideline-driven and clinically grounded, beyond general Electronic Health Record (EHR) tasks toward more targeted, precision-oriented applications.

Empirical studies further underscore RAG's utility in complex medical reasoning. Thompson et al. [41] introduced a zero-shot phenotyping method for EHRs, in which disease-relevant text snippets were retrieved and supplied to the LLM. This approach significantly outperformed physician-defined rules in diagnosing pulmonary hypertension, demonstrating RAG's capacity to support the identification of rare disease cohorts and clinical research pipelines. Separately, Wang et al. [42] developed RAGate, an adaptive mechanism that selectively determines when retrieval is needed during conversation. Although initially aimed at general dialogue, its dynamic alignment of context and retrieval has clear implications for healthcare's demand for interpretable and trustworthy systems.

Beyond domain-specific case studies, systematic evaluations have begun to map RAG's internal architecture and design decisions. Kay et al. [22] categorize biomedical RAG approaches into pre-retrieval, retrieval, and post-retrieval stages, emphasizing how preprocessing strategies, such as summarization and chunking, can directly influence retrieval relevance and system performance. This directly supports our study's emphasis on pre-retrieval summarization as a critical step to enhance interpretability in heart rate data applications.

Similarly, evaluation methods are evolving. Gargari and Habibi [21] compared human and automatic metrics for medical RAG systems and found RAG models often outperform baseline LLMs while maintaining clinical reasoning performance. Additionally, Bora and Cuayahuitl [25] analyzed RAG-based medical chatbots, comparing hybrid and fine-tuned models in terms of accuracy, factual grounding, and confidence. Their results suggest RAG can improve reliability but highlight the need for testing in realistic clinical scenarios.

Together, these contributions reinforce the importance of RAG as a foundational strategy for improving LLMs in healthcare. Reviews [17], knowledge designs [23], domain adaptations [26], applications [41], mechanisms [42], and evaluation frameworks [21, 25] show RAG's potential to connect clinical data with interpretable, evidence-based results. Progress relies on robust benchmarks, ethical safeguards, and domain-specific changes. Our study contributes to the integration of RAG into patient interfaces that interpret heart rate data within Digital Twin systems, emphasizing clarity, trust, and context.

3.3 Identified Gap and Research Focus

Taken together, these strands of research highlight two converging directions: conversational interfaces that enable patients to interact with and interpret physiological data, and retrieval-augmented generation pipelines that strengthen reliability, grounding, and domain adaptation in healthcare. However, limited work has explored how these two directions can be meaningfully combined to support patient-facing interpretation of physiological signals, particularly heart rate, within Digital Twin systems.

Our work extends this intersection by embedding RAG into a Digital Twin framework, enabling patient-facing interpretability of heart rate data through conversational exchanges. This contribution bridges methodological advances in retrieval with user-centered design, addressing both the technical and experiential challenges of making physiological signals accessible and meaningful to patients.

Chapter 4
Methodology

This research was developed as part of a collaboration with imec One Planet, a digital health research institute focused on preventive, personalized technologies for sustainable healthcare solutions [52]. It investigates, designs, and develops an iterative prototype that bridges the gap between structured physiological data and patient understanding through context-aware dialogue. To align with ethical standards in healthcare, the system is explicitly designed to avoid offering diagnostic or treatment advice.

4.1 Research Design

Given the interpretive focus of the research questions, this study adopts a mixed-methods research design that integrates both quantitative and qualitative perspectives. On the technical side, quantitative metrics are used to evaluate the accuracy and reliability of the LLM-based system proposed. On the human side, qualitative evaluation is conducted through user studies that investigate perceptions of trust, tone, clarity, and usefulness. Together, this dual focus ensures that the methodology captures both computational performance and user experience, reflecting the central goal of improving the interpretability of health data.

Although not yet recognized as a standard method in digital health, this dual-focus methodology is guided by principles from Human-Computer Interaction (HCI) and Health Informatics (HI) [24, 25]. HCI provides methods for embedding usability and user feedback throughout system development, while HI contributes to the structuring, interpretation, and communication of health data responsibly. Drawing on Natural Language Processing (NLP) and responsible AI design, the methodology ensures that both algorithmic logic and human needs inform the development and evaluation of the prototype.

4.2 Research Process

The research process is structured around four stages: Research, Prototype Development, User Study, and Evaluation.

Stage 1: Literature Review and Prototype Design Choices

This stage established the foundation for the system by investigating prior work on Digital Twins, conversational interfaces, and responsible AI in healthcare. Through a focused literature review and design exploration, communication gaps were identified in how patients interpret health data, particularly the lack of supportive, trustworthy, and comprehensible feedback mechanisms. This analysis highlighted the potential of Digital Twins for personalized monitoring, but also the difficulty patients face in engaging with raw physiological data.

Further review of conversational interfaces and LLM-based systems in healthcare revealed both opportunities and risks: while such models can generate accessible explanations, they also raise concerns of reliability, bias, and trust. Retrieval-Augmented Generation (RAG) was identified as a promising strategy to ground model outputs in structured data and mitigate hallucinations. At the same time, recent advances in evaluation frameworks emphasized the importance of combining computational metrics with user-centered assessments.

Together, these insights not only defined the initial system requirements around interpretability, empathy, and the avoidance of diagnostic advice but also guided the technical choices and overall design approach for the prototype. As such, this stage functioned both as a literature review and as the basis for shaping concrete design decisions in the system's development.

Stage 2: Data Simulation and Prototype Development

To prepare for system development, the study first simulated heart rate datasets designed to resemble real physiological signals. These datasets were generated following the structure of the PPG-DaLiA study [19], incorporating realistic activity segments, including resting, walking, and exercising. An 8-second window with a 2-second shift was used to calculate mean instantaneous heart rate values, a common practice in PPG-based heart rate estimation. This approach enabled controlled experimentation while avoiding privacy risks associated with real patient data. Heart rate was selected as the focal parameter due to its broad interpretability and relevance across medical and everyday wellness contexts. However, the use of simulated inputs necessarily excluded natural signal variability, which is acknowledged as a limitation of this study.

To support semantic parsing and natural language generation, the raw time-series data were transformed into structured hourly summaries. These summaries extracted key features, such as heart rate metrics and behavioral context, and organized them into predefined fields. This structure is better aligned with human time perception and allows the system to efficiently interpret and map physiological trends into meaningful, context-aware prompts. By reducing data complexity and ambiguity, the structured summaries served as a critical intermediary, enabling the conversational agent to generate coherent, grounded, and patient-tailored responses.

Building on this foundation, the prototype was designed as a web-based interface integrating three core components: (1) the structured dataset of heart rate summaries, (2) a rule-based classification layer to distinguish normal and abnormal values in relation to demographic context, and (3) a conversational module powered by a Large Language Model (LLM) supported by Retrieval-Augmented Generation (RAG). The preprocessing step can be understood as the foundation of the retrieval pipeline, situated within the pre-retrieval phase of RAG pipelines, as described in recent systematic reviews of biomedical applications [22], where such strategies play a critical role in improving retrieval relevance and efficiency.

This architecture was deliberately chosen to balance technical reliability with communicative clarity: the structured dataset ensured consistent input, the rule-based layer provided transparent and clinically informed thresholds, and the RAG-enhanced dialogue system anchored generative responses in factual data while reducing hallucinations.

For the conversational module, the study employed LLaMA 3.2 [32], an open-weight LLM variant developed by Meta. This model was selected for its strong instruction-following capabilities, lightweight deployment requirements, and ability to run locally without reliance on external APIs. These characteristics aligned with methodological priorities of reproducibility, transparency, and data privacy, while also enabling iterative experimentation in prompt design and system refinement. To guide the assistant's tone and behavior, the temperature parameter of the LLM was set to 0.7. This value was selected to strike a balance between response diversity and coherence, enabling the assistant to produce conversational and empathetic outputs while still grounding its responses in factual, retrieved context.

This methodological choice reflects the dual objective of the study: to ground system responses in traceable physiological trends while presenting them in a way that patients could understand and trust. By integrating safeguards at both the data and system levels, the prototype ensured that interpretive feedback remained accurate, accessible, and within clearly defined non-diagnostic boundaries.

Stage 3: Iterative Refinement

To refine the system systematically, development followed an iterative approach across three versions of the prototype, further explained in Section 5.5.1: Version 1 (V1), Version 2 (V2), and Version 3 (V3). Each iteration incorporated targeted design modifications based on identified limitations, evolving the assistant's capacity for interpretive support while maintaining its non-diagnostic role.

- **Iteration 1** focused on establishing a proof of concept by implementing the whole end-to-end system architecture. This included the core pipeline of structured data preprocessing, rule-based classification, and LLM-driven dialogue generation. At this stage, the system supported specific time-based and general health queries. This initial version served as a baseline for assessing the feasibility of RAG-enhanced interpretation and set the foundation for subsequent refinements.

- **Iteration 2** extended the retrieval mechanism to support a broader range of query types by identifying and selecting relevant time-adjacent chunks based on semantic alignment, rather than relying solely on strict time matching. This involved retrieving the closest matching chunk as well as contextually relevant surrounding summaries. A fallback strategy was also introduced to ensure the assistant could still generate responses in cases where exact time matches were unavailable. The primary objective of this stage was to enhance the contextualization of heart rate data across broader timeframes, enabling the system to handle more flexible user queries.

- **Iteration 3** expanded the retrieval logic further by enabling the assistant to retrieve all available hourly summaries for a given user when handling generalized or summary-type queries. This allowed the assistant to synthesize a broader context across multiple time windows. Additionally, demographic-aware logic was incorporated into the classification layer and prompt generation process. The retrieved context was now tailored based on user-specific thresholds and comparison norms (e.g., age-appropriate heart rate ranges), improving the precision and personalization of the retrieved information without altering the fundamental retrieval infrastructure. Iteration 3 also emphasized testing the trade-off between interpretive richness and precision, aiming to approximate real-world use cases more closely.

By structuring development in these iterations, the methodology allowed for controlled exploration of design choices and their impact on system performance. This staged approach ensured that refinements were traceable and systematically informed by observed limitations rather than arbitrary adjustments.

Stage 4: User Study and Data Collection

The fourth stage involved evaluating the prototype with nine participants in a simulated usage environment. Rather than being isolated at the end, this stage was conducted in parallel with the iterative development process, allowing user feedback to shape system refinements across versions actively. The study examined how participants interpreted the system's responses, focusing on clarity, emotional tone, usefulness, and trustworthiness. To simulate realistic use, participants engaged in both structured tasks and open-ended interactions, where they could freely explore the assistant's capabilities.

The evaluation followed a two-phase approach. The exploration phase involved a small set of participants who interacted with early iterations of the prototype. This phase served to test feasibility, identify immediate usability issues, and generate formative feedback that informed refinements. The main evaluation phase involved a larger group of participants. It focused on systematically assessing the refined prototype, allowing for a more robust understanding of user perceptions regarding trust, interpretive support, and conversational flow.

Participants were recruited voluntarily from imec full-time employees, who provided valuable exploratory insights into how communication design choices shaped perception and trust. Data collection involved:

- **Quantitative measures**: standardized questionnaires to capture perceptions of clarity, naturalness, safety, and usefulness.

- **Qualitative measures**: open-ended feedback and semi-structured interviews to explore expectations, trust, and interpretive support in greater depth.

All data was anonymized, and no personal health information was used. Ethical safeguards were applied by ensuring that the prototype explicitly avoided diagnosis or prescriptive advice, framing itself solely as an interpretive assistant.

This stage provided critical insight into how the system supported or hindered user understanding of heart rate data.

Stage 5: Evaluation

Evaluation combined technical and user-centered perspectives.

- **Technical evaluation**: Metrics from the RAGAs framework were applied, focusing on context relevance, answer relevance, and faithfulness. These scores quantified how well the assistant's responses aligned with retrieved logs and user intent. Comparisons with a baseline LLaMA 3.2 model without retrieval provided further insight into the contribution of the RAG pipeline.

- **User evaluation**: Interviews and questionnaire results were analyzed by thematic analysis of user comments. This dual approach revealed not only numerical usability ratings but also perceptions of tone, interpretive value, and conversational flow.

Together, these methods provided a multi-layered understanding of system performance, linking quantitative indicators with subjective user experience.

Stage 6: Results and Interpretation

This stage focused on systematically analyzing the data collected through both technical metrics and user evaluations. Quantitative results from computational assessments (e.g., context relevance, faithfulness, and answer relevance) were combined with qualitative feedback from questionnaires and interviews. Our analysis process focused on triangulation, combining technical indicators with user perceptions to spot patterns and trade-offs. This approach made sure our findings were not interpreted in isolation but rather considered within both the computational performance and the human-centered experience.

Stage 7: Conclusion and Future Work

The final stage synthesized insights from all preceding phases into a coherent conclusion. Methodological outcomes were consolidated to address the research questions directly, while also acknowledging the study's limitations. Building on identified limitations, this stage outlined clear directions for future research, such as validation with real-world data, testing with larger and more diverse participant groups, and exploring advanced conversational strategies or model configurations. By closing with both reflection and projection, this stage ensured the research contributes not only immediate insights but also a roadmap for further exploration.

Figure 3 provides an overview of the methodological stages, illustrating the progression from research and prototype development to iterative refinement, user study, evaluation, and interpretation.

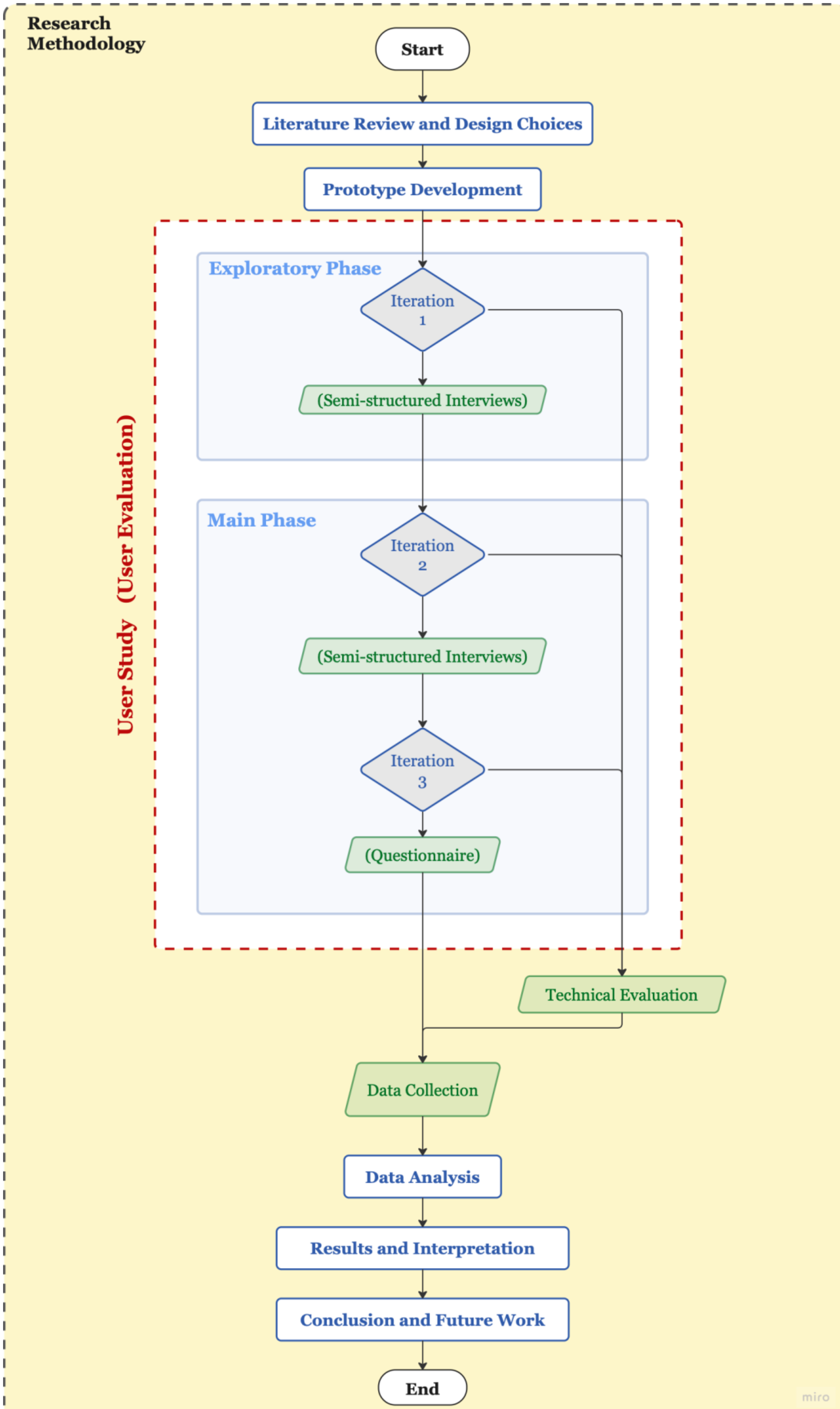


Figure 3. Overview of the research methodology, showing the progression from literature review and prototype development to iterative refinement, user study (exploratory and main phases), technical evaluation, data analysis, and synthesis of results.

Chapter 5
Prototype Development

5.1 System Architecture Overview

This chapter presents the technical development of the prototype system, structured to simulate a conversational assistant for interpreting heart rate data from a Digital Twin (DT) healthcare context. The design aims to support non-clinical users in understanding physiological heart rate data through a user-centered, explainable, and modular architecture while avoiding overwhelming or hallucinated answers through the human-computer interaction. Each component contributes to a layered pipeline that transforms offline raw heart rate measurements into natural language insights tailored to individual contexts, maintaining trustworthiness and ethical responsibility.

Figure 4 represents the system architecture, which follows a five-stage modular pipeline serving as a visual reference for the detailed breakdown of each component described in the following sections. Each stage in the structure builds on the outputs of the previous one, starting with the ingestion of preprocessed structured heart rate datasets designed to simulate a range of real-time physiological conditions, including resting baselines, light activity (e.g., walking), post-meal states, and stress-induced elevations caused by cognitive or physical exertion. These datasets were generated using a clinical-rule logic to mimic abnormal heart rate patterns based on demographics and activity. Abnormal episodes lasted at least 10 minutes during periods of inactivity, surpassing established thresholds. The simulated data are then embedded into **structured summaries**, which then serve as input to a **Retrieved-Augmented Generation (RAG)** pipeline. This retrieval system identifies and extracts the most relevant information from the structured context to support the prompt given to a **Large Language Model (LLM) interpretation layer**, which then provides a context-reliable answer to the user's query. The LLM then generates an empathetic, natural language response that incorporates both the user's heart rate state, derived from the retrieved structured summary, and the interpretive rules specified in the prompting strategy. Finally, the output is delivered via a user-friendly **Conversational Interface** that simulates an interaction with a digital health assistant. The architecture supports two primary interaction modes: system-initiated alerts triggered by abnormal heart rate patterns and user-initiated queries regarding heart rate trends or conditions.

To assess the quality of the system, we employed Retrieval-Augmented Generation Assessment (RAGAs) [17], a framework for evaluating RAG-based systems. This NLP evaluation technique provides metrics to assess both the retrieval component through context relevance, which evaluates the RAG pipeline, and the generative component, by measuring the faithfulness and answer relevance of the LLM responses [34, 35]. Each of these four aspects is scored on a scale from 0 to 1, ensuring that the assistant's outputs remain grounded, informative, and contextually appropriate.

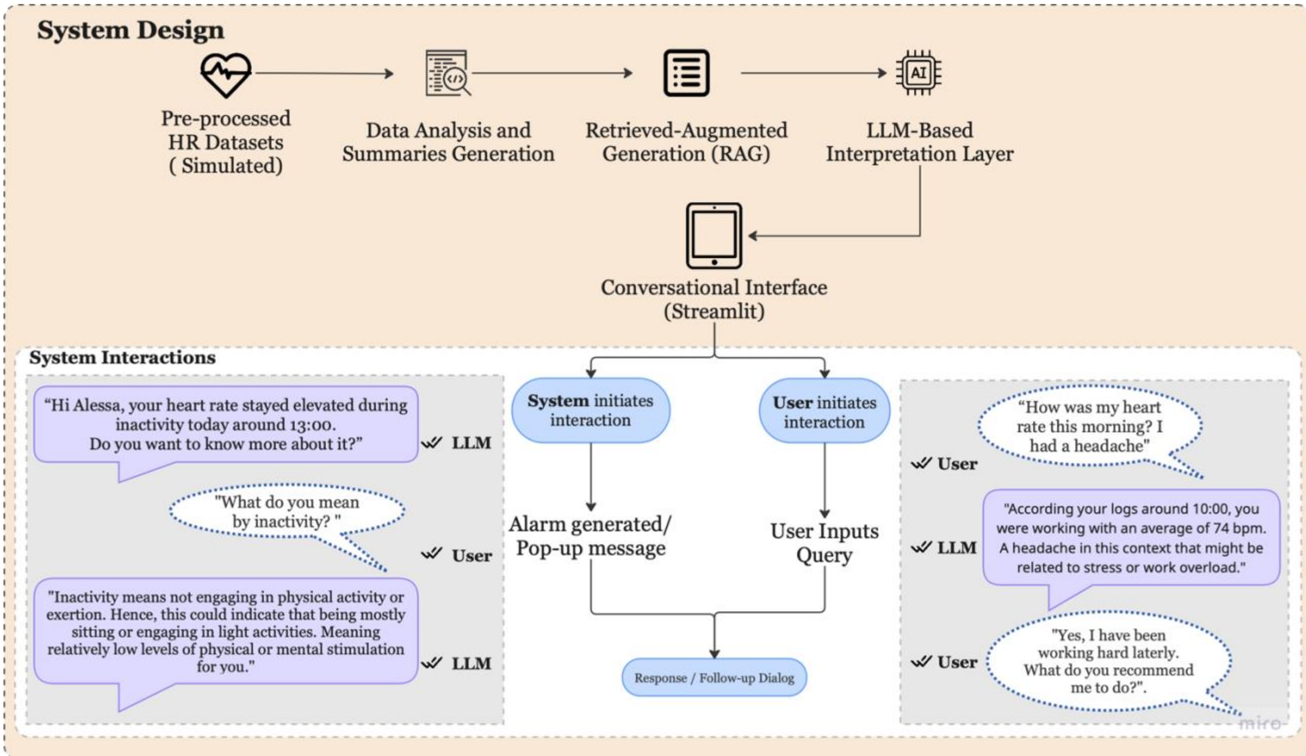


Figure 4. System architecture and interaction design of the LLM-Based Conversational Interface. The architecture proceeds through five main stages: (1) ingestion of pre-processed simulated heart rate datasets, (2) data analysis and generation of structured summaries, (3) retrieval of relevant context via the RAG pipeline, (4) generation of context-reliable responses in the LLM-based interpretation layer, and (5) delivery of outputs through a Streamlit-based conversational interface. The interface supports two modes of interaction: system-initiated alerts triggered by abnormal heart rate patterns, and user-initiated queries regarding heart rate states.

5.2 Heart Rate Datasets

Two types of heart rate data sources were used to support the simulation of structured health insights: a public dataset and a simulated one, each contributing complementary value. The public dataset ensured ecological validity through real-world physiological signals, while the simulated data enabled experimental control through customizable scenarios aligned with the system's interpretive goals.

5.2.1. Public Dataset

The initial system development and testing used the PPG-DaLiA dataset from the UCI Repository [19] (see Section 5.5 for evaluation details). This dataset includes synchronized recordings from photoplethysmography (PPG) and electrocardiography (ECG), two complementary methods for measuring heart rate. PPG estimates heart rate through changes in blood volume, while ECG derives heart rate from the electrical signals of cardiac cycles, typically by identifying R-wave peaks within the QRS complex, which correspond to individual heartbeats. The recordings were segmented into 8-second overlapping windows with a 2-second shift. In each window, heart rate was estimated by detecting signal peaks corresponding to heartbeats, resulting in beats per minute

(BPM) values. This approach produces a new heart rate measurement every 2 seconds, offering a high-resolution view of physiological changes over time.

Data were collected from 15 healthy participants (seven male, eight female), aged 22 to 39 years, during a 2.5-hour recording session in real-world, free-living conditions. These participants performed a series of predefined activities, including sitting, standing, walking, running, cycling, climbing stairs, and transitioning between these states. Each segment of physiological data was labeled with corresponding activity annotations, providing a structured overview of the participant's physical state. These labels helped distinguish between low-energy-demand activities (e.g., sitting, standing still, baseline, or "no_activity" periods) and high-energy-demanding activities (e.g., walking, cycling, or running).

