

Article

EverydAI: Virtual Assistant for Decision-Making in Daily Contexts, Powered by Artificial Intelligence

Carlos E. Pardo B., Oscar I. Iglesias R. , Maicol D. León A. and Christian G. Quintero M. * 

Department of Electrical and Electronics Engineering, Universidad del Norte, Barranquilla 081007, Colombia; ecpardo@uninorte.edu.co (C.E.P.B.); iglesiasso@uninorte.edu.co (O.I.I.R.); maicoll@uninorte.edu.co (M.D.L.A.)

* Correspondence: christianq@uninorte.edu.co

Abstract

In an era of information overload, artificial intelligence plays a pivotal role in supporting everyday decision-making. This paper introduces EverydAI, a virtual AI-powered assistant designed to help users make informed decisions across various daily domains such as cooking, fashion, and fitness. By integrating advanced natural language processing, object detection, augmented reality, contextual understanding, digital 3D avatar models, web scraping, and image generation, EverydAI delivers personalized recommendations and insights tailored to individual needs. The proposed framework addresses challenges related to decision fatigue and information overload by combining real-time object detection and web scraping to enhance the relevance and reliability of its suggestions. EverydAI is evaluated through a two-phase survey, each one involving 30 participants with diverse demographic backgrounds. Results indicate that on average, 92.7% of users agreed or strongly agreed with statements reflecting the system's usefulness, ease of use, and overall performance, indicating a high level of acceptance and perceived effectiveness. Additionally, EverydAI received an average user satisfaction score of 4.53 out of 5, underscoring its effectiveness in supporting users' daily routines.

Keywords: artificial intelligence; augmented reality; digital avatars; large language models; object detection; recommendation systems; virtual assistants; web scraping



Academic Editor: Polinapilinho Katina

Received: 18 July 2025

Revised: 14 August 2025

Accepted: 29 August 2025

Published: 31 August 2025

Citation: Pardo B., C.E.; Iglesias R., O.I.; León A., M.D.; Quintero M., C.G. EverydAI: Virtual Assistant for Decision-Making in Daily Contexts, Powered by Artificial Intelligence. *Systems* **2025**, *13*, 753. <https://doi.org/10.3390/systems13090753>

Correction Statement: This article has been republished with a minor change. The change does not affect the scientific content of the article and further details are available within the backmatter of the website version of this article.

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Decision-making in uncertain or information-rich environments remains one of the most pervasive and cognitively demanding challenges that individuals face in modern society. Whether consciously or unconsciously, people are required to make countless decisions throughout the day, many of which occur under conditions of uncertainty, limited time, or insufficient information. In this context, research conducted in [1] involving a survey of 250 respondents found that approximately 87% of participants reported feeling overwhelmed by the sheer number of available options when making everyday choices.

This sense of cognitive overload is often associated with decision fatigue, confusion, and suboptimal outcomes, ultimately reducing individuals' ability to act efficiently in daily life. This issue manifests in various seemingly simple but recurrent activities such as planning meals, choosing an appropriate exercise routine, or deciding what to wear, where individuals may encounter uncertainty, frustration, or lack of direction. In many cases, the sheer abundance of information and alternatives does not ease the process but instead complicates it, particularly when recommendations are not tailored to the user's personal preferences, habits, or contextual needs. Studies have shown that such decision

environments can trigger impulsive behaviors or lead users to avoid decision-making altogether, further exacerbating the problem [2].

Moreover, even when information is readily available, its utility is limited if users lack the domain-specific knowledge required to interpret it effectively. For example, tasks like selecting ingredients for a recipe or composing a nutritionally balanced meal involve implicit knowledge, such as understanding ingredient compatibility or preparation methods [3].

In light of these challenges, there has been a growing interest in the development of intelligent virtual assistants capable of providing real-time, personalized support in everyday decision-making scenarios. These systems leverage recent advancements in artificial intelligence (AI), such as context-aware recommendation engines, natural language understanding, and real-time interaction models, to offer more meaningful and relevant guidance to users [4]. The integration of cutting-edge technologies, including augmented reality (AR), object detection, and large language models (LLMs), has significantly enhanced the ability of these assistants to interpret both the user's environment and intentions, thereby enabling a more seamless and adaptive user experience. This technological evolution prompts a central research question: Can an AI-powered system be effectively designed to support people in making faster, more informed, and personally relevant decisions in the context of their daily routines? [5].

To address this need, the present work proposes the design and implementation of a virtual assistant powered by advanced AI technologies, specifically tailored to assist users in routine decision-making tasks across diverse areas such as cooking, fashion, and fitness using computer vision in order to extract the contextual information of the objects available to the user, such as ingredients, clothes, or gym equipment, and give recommendations. Figure 1 shows the general idea of the proposed system.

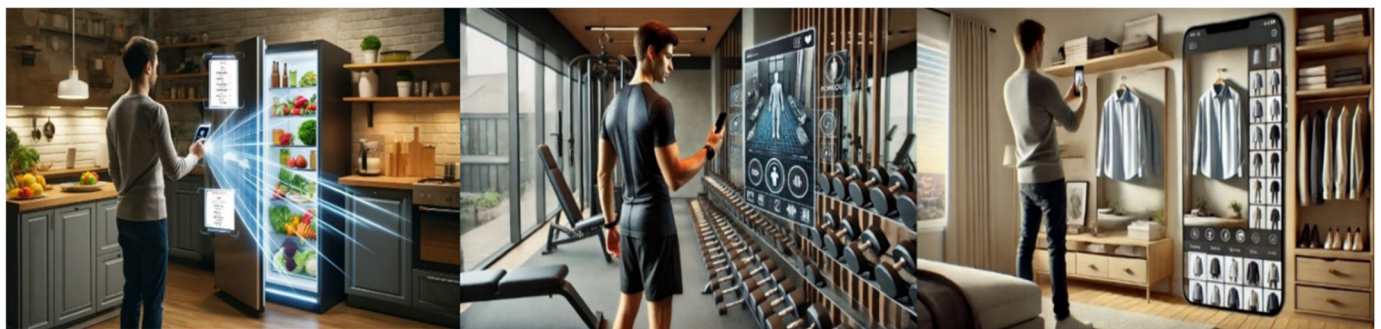


Figure 1. An illustration of the general idea. See the Acknowledgments section for details.

2. Related Works

2.1. Theoretical Framework

The foundational components of the assistant such as LLMs, virtual assistants, object detection, recommender systems, digital avatars, augmented reality, and web scraping are outlined below to provide context for the technologies involved in this research.

2.1.1. Large Language Models

A LLM is an AI model based on transformer architecture, designed to understand and process human language by analyzing vast amounts of text data. These models excel at comprehending context, generating human-like text, providing contextualized responses, and following instructions effectively. In recent years, LLMs have gained significant popularity and are widely used in traditional tasks such as search engines, customer support, translation, and education. Additionally, they have proven valuable in specialized fields like code generation, healthcare, and finance [6–8].

2.1.2. Virtual Assistants

A virtual assistant (VA) is a software agent that performs tasks or services based on user commands, utilizing AI and natural language processing (NLP) to interpret information, provide suggestions, and respond in real time through text or speech. Widely used in industries such as finance, healthcare, education, and marketing, VAs enhance efficiency by offering full-time availability, multilingual capabilities, and extensive knowledge access. Their adoption in Industry 4.0 (I4.0) has grown due to their role in data-driven decision-making, but challenges like user privacy, security risks, and ethical concerns remain obstacles to broader implementation [9].