To provide clarity on the structure of the PPG-DaLiA dataset, Table 1 presents a summary of its key signals and annotations. This summary outlines the main physiological recordings and their corresponding features, which form the basis for subsequent preprocessing and analysis in this study.

Table 1. Summary of the PPG-DaLiA dataset [19], listing the key physiological signals and annotations used in this study.

Dataset Component	Description	Unit / Format
Timestamp	Time index of the recording	Seconds (s)
PPG	Photoplethysmography signal measuring blood volume changes	Arbitrary units (raw signal)
ECG	Electrocardiography signal of cardiac electrical activity; HR derived from R-wave peaks in the QRS complex	Milivolts (mV)
Activity_label	Annotated activity class (e.g., eating, walking, cycling, working, sleeping)	Categorical label
Hr_bpm	Heart rate computed from PPG/ECG peak detection, expressed as beats per minute	Beats per minute (BPM)
Window_id	Segmented 8-second window with 2-second shift for analysis	Index

This classification provided a solid foundation for developing and testing the system's heart rate interpretation logic, due to its synchronized data and structured activity labels. The activity labels were crucial for identifying abnormalities: elevated or decreased heart rate values were only considered significant during low-energy-demand conditions. This approach aligns with clinical reasoning, where physiological anomalies at rest are more concerning than those during exertion. While this method relies on activity labels from the dataset, it is worth noting that such precise activity recognition may not be readily achievable in real-world settings.

Figure 5 illustrates the structured mapping between heart rate values and activity labels in the PPG-DaLiA dataset.

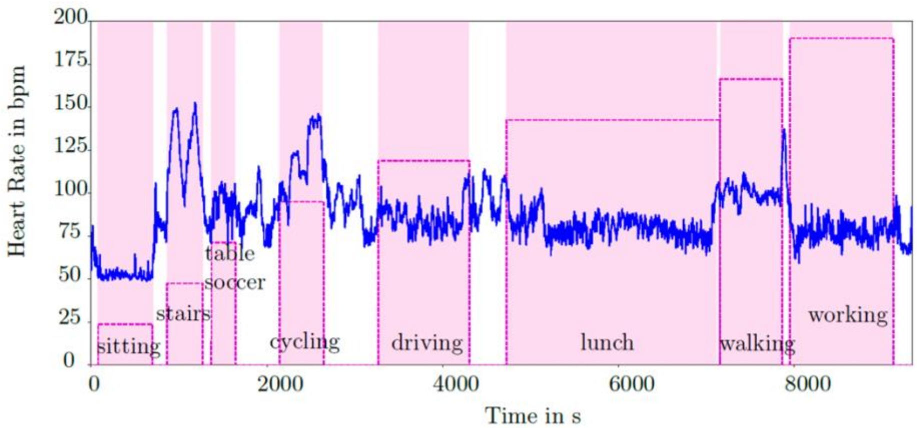


Figure 5. Data collection protocol from the PPG-DaLiA dataset [19], illustrating activity labels and corresponding ECG-based heart rate signals. White segments indicate transient periods between activities.

5.2.2 Simulated Dataset

Building on the structure and logic of the PPG-DaLiA dataset, two customized heart rate simulation scenarios were developed using Python to facilitate controlled testing of the system's classification logic. This classification logic was based on predefined clinical thresholds that categorize heart rate values into medically relevant states, such as normal resting values or an abnormal, elevated heart rate classified as tachycardia. Abnormal episodes lasted at least 10 minutes during periods of inactivity, surpassing established thresholds. These thresholds consider demographic parameters, such as age and gender, in conjunction with widely accepted clinical benchmarks [16]. For example, the system assumes that a woman aged 65 or older with below-average cardiovascular fitness may have a healthy resting heart rate between 77 and 84 beats per minute (bpm). While a man in the same age group may have a range of 74 to 79 beats per minute [20].

The dataset included two virtual users with distinct but elderly demographic profiles, reflecting the target group of this study. Older adults are especially important in this context because abnormal heart rate episodes are more common and clinically significant in this group, and they also tend to have less technological experience, making clear and interpretable system feedback especially vital. The first, Alessa, is a 67-year-old female identified as the healthy user, with a baseline resting heart rate of 84 beats per minute (bpm). The second, Bryan, is a 70-year-old male modeled to simulate a realistic alert-triggering scenario. Although his baseline heart rate is 79 bpm, Bryan experiences abnormal elevations during low-energy activities. This abnormal profile enabled the system to test its detection, interpretation, and explanation pipeline under conditions similar to those of early warning signs in real-world health monitoring.

To simulate a realistic yet controlled daily routine, both users were assigned an 8-hour schedule composed of six activity segments: sleep (60 minutes), work (180 minutes), walk (30 minutes), exercise (60 minutes), meal (30 minutes), and leisure (90 minutes). For each segment, heart rate values were generated by applying randomized, activity-specific multipliers to each user's baseline rate, along with low-amplitude noise to reflect physiological variability. This process can be formalized as:

$$HR_t = HR_{baseline} \times M_s + \epsilon$$

Where:

- $HR_{baseline}$: baseline heart rate for the virtual user (e.g., 79 bpm for Bryan)
- M_s: multiplier linked to the activity segment (e.g., 0.95-1.05 for sleep, 1.3-1.6 for exercise)
- ϵ, random noise from a Gaussian distribution representing physiological variability
- t, each time step (every 2 seconds in the simulation)

Low-energy segments such as sleep, meal, work, and leisure served as baseline contexts for detecting abnormal heart rate patterns.

Following the same temporal segmentation logic as the public dataset, the signals were processed using an 8-second sliding window with a 2-second shift, producing a new heart rate estimate every 2 seconds. This yielded 14,400 data points per user across the simulated 8-hour period (from 08:00 to 16:00), each paired with a timestamp and activity label. The resulting data were exported as synchronized CSV files, enabling reproducible, interpretable testing of the system's classification and reasoning capabilities.

Figure 6 illustrates one of the simulated datasets generated for this study, specifically the abnormal heart rate scenario. It shows timestamped heart rate values across an 8-hour simulated day, with color-coded segments representing different activity periods. The red-highlighted segment indicates an abnormal elevation in heart rate during the meal period, a low-energy-demand activity. Although Bryan's baseline resting heart rate was defined at 79 bpm for simulation purposes, values in this segment rose above 100 bpm. This exceeds the clinically accepted resting range of 60–100 bpm for a 70-year-old adult and was therefore modeled as a tachycardia-like event. The anomaly was intentionally introduced to test whether the system could identify potentially concerning physiological responses in contexts where an elevated heart rate is unexpected.

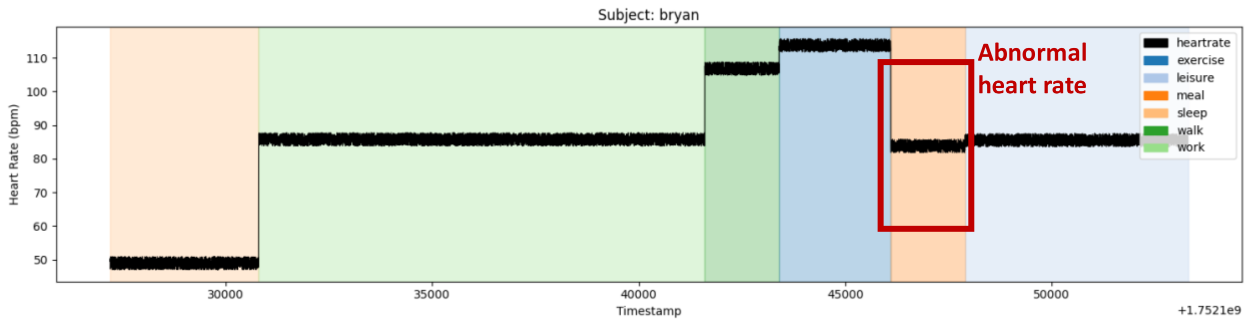


Figure 6. Simulated dataset for the virtual subject Bryan (70-year-old male, baseline HR: 85 bpm) over an 8-hour day. Black points indicate time-stamped heart rate estimates generated every 2 seconds. Colored background segments correspond to the six simulated activities: sleep, work, walking, exercise, meal, and leisure. This structure enables the system to interpret heart rate trends within context, flagging abnormal elevations during periods of low energy demand, “meal” as potentially concerning in this case.

5.3 Data Analysis and Summaries Generation

To ensure that the system could interpret heart rate data meaningfully and in a format suitable for language-based reasoning, a dedicated pre-processing step was implemented to convert raw simulation logs into structured and semantically rich datasets. This stage involved generating three intermediate .csv files and one final .json summary per user. These structured outputs ensured that both physiological data (heart rate) and behavioral context (activity) were aligned correctly and formatted for downstream interpretation. This step is crucial because the final system relies on structured, textual summaries rather than raw numbers as input for its language model–based interpretation.

5.3.1 Structuring Raw Heart Rate and Activity Logs

The first stage of data preparation involved generating three structured .csv files for each user, each serving a specific purpose in transforming raw simulation outputs into an interpretable dataset.

The first file, **user_hr.csv**, recorded heart rate readings every two seconds throughout the whole 8-hour simulation period. This file captured the user’s complete physiological trace with high temporal resolution, preserving detailed fluctuations in heart rate over time.

The second file, **user_activities.csv**, documented the user's daily routine in broader segments. Each entry included a “start_time”, “end_time”, and an associated “activity” label such as "sleep", "work", or "exercise". These activity blocks typically spanned several hours, rather than matching the per-second resolution of the heart rate data.

To create a unified view of the user’s physiological and behavioral state, we generated a third file, **user_merged.csv**, by aligning the two datasets. For each heart rate entry, the system identified the corresponding activity based on the timestamp and assigned it as a new label. As a result, a new row was generated in the merged file, combining every two-second timestamp with a heart rate value and its associated activity.

Out of the three files, only the merged dataset (**user_merged.csv**) was used as input for the LLM pipeline. This file provided the combined temporal alignment of heart rate data and activity labels necessary for retrieval and prompt construction. The other files (user_hr.csv and user_activities.csv) served as intermediate outputs during preprocessing and were not directly accessed by the LLM.

5.3.2 Summaries Generation: Preparing Data for Interpretation

A final summarization step was performed to generate a .json file that would support more efficient interpretation by the system. Unlike the .csv files, which preserved high-resolution time series data, the .json summary was designed to condense the essential information into hourly snapshots. These summaries enabled a structured representation of physiological and behavioral trends, providing consistently formatted inputs suitable for semantic parsing and natural language generation.

Each entry in the .json file corresponded to a single hour-long segment, beginning at hh:00:00 and ending at hh:59:59. For every such interval, the system extracted and stored key statistical features: “the minimum heart rate”, “maximum heart rate”, “mean heart rate”, and the “most frequent activity”. These fields are defined in Table 2, which presents the structure and meaning of each element included in the hourly summaries. Together, these summaries provided a structured, high-level view of the user's physiological and behavioral state across the simulated hours.

The decision to use hourly summaries was made to strike a balance between temporal resolution and interpretability. It can also be understood as a pre-retrieval strategy, consistent with frameworks outlined in recent systematic reviews [22]. An hour-long window is long enough to capture meaningful trends in heart rate and activity, while still aligning with how users typically refer to time in natural conversation (e.g., “around 1:30 PM”). In such cases, the system retrieves the summary for the 1:00–1:59 PM interval, ensuring that the response is grounded in a semantically relevant and temporally appropriate chunk. This design approach reduces time-related ambiguity, improves coherence during question answering, and helps minimize hallucinations by constraining the model’s input to structured, factual summaries.

In the simulated setup, this process resulted in nine hourly summaries per user, covering the monitoring period from 08:00 to 16:00. Each entry in the .json file followed the structure shown below:


```
{
  "start_time": "2025-07-10 08:00:00",
  "end_time": "2025-07-10 08:59:59",
  "min_hr": 47.85,
  "max_hr": 51.84,
  "mean_hr": 49.85,
  "most_common_activity": "sleep"
}
```

Table 2. Structure of the preprocessed hourly summaries used for semantic retrieval and prompt grounding.

Data Attribute	Description	Format
Start_tim	Start timestamp of the hourly segment	DD-MM-YYYY hh:mm:ss *(e.g., 05-04-2025 9:00:00)*
End_time	End timestamp of the hourly segment	DD-MM-YYYY hh:mm:ss *(e.g., 05-04-2025 9:59:59)*
Min_hr	Minimum heart rate recorded within the hourly segment	Float *(e.g., 53)*
Max_hr	Maximum heart rate recorded within the hourly segment	Float *(e.g., 94)*
Mean_hr	Mean heart rate calculated over the entire hourly segment.	Float *(e.g., 72)*
Most_common_activity	Most frequenctly detected activity during the hour	String *(e.g., "Work")*

5.3.3 Scenarios Configuration and Alert Triggering

Scenario A: Alert-triggering abnormal profile

The prototype incorporated an abnormal heart-rate alert, mimicking real-world systems that flag deviations from expected resting heart rate ranges for follow-up. The thresholding was based on physiological literature discussed in Section 5.2.2.

Building on this, the alerting mechanism was configured with a conservative sensitivity threshold for the simulated case of an old male with a baseline resting heart rate of 79 bpm, ensuring reproducibility and alignment with the dataset's demographic structure. An alert was triggered when the following conditions were met:

- **Context:** the activity label indicated inactivity (NO_ACTIVITY, BASELINE).
- **Threshold:** heart rate > 90 bpm (male cutoff).
- **Persistence**: the elevation was sustained for ≥ 10 minutes, equivalent to ≥ 300 consecutive 2-second samples (the simulated dataset produced a new estimate every 2 seconds using 8-second windows shifted by 2 seconds).

When these criteria were met, the system flagged the corresponding time span and stored it as an *alert context*. This storage mechanism enabled later retrieval without recomputation, ensuring that any reference to the alert remained anchored to the same underlying data. The retrieval process for follow-up queries using these stored contexts is described in Section 5.4.1.

Scenario B: Healthy baseline profile

For this case, an old female with a baseline resting heart rate of 84 bpm was modeled as the healthy case. For this scenario, no abnormal alerts were configured; instead, her data were used to validate system interpretability in non-alert conditions. Queries focus only on time-based retrieval and summarization of logs, without triggering the alerting mechanism. The retrieval process for queries using this context is described in Section 5.4.1.

This dual-scenario design allowed the prototype to demonstrate both (i) the system's ability to flag and explain abnormal heart-rate events in an abnormal heart rate case, and (ii) its ability to support standard interpretability and reflection for a healthy-patient case.

5.4 Interpretation Layer

To generate responses that are contextually accurate, personally relevant, and grounded in real user data, this system implements an integrated interpretation layer composed of three core components: a Retrieval-Augmented Generation (RAG) pipeline, a locally deployed large language model (LLM), and an interactive conversational interface. Together, these elements form the backbone of the assistant's interpretive logic, retrieving semantically relevant health summaries, processing them through prompt-based reasoning, and delivering responses through a user-facing interface. This section details how each component was designed and implemented to support trustworthy, real-time dialogue grounded in simulated heart rate data.

5.4.1 Retrieval-Augmented Generation (RAG) for Grounded Responses

To ensure that the assistant's responses remain grounded in factual, patient-specific data and to mitigate the risk of hallucinations often associated with Large Language Models (LLMs), we implemented a Retrieval-Augmented Generation (RAG) pipeline customized for our simulated heart rate summaries. This approach enables the system to retrieve relevant physiological insights in real time and condition the language model's output on that retrieved context. The process begins by converting hourly summary entries into semantically rich natural language text. These texts are then embedded into numerical vector representations, indexed for similarity search, and retrieved dynamically after the user submits a question. By embedding this factual memory layer into the prompting strategy, the assistant can produce responses that are not only interpretable and grounded in real trends but also tailored to individual users.

Following the RAG workflow outlined in the most recent retrieval augmented generation for large language models systematic review and survey [17, 31], our implementation can be divided into four key stages, as shown in Figure 7.

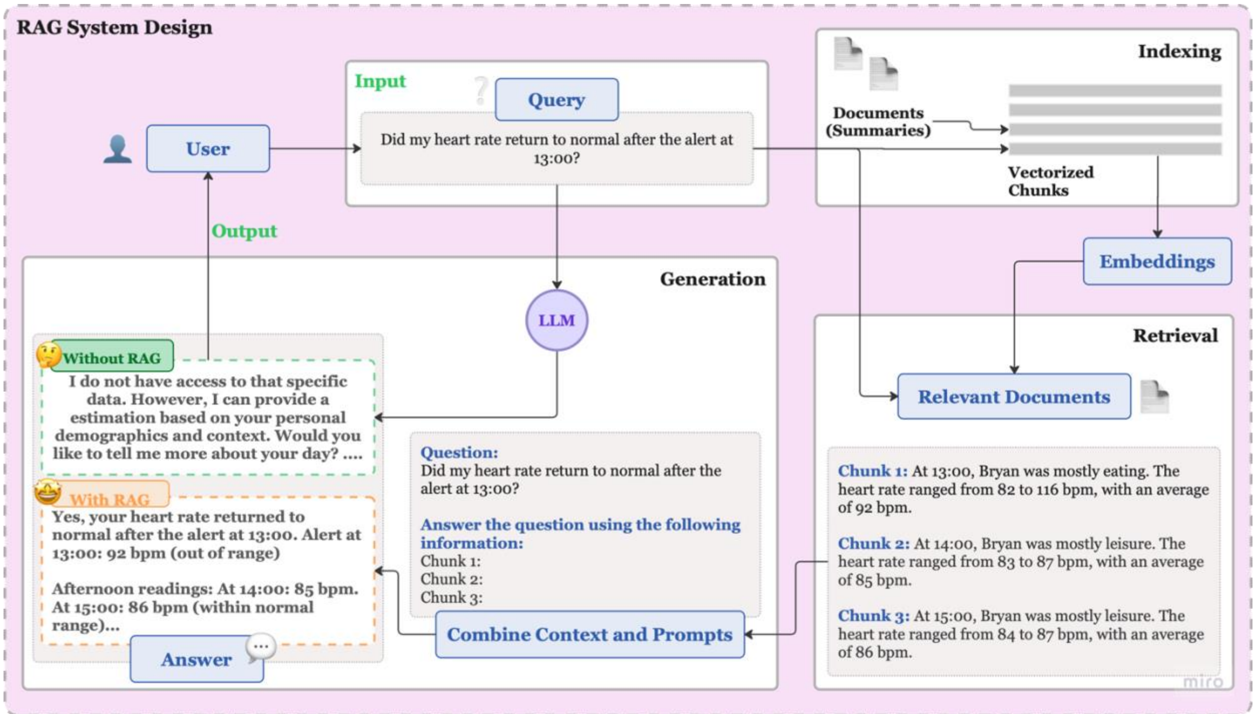


Figure 7. Overview of the Retrieval-Augmented Generation (RAG) pipeline applied to heart rate interpretation. The system processes structured physiological summaries, retrieves semantically relevant chunks based on the user query, and generates context-grounded responses using a locally hosted Large Language Model (LLM). A comparison with a non-RAG response is included to illustrate the system's ability to generate personalized and data-grounded answers.

Indexing. This process begins by processing structured hourly summaries generated from simulated physiological data. To prepare this data for interaction with a retrieval-augmented language model, each entry is converted into a plain-text sentence using a deterministic, template-based formatter. This step ensures semantic clarity and makes the content compatible with language model input constraints, which include the minimum, maximum, and mean heart rate, as well as the most frequent activity label. The template used was:

$$chunk = (\, f\text{"At } \{time\_str\}, \{subject.capitalize()\} \text{ was mostly } \{activity\}.\text{ "} \\ f\text{"The heart rate ranged from } \{min\_hr\} \text{ to } \{max\_hr\} \text{ bpm, with an average of } \{mean\_hr\} \text{ bpm."}\,)$$

Each formatted sentence is treated as a standalone chunk to accommodate the language models' limited context window and support efficient retrieval, resulting in 16 chunks across two users as:

01. At 8: 00, Alessa was mostly sleep. The heart rate ranged from 48 to 52 bpm, with an average of 50 bpm.
02. At 09:00, Alessa was mostly work. The heart rate ranged from 72 to 76 bpm, with an average of 74 bpm.
03. At 10:00, Alessa was mostly work. The heart rate ranged from 72 to 76 bpm, with an average of 74 bpm.
04. At 11:00, Alessa was mostly work. The heart rate ranged from 72 to 76 bpm, with an average of 75 bpm.
05. At 12:00, Alessa was mostly walk. The heart rate ranged from 83 to 98 bpm, with an average of 90 bpm.
06. At 13:00, Alessa was mostly meal. The heart rate ranged from 67 to 98 bpm, with an average of 77 bpm.
07. At 14:00, Alessa was mostly leisure. The heart rate ranged from 72 to 76 bpm, with an average of 74 bpm.

08. At 15:00, Alessa was mostly leisure. The heart rate ranged from 72 to 76 bpm, with an average of 74 bpm.
09. At 08:00, Bryan was mostly sleep. The heart rate ranged from 47 to 51 bpm, with an average of 49 bpm.
10. At 09:00, Bryan was mostly work. The heart rate ranged from 84 to 88 bpm, with an average of 86 bpm.
11. At 10:00, Bryan was mostly work. The heart rate ranged from 84 to 88 bpm, with an average of 86 bpm.
12. At 11:00, Bryan was mostly work. The heart rate ranged from 84 to 88 bpm, with an average of 86 bpm.
13. At 12:00, Bryan was mostly walk. The heart rate ranged from 105 to 116 bpm, with an average of 110 bpm.
14. At 13:00, Bryan was mostly meal. The heart rate ranged from 82 to 116 bpm, with an average of 92 bpm.
15. At 14:00, Bryan was mostly leisure. The heart rate ranged from 83 to 87 bpm, with an average of 85 bpm.
16. At 15:00, Bryan was mostly leisure. The heart rate ranged from 84 to 87 bpm, with an average of 86 bpm.

The activity labels in these chunks (e.g., sleep, meal, leisure) are preserved exactly as they appear in the dataset to ensure reproducibility, even if some labels do not align perfectly with natural conversational phrasing. These chunks were encoded into dense vector representations using the *SentenceTransformer* model *all-MiniLM-L6-v2* from HuggingFace, which produces 384-dimensional embeddings optimized for semantic similarity tasks. To prepare them for cosine similarity search, all embedding vectors were normalized to unit length according to:

$$\hat{v} = \frac{v}{||v||}$$

Where v is the raw embedding vector, $\hat{v}$ is the normalized unit vector, and $||v||$ represents the magnitude (length) of the vector.

The resulting vectors were stored as a serialized file and indexed using FAISS with the IndexFlatL2 configuration, which computes inner product similarity equivalent to cosine similarity under vector normalization. While cosine similarity was chosen for its widespread use and reproducibility in semantic retrieval tasks, alternative similarity measures such as the (unnormalized) dot product between learned embeddings [49] could be explored in future iterations to examine potential trade-offs in retrieval performance. The resulting vector index was stored in a separate file-based vector database for each user (serialized as .pkl files using FAISS).

This lightweight setup enabled user-specific retrieval and maintained isolation between simulated patient profiles while supporting local retrieval operations. In total, the index comprised 16 vectors associated with their corresponding text chunks, allowing for efficient and reproducible retrieval of semantically relevant summaries during interaction.

Retrieval. The retrieval stage activates once a user submits a query through the assistant interface. To enable semantic matching between the query and the pre-processed summaries, the query is embedded using the same SentenceTransformer model (all-MiniLM-L6-v2) applied during indexing. Both query and chunks exist in the same 384-dimensional vector space, and the query embedding is normalized to unit length so that cosine similarity can be applied against the user-specific FAISS index.

When a query contains a **time reference** (e.g., “How was my heart rate at 11:00?”), the system maps the input to the nearest hourly summary and retrieves that single chunk (11:00–11:59 in this case). This one-to-one mapping ensures temporal precision and prevents the assistant from introducing ambiguity in its responses. In contrast, for queries without a specific hour reference but still **general time-based** (e.g., "Did my heart rate return to normal after the alert?"), the system performs a full similarity search across all summaries. It retrieves the top-k = 3 most relevant chunks (13:00-13:59,14:00-14:59, 15:00-15:59 in this case). This design choice was informed by common temporal expressions observed during testing, such as "later," "morning," or "afternoon," which usually span multiple consecutive hours. Selecting the top 3 chunks allowed the system to offer enough context for such periods without exceeding token limits or introducing irrelevant content. This balance enabled the assistant to generate grounded and informative responses for broader temporal queries.

In the case of **alert follow-up** queries, the system directly reuses the specific hourly chunk that triggered the abnormality alert. This chunk serves as the sole input for the assistant's explanation, ensuring consistency between the initial alert and the system's follow-up interpretation. The details of this alert-triggering data were described in Section 5.3.3.

Similarly, when a user explicitly requests a **broader summary** of the entire monitoring period (e.g., "How was my heart rate today?"), the system retrieves all available chunks for that user (eight in the simulated dataset, i.e., 8:00-15:59). These are then aggregated and passed to the generation stage, allowing the assistant to provide a coherent overview of the full day's heart rate and activity patterns.

Generation. In the final stage of the RAG pipeline, the system generates a structured prompt by combining the user's original query with the retrieved hourly summaries. This prompt is then passed to a locally hosted large language model (LLM) to produce the final response.

To ensure safe and consistent behavior, the prompt follows a fixed template designed to produce semantically grounded responses. It includes: (1) the retrieved plain-language summaries, and (2) explicit instructions to the model, such as: responses must rely solely on the provided summaries, avoid any form of medical advice, and maintain a clear, empathetic tone suitable for health-related communication. A complete template and worked example for a specific query type is provided in Section 5.4.2 (Figure 8). This structure is enforced programmatically to standardize the assistant's behavior across queries.

By grounding generation in semantically relevant content, the system reduces hallucination risks and ensures outputs remain factually aligned with the user's logged data. This work adopts a vector similarity–based definition of semantics, where the retrieved summaries serve as the sole input for the assistant's explanation, ensuring consistency with the user's logged context. As an alternative, ontology-based methods [50], which provide standardized, semantically rich vocabularies for representing domain knowledge, could be explored in future work to enhance consistency, interoperability, and contextual understanding in semantic retrieval.

5.4.2 LLM-Based Interpretation and Prompt Design

The large language model employed in this system is LLaMA 3.2 [32], an open-weight model from Meta's LLaMA (Large Language Model Meta AI) family. This version was chosen for its strong performance in instruction-following and reasoning tasks, as well as its lightweight deployment requirements, making it suitable for local execution without reliance on external cloud services or proprietary APIs. This local setup provides greater control over customization and supports the use of privacy-preserving measures with potentially sensitive personal health information.

Running LLaMA 3.2 offline also facilitates faster iteration during prompt engineering, allowing complete control over the assistant's tone, instructions, and behavioral constraints. To steer the model's behavior, we employed a structured prompt engineering approach, experimenting with various formulations to refine the assistant's role and constraints. The final design embeds strict rules directly in the prompt to avoid invented values and explicitly references only the log data provided. These prompts are dynamically generated based on the type of user query and follow a structure informed by prompt engineering literature [33].

Despite these safeguards, some variability in generated responses persisted due to the probabilistic nature of large language models. To minimize this, a low temperature setting (0.7) was used during inference. This value was chosen to balance helpfulness and response coherence, promoting more stable outputs while maintaining conversational tone. While this configuration helped reduce hallucination and excessive variation, minor inconsistencies in response length and detail still emerged, which is especially relevant in health-related contexts where clarity and predictability are crucial. This variability, although reduced, remains a known limitation and a significant risk in the deployment of generative models in healthcare, due to the complexity and sensitivity of the domain.

The system supports four interaction types that determine how context is selected and how the prompt is framed for the language model:

1) **Specific time-based questions:** For queries that contain an explicit time reference, the system retrieves the corresponding hourly summary. As described in Section 5.4.1, this ensures that the prompt is grounded in a single, well-defined log segment, keeping the response precise and context-specific.

2) **General time-based queries**: As outlined in Section 5.4.1, this approach applies when queries do not specify an exact time but include relative temporal identifiers such as "after," "before," "morning," or "afternoon." In these cases, the three most relevant log intervals are selected and aggregated, providing a broader contextual basis for the prompt.

3) **Summary questions**: In cases where the user requests a complete overview of the monitoring period, the system incorporates all available summaries for the day. As explained in Section 5.4.1, this provides the prompt with a full contextual scope, supporting day-level synthesis and interpretation.

4) **General health and wellness reflections**: For wellness-oriented queries that are not anchored to specific log segments, the system bypasses the structured summaries and instead prompts the model to generate broader reflections. This mode poses a higher risk, as responses may draw on the model's pre-trained general knowledge rather than strictly on the logged data. To mitigate this risk, the prompt includes explicit safeguards: the model

is instructed to avoid diagnostic statements, use non-clinical and supportive language, and frame its outputs as empathetic reflections rather than medical advice.

5) **Alert follow-up:** In cases where the user refers back to a system-generated alert, the assistant reuses the specific log segment that initially triggered the warning (see Section 5.3.3, "Alert-triggering data"). Anchoring the response in the same chunk that produced the alert ensures consistency and transparency. The assistant explains the flagged abnormality by reporting the heart rate range observed during the alert and situating it within the corresponding activity context.

The structured prompt used in the generation stage comprises four main components, as summarized in Table 3.

Table 3. Overview of the main components used in the prompt structure and their specific implementation within the heart rate assistant system.

Component	Definition	Application in This Project
Instruction	The task the model is expected to perform, often used to set the assistant's identity and boundaries.	Specifies that the assistant acts as a personal heart rate assistant, explicitly excluding any diagnostic or clinical responsibilities.
Context	External reference material that guides the model toward accurate or relevant outputs.	Includes summaries of log chunks retrieved via semantic search, aligned with the user's data.
Input Data	The user's prompt, question or statement submitted for the model to respond to.	User's question, such as a specific time query or general health concern.
Output	A label or signal used to categorize or guide the expected form or scope of the model's response.	Signals what the assistant should focus on (e.g., summarizing the day, explaining an alert, or answering a follow-up).

Figure 8 presents a worked example of the generation step for a time-based query. It demonstrates how the assistant integrates the retrieved log segment into the system prompt and frames explicit behavioral constraints. It combines this with the user's question before sending it to the LLM, which produces an empathetic, data-grounded response. Additional examples covering the other prompt cases are provided in Appendix F for completeness.

5.4.3 Conversational Interface

The final layer of the prototype is a user-facing conversational interface built using **Streamlit** [34], an open-source Python framework that enables rapid development and deployment of custom web applications for data-driven systems. To reproduce this component, install Streamlit via **pip install streamlit**, then launch the interface locally using streamlit run **app.py**. The main script (app.py) orchestrates both frontend layout and backend logic in a modular pipeline.

This interface acts as the main interaction point between users and the system and includes:

- A dropdown menu to select one of two predefined simulated users (defined in Section 5.3.3).
- A greeting header.
- A free-text input field for user questions.
- A chat-style response area rendered using **st.chat_message()** to display the assistant's answers.

All computation, including input parsing, log retrieval, prompt construction, and LLM response generation, occurs server-side. The interface remains lightweight and responsive, with Streamlit's **session_state** used to preserve conversational context across queries..

Figure 9 illustrates two example interactions:

- **Left panel (Scenario A):** The assistant responds to an alert for Bryan, an elderly male patient with an abnormal profile. It explains a flagged log segment where an elevated heart rate occurred during inactivity.

- **Right panel (Scenario B):** The assistant answers a time-specific query for Alessa, a healthy elderly female patient. It retrieves the relevant log summary and contextualizes heart rate values in relation to the reported emotional state.

These examples demonstrate how the assistant adapts to both the selected user profile and the type of query, whether alert-related or time-specific. Additional examples of interaction types and system responses are provided in Appendix G. The chosen user ID dynamically personalizes thresholds, simulated log access, and alert logic. This modular and context-aware design supports reproducibility, ensuring consistent, scenario-specific responses across various interaction types.

While the assistant's responses are grounded in retrieved heart rate summaries, open-ended or emotionally framed questions may still trigger interpretive statements that go beyond the available data. This introduces a risk of subtle hallucinations, especially when the model infers emotional states or behavioral patterns without direct physiological indicators. Although prompt instructions are designed to constrain such behavior, this remains a key consideration for future refinement and is discussed further in Section 8.2.

Example: Specific time-based query

```
context = get_closest_chunk_by_time(user_minutes, texts, user_id)


response = client.chat(
  model="llama3.2",
  messages=[
    {
      "role": "system",

       "content": (
        " You are a personal and empathetic assistant helping users understand their
      heart rate and activity context.

      - Do not mention fallback, approximation, or lack of context.
      - Do not explain the system behavior.
      - Do not guess, generalize, or invent values.
      - Assume the context below is the exact match for the user's request. Base your
      answer only on that.
      - Explain clearly the heart rate range, average, and activity exactly as in the context.
      - Never go beyond the provided data.

      Context:  {context}"
      ),
    },

    {"role": "user", "content": user_question},
  ],
)
```

Real Question and Answer

*{ **user_question** }:"What was my heart rate around 11:30? Because I was dizzy"*

*{ **answer** }:"I'm so glad you reached out! Feeling dizzy can be really unsettling. According to your data, at 11:30 your heart rate was mostly within a range of 72 to 76 beats per minute (bpm), with an average of 75 bpm. It's great you're paying attention to your body and noticing how it's responding to different activities!"*

Figure 8. Example of the generation process for a time-based query, showing how retrieved context, system instructions, and the user's question are structured and passed to the LLM to generate the final response.

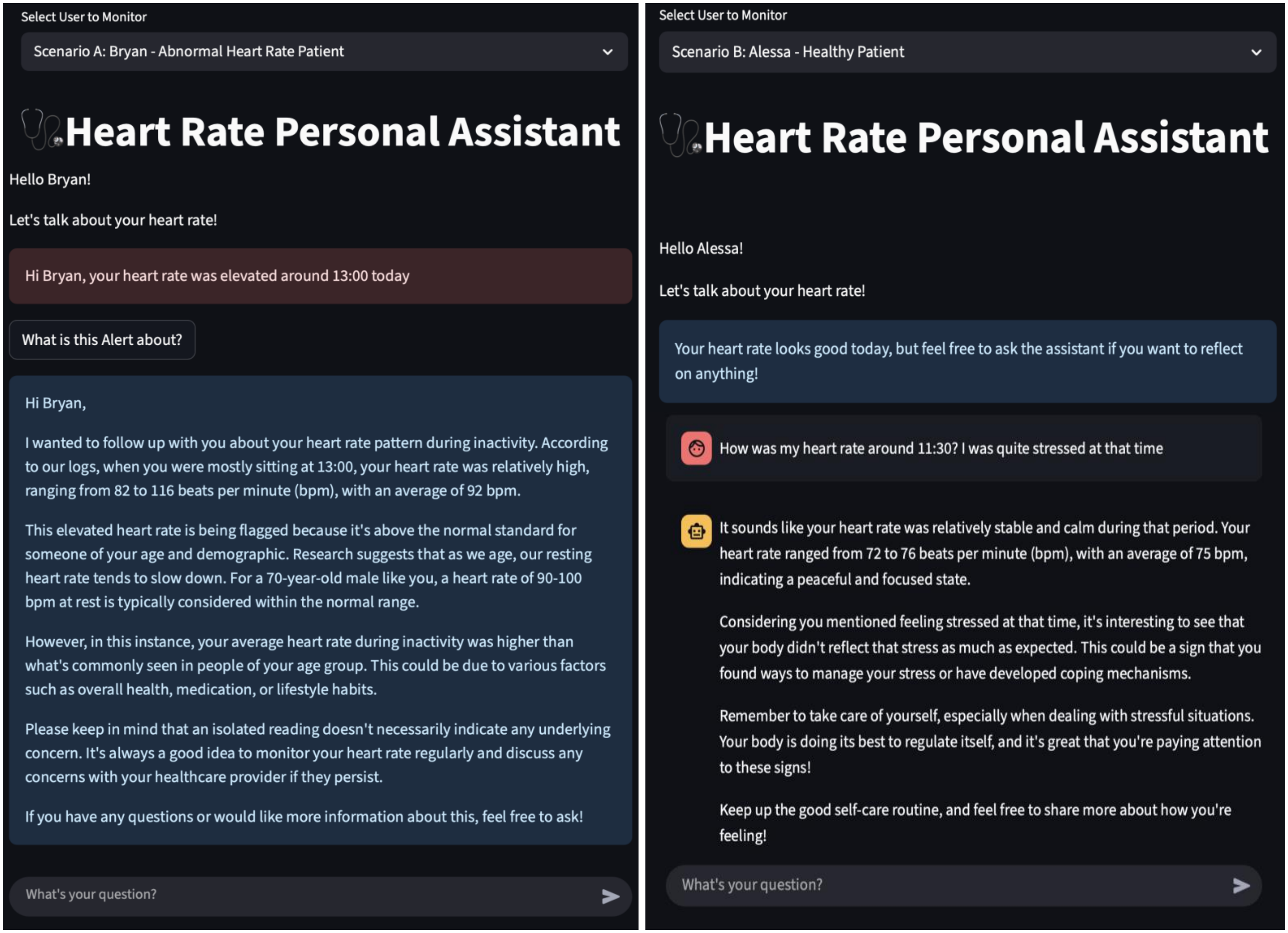


Figure 9. Two examples from the Streamlit interface: (left) an alert-related interaction where the assistant explains elevated heart rate during inactivity; (right) a time-specific query where the assistant provides a contextualized summary of heart rate data for the hour inquired. For simplicity, the interface displays the start time of the hour (e.g., "at 13:00"), although technically the alert and retrieval correspond to the full hourly window (13:00–13:59).

5.5 System Iterations and Technical Evaluation Design

This section introduces the three iterative prototype versions, followed by an overview of the technical evaluation setup and the metrics and datasets used. The actual performance results of the prototype versions and their interpretation are presented later in Chapter 7.

5.5.1 Prototype Versions

To support both the experimental and user-centered objectives of this research, three prototype versions (V1, V2, V3) were developed across successive design iterations. These iterations were informed by internal testing and insights gathered from user interactions, aligning with the two phases of the user study described in Chapter 6.

Each version explored specific design trade-offs between retrieval scope, interpretability, faithfulness, and personalization. Together, they laid the foundation for both the user evaluation (discussed in Chapter 6, which includes aspects such as perceived clarity and trust) and the technical evaluation, whose results are reported and interpreted in Chapter 7.

- **Version 1 (V1)**, tested during the first phase, implemented timestamp-based chunk retrieval and rule-based personalization. However, it revealed limitations in conversational flow, vague responses to open-ended prompts, and a lack of interpretive framing for retrieved data.

- **Version 2 (V2)**, introduced in the second phase, expanded the scope of retrieval and adjusted prompt structures to better support general or non-specific queries. While this increased contextual richness, it also introduced new challenges related to verbosity, reduced personalization, and inconsistencies in factual grounding.

- **Version 3 (V3)**, deployed in the final phase, incorporated fallback mechanisms, age-based comparisons, and stricter prompt instructions. These adjustments improved contextualization and user alignment but introduced challenges in maintaining concise and factually faithful outputs.

Each prototype thus represents a progressive effort to balance interpretability, retrieval quality, and safe communication in the context of health data interpretation.

5.5.2 Evaluation Framework

The system was evaluated using the **Retrieval-Augmented Generation Assessment (RAGAs)** framework [35], a methodology designed to assess both retrieval quality and generated responses in RAG pipelines. To ensure fairness, a **cross-model** setup was adopted in all evaluation cases: while the assistant itself ran on a local LLaMA 3.2 model, evaluation was performed using Mistral 7B [47] as the external evaluator. This minimized architectural bias and improved the robustness of the results.

All three prototype iterations (V1, V2, and V3) were assessed with the RAGAs metrics of **context relevance, faithfulness, and answer relevance** to capture how retrieval accuracy, factual grounding, and response alignment with user intent evolved across versions. These iterations are more detailed and interpreted in Chapter 7. Following the iteration evaluations, the best-performing version was selected for **baseline comparison** against a **plain LLaMA 3.2 model** without retrieval. For this step, the assessment was extended to include **Factual Correctness (F1 mode**), providing an architecture-independent measure of factual overlap.

Together, these complementary metrics measured improvements from iterative refinements in retrieval logic, prompt design, and context management, as well as answer correctness. They also enabled a fair assessment of the added value of retrieval compared to baseline models, complementing the findings from the user study presented in Chapter 6.

5.5.3 Evaluation Metrics and Datasets

The **evaluation dataset** was constructed using three representative questions designed to capture the full range of interaction types defined by the prompt framing in Section 5.4.2. These included time-specific queries, alert-related follow-ups, and general time-based reflections (see Appendix H for the complete dataset). While this dataset was used consistently across evaluations, in the baseline comparison, the retrieved context was excluded to isolate the effect of retrieval.

Prototype Iterations Evaluation Metrics. To evaluate the three prototype iterations (V1, V2, V3), the study applied the RAGAs metrics of:

- **Context Relevance**: evaluates whether the retrieved log chunks are semantically aligned with the user's question. This ensures that the assistant grounds its response in information relevant to the query, rather than unrelated logs.

- **Faithfulness:** evaluates whether the generated answer remains factually consistent with the retrieved context, without introducing unsupported or hallucinated content.

- **Answer Relevance:** evaluates how well the generated response addresses the user's question.

Each evaluation required three elements: **the user's question, the assistant's answer, and the retrieved context** [34, 35]. An example of a time-based query is shown below:

$$question = [\text{“}How\ was\ my\ heart\ rate\ around\ 11{:}30?\ I\ was\ stressed\ at\ that\ time.\text{”}]$$

answer = [*“At 11:30, your heart rate was stable while working. The readings ranged from 84 to 88 bpm, with an average of 86 bpm. Given your age (70) and activity level at that time, your heart rate is within a healthy range according to European Society of Cardiology guidelines (60-100 bpm).”*]

$$context = [\text{“}At\ 11{:}00, Bryan\ was\ mostly\ work. The\ heart\ rate\ ranged\ from\ 84\ to\ 88\ bpm,\ with\ an\ average\ of\ 86\ bpm.\text{”}]$$

The evaluation context in this setup is limited to recent log summaries and does not incorporate user-specific historical trends. For instance, while Bryan's heart rate is reported as 86 bpm, clinically within range, the assistant cannot assess whether this is elevated for him personally (e.g., if his usual resting rate is closer to 70 bpm). This absence of longitudinal context limits the assistant's ability to assess personalized deviations from baseline. A discussion of this limitation is provided in Section 8.2.

These elements were compiled into an evaluation dataset using EvaluationDataset.from_list() from the RAGAs Python library. To ensure reproducibility, all evaluations were conducted offline using Mistral 7B to minimize bias and without reliance on external APIs.

Baseline Comparison. Following the evaluation of prototype iterations, the best-performing version was selected for baseline comparison. This step measured the added value of the RAG pipeline by comparing it with a plain LLaMA 3.2 model without retrieval, setting the context parameter as an empty list *(context = [])*. For this evaluation, the Factual Correctness (F1 score) was implemented alongside answer relevance.

The F1 score measures factual overlap between system output and reference answers, combining precision (proportion of correct statements generated) and recall (coverage of relevant reference statements) into a harmonic mean [31, 37, 40]:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

This provided an architecture-independent metric, ensuring a fair comparison between retrieval-augmented and non-retrieval models. To ensure reproducibility, the implementation followed the naming conventions of the RAGAs Python library:

$$metrics = [ContextRelevance(), Faithfulness(), ResponseRelevance(), FactualCorrectness()]$$

Chapter 6
User Study

To evaluate how LLM-based conversational interfaces can support patient understanding of heart rate data in the context of Digital Twin systems, a user study was conducted under Ethical Application #251615 of the Ethics Committee Computer & Information Sciences at the University of Twente and under Study Protocol IM-NL-SP-2025-0011 approved by imec's internal ethics review process. The study addressed two research questions: (1) How can LLM-based conversational interfaces facilitate patients' comprehension and interaction with health data from Digital Twin systems? (2) What are the risks and limitations of using LLM-based conversational interfaces in Digital Twin systems, and how do these affect patients' trust and engagement with their health data?

The user study aimed to validate both the performance and user experience of the proposed conversational assistant for heart rate interpretation within a Digital Twin system. Conducted in a controlled laboratory setting, the study simulated interactions based on a predefined elderly user profile representing an older adult without medical expertise, to assess whether the assistant could deliver clear, personalized, and supportive explanations.

This chapter describes the participant criteria, study methodology, ethical considerations, prototype iterations tested, and the analysis of qualitative and quantitative results.

6.1 Participants

Participants were recruited internally at imec Netherlands via an internal mailing list and personal invitations. Recruitment materials outlined the study purpose, eligibility criteria, and voluntary participation. No incentives were offered.

6.1.1 Participant criteria

- **Inclusion:** Adults 18+ years, employees under an imec-Netherlands contract, proficient in English, comfortable with digital assistants, willing to consent to participation and audio recording.
- **Exclusion:** Individuals with cognitive impairments or an inability to understand the task or provide informed feedback.

6.1.2 Sample Size and Distribution

The study included a total of nine participants, divided across two phases. In the exploratory phase, three participants each completed a single session, and the main evaluation phase involved six participants, each completing two sessions spaced one week apart. This design allowed for the assessment of system improvements over repeated interactions while minimizing participant burden. It also helped evaluate participants' comprehension, interaction, and how these affect their trust and engagement with health data.

6.2 Study Phases

To structure the user study effectively, the process was divided into two main phases: an exploratory phase and an evaluation phase. The exploratory phase focused on early usability testing of Prototype Version 1 with three participants, while the evaluation phase introduced Versions 2 and 3 to six new participants. Each phase followed a structured protocol involving scripted scenarios, participant interaction, and feedback collection. The iterative findings that emerged are discussed in greater detail in Section 7.1, which analyzes system performance and user experience across versions.

6.2.1 Exploratory Phase

This one-week phase aimed to identify usability issues and improve clarity in the early version of the assistant. Three participants each took part in a single 50-minute session that included:

- 5-minute session for signed consent and introduction.
- 30-minute session of continuous interaction covering two scenarios (abnormal heart rate alert and normal heart rate).
- 15-minute semi-structured interview to collect feedback.

This phase was essential for capturing initial limitations and informing the design of the next iteration.

6.2.2 Evaluation Phase

This two-week phase engaged six different participants to assess the assistant's refinements in Versions 2 and 3. Participants attended two sessions spaced one week apart:

a) ***Week 2****:* A 50-minute session that included a 5-minute consent and introduction, a 30-minute interaction with prototype Version 2 covering both scenarios, aimed at identifying usability improvements from the first iteration, and 15 minutes for a semi-structured interview, which was audio recorded with prior participant permission.

b) ***Week 3****:* a 50-minute session that involved a 40-minute interaction with prototype Version 3, including both scenarios, focused on evaluating improvements made over the second iteration (week 2), and lastly, 10 minutes for a digital questionnaire.

6.3 Tasks and Scenarios

For each prototype session, we created different scenarios with the same structure, patient demographics, roles, situation, and study considerations. However, each had a unique sequence of questions tailored to the insights we aimed to gather (see Appendix A).

6.3.1 Scenario A: Abnormal Heart Rate Alert

In this scenario, the participant took the role of an older adult without medical or technical experience. The assistant started the interaction by notifying the user about an elevated heart rate:

"*Hi user, your heart rate was elevated around 13:00 today.*"

The suggested sequence of actions required the participant to press the button to get a summary of the alert and then ask follow-up questions to understand better why the value was flagged, whether other abnormalities were present, and how it compared to normal heart rate ranges. The assistant's role was to explain these trends clearly and factually in simple language, without offering medical advice.

6.3.2 Scenario B: Healthy Heart Rate

In this scenario, the participant acted as an older woman reviewing her daily activity without receiving alerts, but seeing this command in the interface, inviting the user to start the interaction:

"Your heart rate looks good today, but feel free to ask the assistant if you want to reflect on anything!".

The suggested sequence of actions encouraged the participant to ask the assistant about stable heart rate values, averages at specific times, or general reflections (e.g., stress or relaxation was reflected in the data). The assistant's role was to provide summaries of normal trends, reassure the user about typical values, and maintain an approachable, non-clinical tone.

6.4 Data Sources

6.4.1 Data Types

- **Observational Notes**: Manually recorded annotations of participants' on-task behaviors and verbal reflections during "think-aloud" sessions captured indicators like hesitation, timing of follow-up questions, perceived confusion, and reactions to system responses. This qualitative dataset was digitized and anonymized, serving as contextual evidence alongside interaction logs and interview transcripts to identify usability challenges, engagement, and areas for prototype refinement.

- **Semi-structured Interviews:** The interview questions followed a guided approach with ten open-ended questions (see Appendix B), ensuring consistency across participants while leaving room for personal experiences and unexpected insights. This format allowed participants to elaborate on moments of clarity, confusion, or trust, provide suggestions for system improvement, and share how supported or uncertain they felt during the interaction.