2.1.3. Object Detection

Object detection is a fundamental task in computer vision that involves identifying and locating instances of objects within digital images. It is used to recognize various object classes, such as humans, animals, or vehicles, and plays a crucial role in numerous applications. The effectiveness of object detection is evaluated using metrics such as precision, recall and the mean average precision (mAP), which summarizes precision and recall across all object classes to provide a single performance score [10].

2.1.4. Recommendation Systems

Recommender systems are AI-driven algorithms designed to analyze data and suggest relevant content to users. They are widely used in platforms such as streaming services, e-commerce, and social media to personalize user experiences. These systems operate through techniques like collaborative filtering, content-based filtering, context-based filtering, or hybrid approaches that combine multiple methods. Their goal is to enhance user experience, increase engagement, and optimize decision-making. However, they also face challenges such as privacy concerns, bias in recommendations, and transparency in their decision-making process [11–13].

2.1.5. Digital Avatars

Digital 3D avatars are virtual representations of individuals that are created using three-dimensional graphics. These avatars can be customized to resemble real people or take on entirely fictional appearances, often used in video games, virtual reality (VR), AR, metaverse platforms, and digital assistants. They can range from static models to highly interactive avatars capable of mimicking facial expressions, body movements, and even speech using AI and motion-capture technology. Digital 3D avatars enhance user interaction, social presence, and personalization in digital environments, playing a crucial role in entertainment, online communication, and immersive experiences [14].

2.1.6. Augmented Reality

AR is technology that overlays digital elements, such as images, sounds, or 3D objects, onto the real world in real time. Unlike VR, which immerses users in a completely digital environment, AR enhances the physical world by integrating virtual content through devices like smartphones, AR glasses, and headsets. AR relies on computer vision, sensors, and AI to recognize the surrounding environment and seamlessly place digital information within it [15].

2.1.7. Web Scraping

Web scraping is the automated process of extracting data from websites to gather and analyze information efficiently. It is widely used in industries such as market research, finance, e-commerce, and social media analysis to collect data for insights, trend monitoring, and decision-making [16].

2.2. State of the Art

In this section, we explore previous research on virtual assistants, object-detection, and web-scraping techniques within the domains of cooking, fashion, and fitness to understand key methodologies and insights that contribute to the system development.

First, the study in [17] introduces GptVoiceTasker, an advanced virtual assistant leveraging LLMs to enhance task efficiency and user experience on mobile devices. By employing prompt engineering, GptVoiceTasker interprets user commands with high accuracy, automates repetitive tasks based on historical usage, and streamlines device interactions. This virtual assistant is designed to improve smartphone interaction through voice commands; however, it faces several limitations, such as reliance on prior interactions with applications, difficulties handling complex tasks, interference from dynamic user interfaces, latency issues, and privacy concerns. Looking ahead, the goal is to generalize common tasks, adapt the system to devices like smartwatches and AR/VR platforms, improve handling of unexpected real-time interface elements, and optimize the user experience with features such as live transcriptions, while strengthening data security. Key research questions include how to enhance adaptability to new applications, manage dynamic interfaces, and ensure privacy while minimizing latency.

Secondly, the research in [18] introduces BIM-GPT, a prompt-based virtual assistant framework that integrates Building Information Models (BIMs) with generative pre-trained transformers (GPT) to enable efficient natural language-based information retrieval. The BIM-GPT framework presents several limitations, including its evaluation on a single dataset that does not encompass all possible query types, the exclusion of geometric information such as width, length, and height, a latency of approximately five seconds due to multiple API calls, the lack of support for multi-turn conversations, and the risk of generating irrelevant or inaccurate information without carefully designed prompts. To address these limitations, future work suggests expanding datasets to include more diverse information, optimizing performance to reduce latency, developing multi-turn interaction capabilities, refining prompts to improve control over text generation, and exploring the impact of different GPT models on framework performance.

The research presented in [19] introduces a Python-based Voice Assistant designed to perform routine tasks such as checking the weather, streaming music, searching Wikipedia, and opening desktop applications. Leveraging AI technology, this assistant simplifies user interactions by integrating functionalities similar to popular virtual assistants like Alexa, Cortana, Siri, and Google Assistant. However, the current system's functionality is limited to working exclusively with application-based data, restricting its ability to handle more diverse or real-time external information sources.

The study in [20] presents a virtual assistant developed using open-source technologies, integrating natural language processing (NLP) and speech recognition to interpret user requests, while Machine Learning (ML) improves responsiveness over time. Trained on a dataset of queries and responses, and enhanced with various Python libraries, the system supports tasks such as sending messages and emails, making calls, playing YouTube videos, automating Chrome usage, opening desktop applications, and engaging in conversation. Performance evaluation showed a 90% accuracy rate in interpreting and responding to queries, response times of only a few seconds, and high user satisfaction in terms of ease of use, helpfulness, and overall performance. These results demonstrate the assistant's accuracy, responsiveness, and user-friendliness, highlighting its potential for applications across domains such as healthcare, education, and customer service.

The study in [21] explores how generative AI can enhance Human–Machine Interaction (HMI) by enabling virtual assistants to generate contextually relevant responses and adapt to diverse user inputs. The proposed generative AI virtual assistant is developed

using a modular methodology that integrates multiple open-source tools, each selected for its unique strengths: Gradio for creating intuitive user interfaces, Play.ht for natural-sounding text-to-speech, Hugging Face for access to robust pre-trained models, OpenAI for advanced language generation, LangChain for improved contextual understanding, Google Colab for scalable cloud-based model training, and blockchain technology for secure and transparent data handling. The system is trained and refined on datasets from Hugging Face and evaluated using metrics such as accuracy, precision, recall, and F1 score, achieving high contextual understanding and responsiveness. Additional functionalities included web-based interactivity via Gradio components, immersive voice feedback from Play.ht, and iterative performance monitoring for continuous improvement. Limitations identified include scalability challenges with Gradio in complex scenarios, the occasional inability of Play.ht to capture subtle emotional nuances, customization constraints from Hugging Face's reliance on existing datasets, inherent biases, the computational costs of large language models, and resource restrictions in Google Colab. Future directions involve optimizing scalability, improving emotional expressiveness in speech synthesis, enhancing adaptability to evolving conversational patterns, refining bias mitigation strategies, and expanding applicability across diverse domains while ensuring user privacy and maintaining low latency.

The study in [22] introduces LLaVA-Med, a vision–language conversational assistant designed to answer open-ended biomedical research questions by combining multimodal AI with domain-specific datasets. The model is trained cost-efficiently using a large-scale biomedical figure-caption dataset from PubMed Central and GPT-4-generated instruction-following data, applying a curriculum learning strategy where it first aligns biomedical vocabulary from captions, then learns open-ended conversational semantics. Fine-tuning is performed on a general-domain vision–language model using eight A100 GPUs in under 15 h, achieving strong results and outperforming prior supervised state-of-the-art on several biomedical visual question-answering benchmarks. LLaVA-Med integrates a self-instruct data curation pipeline leveraging GPT-4 and external knowledge to enhance contextual understanding and chat capabilities. Limitations include domain specificity that hinders generalization beyond biomedical tasks, susceptibility to hallucinations and inherited data biases, dependency on input quality and image resolution (currently 224×224), and reduced reliability for unseen question types. Future directions aim to improve reasoning depth, mitigate hallucinations, enhance resolution and data quality, and expand the methodology to other vertical domains, while ensuring outputs remain reliable, interpretable, and clinically useful.