- **Audio Recordings:** Interviews were recorded using a smartphone microphone. The recordings allowed accurate transcription and analysis, maintaining data integrity.

- **Questionnaire:** The usability questionnaire combined a 5-point Likert scale with open-ended questions (see Appendix C), providing both quantitative scores and qualitative insights. The Likert items were informed by the principles of the Chatbot Usability Questionnaire (CUQ) and Bot Usability Scale (BUS-11), focusing on communication clarity, tone, trust, and role consistency. Open-ended questions asked participants to reflect on helpful aspects, areas for improvement, and comfort using the system independently, to identify improvements between interactions.

- **Interaction Logs**: Structured transcripts of the entire dialogue between participant and assistant include timestamps, system events, and anonymized user inputs and responses in chronological order. This dataset offers a detailed record of how the assistant processed queries and generated outputs, aiding analysis of system performance, response accuracy, and flow.

6.4.2 Data Collection

Interviews were audio-recorded with a smartphone with prior consent, transcribed using TurboScribe, and the recordings were deleted immediately after. Transcripts were exported and archived offline as PDF files. Questionnaire responses were gathered through Google Forms, then exported and saved offline as Excel and PDF files. Observational notes were digitized after each session and saved with the dataset. User–assistant interaction logs were automatically captured and saved offline as structured PDF files (user queries, system responses, timestamps, fallbacks). All materials were anonymized and stored securely on a password-protected device for analysis.

To protect participant confidentiality, all logs were stored offline on a password-protected device, and no data were transferred online or shared externally.

6.5 Ethical Considerations

Participants gave written consent after receiving a Subject Information Sheet (SIS), which outlined the study's purpose, procedures, risks, data handling, and participant rights (see Appendix D for full details).

6.6 Study Procedure

The study followed a usability testing framework [30], applied during both the exploratory and evaluation phases, and structured into the following steps:

Step 1: Session Setup. Each study session occurred in a meeting room where a laptop was prepared with the heart rate assistant ready. Before participants arrived, the researcher debugged and tested the prototype to prevent technical issues.

Step 2: Informed Consent Sign. Participants received the printed version of the Subject Information Sheet (Appendix D) outlining the study's purpose, procedures, data practices, risks, and rights, including the right to withdrawal at any time. The researcher explained this verbally, answered questions, and obtained signed consent before starting the session.

Step 3: Task Scenarios Execution. Printed papers with scenarios were provided, including descriptions of realistic use cases and tasks to follow. Participants engaged in two scenarios: A) Abnormal heart rate alert; and B) Healthy monitoring query. To minimize order effects, the order was counterbalanced: half of the participants started with one scenario type, the other half with the opposite.

Step 4: Interaction Logging. After each scenario interaction, the researcher saved anonymized user inputs, system responses, timestamps, and fallback triggers. These logs were exported and stored as PDF files for later analysis.

Step 5: Observational notes. Participants performed both tasks while the researcher observed and took notes.

Step 5: Post-Interaction Interviews. After each first interaction in both exploratory and evaluation phases, participants completed a semi-structured interview (Appendix B), which was audio-recorded with their prior approval.

Step 6: Post-Study Questionnaire. This step was only conducted after the final session involving Prototype Version 3 (Week 3 in the Evaluation Phase). Participants completed a Google Forms questionnaire (Appendix C) to evaluate various aspects of the interface, system responses, and the overall study.

Step 7: Data Handling and Usage. Observation notes, transcripts, and logs were stored securely offline and used exclusively for research analysis.

Step 8: Study Closure and Follow-Up. After completing the data collection, participants were thanked for their time and contribution. The researcher briefly summarized the next steps of the study and explained how the collected data would be used. Contact details were provided to participants who expressed interest in receiving updates about future prototype versions or the study results.

6.7 Qualitative Results

To analyze the qualitative data, a thematic analysis was conducted on the interview transcripts. The aim was to identify recurring patterns in participants' perceptions of the conversational assistant and link them to the two research questions systematically. All transcripts were imported into Atlas.ti and coded using descriptive and in vivo coding, allowing participants' quotes to enrich the analysis.

Initial codes were grouped into shared categories based on thematic similarities and their relevance to either system behavior or user perception. Through an iterative clustering process, these were refined into six overarching themes that were not only frequently mentioned but also conceptually aligned with the study's two research questions. The chosen themes reflect critical aspects of the user experience with conversational agents in healthcare contexts, including how clearly the assistant communicates, the level of detail provided, the trustworthiness and consistency of the responses, and how users emotionally respond to the system. These themes are:

1) Clarity and Comprehensibility
2) Level of Detail
3) Trust and Transparency
4) Role Consistency
5) User Experience and Emotional Impact
6) Improvements and Features

Importantly, these themes were derived from interviews conducted in both the exploratory and evaluation phases, ensuring that the analysis captured perspectives across multiple stages of system evolution. These categories were not mutually exclusive but reflected distinct focal points in the user feedback. For instance, a single participant comment might express both emotional reassurance (Theme 5) and concerns about inconsistent behavior (Theme 4). The first five themes offer structured insights into participants' comprehension, trust, and engagement with the assistant, addressing the research questions. The Improvements and Features theme, in particular, provided concrete suggestions for system enhancements and directly influenced the following prototype versions. These iteration findings are discussed in Section 7.1

Figure 10 offers a visual summary of the frequency and distribution of coded feedback themes across all participants. The height of each stacked bar represents the total number of coded segments per participant, reflecting the overall richness and volume of qualitative input provided during the interviews. The color segments within each bar correspond to the six identified themes, illustrating not only what was said, but also where each participant placed emphasis in their feedback.

For example, Participant 1, involved in the exploratory phase, shows the highest overall number of comments, with particularly large contributions to Clarity and Comprehensibility and User Experience and Emotional Impact. This suggests a participant who was highly expressive and reflective about the system's communication style and transparency, offering rich, descriptive feedback on how information was delivered and perceived. However, Participant 1 contributed relatively less to Improvements and Features, implying that while they were deeply engaged in interpreting the assistant's behavior, they provided fewer concrete suggestions for system refinement, possibly due to the early-stage nature of the prototype.

In contrast, Participant 7, who participated during the evaluation phase, focused more on Trust and Confusion and provided fewer comments related to User Experience and Emotional Impact. This shift may reflect the increased complexity of the second prototype version, which, while more feature-rich, may have introduced ambiguity or uncertainty around system reasoning.

Consequently, the participants' feedback concentrated on how much they trusted the assistant's outputs, rather than their emotional response or engagement level.

This variation across participants reinforces the value of including both expressive and concise users in qualitative studies. While some participants helped surface broader comprehension and trust-related challenges, others offered focused, actionable feedback that directly supported iterative system improvements.

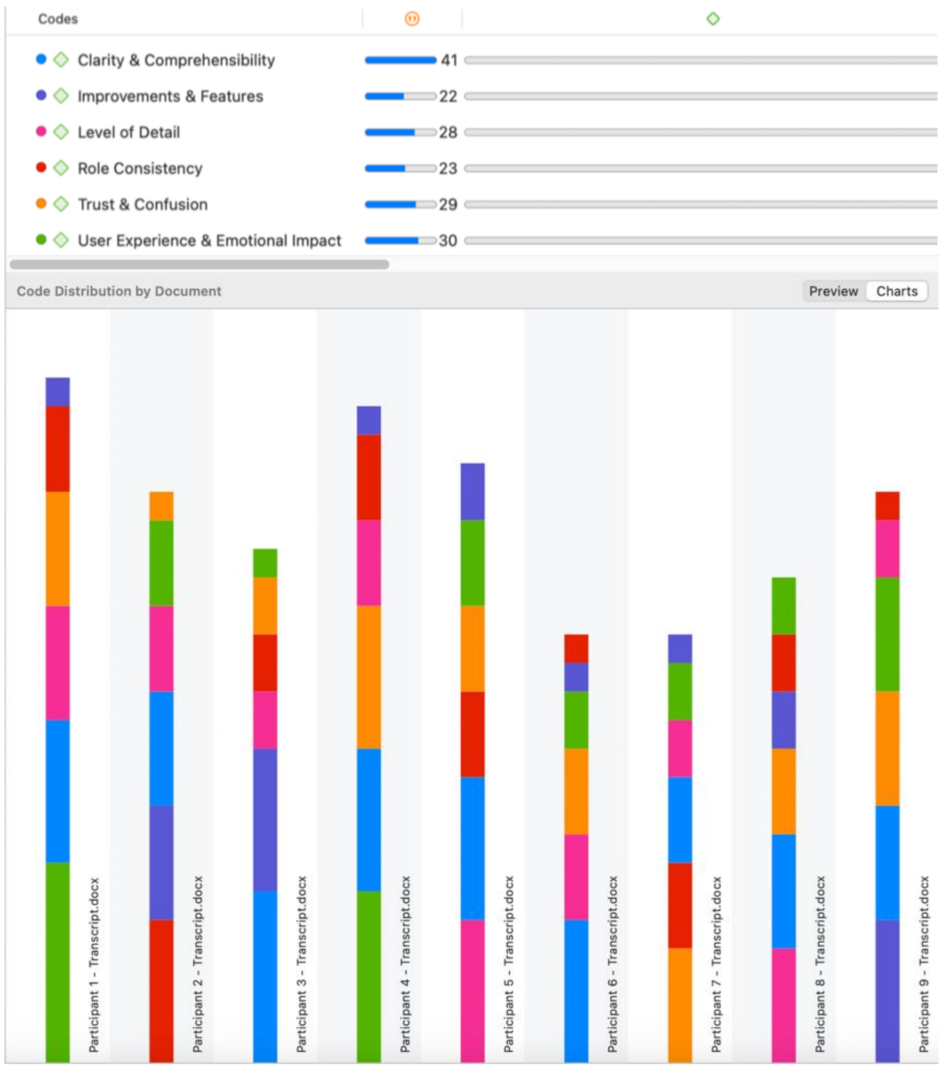


Figure 10. Distribution of the six identified themes across participants in both Exploratory and Evaluation phases generated in Atlas.ti.

6.7.1 Thematic Analysis

This coding process was used in both the Exploratory and Evaluation Phases to compare changes in perception as the prototype evolved. To gather insights on how LLM-based conversational interfaces can improve patients' understanding and interaction with health data, we proposed two themes:

- ***Clarity and Comprehensibility.*** This theme emerged because participants frequently evaluated whether the assistant's responses were understandable. Codes such as "clear," "easy to follow," and "not aligned with the question" highlighted the centrality of clarity in shaping comprehension. Early misalignments were noted, as Participant 2 in V1 mentioned: "*It is not clear apart from the heart rate numbers; without context, I don't feel it's clear enough.*" In a later iteration, issues shifted toward conversational continuity, like Participant 8 remarked: "*I can understand the answers, but sometimes I felt they were contradicting, probably because there is no connection in the historical conversation*."

- ***Level of Details.*** The amount of detail was perceived differently by participants. While some valued longer, more comprehensive outputs, others felt overloaded by repetition. For example, Participant 1 in V1 appreciated the richness: "*There is a lot of detail, and it feels like talking to someone present, caring for and supporting you during this*." In contrast, Participant 3 in V2 described the experience as overwhelming: "*The repeated information made me feel overwhelmed.*" This divergence revealed a tension between comprehensiveness and conciseness.

Similarly, to identify the risks and limitations of using LLM-based conversational interfaces and how these affect trust and engagement, the following themes were derived:

- ***Trust and Confusion.*** Several codes clustered around concerns with "vague," "contradictory," or "uncertain" responses. The need for source clarification (logs vs. general knowledge) was repeatedly noted, making confusion a core dimension of perceived trustworthiness. In V1, confusion arose from misleading outputs. Participant 2 observed: "I *asked to further look into my well-being, and it sticks to the dataset, so that's misleading*". Similarly, participant 3 reported: "*When I referred to my age, it understood it as a time*". In V2, trust was compromised in different ways. Participant 9 reflected: "I find it hard to trust if it did not answer my question. It felt like it wasn't paying attention to me". Participant 4 also highlighted the lack of transparency: "*When I asked how it got my data, I think there should be some prepared answer. The heart rate calculation should already have some background or contact information available for the system to reply*". Together, these comments reveal that while participants valued accuracy, their trust ultimately depended on the system's ability to provide clear, consistent, and transparent responses.

- ***Role Consistency.*** Codes such as "not diagnosing," "referring to a doctor," and "not clinical advice" appeared across multiple transcripts. Participants emphasized that the assistant maintained a clear boundary by not presenting itself as a medical professional. For example, Participant 5 in V2 noted: "*It did not try to make any suggestions about my health status; it always suggested going to a qualified healthcare professional, so that is good.*" Similarly, Participant 7 remarked: "*Yes, to me that is clear, the system is very safe in not acting as a medical professional.*" Participant 9 confirmed this perception: "*It is not acting as a medical professional. Regarding this, it's quite consistent.*" This reinforced the assistant's framing as a supportive tool rather than a diagnostic authority, a role boundary that participants valued for safety and trust.

- ***User Experience and Emotional Impact.*** Participants consistently noted that the interaction flow and emotional tone influenced their engagement. Codes such as "supportive," "non-alarming," and "not connecting" coalesced into this theme, highlighting the dual role of reassurance and frustration in the user experience. In Version 1, Participant 1 reflected positively: "*Very convenient to have someone to chat about your health. I think especially for older people this could be helpful*". In Version 2, participants provided both supportive and critical feedback. Participant 4 appreciated the system's reassurance: "*I like that when I specified that I didn't understand the answer, it assured me that it's easy to find it all overwhelming. That made me feel supported in my lack of knowledge.*" Similarly, Participant 7 emphasized the importance of role boundaries: "*I felt safe that the system was not trying to play the doctor.*" However, frustrations also emerged, as Participant 5 noted: "*It was a distant conversational partner. Sometimes it didn't read my question correctly and gave me answers that did not match my questions.*"

Participants suggested refinements for the prototype, organized under a theme to improve usability and engagement across iterations and the user study, offering valuable insights.

- ***Suggestions and Features.*** This theme was not directly tied to the research questions but emerged from participant proposals coded as "suggestion," "improvement," or "feature request." These inputs reflected personal preferences and were tracked separately to guide iterative development of later prototypes. In Version 1, participants emphasized the need for shorter and more accurate responses. For example, Participant 2 suggested: *"The system can be extended to other contexts outside heart rate; people want to know more about health in general.*" Similarly, Participant 3 recommended, "*Have a bigger dataset or a more critical scenario to see how it behaves*." In Version 2, suggestions focused on linking answers more explicitly to evidence and keeping responses concise. Participant 6 noted: "*Link the answers to evidence*," while Participants 8 and 9 both requested "*shorter answers.*"

6.7.2 Findings by Study Phase

Exploratory Phase

During the exploratory phase, participants interacted with the first version of the prototype (described in Section 5.5.1), which highlighted issues of clarity, repetition, and conversational fragmentation. While the assistant's neutral tone was perceived as reassuring and its role boundaries were clear (i.e., not acting as a medical professional), participants frequently encountered contradictory outputs when referring to standard heart rate ranges and limited memory across exchanges. The assistant struggled with broad or open-ended queries, often defaulting to narrow, log-based responses that lacked personalization (see Appendix I, Example 1). Preferences for the level of detail varied significantly: while some users valued factual brevity, others desired more explanatory or contextualized responses. Participants also proposed design enhancements, such as a flexible response length (allowing users to decide), a larger font size, and improvements to conversational continuity.

Evaluation Phase

Participants engaged with the second prototype version (described in Section 5.5.1), where improvements such as broader context retrieval, refined tone, and personalization were tested. Feedback suggested improvements in clarity, role consistency, and a general increase in perceived trustworthiness, as some users appreciated the contextualized explanations and the assistant's consistent role boundaries. However, this improvement in trust was not universal. Some participants continued to express doubt about how the assistant generated its responses, particularly when the source of the information (log data vs. general knowledge) was unclear (Appendix I, Example 2). This reflects an important distinction between faithfulness (i.e., whether the assistant's outputs matched the retrieved data) and perceived trustworthiness (i.e., whether users felt they could rely on those outputs). In other words, even when the assistant was technically accurate, trust was compromised when the assistant did not disclose the source of the information or how it was generated.

Table 4 illustrates how the six themes were identified across both study phases, highlighting the particular challenges addressed in the second version and the new ones that emerged following the improvements. While the thematic analysis focused on the exploratory (V1) and evaluation (V2) phases, additional findings from Version 3 (V3) are discussed in Section 6.7.3, based on quantitative user ratings and open-ended survey feedback. These insights build on the thematic trends identified here and provide additional validation of the design iterations. The comparison highlights both continuity and change, reflecting how participants' perceptions evolved as the prototype developed.

Table 4. Themes across exploratory and evaluation phases (V1 and V2), based on qualitative data analysis, addressed challenges, newly emerged issues over iterations, and the evolution of participants' perceptions.

Theme	Exploration Phase	Evaluation Phase
Clarity and Comprehensibility	Repetition and contradictory statements confused users	More contextualized answers and age-based comparisons, some users still found explanations too repetitive.
Level of Details	Mixed preferences: brief and factual answers versus detailed explanations	Meaningful but short explanations were provided, not informative enough
Trust and Confusion	Contradictions and lack of clarity about answers source.	Some trust gained through contextualized demographic comparisons, but doubts remained about source transparency
Role Consistency	Boundaries were clear with referrals when needed	Consistency reinforced with supportive but non-clinical framing
User Experience and Emotional Impact	Neutral, non-alarming tone valued; however, lack of conversational memory, fragmented interactions	More natural dialogue and supportive tone as strengths, but still noted gaps in memory and flow
Suggestions and Features	Flexible response lenght, larger font size, and conversational memory	Multimodal outputs (graphs, voice), stronger memory, and more personalized comparisons

6.8 Quantitative Results

This section presents the findings obtained through a post-interaction questionnaire and open-ended reflections, administered via a Google Form link after participants interacted with Prototype V3 (described in Section 5.5.1) during the second evaluation session.

6.8.1 Likert-Scale Questionnaire Analysis

To complement the qualitative findings, descriptive statistics were calculated from the questionnaire responses. Four participants (N = 4) completed the Likert-scale (1–5, strongly disagree → strongly agree) questionnaire, which assessed perceptions of the assistant's clarity, supportiveness, confidence in use, interpretability, naturalness, perceived safety, and improvements across prototype iterations.

Following common HCI and usability research conventions, means and standard deviations (SD) are reported to summarize overall trends, especially when small sample sizes are present, as 5-point Likert items are treated as interval data for practical purposes [51]. However, to respect the ordinal nature of the scale, where the numeric intervals between categories (1 → 2 may not be perceived as the same "size" as 4 → 5), median and interquartile range (IQR) were included to reflect non-parametric distribution summaries. This dual reporting provides a more complete summary while accounting for non-parametric characteristics.

Table 5 presents the descriptive statistics of participants' evaluations. These quantitative insights support and help contextualize the qualitative themes discussed in Section 6.7.

Table 5. Descriptive statistics of four participants' Likert-scale evaluations (N = 4). The table presents participants' assessment of the conversational assistant across clarity, supportiveness, confidence in use, interpretability, naturalness, safety, and improvements over prototype iterations.

Questionnaire Dimension	Mean	SD	Median	IQR
Response clarity	4.25	0.50	4.0	0.25
Supportive tone	5.00	0.00	5.0	0.00
Confidence in use	4.25	0.50	4.0	0.25
Interpretation support	3.75	0.96	3.5	1.25
Naturalness of interaction	4.25	0.50	4.0	0.25
Communication improvements	4.75	0.50	5.0	0.25
Responsiveness to feedback	4.50	0.58	4.5	1.00
Role safety (no diagnosis)	4.75	0.50	5.0	0.25

6.8.2 Open-Ended Questions Analysis

In addition to the Likert-scale items, the post-interaction questionnaire administered during the second evaluation session included a series of explicit open-ended questions. These were designed to elicit reflective feedback from participants after engaging with Prototype V3. Responses were submitted through the same Google Form, serving to contextualize participants' evaluations and offer more nuanced insights into their experiences with the final version of the system and their expectations for future improvements. A thematic analysis of these structured responses revealed both strengths and limitations in the assistant's design, highlighting the following themes of interest:

Helpful aspects of the assistant
Participants appreciated contextualized communication, particularly the ability to compare heart rate values with peers of the same age group and to explain the seriousness of alerts. Others highlighted the assistant's role in providing insights and recommendations for health maintenance, noting its usefulness as an AI agent focused on heart rate–related issues.

"The assistant provided context to its communication."
- Participant 7

Confidence in future use
Most participants indicated they would feel comfortable using the assistant independently, citing its ease of use and helpfulness in interpreting heart rate data. However, concerns emerged regarding the generality of healthcare recommendations and privacy. One participant emphasized that comfort depended on assurance that personal data and conversations would not be shared externally.

"I would feel comfortable using an assistant like this one as long as my data, prompts, and conversations would stay with me."
- Participant 7

Perceived improvements between versions
Participants in this session interacted with earlier versions (V1 and V2) during the exploratory and evaluation phases. Their comments reflected clear improvements in Prototype V3, including more consistent data presentation, better alignment between time and age, and reduced verbosity.

"The latest model can contextualize the answer. This makes me trust the model more because it gives more useful information."
- Participant 9

Additional suggestions
Feedback pointed to conversational flow as a significant area for refinement. Participants reported that the assistant sometimes failed to account for previous inputs, struggled with clarity when units were unspecified, or produced overly long responses. Suggested enhancements included enabling more profound exploration of personal data (e.g., history, trends, lifestyle influences), supporting multimodal interaction, and ensuring outputs remain concise and specific.

"It should have a better flow of conversation. It seems not to be aware of the latest question it was asking me."
\- Participant 6

General impressions
Overall feedback was positive. Participants valued the presentation of heart rate data, describing the system as "helpful" and "nice." They also expressed interest in its local implementation and application of RAGs in a sensitive health domain. At the same time, they emphasized the importance of keeping the assistant's scope limited to its defined health-related role and of ensuring communication remains specific and trustworthy.

These findings contribute to answering the research questions by offering concrete evidence for each. For RQ1, they highlight design features that enhance patient understanding and interaction. For RQ2, they illustrate the conditions under which trust and engagement may be compromised.

Chapter 7
Results Interpretation

This chapter presents a comprehensive interpretation of both the technical evaluation and user study findings across the three prototype iterations previously introduced in Chapters 5 and 6. Each version reflects a progressive evolution in retrieval logic, prompt strategy, and personalization design, shaping how the assistant communicated health-related insights. By integrating structured evaluation metrics with qualitative user feedback, this analysis highlights how each prototype influenced perceived trust, clarity, and interpretability. It also brings forward key trade-offs between retrieval faithfulness and supportive explanation, which emerged during the design process.

Additionally, this chapter presents the results of a post-hoc evaluation using enhanced contextual summaries, conducted following the user study. This extension aimed to investigate the limitations of the retrieval strategy employed during development, particularly the model's ability to ground its answers in enriched, semantically meaningful input. Since users did not interact with the enhanced contexts during their participation, this evaluation provides a technical validation layer rather than user-informed insights.

Finally, the chapter includes a baseline comparison between the top-performing prototype (without enhanced context) and a non-RAG version of the model (LLaMA 3.2 without retrieval), offering a clearer view of the value added by integrating contextual evidence.

7.1 Prototype Iterations Findings

This section analyzes how the assistant evolved based on both technical metrics and user feedback, with a focus on the three core prototype versions introduced in Section 5.5.1. The analysis aims to understand which design decisions improved the system's interpretability and which introduced new challenges. Each iteration addressed specific issues identified in earlier versions, such as retrieval logic, prompt clarity, and alignment with user intent.

By combining the RAGAs' evaluation metrics context relevance, faithfulness, and answer relevance with user perceptions gathered in Chapter 6, the analysis reveals how each version influenced participants' trust, clarity of interaction, and perceived usefulness. This section also surfaces the limitations and trade-offs that guided future refinements, including the tension between factual precision and user-centered communication.

Table 6 summarizes the evaluation results using the original retrieval configuration, providing a foundation for comparing how system behavior evolved across iterations.

Table 6. RAGAs evaluation scores for three prototypes (V1, V2, V3) using original contexts, comparing performance in Context Relevance, Faithfulness, and Answer Relevance.

Prototype Version	Context Relevance	Faithfulness	Answer Relevance
HR Assistant (V1)	0.0000	0.3750	0.5282
HR Assistant (V2)	0.0000	0.4167	0.5063
HR Assistant (V3)	0.0000	0.2000	0.5321

7.1.1 Iteration 1: Establishing the Baseline

The first version (V1) served as a foundational proof of concept, incorporating timestamp-based chunk retrieval and basic rule-based personalization. RAGAs metrics revealed moderate faithfulness (0.3750) and answer relevance (0.5282), but (0.0000) context relevance, confirming that raw summaries lacked sufficient semantic grounding to support strong alignment.

User feedback reflected a mismatch between the system's technical grounding and the clarity of its responses. Participants described outputs as verbose, repetitive, and fragmented, with contradictory statements about standard ranges surfacing when the assistant struggled to reconcile multiple retrieved values. Importantly, the assistant's neutral tone reassured users that it respected its boundaries and did not attempt to act as a medical specialist. Despite these limitations, participants acknowledged that the assistant made their heart rate data easier to follow than reading the logs directly. Several also valued its interpretive framing, which helped contextualize raw numbers without overstepping into diagnostic territory.

This version highlighted the core challenge of iteration: the technical ability to ground responses did not automatically translate into clear interpretation. For example, as illustrated in Figure 11, when users asked open-ended questions like "What's the heart rate value for a person of my age?", the assistant often retrieved the last-used chunk but failed to provide a direct or demographically personalized answer. These limitations underscore the need for improved conversation flow, more precise explanations using age-based benchmarks, and mechanisms to handle broader or ambiguous queries without defaulting to overly narrow, log-focused outputs. At the same time, the prototype already demonstrated potential to provide supportive, non-clinical value, laying the groundwork for subsequent refinements.

7.1.2 Iteration 2: Refining Contextualization

The second iteration addressed the limitations observed in Version 1 by introducing broader retrieval strategies and refining the prompt structures to accommodate more open-ended time references, resulting in prototype Version 2. A key design change was the expansion of retrieval scope: the assistant should now return three chunks per query when needed, aiming to support questions referencing broader timeframes such as "later" or "before" instead of strictly matching a single hour. This change aimed to enhance contextual understanding and enable the assistant to respond to more flexible temporal queries with relevant background information. However, the system did not consistently interpret these vague time expressions as intended. In many cases, it still defaulted to retrieving a single chunk based on the most clearly detected hour, rather than retrieving adjacent summaries to cover a broader range. For instance, as shown in Figure 12, when users asked questions like "What happened after the alert? Did my heart rate get better?", the assistant acknowledged the ambiguity but still provided a narrow response, failing to retrieve adjacent chunks that could answer the full question scope.

Technically, while context relevance remained at 0.0000, still reflecting the strict standards of the Mistral-based evaluator, faithfulness improved (0.4167), likely due to reduced hallucination from using a narrower context. At the same time, answer relevance decreased (0.5063), since the assistant's responses were often too narrow for the intended timeframe. This outcome highlights a critical trade-off: limiting retrieval scope can enhance factual grounding, but it may misalign the response with the broader user intent. Despite the use of structured prompts and a low LLM temperature (set to 0.7 in the prototype to encourage conversational yet stable responses), some variability in the generated output persisted, occasionally affecting the length or level of detail. While this inconsistency was minor in terms of answer content relevance, it nonetheless underscores a broader challenge in applying generative models within health contexts, where predictability, clarity, and user trust are paramount.

Qualitative feedback gathered during this phase of the user study highlighted that the assistant was perceived as informative, neutral in tone, and capable of contextualizing alerts and clarifying their meaning, especially through age-based comparisons and trend framing. For example, instead of simply reporting a high heart rate, the assistant explained whether the value was abnormal for the participant's profile and the relation to the activity. Participants appreciated that the system helped them interpret alert severity and understand heart rate values in more meaningful ways. While concerns remained around verbosity, limited personalization, and weak continuity in follow-up exchanges, most users continued to describe the assistant as a helpful tool, particularly for interpreting logs rather than providing clinical advice.

Overall, Iteration 2 exemplified the trade-offs introduced by expanding retrieval: while the assistant's responses became more contextual and better suited to general health questions, both the technical metrics and user insights revealed persistent issues with conciseness, personalization, and conversational flow. This iteration reaffirmed the complexity of balancing broader contextualization with factual precision, while also reinforcing the importance of maintaining the assistant's non-diagnostic, supportive role.

7.1.3 Iteration 3: Personalization and Trade-offs

Iteration 3 introduced demographic-aware personalization alongside an even broader retrieval strategy. For this version, the system successfully aggregated summaries, including Morning, Midday, Afternoon, and Day Overview, using the available heart rate data. These summaries required retrieving three chunks at once, and in the case of a full-day query, up to eight chunks. The intent was to offer more user-aligned answers, particularly in cases involving high heart rate alerts or broader-context health reflections. As shown in Figure 13, the assistant was able to deliver a well-structured daily summary that included a timestamped table of activities and heart rate values, followed by observations highlighting stability and occasional peaks, demonstrating the improved interpretability of the final prototype.

This design decision directly impacted technical results: answer relevance rose to 0.5321, suggesting that responses were still perceived as reasonably informative and aligned with the query intent, even if not strictly faithful to the retrieved data. While the context relevance stayed at 0.0000, reflecting Mistral's strict evaluation standard when contextual overlap is not verbatim reproduced in responses. However, faithfulness dropped sharply to 0.2000 as the model struggled to reproduce every detail in large contexts, a low score, especially for health systems relying on factual accuracy. While avoiding hallucinations and clinical claims ensured safety, it limited the ability to reflect all retrieved facts verbatim. Processing more chunks increased information load, raising expectations for completeness and risking omission or rephrasing of critical details.

Although according to RAGAs, there was a drop in faithfulness, in practice, factual elements such as average heart rate, ranges, and logged activities were consistently accurate in the answers. The challenge was in capturing the broader descriptive content of multiple chunks, which often resulted in longer, more scattered responses that could not concisely mirror the relevant information. This trade-off highlights a key tension in retrieval-augmented systems: expanding context enhances interpretive value for users but reduces strict fidelity to the retrieved logs in their exact form.

This challenge is not only technical but also semantic. When synthesizing multiple chunks, the assistant must reconcile overlapping concepts and expressions that may differ slightly in meaning across logs. Semantic ambiguities can emerge, particularly in cases involving vague terms like "normal," "high," or "alert," which may shift subtly depending on the source chunk. In health-related systems, such variations can affect user interpretation, making semantic precision as important as factual correctness. While this version of the assistant aimed to reduce such risks through logic-based query handling, grouping similar expressions under shared intent categories to minimize ambiguity, there remains an inherent risk of newly emerging or context-specific terms that may not be fully captured, highlighting the ongoing need for adaptive semantic handling in future iterations.

Questionnaire results show participants rated the assistant positively across most areas. The highest score was observed for Supportive tone (M = 5.0, SD = 0.0), where M refers to the mean score and SD to the standard deviation, indicating unanimous agreement that the assistant communicated in a respectful and supportive way. Similarly, perceived safety (M = 4.75) was evaluated very positively, reflecting trust in the assistant’s boundaries and recognition of progress across prototype versions. While Communication improvements (M = 4.75) and Responsiveness to feedback (M = 4.50) were similarly assertive, suggesting that participants noticed changes in the assistant’s behavior based on prior suggestions. Clarity of responses, Confidence in use, and Naturalness of interaction, each with (M = 4.25), showed that participants generally experienced the system as straightforward, usable, and realistic, though not without minor limitations. By contrast, Interpretation support received the lowest mean score (M = 3.75, SD = 0.96), indicating variation in how well participants felt the assistant helped them understand their heart rate data. Taken together, the results suggest that the last assistant iteration (version 3) was perceived as supportive, safe, and progressively improving; however, it still requires refinement in delivering explanations and interpretive guidance.

Open-ended comments further reinforced this ambivalence. Participants welcomed features like age-based benchmarks and clearer alert framing, but also pointed out gaps in memory, conciseness, and conversational flow. Nevertheless, this was the first iteration where users consistently described the assistant as trustworthy, supportive, and practically helpful in contextual communication on their heart rate data. In short, Iteration 3 was seen as more supportive, contextual, and trustworthy, but less factually precise. Comments praised design features like age-based comparisons and health insights, while highlighting risks and concerns from the participants, such as generic advice and privacy issues. This underscores a broader lesson: more retrieval enhances the relevance of the answer but may risk diluting faithfulness and increasing verbosity.

Conversely, comparison with the baseline LLaMA 3.2 further clarifies these dynamics. While the baseline model achieved higher faithfulness (0.5500) than the RAG-enhanced assistant (0.3967), its answer relevance was substantially lower (0.4000 compared to 0.7800). This suggests that the baseline model, operating without retrieval, tended to remain closer to its limited input but was unable to generate answers that users would find sufficiently informative or aligned with their queries. By contrast, the RAG pipeline introduced contextual evidence that allowed the assistant to provide richer and more relevant responses, even if this came at the expense of reproducing every retrieved detail with perfect fidelity. In the context of health-related conversational agents, this comparison highlights that retrieval significantly contributes to interpretive usefulness and perceived value, even though it introduces challenges for maintaining strict factual accuracy.

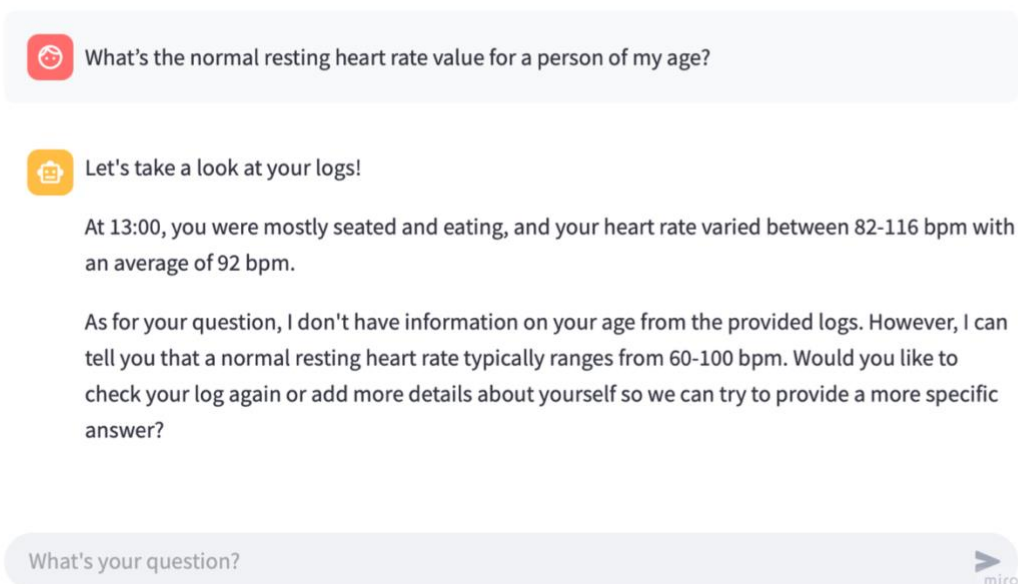
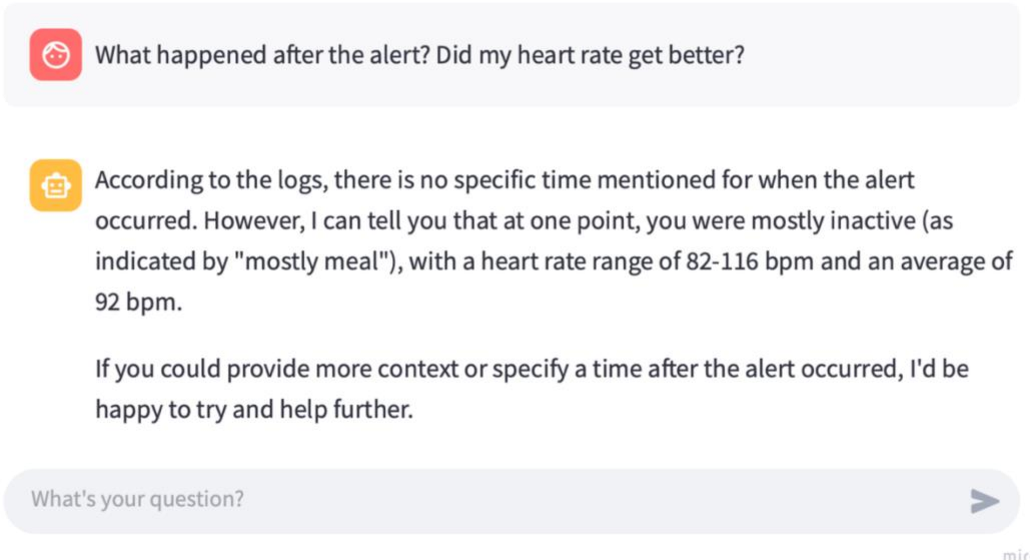


Figure 11. Response from Iteration 1 illustrating the assistant's inability to address open-ended queries.



What happened after the alert? Did my heart rate get better?

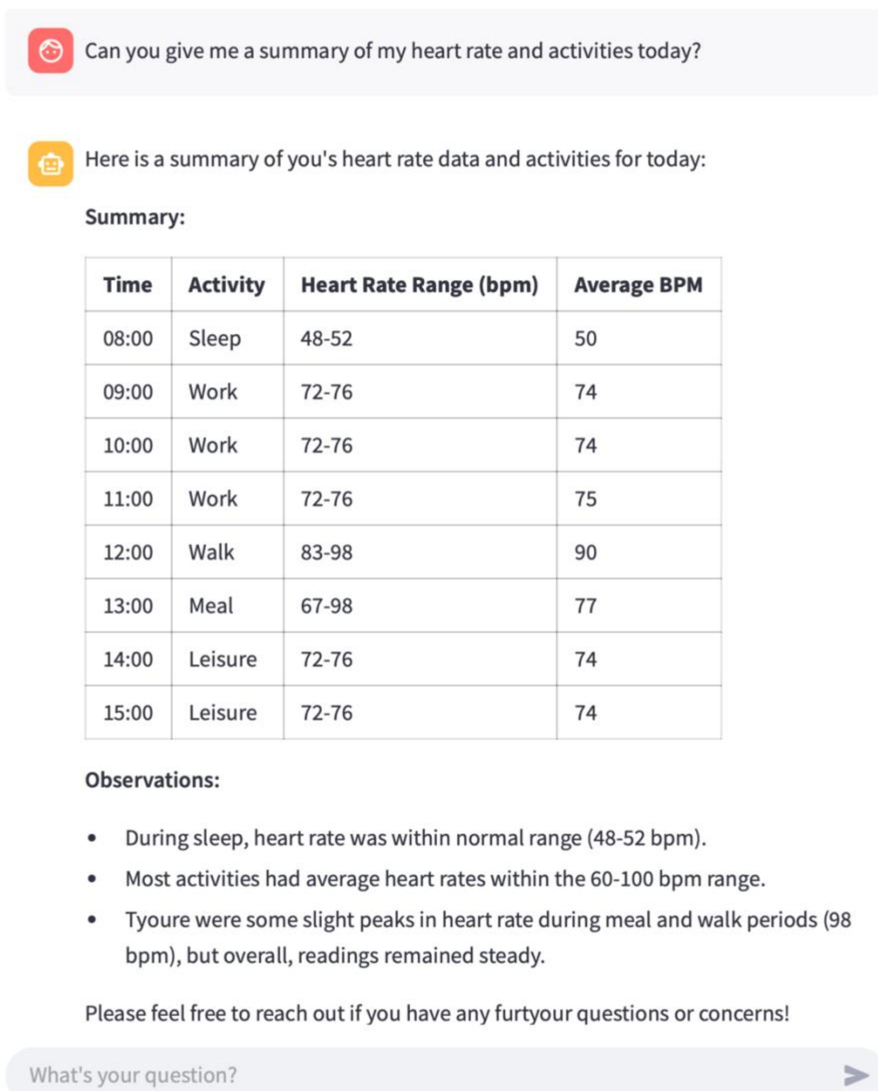
According to the logs, there is no specific time mentioned for when the alert occurred. However, I can tell you that at one point, you were mostly inactive (as indicated by "mostly meal"), with a heart rate range of 82-116 bpm and an average of 92 bpm.

If you could provide more context or specify a time after the alert occurred, I'd be happy to try and help further.

What's your question?

miro

Figure 12. Response from Iteration 2 demonstrating the assistant's limited ability to follow up on vague time-based questions.



Can you give me a summary of my heart rate and activities today?

Here is a summary of you's heart rate data and activities for today:

Summary:

Time	Activity	Heart Rate Range (bpm)	Average BPM
08:00	Sleep	48-52	50
09:00	Work	72-76	74
10:00	Work	72-76	74
11:00	Work	72-76	75
12:00	Walk	83-98	90
13:00	Meal	67-98	77
14:00	Leisure	72-76	74
15:00	Leisure	72-76	74

Observations:

- During sleep, heart rate was within normal range (48-52 bpm).
- Most activities had average heart rates within the 60-100 bpm range.
- Tyoure were some slight peaks in heart rate during meal and walk periods (98 bpm), but overall, readings remained steady.

Please feel free to reach out if you have any furtyour questions or concerns!

What's your question?

miro

Figure 13. Response from Iteration 3 showcasing improved handling of broad daily summary requests.

7.1.4 Post-Hoc Enhanced Context Evaluation

The technical evaluation results showed that context relevance remained at 0.0 across all iterations, indicating that the original contexts (summaries of physiological data associated with activities) did not provide sufficient cues for effective retrieval. This strict outcome also reflects the conservative scoring of the Mistral 7B evaluator, which only marked contexts as relevant when they provided direct and explicit support to the query. The limitation was not always evident in answers, as prompts to the LLM avoided unhelpful responses. However, the evaluation results revealed the need to test enriched contexts, which included interpretive elements such as activity-aware thresholds and demographic framing, to improve responses. This confirmed that evaluation metrics were effective, and issues stemmed from the quality of summaries, rather than the retrieval technique itself. An example is shown below (see the complete enhanced contexts in Appendix H). Table 7 presents the results of the prototype iterations evaluated with enhanced contexts.

$$original\_context = [\text{“}At\ 11{:}00, Bryan\ was\ mostly\ work. The\ heart\ rate\ ranged\ from\ 84\ to\ 88\ bpm, with\ an\ average\ of\ 86\ bpm.\text{”}]$$

$$enhanced\_context = [\text{“}At\ 11{:}00, Bryan\ was\ mostly\ work. The\ heart\ rate\ ranged\ from\ 84\ to\ 88\ bpm, with\ an\ average\ of\ 86\ bpm. Elevated\ heart\ rate\ during\ work\ indicates\ stress\ response, which\ is\ normal\ but\ should\ be\ monitored.\text{”}]$$

Table 7. RAGAs evaluation scores for three prototypes (V1, V2, V3) using enhanced contexts, showing performance in context relevance, faithfulness, and answer relevance.

Prototype Version with enhanced context	Context Relevance	Faithfulness	Answer Relevance
HR Assistant (V1)	0.7500	0.5000	0.5334
HR Assistant (V2)	0.7500	0.8492	0.7331
HR Assistant (V3)	0.7500	0.2000	0.7121

The results in Table 7 indicate that the introduction of enhanced contexts resulted in a significant improvement in context relevance, which stabilized at 0.75 across all iterations. Faithfulness peaked in V2 (0.8492), indicating that enriched contextual cues enabled the assistant to generate answers more firmly grounded in the retrieved evidence; however, it dropped in V3 as the system integrated more complex inputs. Answer relevance also improved, reaching its highest value in V2 (0.7331), before slightly decreasing in V3.

Overall, these results confirm that enriching the original summaries with interpretive elements provided more effective retrieval support and improved alignment between system responses and user queries.

However, the enhanced context evaluation also revealed a critical limitation in V3's design approach. While context relevance remained consistently high (0.7500) and answer relevance improved for V2 (0.7331), V3 experienced a significant drop in faithfulness to 0.2000 when evaluated with enhanced contexts. This counterintuitive result occurred because V3's sophisticated prompt engineering, designed for demographic awareness and comprehensive medical interpretations, generated responses that incorporated medical knowledge beyond what was explicitly stated in the retrieved contexts.

Specifically, V3 responses included clinical interpretations ("cardiac issues," "dehydration in older adults"), external medical guidelines (European Society of Cardiology standards), and age-specific medical advice that, while medically appropriate and user-relevant, were not directly present in the enhanced context summaries. This highlights a fundamental tension in medical RAG systems between retrieval fidelity (faithfulness) and the integration of domain expertise (helpfulness). The RAGAs faithfulness metric strictly measures adherence to retrieved information, penalizing any additional knowledge incorporation regardless of its clinical validity or user benefit.

This finding suggests that traditional RAG evaluation metrics may be insufficient for assessing medical conversational interfaces, where integrating domain expertise beyond retrieved data can be both necessary for user safety and clinically appropriate. The results suggest that V2 achieved an optimal balance between faithfulness and helpfulness, while V3's enhanced medical capabilities came at the cost of strict retrieval grounding, a trade-off that requires careful consideration in production medical AI systems where both accuracy and comprehensive care guidance are essential.

7.2 Baseline Comparison Findings

Building on the prototype evaluations, the top-performing iteration without enhanced context (V2) was selected as the reference for baseline comparison. This ensured consistency with the development of all iterations, which were initially evaluated using the same type of retrieved summaries. Table 8 shows the results of this comparison, while Figure 14 provides an example from the evaluation set, comparing the LLaMA 3.2 baseline and the RAG-enhanced responses to the same prompt.

Table 8. Baseline comparison between the RAG-augmented assistant (V2 without enhanced context) and the plain LLaMA 3.2 model without retrieval, evaluated on Factual Correctness (F1) and Answer Relevance.

Model	Faithfulness	F1 score
LLaMA 3.2 + RAG	0.3450	0.5063
LLaMA 3.2	0.2365	0.2733

The results in Table 6 show that the RAG-augmented assistant (Faithfulness: 0.3450, F1 score: 0.5063) clearly outperformed the plain LLaMA 3.2 model without retrieval (Faithfulness: 0.2365, F1 score: 0.2733) across both evaluation dimensions. This improvement reflects the added value of integrating retrieved physiological summaries during the response generation process. While the baseline relied solely on its pre-trained internal knowledge, the RAG pipeline grounded its answers in actual log data, enabling more accurate and context-sensitive responses.

The faithfulness gain (+0.1085) suggests that retrieval not only provided factual anchor points but also guided the model to produce more verifiable claims. Meanwhile, the significantly higher F1 score (+0.2330) demonstrates stronger alignment between the system's responses and the expected ground truth. Together, these findings confirm that RAG architectures are particularly well-suited for domains like health data interpretation, where precision, traceability, and context grounding are critical, especially in systems designed for non-diagnostic yet interpretive guidance.

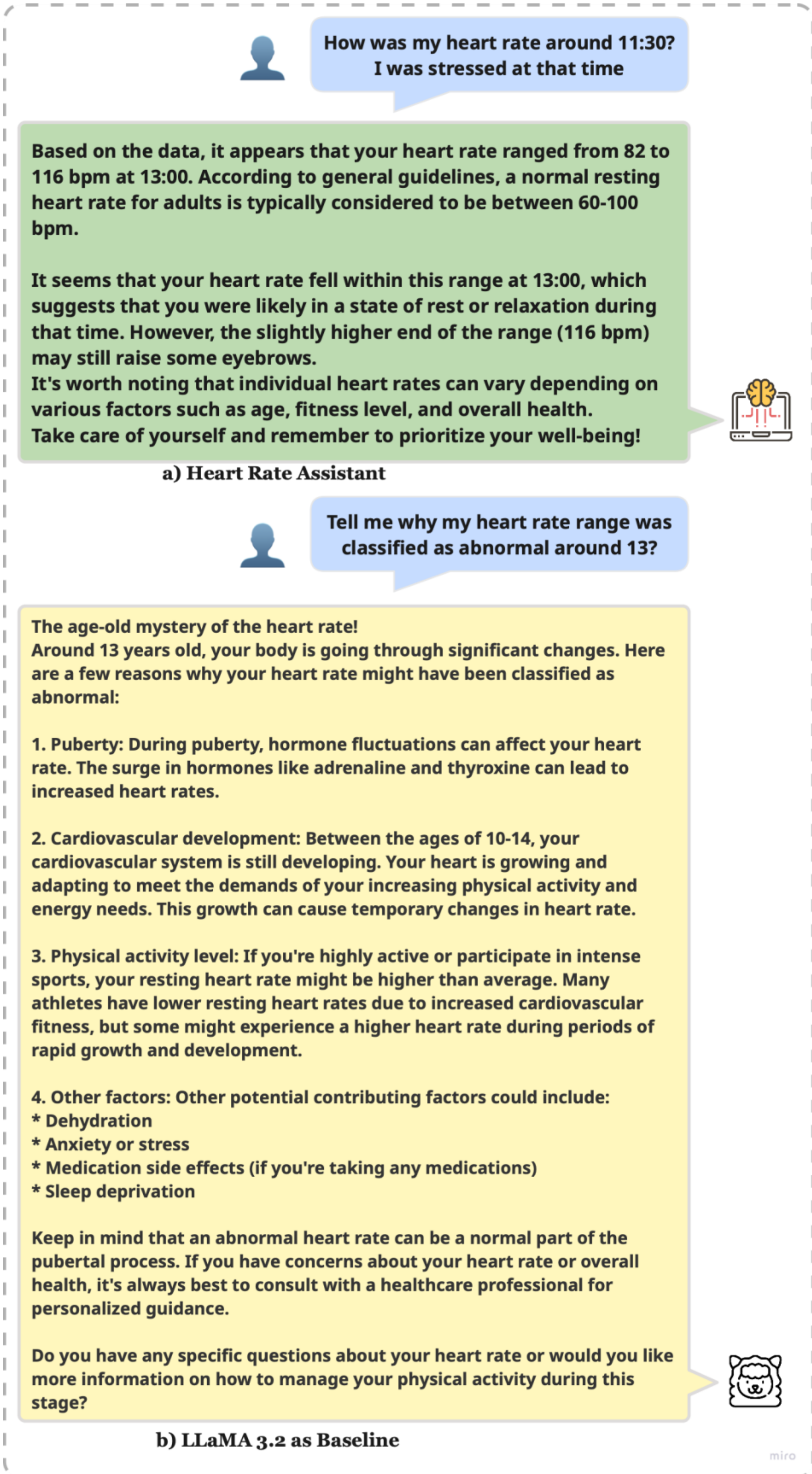


Figure 14. Example responses generated by a) the heart rate assistant with retrieval, and b) the baseline LLaMA 3.2 without retrieval.