The research presented in [23] introduces an AI-based food quality-control system that integrates the YOLOv8 object-detection model to identify and locate spoiled food in real time, leveraging high-resolution images and cloud-based visualization tools. The system uses YOLOv8's advanced recognition capabilities to detect damaged products in complex production flows, providing rapid quality assessments that improve efficiency and reduce product losses. Cloud integration enables dynamic visualization of detection accuracy, model behavior, and potential biases, allowing for continuous monitoring and real-time adjustments to optimize performance. While the approach offers significant benefits, its effectiveness depends on the quality and diversity of training images, the robustness of cloud connectivity, and the model's ability to handle unseen food types or extreme environmental conditions. Future directions include expanding the dataset to improve generalization, integrating predictive analytics for early spoilage detection, enhancing explainability for quality assurance compliance, and scaling the system for deployment across diverse food production environments.

The study in [24] introduces BiTrains-YOLOv8, an enhanced food-detection model built on YOLOv8 and optimized for real-time applications through advanced supervision techniques that improve accuracy, robustness, and generalization to unseen food items. The model integrates dual training strategies to enhance detection performance, achieving an average accuracy of 96.7%, precision of 96.4%, recall of 95.7%, and F1 score of 96.04%, outperforming baseline YOLOv8 in speed and resilience, even with partially obscured or hidden food items. Its rapid inference capabilities make it suitable for applications such as dietary assessment, calorie counting, smart kitchen automation, and personalized nutrition.

The research presented in [25] proposes an improved YOLO-based food packaging safety-seal detection system, addressing inefficiencies of traditional manual inspection through an optimized network structure, targeted training strategies, and anchor frame design tailored for small object detection. Comparative experiments against a CNN-based closure detection system show consistently higher recall and F1 scores for the improved YOLO model under identical datasets and iterations, with performance gaps increasing over time, demonstrating superior accuracy, reliability, and efficiency in identifying subtle packaging defects. The authors outline future enhancements by integrating advanced sensing technologies such as high-resolution cameras and multispectral imaging to increase detection resolution, expand coverage, and enable more comprehensive and refined defect recognition.

Reference [26] proposes an automated billing system for self-service restaurants, tailored for Indian cuisine, that leverages computer vision and the YOLO object-detection algorithm—specifically YOLOv8, with comparative use of YOLOv5—to identify food items from images and extract details such as names and prices. A graphical user interface visually demonstrates the process, enabling waiters or customers to generate itemized bills from a single photograph, thereby reducing manual entry errors and alleviating cashier congestion during peak times. Tested on the IndianFood10 dataset, the system achieved a 91.9% mean average precision, surpassing previous state-of-the-art results. Future work aims to extend capabilities to volumetric detection of food items and fully automate self-service billing through menu integration supported by geolocation technologies.

The study in [27] presents a computer vision-based system for clothing recognition and real-time recommendations in e-commerce, leveraging an improved YOLOv7 architecture. To address the challenges of accurate and efficient clothing detection, the authors redesign the YOLOv7 Backbone and integrate the PConv (Partial Convolution) operator to boost inference speed without compromising accuracy. The system also incorporates the CARAFE (Content-Aware ReAssembly of FEatures) upsampling operator to preserve crucial semantic information and the MPDIoU (Minimum Point Distance Intersection over Union) loss function to optimize recognition frame selection. Experimental results show a 9.2% increase in inference speed (95.3 FPS), a 20.5% reduction in GFLOPs, and improved detection accuracy compared to the native YOLOv7, demonstrating the model's superior performance and effectiveness in clothing recognition for e-commerce and live-streaming applications.

The study in [28] presents FashionAI, a YOLO-based computer-vision system designed to detect clothing items from images and provide real-time recommendations. Leveraging YOLOv5 for object detection, a custom CNN for clothing pattern identification, and ColourThief for dominant color extraction, the system analyzes user-uploaded images to recognize garments such as shirts, pants, and dresses. It then employs web scraping to match these features with similar products on e-commerce platforms like Myntra, enhancing the online shopping experience by offering personalized and visually similar clothing suggestions.

Finally, the research in [29] conducts a comparative experimental evaluation of Selenium and Playwright for automated web scraping, focusing on reliability metrics such

as uptime and the Rate of Occurrence of Failures (ROCOF) under controlled 24 h tests on HDD- and SSD-equipped laptops. Both tools executed identical Python scripts, with uptime and ROCOF measured at multiple time intervals to account for hardware and time-of-day effects. Selenium achieved 100% uptime with no failures, while Playwright reached 99.72% uptime with four downtimes but demonstrated more stable and predictable execution times. Although both had one failure per 10-test sequence, Selenium’s faster execution times resulted in a higher failure rate per second, whereas Playwright’s slower but consistent performance proved advantageous for dynamic web application testing. The results also confirmed SSD hardware’s superiority over HDD in speed and stability.

Table 1 presents a summary of the related works.

Table 1. A summary of related works, highlighting for each reference the study’s main contributions, its potential impact on the proposed system design, and the corresponding research area.

Ref.	Work	Impact	Area
[17]	A mobile virtual assistant that automates tasks by learning from user commands	Encourages Open AI and function calling	General task automation on mobile devices.
[18]	A virtual assistant for retrieving and summarizing BIM-related information	Encourages usage of GPT models	Construction industry
[19]	General-purpose virtual assistant for task automation	Recommends python as programming language	General services
[20]	A prototype virtual assistant with NLP, speech recognition, and ML-based improvements	Python as programming language	General services
[21]	Developing a generative AI-powered virtual assistant capable of contextual adaptation.	Encourages usage of Open AI models to power virtual assistants	Human–Machine Interaction
[22]	A multimodal conversational AI assistant for biomedical image interpretation	Recommend usage of GPT 4o model from Open AI	Healthcare
[23]	An AI-powered food using YOLOv8 for real-time detection of spoiled food	Encourages usage of Yolo architecture in food-based tasks of CV	Food production and quality assurance.
[24]	Real-time food detection for dietary evaluation, healthcare, and smart kitchen automation	Suggests that YOLOv8 is a well performed option for food detection	Food industry
[25]	Comparing YOLO vs. CNN	Suggests that Yolo architectures can outperform other state of the art options	Food industry
[26]	Automated food recognition and billing in self-service restaurants	Yolo algorithms achieve consistent metrics on food approaches	E-commerce and fashion
[27]	Clothing recognition and recommendation system for e-commerce.	Integrating computer vision with recommendation algorithms enhances online clothes related experiences	Fashion retail
[28]	Clothing recognition and recommendation	Yolo models perform well on clothes applications	Fashion
[29]	Comparison of Selenium and Playwright	Playwright outperforms Selenium in response times	Web scraping

The review of related works highlights key trends in the development of AI-powered virtual assistants and real-time object-detection systems. A predominant approach in recent research is the integration of OpenAI’s LLM services to enhance the intelligence and

responsiveness of virtual assistants. Additionally, Python has emerged as the preferred programming language for backend development, providing flexibility and robust AI and automation capabilities. In the field of object detection, YOLO-based architectures are widely adopted for their efficiency and accuracy in real-time classification of domain-specific resources. These insights underscore the prevailing methodologies shaping current advancements and inform the design choices for our proposed system.