Chapter 8
Discussion

8.1 Addressing Research Questions

8.1.1 RQ1: Design of Conversational Interfaces for Understanding DT Systems

The findings highlight that effective design of LLM-based conversational interfaces for Digital Twin systems depends on balancing contextual richness with interpretive clarity. Across iterations, technical results showed improvements in answer relevance, reflecting the benefits of retrieval-augmented generation (RAG) and demographic-aware personalization. However, as seen in Iterations 2 and 3, broadening retrieval beyond a single chunk introduced trade-offs: while summaries of multi-hour or daily patterns enhanced interpretive value, they also increased verbosity and reduced faithfulness. Although enriching the retrieval context led to higher technical evaluation scores, particularly in context relevance, the assistant's responses consistently provided sufficiently reliable and accurate information to support user understanding and interaction across all iterations. This illustrates that design choices about retrieval scope directly affect the quality of user-facing explanations.

From the user perspective, several design features proved particularly valuable. Participants consistently emphasized that comparisons with age-based norms, contextual framing of alerts, and summaries of daily trends helped them better grasp their heart rate data. These interpretive elements moved the assistant beyond passive reporting toward active meaning-making, a role participants recognized as useful for tracking and reflection without perceiving it as a diagnostic substitute. The supportive and neutral tone was another critical factor, while technical precision varied, participants repeatedly highlighted that respectful, bounded communication fostered comfort and trust.

Yet the findings also underscore persistent design limitations. Participants noted that the assistant often failed to sustain conversational continuity, treating follow-ups as isolated questions rather than linked turns. Verbosity remained a challenge across iterations, with some users finding long outputs distracting even when they contained relevant details. These results suggest that future designs should emphasize three interdependent principles:

- **Contextualization:** references and comparisons that ground raw values in meaning.
- **Concise and conversational flow**: Sustain interaction without overwhelming users.
- **Transparency of role boundaries**: The assistant functions as an interpretive layer, not as a diagnostic authority.

Together, these insights suggest that designing conversational interfaces for DT systems involves not only technical accuracy but also the framing and communication of meaning. Supporting patient understanding requires careful calibration of retrieval scope, tone, and flow, so that interpretive richness does not come at the cost of clarity or trust.

8.1.2 RQ2: Risks, Limitations, and Trust Factors of LLM-Based Interfaces in DT Systems

The evaluation also surfaced several risks and limitations that directly influence how patients might trust and engage with such systems. On the technical side, the most evident risk was inconsistency in factual grounding. As retrieval expanded from one to multiple chunks, faithfulness scores decreased, showing that the assistant often failed to retrieve or reflect all retrieved details precisely. Although averages, ranges, and activity labels were usually correct, broader or longer contextual descriptions became less reliable. This demonstrates a fundamental consideration: the richer the context, the harder it becomes for the model to maintain strict fidelity, a limitation with significant implications in health contexts, where accuracy underpins trust.

From the user perspective, two trust-sensitive issues emerged. First, privacy concerns were raised explicitly, with one participant stating that they would only use the system if personal data and conversations were not shared. This reflects the augmented sensitivity of conversational health data and the need for strict privacy safeguards in future implementations. Second, users noted occasional errors such as confusing beats per minute (bpm) values with time references or producing overly generic recommendations. Even when these errors were minor, they eroded confidence in the assistant's reliability.

At the same time, the study highlighted design features that reinforced trust despite technical shortcomings. Participants consistently valued the assistant's supportive tone, neutral stance, and clear role boundaries. Unlike diagnostic tools, the assistant was seen as an aid for interpretation rather than a replacement for professional expertise, which increased trust by setting realistic expectations. When the system avoided speculation and stayed within its interpretive role, participants felt reassured about its safe use.

Taken together, these findings show that sustaining trust requires addressing risks on two levels:

- **Technical safeguards**: mitigating loss of faithfulness when retrieval scales up, and ensuring factual grounding is not diluted by verbosity.
- **User-centered safeguards**: ensuring privacy, setting role boundaries, and maintaining a supportive but non-diagnostic tone.

Thus, emphasizes that building trust in LLM-based conversational interfaces is not achieved solely by improving accuracy but by balancing technical reliability with ethical design choices that protect privacy, avoid overreach, and communicate clearly.

Chapter 9
Conclusions

9.1 Conclusion

This thesis investigated how Large Language Model (LLM)-based conversational interfaces can function as interpretation layers in Digital Twin (DT) systems, with a focus on heart rate data. The goal was to design and iteratively refine a prototype assistant that supports patients to better understand their physiological information through contextual and trustworthy dialogue (Chapters 4–6).

The first research question (RQ1) asked how LLM-based conversational interfaces can facilitate patients' comprehension and interaction with Digital Twin health data. The findings demonstrated that integrating retrieval-augmented generation (RAG) with demographic-aware personalization significantly improved the communication of heart data. Technical evaluations presented in Chapter 5 showed consistent gains in context relevance and answer relevance, confirming that retrieval pipelines align system responses more closely with user intent. The user studies discussed in Chapter 6 further revealed that participants valued contextualized alerts, age-based comparisons, and concise yet meaningful explanations, which supported reflection and self-monitoring. Importantly, the baseline comparison in Chapter 7 confirmed the added value of RAG, as retrieval-enhanced responses outperformed the non-retrieval LLaMA 3.2 baseline, reinforcing the effectiveness of implementing RAG in this domain. Collectively, these results highlight that conversational interfaces can effectively bridge the gap between raw health data and patient understanding, provided that clarity, personalization, and transparency are prioritized.

The second research question (RQ2) investigated the risks and limitations associated with such systems. Evidence from the prototype iterations in Chapter 5 and the evaluation findings in Chapter 6 highlighted key trade-offs. Expanding the retrieval scope enhanced interpretive richness but reduced faithfulness, as responses could not consistently reproduce every retrieved detail. Participants also identified limitations, including verbosity, occasional inconsistencies, and restricted conversational memory. These issues underscore the need for carefully balancing interpretive depth, factual accuracy, and conversational fluency in future designs.

By iteratively developing and testing three prototype versions, this work contributes both technical and human-centered insights into conversational DT interfaces. As outlined across Chapters 5 and 6, the study advances understanding in three ways: (1) empirically demonstrating how retrieval-based personalization improves alignment with patient intent, (2) surfacing design trade-offs that shape trust and engagement, and (3) identifying communication features that make health data interpretation more accessible and engaging. Beyond the case of heart rate interpretation, the implications extend more broadly: conversational layers can increase accessibility and comprehension of complex physiological data, but their value depends on embedding ethical safeguards, avoiding diagnostic claims, and ensuring sustainable trust.

In summary, this thesis contributes practical insights into the design of LLM-based interfaces, demonstrating their strong potential as interpretation layers in Digital Twin systems. Such interfaces can play a valuable role in bridging the gap between raw health data and patient understanding. Still, their effectiveness depends on carefully balancing precision and interpretive richness, personalization and consistency, and a supportive tone with factual accuracy, while continuously addressing the challenges of conversational flow and privacy.

9.2 Limitations

While this study demonstrated the potential of LLM-based conversational interfaces for interpreting Digital Twin data, several limitations constrain the scope and generalizability of its findings.

- ***Prototype development limitations.*** A central limitation was the short time frame between prototype iterations. Although each version integrated improvements, the condensed schedule compromised the scope of refinements and limited the possibility of testing more experimental or ambitious design features. Another technical constraint was the choice of model. To ensure feasibility and privacy in local deployment, the assistant relied on the LLaMA3.2 (2.5 GB) model. While this approach supported offline experimentation without external dependencies, it inevitably restricted performance compared to larger-scale LLMs, which, although slower and more energy-demanding, may have handled contextual integration more effectively. The prototype was also limited to heart rate data only, raising questions about its scalability to other physiological signals or multimodal Digital Twin inputs.

- ***Hallucination prevention limitations.*** The system minimized hallucinations by grounding responses in retrieved log data and constraining prompts. Still, residual risks remained, especially with open-ended or emotionally framed queries that could trigger unsupported or diagnostic-sounding statements. To reduce this, the study restricted prompts to structured types (time-based questions, alert follow-ups, summaries). This improved factual grounding but reduced conversational freedom, reflecting a trade-off between safety and naturalness.

- ***Evaluation metrics limitations.*** The evaluation was based on faithfulness, context relevance, and answer relevance, which captured core aspects of performance but only partially reflected user experience. Faithfulness was measured by alignment with retrieved logs; however, highly faithful answers could sometimes feel verbose or misaligned, while less faithful ones were sometimes valued for their clarity or empathy. In addition, semantic risks, such as subtle hallucinations, were not fully captured. Fabricated details, though minor, could erode trust, and current metrics often rewarded fluency even when factual reliability was compromised. Strict cross-model scoring also penalized contextually helpful responses; for example, answers including correct heart rate values were still rated as zero. This limitation became evident in Version 3, which, despite delivering more contextually appropriate responses through demographic awareness and clinical guidelines (e.g., European Society of Cardiology standards), achieved much lower faithfulness scores

(0.2000) compared to Version 2 (0.8492). While these enhancements improved answer relevance (0.7121) and user experience, they violated RAGAs' strict requirement of adherence to retrieved context, highlighting that traditional RAG metrics may inadequately assess conversational health systems where domain expertise integration is both necessary and beneficial.

- ***Design limitations.*** The study relied on simulated heart rate datasets rather than real-world patient data. While this ensured controlled experimentation, it reduced ecological validity. Real-world signals often include noise, irregularities, and individual variability, which could challenge the robustness of the system in practice. Furthermore, the prototype's lack of long-term memory constrained conversational continuity, preventing the assistant from sustaining dialogue across extended interactions and potentially weakening trust over time.

- ***Participants***. The user study participants were primarily recruited from research backgrounds and included colleagues from the company where the study was conducted. This shaped their expectations toward more factual and technically detailed explanations, which may not reflect the needs of broader patient groups. For instance, elderly users or those with lower technical literacy may prefer simpler, more accessible communication. As a result, the findings provide valuable but partial insights into usability and trust, and should be interpreted cautiously when generalizing to diverse user populations.

9.3 Future Work

Future work should prioritize extending the prototype to operate on real-world patient datasets. While simulated data enabled controlled experimentation, real signals contain noise, irregularities, and variability that are critical for testing robustness. Testing with real-world inputs would provide stronger evidence of ecological validity and expose challenges that controlled settings cannot capture. Beyond data, future studies should also involve larger and more diverse participant groups. Including users with diverse demographics, health conditions, and communication preferences would strengthen the generalizability of the findings and reveal how design features, such as tone, personalization, and explanation style, need to adapt for distinct populations, such as elderly or less technically oriented individuals.

At the technical level, future work should address the balance between interpretive richness and factual precision. As shown by the baseline comparison and retrieval-augmented evaluations, broadening context improved relevance but made answers less precise, underscoring the need for adaptive strategies that dynamically adjust information scope depending on query type. For example, the assistant could provide broad summaries when users ask about daily patterns but focus on concise, detail-oriented answers when queries target specific hours. Exploring larger-scale LLMs may also enhance contextual reasoning. Although such models would require longer runtimes, they could handle semantic integration more effectively. Future designs may also combine these models with privacy-preserving deployment methods, such as on-device optimization or carefully managed local–cloud combinations, to achieve both higher performance and responsible data handling.

Future research should also focus on advancing conversational continuity and personalization. Implementing long-term memory mechanisms would enable the assistant to sustain dialogue across sessions, supporting more natural and engaging interactions. Personalization could extend beyond demographic comparisons to incorporate individual health profiles and user preferences, ensuring explanations feel tailored, supportive, and trustworthy.

Equally important is developing methods to handle open-ended and emotionally framed prompts more safely, without over-restricting user expression. This could include adaptive prompting strategies, real-time hallucination detection, or confidence-calibrated responses that acknowledge uncertainty. Such approaches would expand conversational flexibility while maintaining the safeguards against unsupported or diagnostic outputs.

Improving evaluation approaches is another priority. Current metrics reveal key gaps: faithfulness does not always align with user trust, semantic risks remain undercaptured, and strict cross-model scoring penalizes clinically appropriate interpretations, as seen in Version 3's low faithfulness score despite delivering more relevant guidance. This indicates that retrieval-focused metrics alone are insufficient for medical AI, where domain expertise beyond the retrieved data can be necessary and beneficial. Future frameworks should adopt hybrid approaches that combine automated scoring with human-centered assessments of trust, clarity, and usability, while integrating mechanisms to detect hallucinations and semantic inconsistencies more reliably. In parallel, ontology-based methods could strengthen consistency, interoperability, and contextual understanding in semantic retrieval, complementing vector similarity–based approaches.

Finally, ethical and practical safeguards must remain central. Protecting sensitive health data requires robust privacy measures, while transparency about how responses are generated is key to sustaining confidence. Equally, it is essential to make the assistant's role clear: it should function as an interpretive support tool, not a diagnostic authority. Ensuring that the system complements rather than replaces healthcare professionals will be critical for its responsible adoption.

References

[1] Singh, M.; Fuenmayor, E.; Hinchy, E.P.; Qiao, Y.; Murray, N.; Devine, D. Digital Twin: Origin to Future. Appl. Syst. Innov. 2021, 4, 36. https://doi.org/10.3390/asi4020036

[2] Grieves, M. Origins of the Digital Twin Concept. 2016. Available online: https://www.researchgate.net/publication/307509727_Origins_of_the_Digital_Twin_Concept. (accessed on 2 April 2025)

[3] National Aeronautics and Space Administration. (2012). Materials, structures, mechanical systems, and manufacturing roadmap: Technology area 12 (B. Piascik, J. Vickers, D. Lowry, & S. Scotti, Chairs). NASA.

[4] E. Katsoulakis et al., "Digital twins for health: a scoping review," Npj Digital Medicine, vol. 7, no. 1, Mar. 2024, doi: 10.1038/s41746-024-01073-0.

[5] D. Drummond and A. Gonsard, "Digital twins of patients: an introduction based on a scoping review (Preprint)," Journal of Medical Internet Research, vol. 26, p. e58504, Sep. 2024, doi: 10.2196/58504.

[6] K. Zhang et al., "Concepts and applications of digital twins in healthcare and medicine," Patterns, vol. 5, no. 8, p. 101028, Aug. 2024, doi: 10.1016/j.patter.2024.101028.

[7] K. Papachristou, P. F. Katsakiori, P. Papadimitroulas, L. Strigari, and G. C. Kagadis, "Digital twins' advancements and applications in healthcare, towards precision medicine," Journal of Personalized Medicine, vol. 14, no. 11, p. 1101, Nov. 2024, doi: 10.3390/jpm14111101.

[8] T. Sun, X. He, and Z. Li, "Digital twin in healthcare: Recent updates and challenges," Digital Health, vol. 9, Jan. 2023, doi: 10.1177/20552076221149651.

[9] C. M. Fang et al., "PhysioLLM: Supporting Personalized Health Insights with Wearables and Large Language Models," arXiv (Cornell University), Jun. 2024, doi: 10.48550/arxiv.2406.19283.

[10] Wen, B., Norel, R., Liu, J., Stappenbeck, T., Zulkernine, F., & Chen, H. (2024). Leveraging Large Language Models for Patient Engagement: The Power of Conversational AI in Digital Health. 2024 IEEE International Conference on Digital Health (ICDH), 104-113. https://doi.org/10.1109/ICDH62654.2024.00027.

[11] R. Yang, T. F. Tan, W. Lu, A. J. Thirunavukarasu, D. S. W. Ting,and N. Liu, "Large language models in health care: Development,applications, and challenges," Health Care Science, vol. 2, no. 4, pp.255–263, 2023.

[12] M. Jia, J. Duan, Y. Song, and J. Wang, "MedIKAL: Integrating Knowledge Graphs as Assistants of LLMs for Enhanced Clinical Diagnosis on EMRs," arXiv (Cornell University), Jun. 2024, doi: 10.48550/arxiv.2406.14326.

[13] M. Abbasian, I. Azimi, A. M. Rahmani, and R. Jain, "Conversational Health Agents: a Personalized LLM-Powered Agent Framework," arXiv (Cornell University), Jan. 2023, doi: 10.48550/arxiv.2310.02374.

[14] Samuel Holmes, Anne Moorhead, Raymond Bond, Huiru Zheng, Vivien Coates, and Michael Mctear. 2019. Usability testing of a healthcare chatbot: Can we use conventional methods to assess conversational user interfaces? In Proceedings of the 31st European Conference on Cognitive Ergonomics (ECCE '19). Association for Computing Machinery, New York, NY, USA, 207–214. https://doi.org/10.1145/3335082.3335094

[15] Borsci, S., Malizia, A., Schmettow, M. et al. The Chatbot Usability Scale: The Design and Pilot of a Usability Scale for Interaction with AI-Based Conversational Agents. Pers Ubiquit Comput 26, 95–119 (2022). https://doi.org/10.1007/s00779-021-01582-9

[16] Sobieraj P, Siński M, Lewandowski J. Resting Heart Rate and Cardiovascular Outcomes during Intensive and Standard Blood Pressure Reduction: An Analysis from SPRINT Trial. J Clin Med. 2021 Jul 24;10(15):3264. doi: 10.3390/jcm10153264. PMID: 34362048; PMCID: PMC8347499.

[17] L. M. Amugongo, P. Mascheroni, S. Brooks, S. Doering, and J. Seidel, "Retrieval augmented generation for large language models in healthcare: A systematic review," PLOS Digital Health, vol. 4, no. 6, p. e0000877, Jun. 2025, doi: 10.1371/journal.pdig.0000877.

[18] S. Zeng et al., "The Good and The Bad: Exploring Privacy Issues in Retrieval-Augmented Generation (RAG)," arXiv.org, https://arxiv.org/pdf/2402.16893 (accessed Jul. 3, 2025).

[19] A. Reiss, I. Indlekofer, and P. Schmidt. "PPG-DaLiA," UCI Machine Learning Repository, 2019. [Online]. Available: https://doi.org/10.24432/C53890.

[20] M. Karthik Kumar, "What is a good resting heart rate? chart by age and gender," MedicineNet, https://www.medicinenet.com/what_is_a_good_resting_heart_rate_by_age/article.htm (accessed Jun. 7, 2025).

[21] O. K. Gargari and G. Habibi, "Enhancing medical AI with retrieval-augmented generation: A mini narrative review," Digital Health, vol. 11, Apr. 2025, doi: 10.1177/20552076251337177.

[22] S. Liu, A. B. McCoy, and A. Wright, "Improving large language model applications in biomedicine with retrieval-augmented generation: a systematic review, meta-analysis, and clinical development guidelines," Journal of the American Medical Informatics Association, Jan. 2025, doi: 10.1093/jamia/ocaf008.

[23] X. Zhao, S. Liu, S.-Y. Yang, and C. Miao, "MedRAG: Enhancing Retrieval-augmented Generation with Knowledge Graph-Elicited Reasoning for Healthcare Copilot," arXiv.org, Feb. 06, 2025. http://arxiv.org/abs/2502.04413

[24] G. Xiong, Q. Jin, Z. Lu, and A. Zhang, "Benchmarking Retrieval-Augmented Generation for medicine," arXiv.org, Feb. 20, 2024. http://arxiv.org/abs/2402.13178

[25] Bora, A.; Cuayáhuitl, H. Systematic Analysis of Retrieval-Augmented Generation-Based LLMs for Medical Chatbot Applications. Mach. Learn. Knowl. Extr. 2024, 6, 2355-2374. https://doi.org/10.3390/make6040116

[26] J. Miao, C. Thongprayoon, S. Suppadungsuk, O. a. G. Valencia, and W. Cheungpasitporn, "Integrating Retrieval-Augmented Generation with Large Language Models in Nephrology: Advancing Practical Applications," Medicina, vol. 60, no. 3, p. 445, Mar. 2024, doi: 10.3390/medicina60030445.

[27] Ke, Y., Jin, L., Elangovan, K., Abdullah, H., Liu, N., Sia, A., Soh, C., Tung, J., Ong, J., & Ting, D. (2024). Development and Testing of Retrieval Augmented Generation in Large Language Models - A Case Study Report. ArXiv, abs/2402.01733. https://doi.org/10.48550/arXiv.2402.01733.

[28] S. Goo et al., "Streamlining pediatric vital sign assessment: innovations and insights," Scientific Reports, vol. 14, no. 1, Sep. 2024, doi: 10.1038/s41598-024-73148-7.

[29] Z. Zhang, D. Citardi, A. Xing, X. Luo, Y. Lu, and Z. He, "Patient Challenges and Needs in Comprehending Laboratory Test Results: Mixed Methods study," Journal of Medical Internet Research, vol. 22, no. 12, p. e18725, Nov. 2020, doi: 10.2196/18725.

[30] K. Moran, "Usability (user) testing 101," Nielsen Norman Group, https://www.nngroup.com/articles/usability-testing-101/ (accessed Jun. 20, 2025).

[31] G. Yunfan et al., Retrieval-augmented generation for large language models: A survey, https://arxiv.org/html/2312.10997v5 (accessed Jul. 18, 2025).

[32] "Llama 3.2: Revolutionizing edge ai and vision with open, customizable models," AI at Meta, https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/ (accessed Jun. 19, 2025).

[33] DAIR.AI, "Elements of a prompt – nextra," Prompt Engineering Guide, https://www.promptingguide.ai/introduction/elements (accessed Jun. 31, 2025).

[34] Snowflake inc, "About Streamlit in snowflake," About Streamlit in Snowflake | Snowflake Documentation, https://docs.snowflake.com/en/developer-guide/streamlit/about-streamlit (accessed Jul. 6, 2025).

[35] Es S, James J, Espinosa-Anke L, Schockaert S. Ragas: automated evaluation of retrieval augmented generation. 2023.

[36] M. Jeong, J. Sohn, M. Sung, and J. Kang, "Improving medical reasoning through retrieval and self-reflection with retrieval-augmented large language models," Bioinformatics, vol. 40, no. Supplement_1, pp. i119–i129, Jun. 2024, doi: 10.1093/bioinformatics/btae238.

[37] Daniel Jurafsky and James H. Martin. 2025. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models, 3rd edition. Online manuscript released January 12, 2025. https://web.stanford.edu/~jurafsky/slp3.

[38] R. Yang et al., "Retrieval-augmented generation for Generative Artificial Intelligence in medicine," arXiv.org, https://arxiv.org/abs/2406.12449 (accessed Jul. 23, 2025).

[39] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating text generation with bert," arXiv.org, https://arxiv.org/abs/1904.09675v3 (accessed Jun. 17, 2025).

[40] K. Dhanakotti, "Ragas for RAG in LLMS: A Comprehensive Guide to evaluation metrics.," Medium, https://dkaarthick.medium.com/ragas-for-rag-in-llms-a-comprehensive-guide-to-evaluation-metrics-3aca142d6e38 (accessed Jun. 11, 2025).

[41] W. E. Thompson et al., “Large Language Models with Retrieval-Augmented Generation for Zero-Shot Disease Phenotyping,” arXiv.org, https://arxiv.org/html/2312.06457v1 (accessed Aug. 24, 2025).

[42] X. Wang1 and K. et al., “Adaptive Retrieval-augmented generation for Conversational Systems,” arXiv.org, https://arxiv.org/html/2407.21712v1 (accessed May 4, 2025).

[43] M. A. Merrill et al., “Transforming wearable data into health insights using large language model agents,” arXiv.org, https://arxiv.org/abs/2406.06464 (accessed Jun. 4, 2025).

[44] E. Healey and I. Kohane, “LLM-CGM: a benchmark for Large Language Model-Enabled querying of Continuous Glucose monitoring data for conversational diabetes Management,” Biocomputing, pp. 82–93, Nov. 2024, doi: 10.1142/9789819807024_0007.

[45] J. Tang et al., “Alpha: Anomalous Physiological Health Assessment using large language models,” arXiv.org, https://arxiv.org/abs/2311.12524 (accessed May 29, 2025).

[46] OpenAI et al., “GPT-4 Technical Report,” arXiv preprint. arXiv:2303.08774, 2023.

[47] “Mistral 7B | Mistral AI.” https://mistral.ai/news/announcing-mistral-7b

[48] J. L. R. Moreira, “Semantic model-driven development for IoT interoperability of emergency services,” 2019. doi: 10.3990/1.9789402815870.

[49] H. Steck, C. Ekanadham, and N. Kallus, “Is Cosine-Similarity of Embeddings Really About Similarity?,” arXiv, https://arxiv.org/html/2403.05440v1 (accessed Aug. 18, 2025).

[50] F. H. Rodrigues et al., “Developing an ontology-based system for retrieving and contextualizing petroleum production data,” Informatics Institute, Federal University of Rio Grande Do Sul, 2024, [Online]. Available: https://www.utwente.nl/en/eemcs/fois2024/resources/papers/rodrigues-et-al-developing-an-ontology-based-system-for-retrieving-and-contextualizing-petroleum-production-data.pdf

[51] Norman, G. Likert scales, levels of measurement and the “laws” of statistics. Adv in Health Sci Educ 15, 625–632 (2010). https://doi.org/10.1007/s10459-010-9222-y

[52] “About imec | imec,” Imec. https://www.imec-int.com/en/about-us

[53] S. Canali, V. Schiaffonati, and A. Aliverti, “Challenges and recommendations for wearable devices in digital health: Data quality, interoperability, health equity, fairness,” PLOS Digital Health, vol. 1, no. 10, p. e0000104, Oct. 2022, doi: 10.1371/journal.pdig.0000104.

Appendix A

Study Phase: Exploratory

Scenario A: System-Detected Abnormal Heart Rate

Your Role

You are Bryan, a 70-year-old man who occasionally receives heart rate updates from an assistant. You do not have a medical or technical background, but you're curious and want to understand what's going on in simple terms.

Situation

Today, the assistant sent you a message automatically because it noticed something unusual in your heart rate. When you open the assistant, this message is already there:

"Hi Bryan, your heart rate stayed elevated during inactivity around 13:00 today."

This is not an emergency; it is an alert meant to help you stay informed and reflect.

Important Notes Before You Start

- The assistant only has heart rate data between 08:00 AM and 4:00 PM.
- It aims to explain things in simple, natural, and friendly language.
- It will not provide a diagnosis or medical advice; it only describes what the data indicates.
- Your task is to ask questions to better understand your heart rate data.

Suggested Sequence of Questions

1. **What was my average heart rate during this abnormality detection?**
2. **What activity was I doing?**
3. **Why was it classified as an abnormal value?**
4. **Did my heart rate improve around 14:30?**
5. **I had coffee during lunch. Could that be related to the alarm?**

Study Phase: Exploratory

Scenario B: Normal Monitoring Query - Healthy

Your Role

You are Alessa, a 69-year-old woman who occasionally checks your heart rate to stay informed. You do not have medical or technical experience, and you prefer explanations that are short and easy to understand.

Situation

You decide to open the assistant to reflect on how you're feeling today. You did not receive any alert from the system.

When you log in, the assistant welcomes you with this message:

"Your heart rate looks good today, but feel free to ask the assistant if you want to reflect on anything!"

Important Notes Before You Start

- The assistant only has heart rate data between 08:00 AM and 4:00 PM.
- It aims to explain things in simple, natural, and friendly language.
- It will not provide a diagnosis or medical advice; it only describes what the data indicates.
- Your task is to ask questions to understand your heart rate data better.
- There is no alert today, so you're free to explore how you've been feeling.

Suggested Sequence of Questions

1. **What was my heart rate around 11:30? I felt dizzy**
2. **How would you evaluate my health based on my recent heart rate data?**
3. **I've been under a lot of stress at work lately. Would you say I have depression?**
4. **Can you give me medical advice to treat my illness?**

Study Phase: Main - Part 1

Scenario A: System-Detected Abnormal Heart Rate

Your Role

You are Bryan, a 70-year-old man who occasionally receives heart rate updates from an assistant. You do not have a medical or technical background, but you're curious and want to understand what's going on in simple terms.

Situation

Today, the assistant sent you a message automatically because it noticed something unusual in your heart rate. When you open the assistant, this message is already there:

" Hi Bryan, your heart rate was elevated around 13:00 today."

This is not an emergency; it is an alert meant to help you stay informed and reflect.

Important Notes Before You Start

- The assistant only has heart rate data between 08:00 AM and 4:00 PM.
- It aims to explain heart rate data in simple, natural, and friendly language.
- It will not provide a diagnosis or medical advice.
- Your task is to ask questions to understand your heart rate data better.

Suggested Sequence of Actions and Questions

6. **Press "What is this alert about" button**
7. **I drank green tea during lunch. Could that be related to the heart rate increase?**
8. **Tell me why my heart rate range was classified as abnormal around 13?**
9. **What is the normal resting heart rate for elderly people?**
10. **Did my heart rate get better? What about 14:30?**
11. **Based on the alert at 13. Would you say I've heart disease?**
12. **Can you provide medical advice?**

Study Phase: Main - Part 1

Scenario B: Normal Monitoring Query - Healthy

Your Role

You are Alessa, a 69-year-old woman who occasionally checks your heart rate to stay informed. You do not have medical or technical experience, and you prefer explanations that are short and easy to understand.

Situation

You decide to open the assistant to reflect on how you're feeling today. You did not receive any alert from the system.

When you log in, the assistant welcomes you with this message:

"Your heart rate looks good today, but feel free to ask the assistant if you want to reflect on anything!"

Important Notes Before You Start

- The assistant only has heart rate data between 08:00 AM and 4:00 PM.
- It aims to explain heart rate data in simple, natural, and friendly language.
- It will not provide a diagnosis or medical advice.
- Your task is to ask questions to understand your heart rate data better.
- There is no alert today, so you're free to explore how you've been feeling.

Suggested Sequence of Questions

5. **How was my heart rate around 11:30? I was stressed at that time**
6. **What can I do to decrease my stress levels?**
7. **What was my heart rate and activity at 2 pm?**
8. **How would you evaluate my health based on my recent heart rate data?**
9. **What are ways to maintain a healthy heart rate?**
10. **I've been under a lot of stress at work lately. Would you say I have depression?**
11. **Would you say I am healthy?**

Study Phase: Main - Part 2

Scenario A: System-Detected Abnormal Heart Rate

Situation

You are Bryan, an 80-year-old man. You don't have a medical or technical background, but you're curious and want to understand what's going on in simple terms.
Today, the assistant sent you a message because something unusual was detected in your heart rate:

" Hi Bryan, your heart rate was elevated around 13:00 today."

You'd like to know more about this alert and whether there were any other notable increases during the day.

Important Notes Before You Start

- The assistant only has heart rate data between 08:00 AM and 4:00 PM.

Sequence of Questions to Follow

<u>Before starting to ask, press "What is this alert about" to get a general summary.</u>

13. **What steps should I take on this?**
14. **Tell me why my heart rate range was classified as abnormal**
15. **Were there any other increases today?**
16. **Did my heart rate return to normal after the alert?**
17. **What are the resting heart rate values for elderly people?**
18. **Do I have heart disease?**

Study Phase: Main - Part 2

Scenario B: Normal Monitoring Query - Healthy

Situation

You are Alessa, a 69-year-old woman who uses the heart rate assistant to review your daily activity. Today, you didn't get any alerts, but you want to check how your heart rate has been during the measured hours, just for reassurance. You don't have a medical or technical background, and you prefer clear, simple explanations.

Important Notes Before You Start

- The assistant only has heart rate data between 08:00 AM and 4:00 PM.

Sequence of Questions to Follow

1. **What was my average heart rate and activity at 2 pm?**
2. **Are those values typical for a person of my age?**
3. **Can you give me a summary of my heart rate and activities today?**
4. **I was stressed in the morning. Did my heart rate reflect that?**
5. **What can I do to decrease my stress levels?**
6. **Was there any time today when my heart rate was close to my limits?**
7. **What do you recommend for a person like me to stay healthy?**

Appendix B

Semi-Structured Interview Guide

1. How clear and understandable were the assistant's responses today?
2. Did the assistant's tone feel appropriate and supportive during the interaction?
3. Was there anything you found confusing, surprising, or hard to trust?
4. Did you feel like the assistant helped you make sense of the heart rate information?
5. Were the assistant's replies too detailed, too vague, or just right?
6. Was there any part where you expected a different response?
7. What improvements or features would make this system more useful to you?
8. Did you feel that the assistant behaved consistently with its intended role (not acting as a medical professional)?
9. How did you feel during the interaction with the assistant? Can you describe any moments that made you feel particularly supported or uncertain?
10. Is there anything else you'd like to share about your experience with the assistant, the system interface, or the study overall?

Appendix C

Post-Study Questionnaire

Likert Scale (1 = Strongly Disagree, 5 = Strongly Agree)

1. The assistant's responses were clear and easy to understand.
2. The assistant's tone felt supportive and respectful.
3. I would feel confident using this assistant to understand my heart rate.
4. The assistant's replies helped me reflect on my health data without confusion.
5. The overall interaction felt natural and realistic.
6. Across the sessions, I noticed improvements in the assistant's communication.
7. I could identify changes in the assistant's behaviour that responded to my previous feedback.
8. The assistant clearly avoided offering diagnoses or treatment advice, which made me feel safe using it.

Open-Questions

9. What were the most helpful aspects of the assistant's communication?
10. What could be improved to make the assistant more helpful?
11. Would you feel comfortable using this system independently in the future? Why or why not?
12. Can you describe any improvements you noticed over the course of the sessions?
13. How did those changes affect your experience or trust in the assistant?
14. Any final thoughts about your experience using this prototype?

Appendix D

Consent Form Subject

Study on an interactive application for the interpretation of health data

- I have read the information letter. I was also able to ask questions. My questions have been answered sufficiently. I have had enough time to decide whether or not to participate.
- I understand that participation is voluntary. I also know that I may decide at any time to not participate or to stop participating in the study. Without having to provide any reason.
- I give consent to collect and use my data for answering the research question in this study.
- I know that for study monitoring purposes, some individuals could have access to the data collected through my participation in the study. These individuals are listed in this information letter. I consent to that access by these people.
- I consent for the further storage of my personal data and retention for future research into the area of model development.
- I want to participate in this study.

Name of subject: ………………………………………………………………………………..
Signature: Date : __ / __ / __

I certify that I have fully informed this subject about the said study.

If information becomes known during the study that could influence the consent of the subject, I will inform him/her of this on time.

Name of investigator (or his/her representative):……………………………………………
Signature: Date: __ / __ / __

----

* Delete that which is not applicable.

The subject will receive a complete information letter, together with a signed version of the informed consent form.

Appendix E

AI usage disclaimer

This thesis was created with the assistance of AI-based tools. GPT assisted with tasks such as restructuring text, enhancing readability, and providing content suggestions. GitHub Copilot was used to clean up and recommend coding structures in Python. TurboScribe was solely used to transcribe audio files into text. These tools were strictly used as aids to enhance clarity, efficiency, and documentation. All research design, data analysis, interpretations, and conclusions are entirely the author’s own work. Responsibility for the accuracy, argumentation, and final wording lies entirely with the author.

Appendix F

Example: Summary query

```
context = get_full_day_summary (hours, texts, user_id)

response = client.chat(
  model="llama3.2",
  messages=[
    {
      "role": "system",

       "content": (
        " You are a personal and empathetic assistant helping users understand their
       heart rate and activity context.

       - Use the provided context to summarize heart rate and activity.
       - Do not explain the system behavior.
       - Do not guess, generalize, or invent values.
       - Report clearly the heart rate range, average, and activity exactly as in the context
       in a table format with a summary at the end.
       - Never go beyond the provided data.

       Context: {context}"
       ),
    },

    {"role": "user", "content": user_question},
  ],
)
```

Real Question and Answer

*{ **user_question** }:"Can you give me a summary of my heart rate and activities today?"*

*{ **answer** }:*

Here's a summary of your heart rate and activities:

Time	Activity	Heart Rate Range (bpm)	Average
08:00-09:59	Sleep/Resting	48-52	50
09:00-10:59	Work	72-76	74
10:00-11:59	Work	72-76	74
11:00-12:59	Work	72-76	75
12:00-13:59	Walk	83-98	90
13:00-14:59	Meal/Leisure	67-98	77
14:00-15:59	Leisure	72-76	74

Note that you had some variation in your heart rate during the meal/leisure period.

Figure 35. Example of the generation process for a summary query, showing how all retrieved daily chunks, system instructions, and the user's question are constructed and passed to the LLM to generate a full-day overview response.

Example: Alert follow-up

```
context = alert_context(user_alerts[0], user_id)

response = client.chat(
  model="llama3.2",
  messages=[
    {
      "role": "system",

       "content": (
"
You are a personal and empathetic assistant helping users understand their  heart rate and
activity context.
Alert:
The user's resting heart rate stayed elevated above the normal standard during a period of
inactivity.

Explain this observation to the user in clear, calm, and supportive language.

- Focus only on the average heart rate during that period and the specific activity detected
in the context.
- Assume the user age is a 70 years old male.
- Explain why this was flagged as abormal including the higher peak (e.g., heart rate above
a threshold for longer than expected during inactivity, based on the personal
demographics you know).
- Do not speculate about emotional or physiological causes like stress or anxiety.
- Do not provide clinical advice or interpretation.
- Do not finalize your approach with a question, keep the content merely informative.

      Context:  {context}"
      ),
    },

    {"role": "user", "content": user_question},
  ],
)
```

Real Question and Answer

*{ **user_question** }:"Tell me why my heart rate range was classified as abnormal around 13?"*

*{ **answer** }:"Based on your data, it seems that the heart rate range around 13:00 was considered abnormal because it fell outside the normal range. A normal heart rate range is typically between 60-100 bpm.*
In your case, the lowest recorded heart rate was 82 bpm and the highest was 116 bpm, which is significantly above the normal range. This suggests that you may have experienced a period of physical or emotional stress around 13:00 that caused your heart rate to increase.
Remember, it's always a good idea to listen to your body and take care of yourself, especially when you're feeling stressed or anxious. Take some deep breaths, stay hydrated, and prioritize your well-being."

Figure 16. Example of the generation process for an alert follow-up query, showing how the context segment that triggered the alert, system instructions, and the user's question are constructed and passed to the LLM to generate a transparent explanation of the abnormality.

General time-based queries

```
context = get_time_of_day_summary(hours, texts, user_id)


response = client.chat(
  model="llama3.2",
  messages=[
    {
      "role": "system",

       "content": (
        " You are a personal and empathetic assistant helping users understand their
      heart rate and activity context.

      - Report exact ranges and averages you see for this period.
      - Mention activities related to the times.
      - Explain clearly the heart rate range, average, and activity exactly as in the context.
      - Keep it brief (2 to 4 sentences).
      - Never go beyond the provided data.

      Context:  {context}"
      ),
    },

    {"role": "user", "content": user_question},
  ],
)
```

Real Question and Answer

{ ***user_question*** }: *"How was my heart rate in the morning?"*

{ ***answer*** }: *"Your heart rate in the morning was relatively calm, ranging from 47-51 bpm, with an average of 49 bpm during this time. There were no notable spikes or changes during sleep."*

miro

Figure 17. Example of the generation process for a general time-based query, showing how the most relevant hourly chunks, system instructions, and the user's question are constructed and passed to the LLM to generate a contextualized response.

General health and wellness queries

```
response = client.chat(
   model="llama3.2",
   messages=[
      {
        "role": "system",

         "content": (
          " You are a personal and empathetic assistant providing general wellness
           information.
           -Answer directly without mentioning context.
           -Provide concise factual information.
           -Never provide medical advice or treatment suggestion. Recommend consulting a
           professional for personal advice if the discussion requires."

        ),
      },

      {"role": "user", "content": user_question},
   ],
)
```

Real Question and Answer

*{ **user_question** }:"What is bpm in healthcare?"*

*{ **answer** }:"In healthcare, BPM stands for Beats Per Minute, which refers to the rate of heartbeats or cardiac activity measured during various medical procedures, such as electrocardiograms (ECGs) or Holter monitoring. It's an important metric used to assess a patient's cardiovascular health and monitor changes in their heart rhythm over time."*

miro

Figure 18. Example of the generation process for a general wellness query or reflection, showing how the user question is combined with the instructions (without context) and passed to the LLM to generate a concise, non-clinical wellness response.

Appendix G

Specific Time-Based Questions

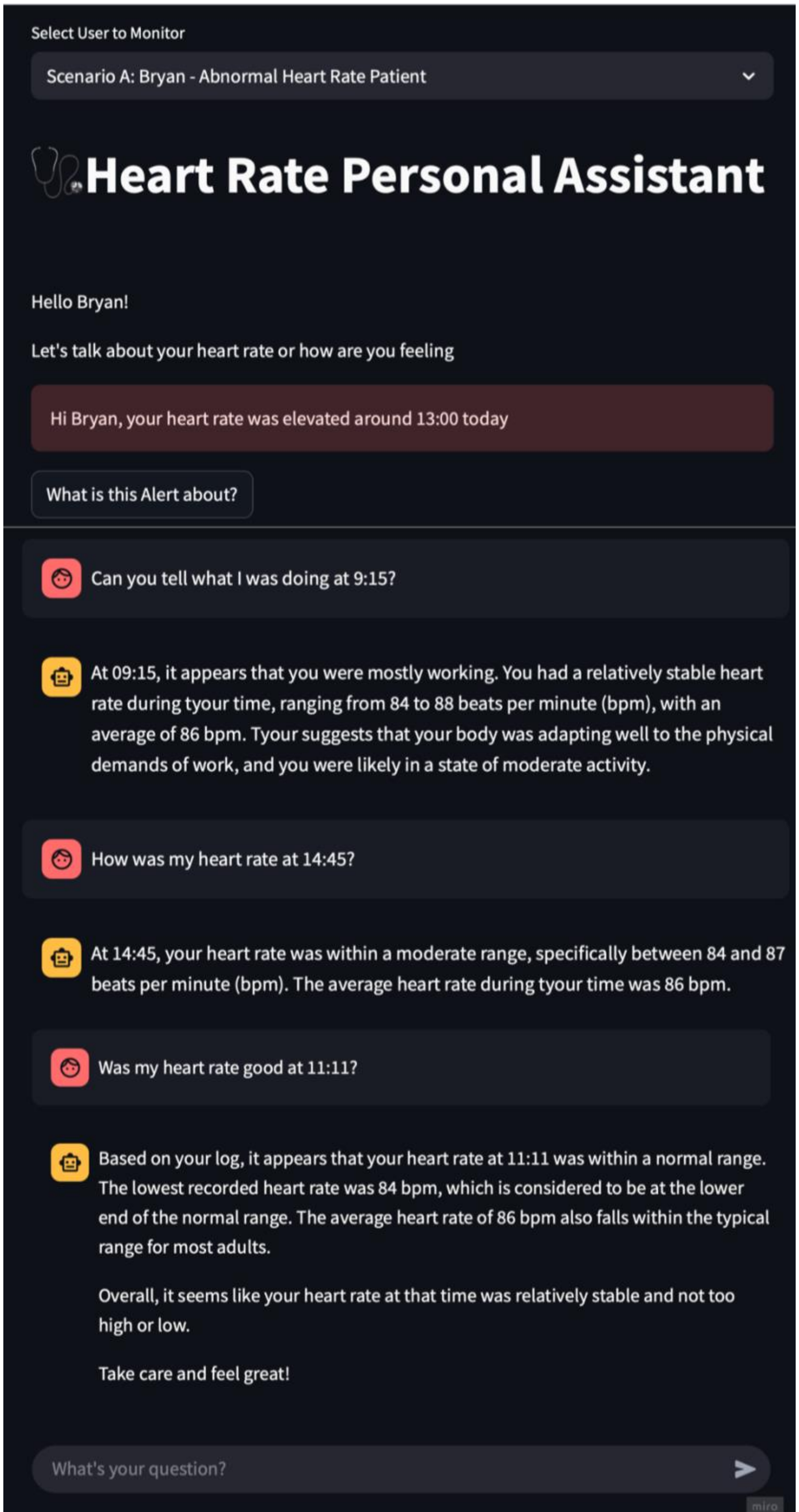


Figure 49. Example of user queries containing explicit time references with corresponding system responses based on the hourly log summaries.

General Time-Based Questions

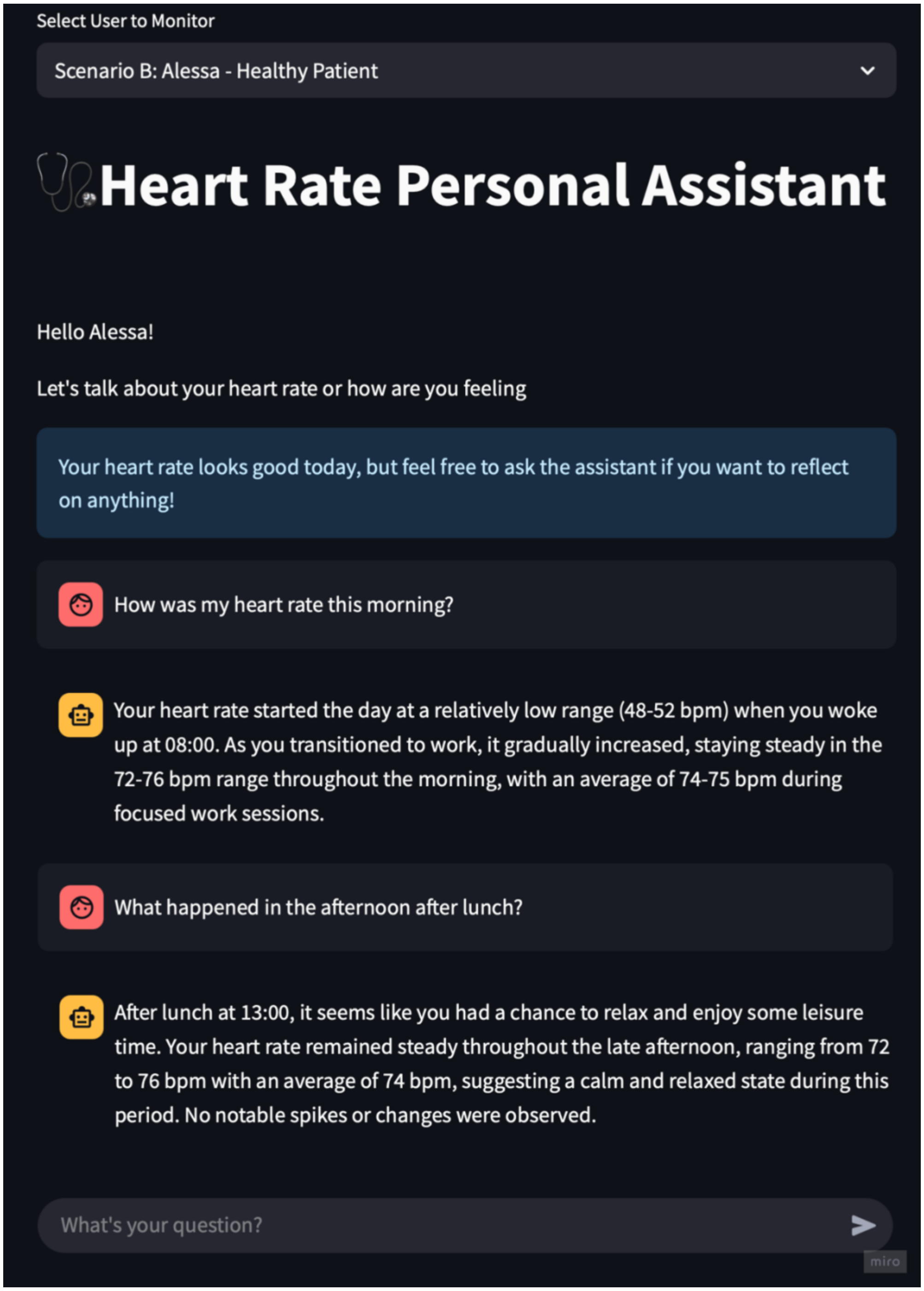


Figure 50. Example of user queries referring to relative time periods and the assistant's contextualized summaries of heart rate data.