3. Proposed Approach

This paper proposes a virtual assistant called EverydAI. This intelligent assistant is designed to support users in daily decisions across domains such as cooking, fashion, and fitness by leveraging a combination of state-of-the-art technologies. EverydAI integrates components like image recognition, AR, NLP, and recommendation systems to deliver intelligent, context-aware, and personalized suggestions.

Unlike conventional recommendation platforms that typically generate generic or static suggestions, EverydAI adapts its outputs based on the user's immediate context. For instance, through object-detection capabilities, the system identifies available resources such as ingredients, clothing items, or fitness equipment to ensure that its recommendations are feasible and personalized.

User interaction with EverydAI is facilitated through a multimodal conversational interface, supporting both voice and text input. This design allows for natural and seamless communication, enabling users not only to receive recommendations but also to ask clarifying questions, adjust preferences, and provide real-time feedback. This iterative loop fosters continuous learning and adaptation, making the assistant more responsive and accurate over time.

Furthermore, EverydAI aims to enhance user engagement and accessibility by incorporating digital avatars and voice synthesis. By contextualizing recommendations within the user's real-world environment such as suggesting outfits based on wardrobe contents, recommending exercises compatible with available equipment, or proposing meals with on-hand ingredients, EverydAI seeks to solve the decision-making problem. The next sections explain the design and development of the system.

3.1. Requirements Definition

The stage at which the requirements are defined is crucial for the successful structuring and execution of EverydAI. During this phase, specific features and functionalities are established to ensure the system meets the proposed objectives. The key requirements for the virtual assistant are detailed below:

3.1.1. Human–Machine Interaction

1. The assistant must maintain a fluent and coherent conversation to identify the user's needs.
2. The assistant's input should be through the user's voice or written text.
3. The assistant's output should be audible or visible, as appropriate.
4. The system must support English and Spanish as interaction languages.
5. The assistant must have knowledge of the domains for which it is developed.

3.1.2. Object-Detection Capability

1. The system must be capable of real-time object detection.
2. The assistant must identify objects within the following domains: cooking, fashion, and fitness. This aims to acquire contextual information about the user before making a recommendation.

3.1.3. User Identification

1. The system must be able to identify the user it interacts with through prior registration.
2. The assistant must store relevant user information to personalize recommendations.
3. User information must be protected and managed following privacy policies, ensuring the security of personal data.

3.1.4. Multi-Domain Support

1. Each domain represents a field where the user can receive recommendations to make informed decisions based on their requirements.
2. The domains must reflect everyday situations where users commonly face decision-making.
3. The system must extract contextual information from the domain to generate personalized recommendations.

3.1.5. Digital Avatar

1. A digital avatar will be created to personify an assistant for each domain.
2. To provide a more human-like interaction experience, the avatars must be capable of performing animations that complement verbal communication, making gestures and facial expressions aligned with the conversation's content, and maintaining proper synchronization between the avatar's movements and its voice.

3.1.6. Technical and Performance Requirements

1. Interaction with the system must occur in real time; therefore, the assistant's responses should not exceed a latency of 6 s unless executing functions or requiring responses from external services.
2. The system must be web-responsive and adaptable to different screen sizes.
3. The system must allow users to maintain interaction continuity when switching from one device to another.

3.2. Design Specifications Definition

The proposed system is designed to operate across multiple domains where decision-making often becomes burdensome due to information overload or lack of contextual assistance.

Cooking: Helps users by suggesting food recipes based on the ingredients available to the user, giving detailed step by step guidance and also suggesting web tutorials on dishes requested by users.

Fashion: Suggest outfits to the user based on the available clothes and suggest clothing products from the internet that are related to the context of the user.

Fitness: Aids the user by explaining physical exercises based on the gym items available at the moment, giving guides and web tutorials if the user needs it.

The overall flow of interaction encompassing input recognition, context processing, and recommendation generation is visually summarized in Figure 2.

The different interactions with the assistant will take place according to the user's intention within three scenarios, in such a way that a result related to the domains (recipe, outfit, workout routine) is achieved. The scenarios are as follows:

Immediate recommendation: The user is looking for a result to be carried out at that moment, taking into account the resources available in the inventory and those detected in real time.

Late recommendation: The user wants to plan for a future outcome, considering the resources available to them and the missing ones that can be acquired.

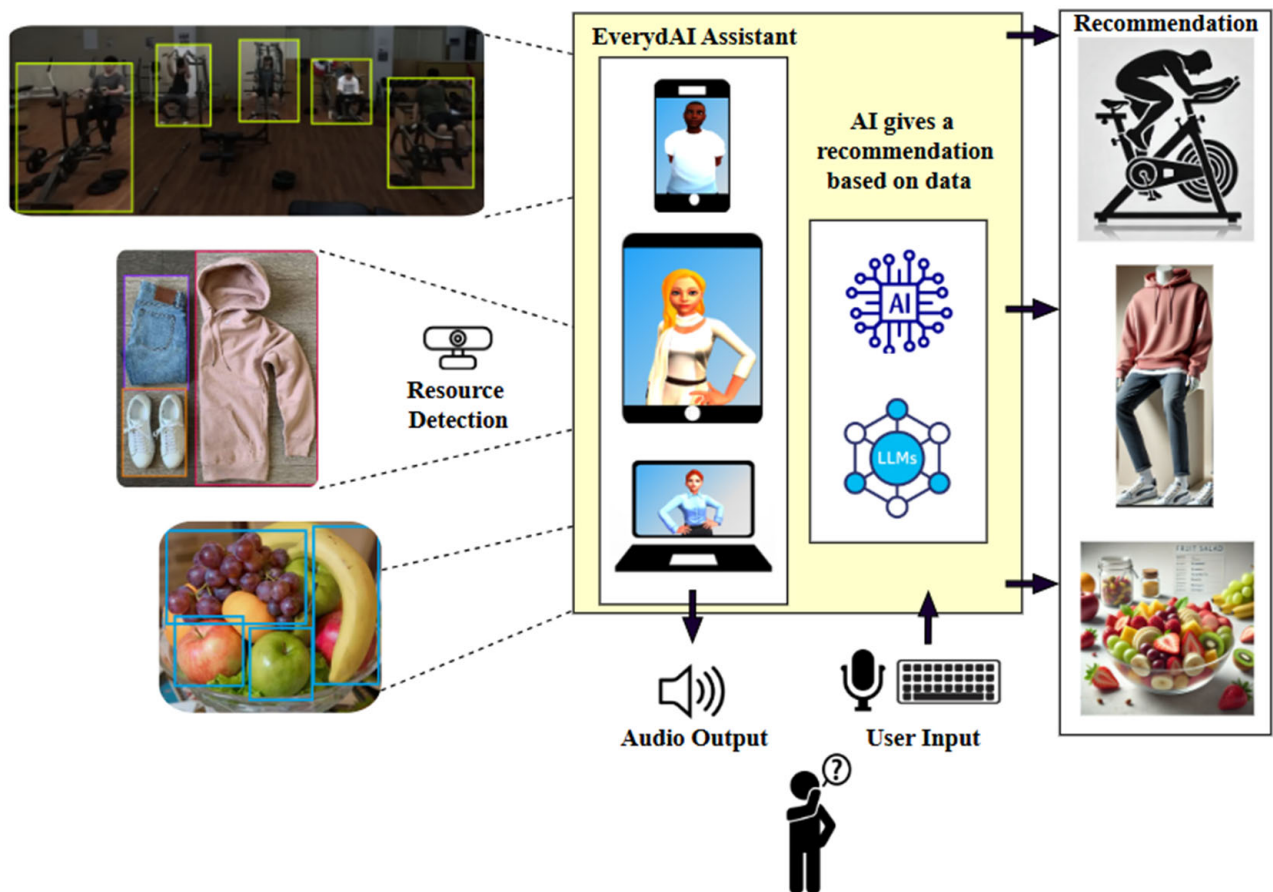


Figure 2. General diagram of EverydAI interaction with users.

Comeback: The user wants to carry out a previously planned result, now taking into account that the missing resources from the previous scenario have been acquired.

These interactions must be addressed as shown in the flow diagram in Figure 3.

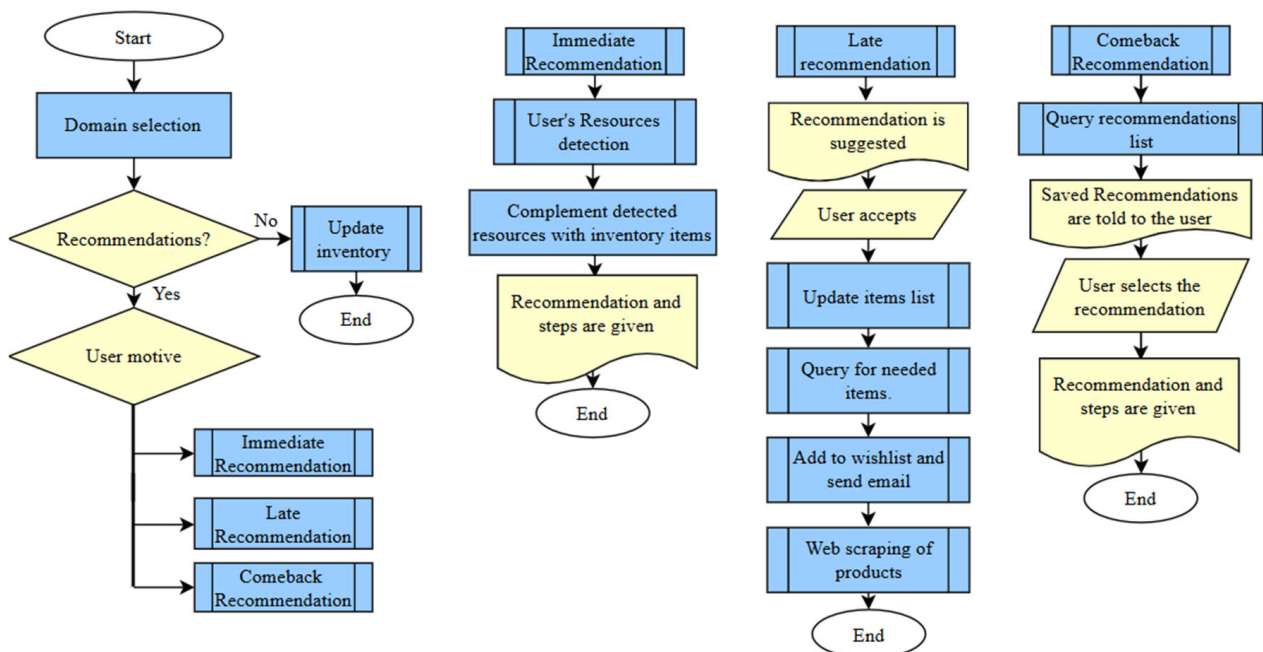


Figure 3. Flow diagram of EverydAI interactions.

3.3. Selection of Technologies

3.3.1. Language Model

The assistant must be able to understand and generate natural language text to communicate and interact with the user. Table 2 presents a comparison of different LLMs based on several criteria relevant to their performance and applicability in various contexts. Some of the technical features considered are explained below:

Table 2. Comparison of different services and platforms offering LLMs.

Tool	Multilingual	Follow Instructions	Open-Source	Function Calling	Related Works	Fluent Interactions [30]
Gemini	X	X		X		64.63%
Open AI	X	X		X	X	65.05%
Mistral	X	X		X		35.95%
Llama	X		X	X		41.73%

Follows Instructions: This refers to whether the model incorporates an assistant-like framework—such as advanced prompting systems or assistant architectures—that enables it to focus on specific tasks or domains. This capability enhances the model’s practical usefulness by allowing it to act, for example, as a programming assistant, educational tutor, or conversational agent within a particular area, consistently following specialized instructions and providing relevant, context-aware responses.

Function Calling: Refers to the model’s ability to interact with APIs and execute functions based on natural language requests.

Fluent Interactions: Refers to the model’s ability to engage in smooth and natural interactions with the user, as measured by their performance on the intelligence index metric [30] at the time of writing this manuscript. This metric is based on eight evaluations, including MMLU-Pro, GPQA Diamond, Humanity’s Last Exam, LiveCodeBench, SciCode, AIME, IFBench, and AA-LCR, which together provide a comprehensive assessment of the model’s performance. The models compared in reference [30] are specifically selected as the closest or most similar representatives from each LLM provider based on this intelligence index ranking, ensuring a fair and relevant comparison among leading large language models.

Although OpenAI models are not free, their low cost and the fact that they meet all other required features make them the ideal option for developing the assistant. In particular, among the models offered by OpenAI, GPT-4o Mini is the one that best fits the requirements. Being multilingual, it can understand and respond in the user’s language. Its ability to handle function calling allows it to detect when a function needs to be executed, enabling the assistant to perform tasks such as web scraping or online searches automatically as needed. Additionally, as it is trained to follow instructions, the model can assume different roles within the domains defined in the proposed approach scope, providing responses and recommendations appropriate to the context.

EverydAI leverages the assistant function of OpenAI’s GPT-4o mini model to bring to life the three specialized assistants within the predefined domains. By utilizing this assistant framework, EverydAI is able to interpret user requirements more effectively and generate responses tailored to specific contexts. This approach allows the system to maintain coherent, domain-focused interactions, ensuring that each assistant operates with expertise relevant to its area. Through structured prompts that incorporate user input, available resources, and contextual data, the assistant function enables EverydAI to deliver precise and context-aware recommendations across its various domains.

3.3.2. Voice-to-Text Conversion

To enable the user to interact with the assistant through voice, it is necessary to recognize what the user says for further processing. Table 3 presents a comparison of different speech-to-text conversion tools.

Table 3. Comparison of platforms offering voice-to-text conversion services.

Tool	Open Source	Multilingual	Requires Audio File	Detects No Speech
Google STT		X	X	X
AZURE Speech recognition		X	X	
Web Speech API	X	X		X
AWS transcribe		X	X	

Requires audio file: Determines whether the tool needs a pre-recorded audio file instead of supporting real-time transcription.

Detects no speech: Indicates whether the tool can identify moments of silence or absence of voice, which improves transcription accuracy and prevents erroneous responses.

The Web Speech API is chosen as the speech-to-text conversion tool due to its free availability and its ability to operate in real time without requiring pre-recorded audio files. Thanks to its capability to detect no speech, it avoids incorrect transcriptions. Additionally, it is a native tool for web environments.