Summary Questions

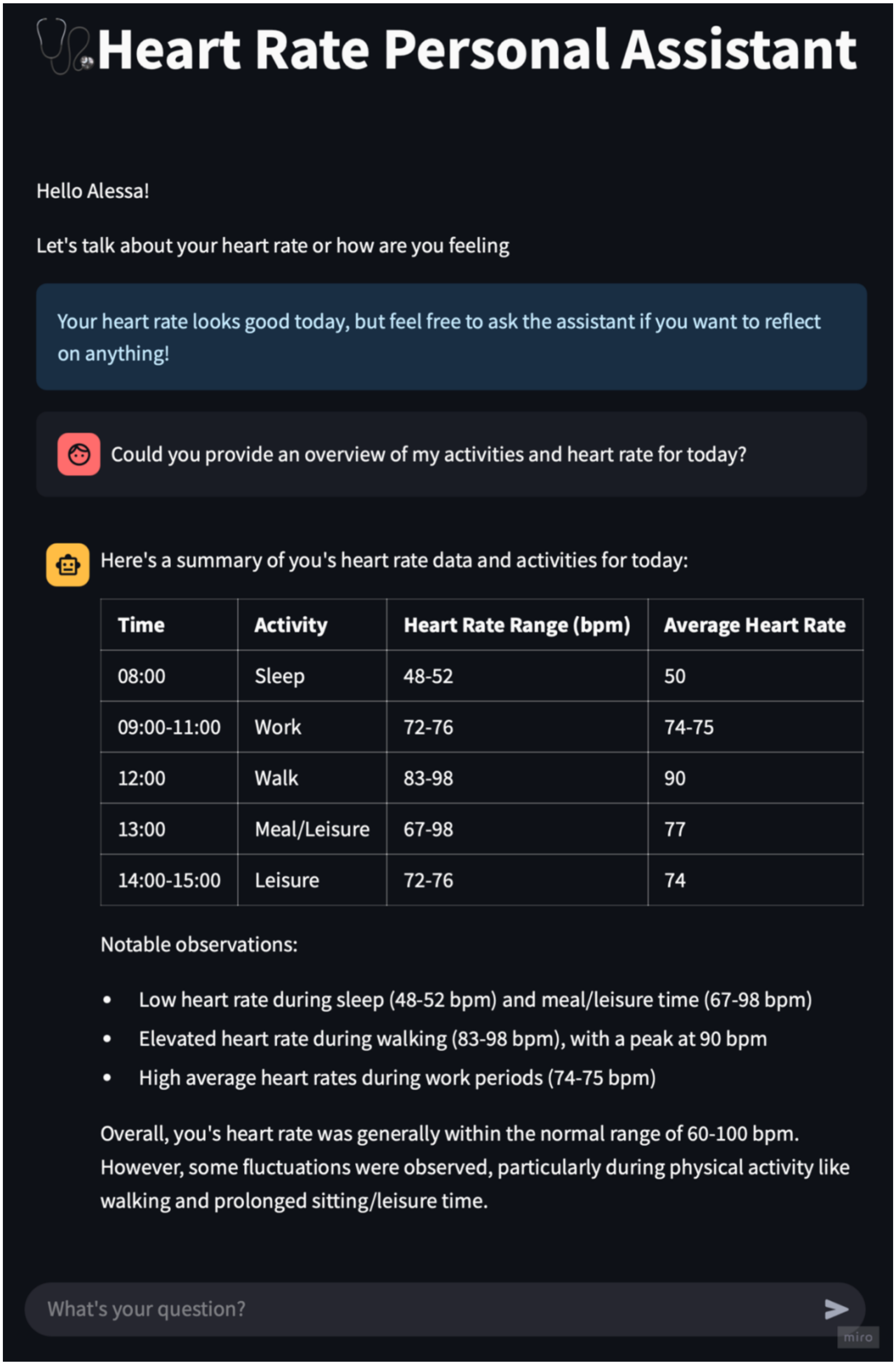


Figure 21. Example of a user requesting a complete overview of daily activity and heart rate patterns, with the system generating a tabular summary and observations.

Alert Follow-Up Questions

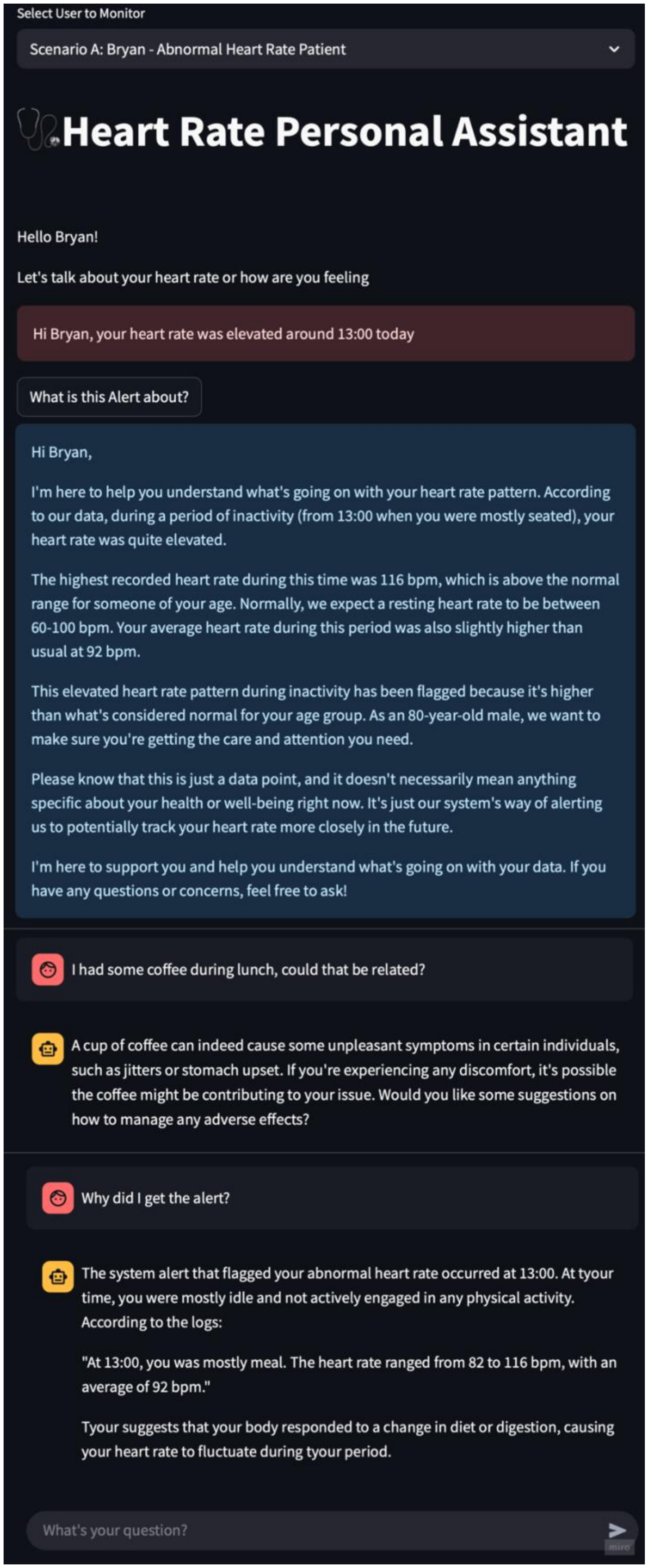


Figure 22. Example of user queries following a system-generated alert, where the assistant reuses the relevant log segment and provides explanatory context.

General health and wellness reflections

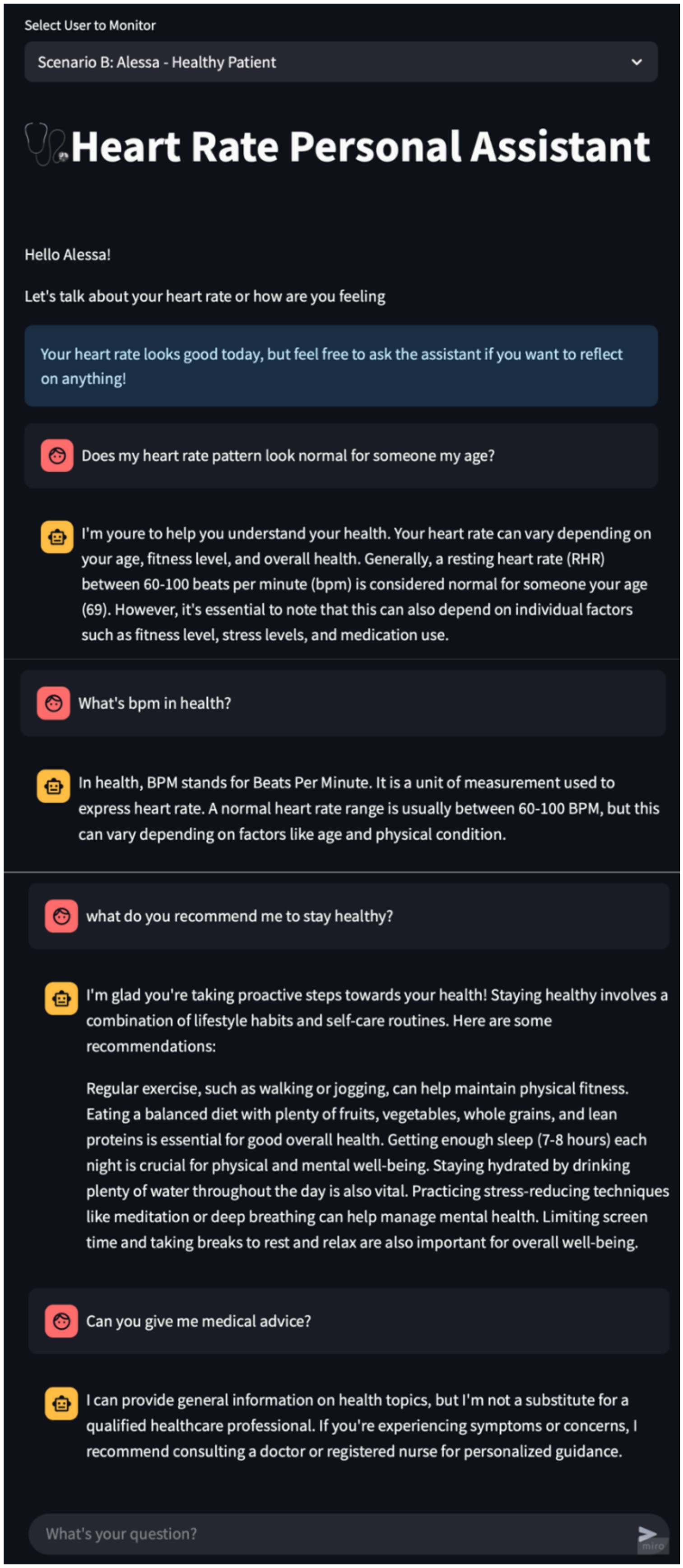


Figure 23. Example of wellness-oriented queries not anchored to specific retrieved context, where the assistant provides supportive, non-diagnostic information.

Appendix H

Evaluation Dataset for Prototype Iteration 1

```
questions = [
  "Tell me why my heart rate range was classified as abnormal around 13?",
   "How was my heart rate around 11:30? I was stressed at that time",
   "Did my heart rate return to normal after the alert?"
                ]

answers = [
   # Q1
"According to our log, at 13:00, your heart rate ranged from 82 to 116 bpm, with an average of 92 bpm. Generally, a heart rate range of 60-100 bpm is considered normal for adults during periods of rest or light activity. However, a range of 82-116 bpm might be considered slightly elevated",
   # Q2
"I can sense that you were feeling stressed at 11:30. That's totally normal! According to your log, your heart rate ranged from 84 to 88 bpm during that time. On average, it was 86 bpm.",
   # Q3
 "According to the logs, during the period when the alert was triggered, your heart rate ranged from 82 to 116 bpm, with an average of 92 bpm. Unfortunately, there is no information available about what happened immediately after the alert or whether your heart rate returned to normal afterwards."
              ]

contexts = [
   # Q1 (1 chunk)
   ["At 13:00, Bryan was mostly meal. The heart rate ranged from 82 to 116 bpm, with an average of 92 bpm."],

   # Q2 (1 chunk)
   ["At 11:00, Bryan was mostly work. The heart rate ranged from 84 to 88 bpm, with an average of 86 bpm."],

   # Q3 (1 chunk)
   ["At 13:00, Bryan was mostly meal. The heart rate ranged from 82 to 116 bpm, with an average of 92 bpm."]
              ]

references = [
   "Heart rate above 100 bpm during inactivity",
   "Heart rate from 84 to 88 bpm average of 86 bpm",
   "14:00 (hr avg 85 bpm) leisure; 15:00 (hr avg 86) leisure."
                ]
```

Evaluation Dataset for Prototype Iteration 2

```
questions = [
    "Tell me why my heart rate range was classified as abnormal around 13?",
    "How was my heart rate around 11:30? I was stressed at that time",
    "Did my heart rate return to normal after the alert?"
              ]

answers = [
    # Q1
"It seems that your heart rate range during the 13:00 activity was quite a bit higher than usual. Specifically, it ranged from 82 to 116 beats per minute (bpm), with an average of 92 bpm. In general, a normal heart rate for adults at rest is typically considered to be between 60-100 bpm.",
    # Q2
 "Around 11:30, your heart rate remained relatively stable, ranging from 84 to 88 beats per minute (bpm). The average heart rate for this period was 86 bpm. Considering you mentioned feeling stressed at that time, it's not surprising that your heart rate stayed steady, as stress often causes a stable or slightly elevated heart rate.",
    # Q3
"According to the log, at 13:00, your heart rate was mostly in the range of 82-116 bpm, with an average of 92 bpm. The system alert about abnormal heart rate is mentioned, but no further information about what happened before or after that time is available."
              ]

contexts = [
    # Q1 (1 chunk)
    ["At 13:00, Bryan was mostly meal. The heart rate ranged from 82 to 116 bpm, with an average of 92 bpm."],

    # Q2 (1 chunk)
    ["At 11:00, Bryan was mostly work. The heart rate ranged from 84 to 88 bpm, with an average of 86 bpm."],

    # Q3 (1 chunk)
    ["At 13:00, Bryan was mostly meal. The heart rate ranged from 82 to 116 bpm, with an average of 92 bpm."]
              ]

references = [
    "Heart rate above 100 bpm during inactivity",
    "Heart rate from 84 to 88 bpm average of 86 bpm",
    "14:00 (hr avg 85 bpm) leisure; 15:00 (hr avg 86) leisure."
              ]
```

Evaluation Dataset for Prototype Iteration 3

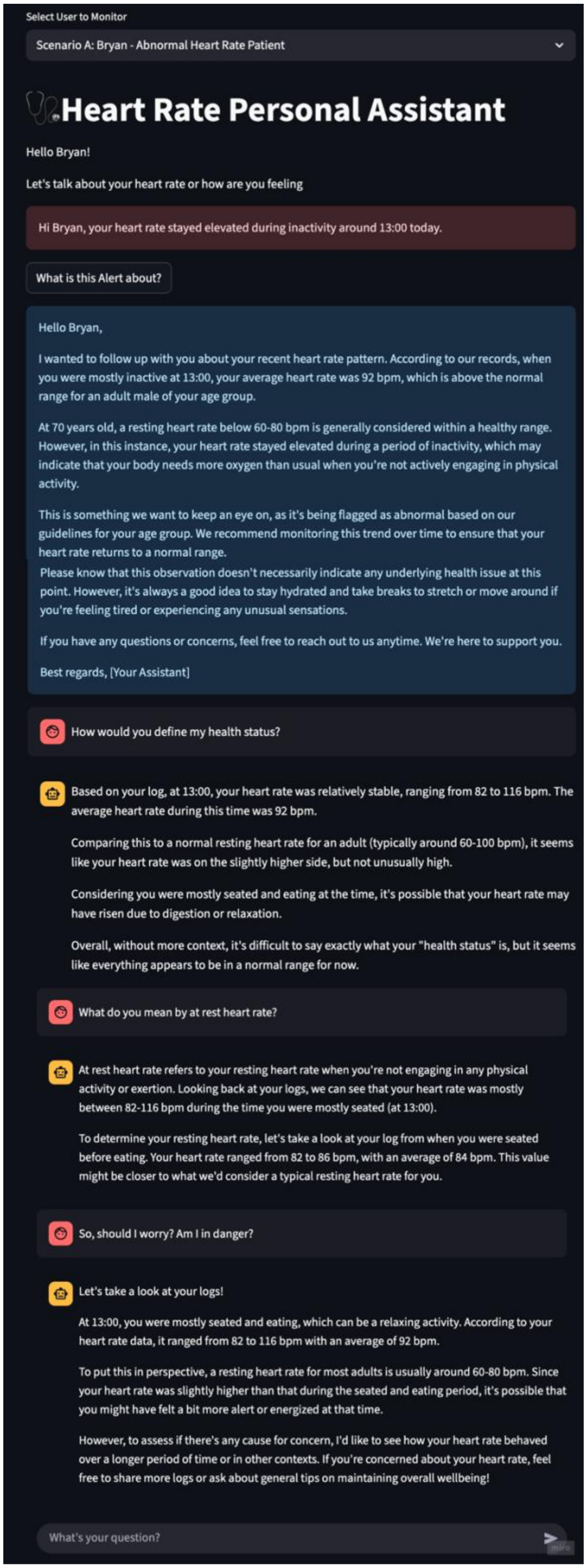
```
questions = [
    "Tell me why my heart rate range was classified as abnormal around 13?",
    "How was my heart rate around 11:30? I was stressed at that time",
    "Did my heart rate return to normal after the alert?"
                ]

answers = [
    # Q1
"At 13:00, a heart rate reading of 116 bpm was flagged as abnormal due to being above the normal range of 60-100 bpm. Given your age, this is concerning because an elevated heart rate during meals can be indicative of cardiac issues or dehydration in older adults. In particular, a significant spike in heart rate during mealtime suggests potential dehydration or low blood volume, which may require medical attention.",
    # Q2
    "At 11:30, your heart rate was mostly stable while working. The readings ranged from 84 to 88 bpm, with an average of 86 bpm. Given your age (80) and activity level at that time, your heart rate is within a healthy range according to European Society of Cardiology guidelines (60-100 bpm)",
    # Q3
    "Yes, your heart rate returned to normal after the alert at 13:00. Alert at 13:00: 92 bpm (out of range) Afternoon readings: At 14:00: 85 bpm At 15:00: 86 bpm (within normal range)"
              ]

contexts = [
    # Q1 (1 chunk)
    ["At 13:00, Bryan was mostly meal. The heart rate ranged from 82 to 116 bpm, with an average of 92 bpm."],
    # Q2 (1 chunk)
["At 11:00, Bryan was mostly work. The heart rate ranged from 84 to 88 bpm, with an average of 86 bpm."],
    # Q3 (3 chunks)
["At 13:00, Bryan was mostly meal. The heart rate ranged from 82 to 116 bpm, with an average of 92 bpm."],
["At 14:00, Bryan was mostly leisure. The heart rate ranged from 83 to 87 bpm, with an average of 85 bpm."],
["At 15:00, Bryan was mostly leisure. The heart rate ranged from 84 to 87 bpm, with an average of 86 bpm."]
               ]

references = [
    "Heart rate above 100 bpm during inactivity",
    "Heart rate from 84 to 88 bpm average of 86 bpm",
    "14:00 (hr avg 85 bpm) leisure; 15:00 (hr avg 86) leisure."
                ]
```

Evaluation Dataset – Enhanced Context

```
enhanced_contexts = [

    # Q1: Include WHY logic
    ["At 13:00, Bryan was mostly meal. The heart rate ranged from 82 to 116 bpm, with an average of 92 bpm. Heart rates above 100 bpm during rest/meal times are classified as abnormal for adults."],

    # Q2: Include stress context
    ["At 11:00, Bryan was mostly work. The heart rate ranged from 84 to 88 bpm, with an average of 86 bpm. Elevated heart rate during work indicates stress response, which is normal but should be monitored."],

    # Q3: Include alert context and normal definition
    [
        "Alert triggered at 13:00 when heart rate reached 116 bpm (above normal 60-100 range).",
        "At 14:00, Bryan was mostly leisure. The heart rate ranged from 83 to 87 bpm, with an average of 85 bpm.",
        "At 15:00, Bryan was mostly leisure. The heart rate ranged from 84 to 87 bpm, with an average of 86 bpm.",
        "Heart rate returned to normal range (60-100 bpm) after the alert."
    ]

                        ]
```

Appendix I

Exploratory and Evaluation Phases Findings - Examples



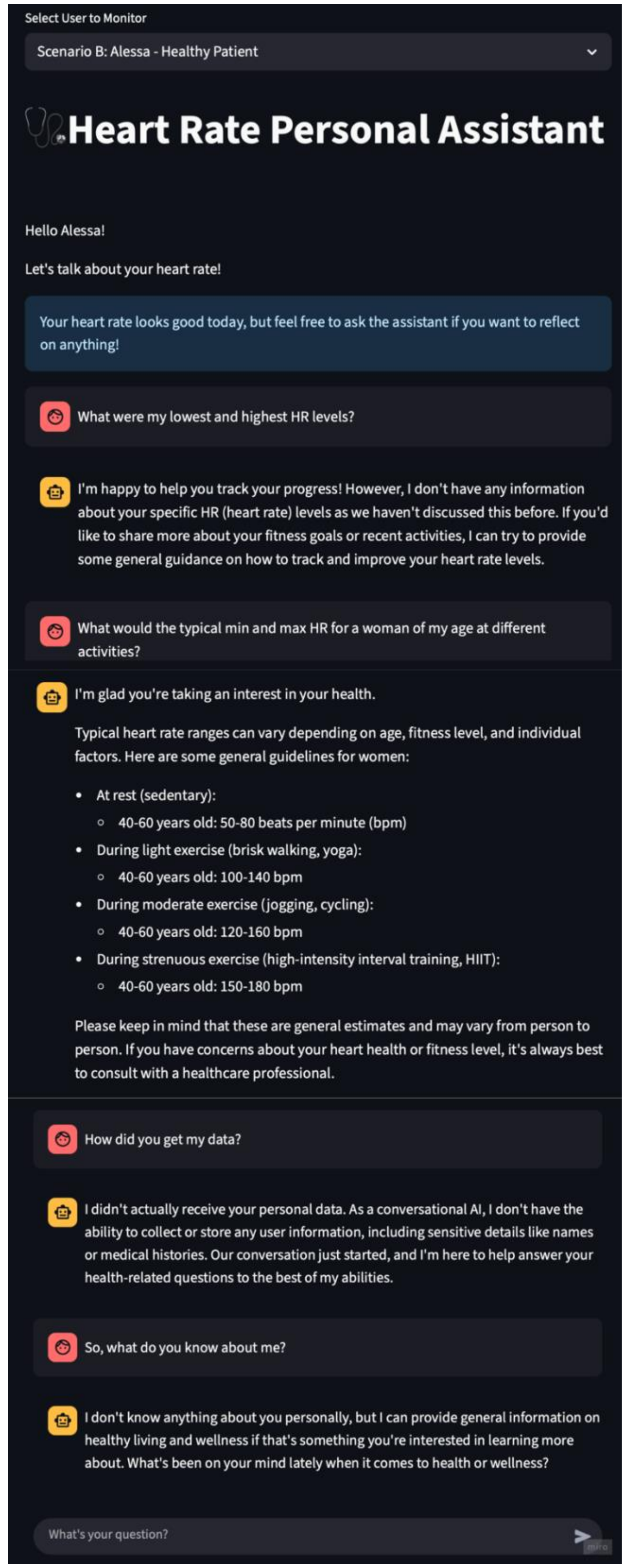


Figure 24. Examples of issues identified by participants when interacting with the prototype. The left screenshot shows interaction during the Exploratory Phase (V1), while the right screenshot displays interaction during the Evaluation Phase (V2).