3.3.3. Three-Dimensional Avatar Models

The assistant must be personified through an avatar to enable natural interactions with the user. Table 4 compares various platforms that provide 3D digital avatars.

Table 4. Comparison of platforms providing 3D digital avatar deployment.

Tool	Open Source	Customization	Facial Expressions	Natural Models
Mixamo	X			X
Ready Player Me	X	X	X	X
Free3D	X			
TurboSquid			X	X

Customization: Assesses whether the platform allows avatars to be modified, either in appearance or clothing.

Facial expressions: Determines whether avatars can display gestures and expressions.

Ready Player Me is selected as the tool for generating 3D digital avatars due to its customization capabilities and facial expressiveness as essential features for achieving more natural interactions with the assistant.

Each domain is represented by a distinctive 3D avatar deployed using Ready Player Me and managed with Blender. These avatars are designed to enhance natural interactions through animations relevant to their domain, and these animations are implemented with Mixamo. For instance, the fitness assistant initiates interactions by stretching his arms. Additionally, all avatars feature poses, facial expressions, blinking, and lip-syncing synchronized with their speech for a more natural interaction. Each one of the avatars used in EverydAI are shown in Figure 4, and the entire process of deploying a digital 3D avatar is shown in Figure 5.



Figure 4. EverydAI assistant avatars.

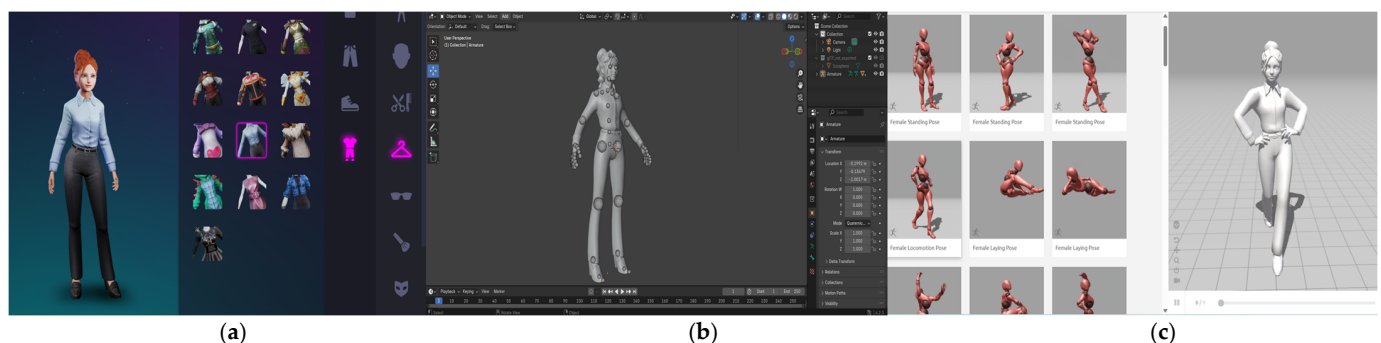


Figure 5. Deploying an avatar process. (a) Personalizing the avatar using Ready Player Me. (b) Processing the 3D file in Blender. (c) Implementing animations in Mixamo.

3.3.4. Text-to-Voice Conversion

To enable the avatar to express itself naturally, it must be equipped with a voice, allowing the assistant to communicate verbally. Table 5 presents a comparison of various text-to-speech (TTS) platforms based on their features and suitability for integration into the system. For text-to-speech conversion in EverydAI, ElevenLabs is chosen due to its outstanding voice quality, which enables the generation of more natural and expressive responses. Additionally, its automatic language detection enhances adaptability to different linguistic contexts without the need for additional configurations. It also offers a wide variety of voices, allowing for the customization of the assistant to suit different scenarios. Voice synthesis is powered by the Eleven Multilingual v2 model.

Table 5. Comparison of platforms offering text-to-voice conversion services.

Tool	Voice Quality	Automatic Language Detection	Variety of Voices	Multilingual
ElevenLabs	X	X	X	X
IBM Watson TTS	X		X	X
Google Cloud TTS	X		X	X

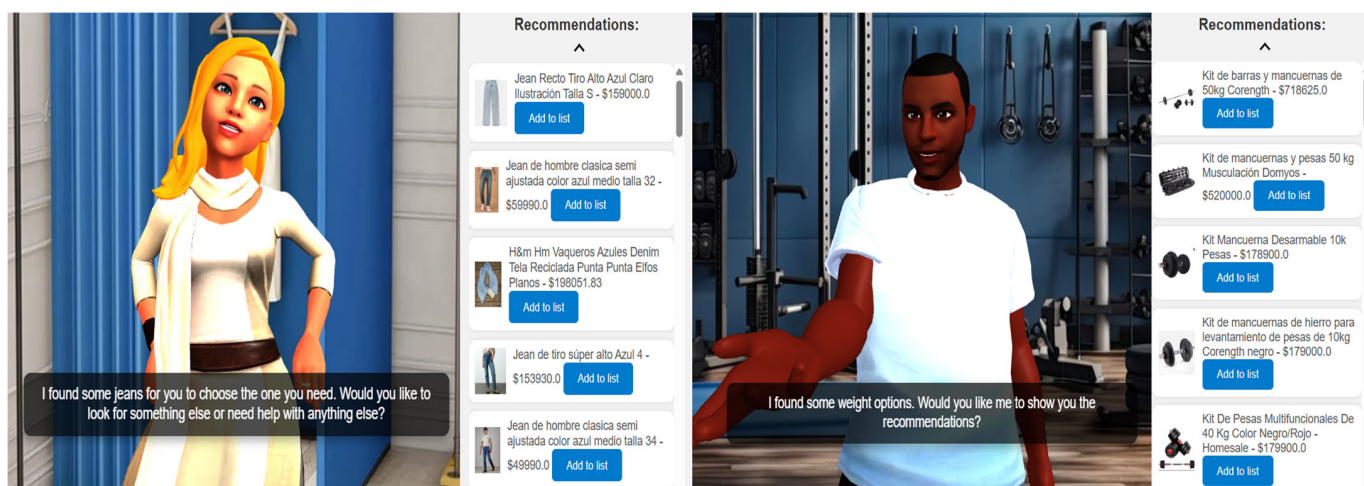
3.3.5. Web Scraping Technology

To extract relevant product information across different domains, the assistant must utilize web-data-extraction technologies. Table 6 presents a comparison of various tools of this kind. For data extraction in EverydAI, the Google Shopping API is selected due to its ability to retrieve product information directly, without the need to load or execute website content. Unlike tools such as Selenium or Playwright, which require processing and waiting for dynamically rendered pages, this API enables faster access to data.

Table 6. Comparison of technologies for web scraping.

Technologies	Dynamic Pages	Specific Sites	Speed
Request		X	
Selenium	X	X	
Playwright	X	X	
SERPAPI Google Shopping API	X		X

In addition to data extraction, EverydAI can provide web-based suggestions, such as tutorials or videos, upon user request. This functionality is powered by the SERPAPI search engine API, which supports voice-based searches for specific tutorials related to the user’s domain of interest. By combining NLP with intelligent search capabilities, EverydAI ensures comprehensive and personalized decision support. The scraping process is triggered through OpenAI’s function-calling mechanism. When a user requests information about specific items, EverydAI generates a structured query and activates the corresponding web-scraping function. This allows the system to fetch and present real-time data on products. Examples of this feature are shown in Figure 6.

**Figure 6.** An example of the web-scraping results.

3.3.6. Computer-Vision Platform—Detection of User Resources

EverydAI utilizes two (2) methods to identify user resources, ensuring flexibility and adaptability to different user needs.

Image-Based Detection: Users can upload an image, which is processed through the Gemini-1.5-Flash model via the Gemini API. The model identifies each resource in the image and sends a structured list to the backend for further processing.

Real-Time Object Detection: Integrated into EverydAI’s augmented experience, this method allows users to point their cameras at resources in real time, detecting objects and surrounding them with bounding boxes.

YOLO is selected as the object-detection architecture due to its support in the state of the art, where it stands out for its efficiency and accuracy in real-time computer-vision tasks. Initially, custom models are trained for different domains using transfer learning; however, the results obtained are not consistent in either practical tests or performance metrics.

Given these limitations, integrating well-performing pre-trained models from a computer vision (CV) platform is necessary. Table 7 presents a comparison between some popular CV platforms. The Roboflow platform is selected, which offers YOLO models previously trained by other users and optimized for various applications through APIs.

These models already perform robustly and present better metrics than those obtained with the custom training. Additionally, these APIs work in such a way that they can be accessed directly from the web page's frontend, where images are collected in real time.

Table 7. Comparison of platforms providing CV model deployment.

Platform	YOLO	Open Source	Remote Inferences
Roboflow	X	X	X
MakeML	X		
Google AutoML			X
AWS Recognition			X
CVAT	X		

This allows the information to be processed without first being sent to the backend, reducing response time. In conclusion, the technology adopted for object detection is Roboflow API + YOLO.

Each domain within the system employs a specific YOLO model trained on a dataset of relevant items and resources. Table 8 provides a comparison of different trained model APIs offered by Roboflow, evaluating key metrics and the number of classes to determine the most suitable model for each use case.

Table 8. Comparison of models provided by the Roboflow platform.

	APIs	Classes	mAP	Precision	Recall
Cooking	[31]	35	87.1%	83.4%	79.5%
	[32]	31	97.5%	94.4%	94.5%
Fashion	[33]	8	69.4%	53.7%	75.8%
	[34]	8	81.1%	78.8%	74.9%
	[35]	8	81.9%	73.0%	79.5%
Fitness	[36]	29	79.2%	85.1%	70.0%
	[37]	23	94.8%	91.8%	91.1%
	[38]	21	87.0%	82.3%	81.2%

The flow diagram of the user resource detection task is shown in Figure 7 and some examples of real-time resource detection using these models are illustrated in Figure 8.

3.3.7. User Inventory and Historical Information

Based on the functionality of detecting user resources, three additional concepts have been developed for each domain: Inventory, which represents the list of resources already known by the assistant and possessed by the user prior to the interaction; Wishlist, which groups the missing items the user needs to achieve a planned outcome; and Recommendations List, which contains the recommendations planned by the user to revisit in the future once the necessary items have been acquired.

The proposed system to address these functionalities utilizes a MySQL database to provide each user with a personalized inventory, where every detected item is stored and managed. This inventory enables the system to track previously recognized objects, allowing for context-aware recommendations. By maintaining a historical record of detected items, EverydAI enhances decision-making by considering past user interactions and preferences.

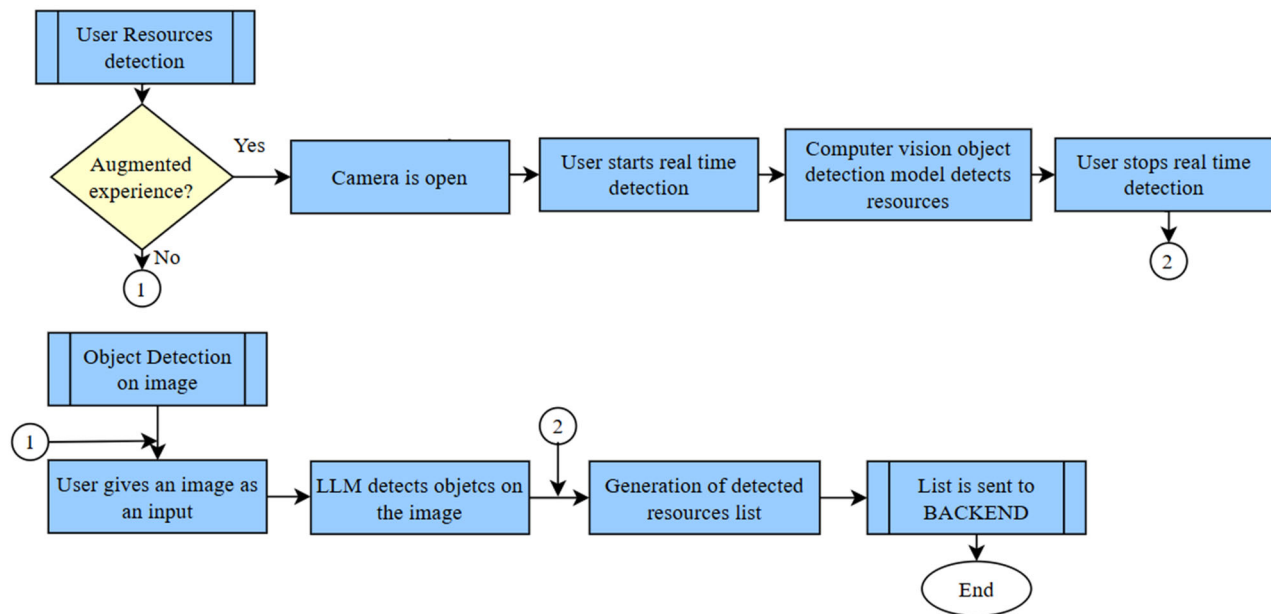


Figure 7. Flow diagram of the object-detection process.



Figure 8. Examples of resource detection.

The database includes tables for user identification, key conversation features for memory retention, a wishlist, and a user inventory for each domain. The backend interacts with this database using the SQLAlchemy framework, ensuring efficient data management and retrieval. Additionally, elements from the inventory are temporarily stored in an in-memory database implemented with Redis, allowing for faster access and reducing the load on the main database during real-time interactions. The database design diagram is shown in Figure 9.

3.3.8. Image Generation

EverydAI incorporates image-generation functionalities through the Runway API to produce visualizations of its recommendations. Based on the recipes or outfits suggested by EverydAI taking into account the user's inventory and contextual preferences, the system formulates descriptive prompts that are sent to Runway.

The resulting images allow users to clearly visualize what the final outcome should look like, such as a completed dish or a fully assembled outfit. This enhances user engagement and decision confidence by translating abstract suggestions into concrete visual references.

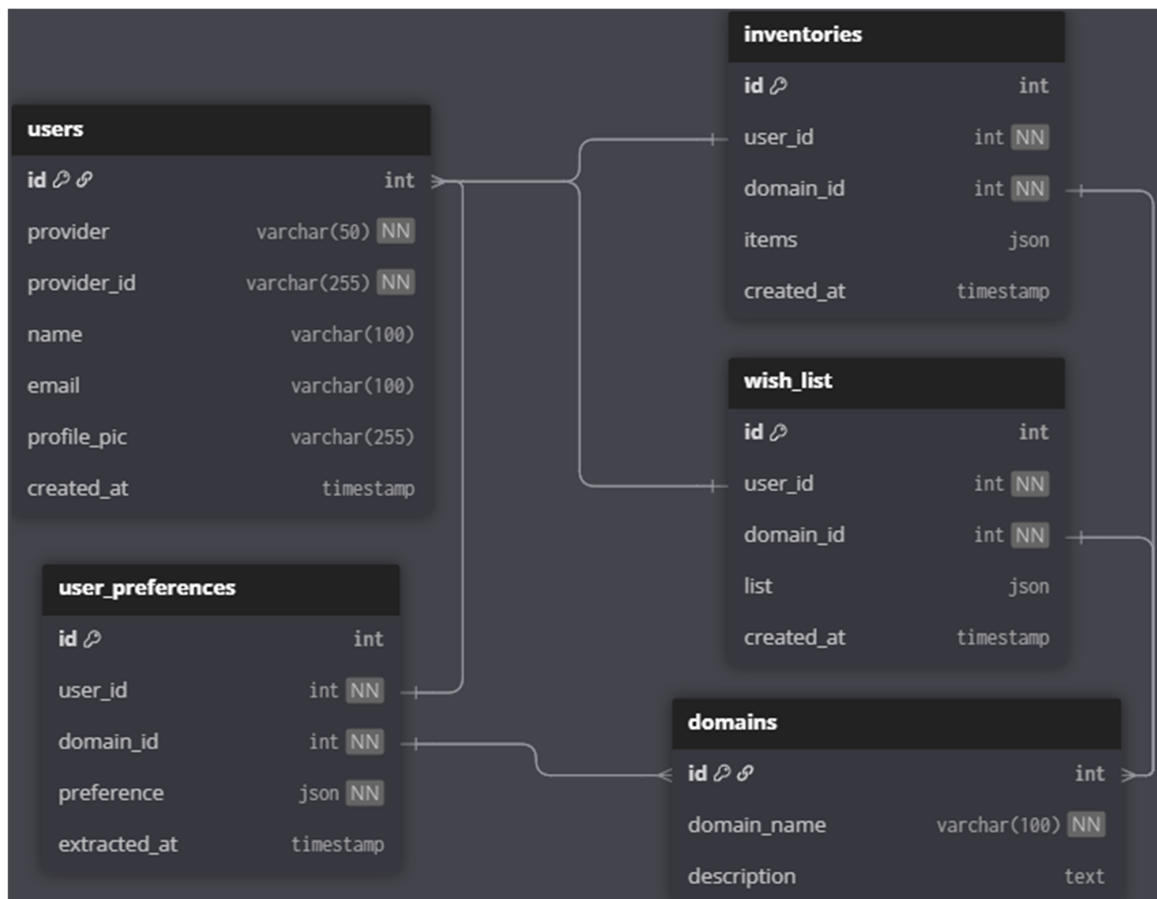


Figure 9. Database structure diagram.

Importantly, before generating these images, users provided or showed EverydAI the contextual resources they had at hand—such as a list of ingredients available in their kitchen or the clothing items in their wardrobe. This contextual information is crucial for the assistant to offer tailored and realistic recommendations aligned with what users could actually execute. In addition, EverydAI is also aware of the user’s broader goal or scenario, such as preparing for an elegant dinner or a themed party. By combining knowledge of both the available resources and the intended purpose, the system is able to generate highly contextualized visual suggestions that match the user’s situation.

Some examples of images of recipes and outfits generated based on the user’s context are shown in Figure 10.

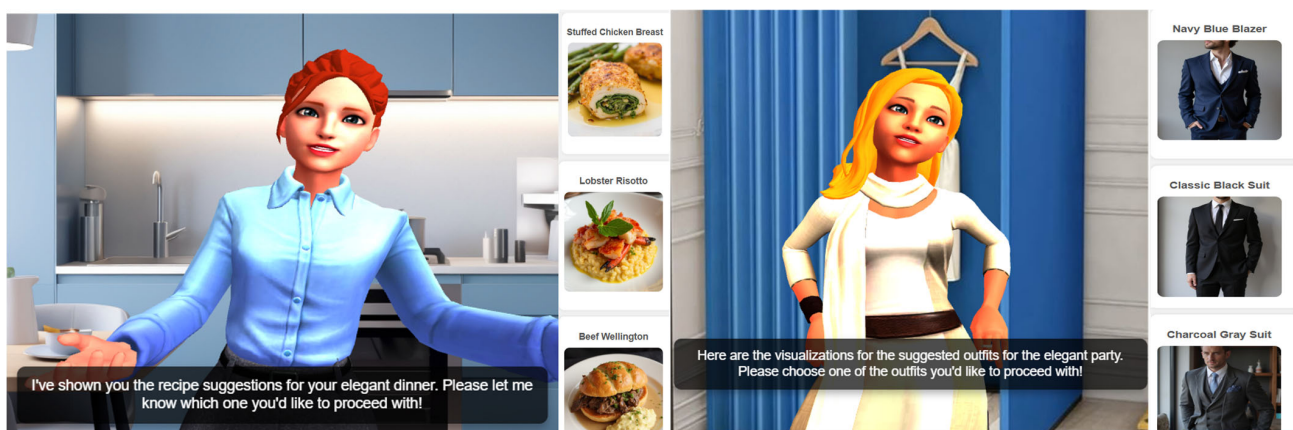


Figure 10. Generation and visualization of recipes and outfits.

Figures 11–13 show a series of interactions with EverydAI once the system is fully implemented with all integrated technologies.

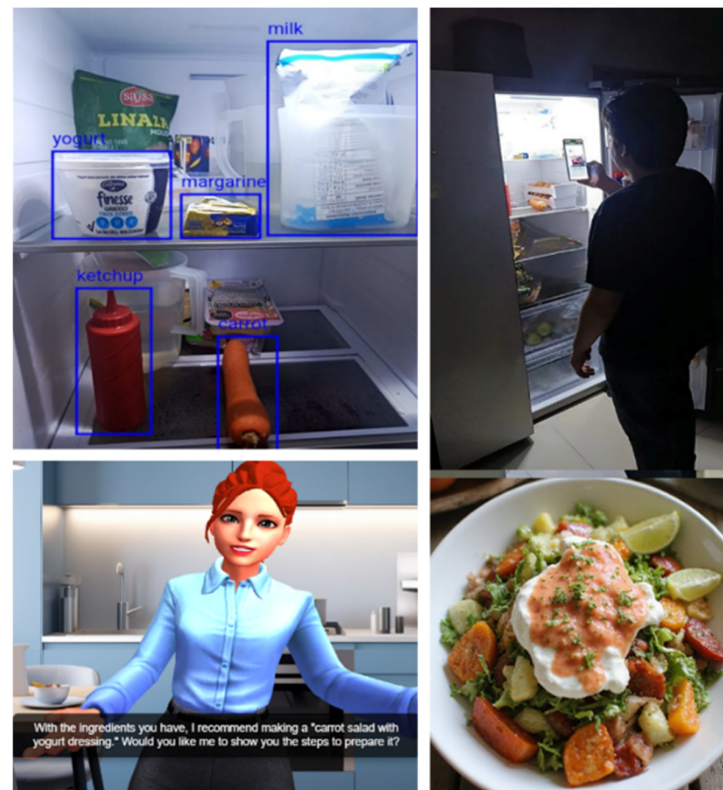


Figure 11. Interaction with cooking assistant.

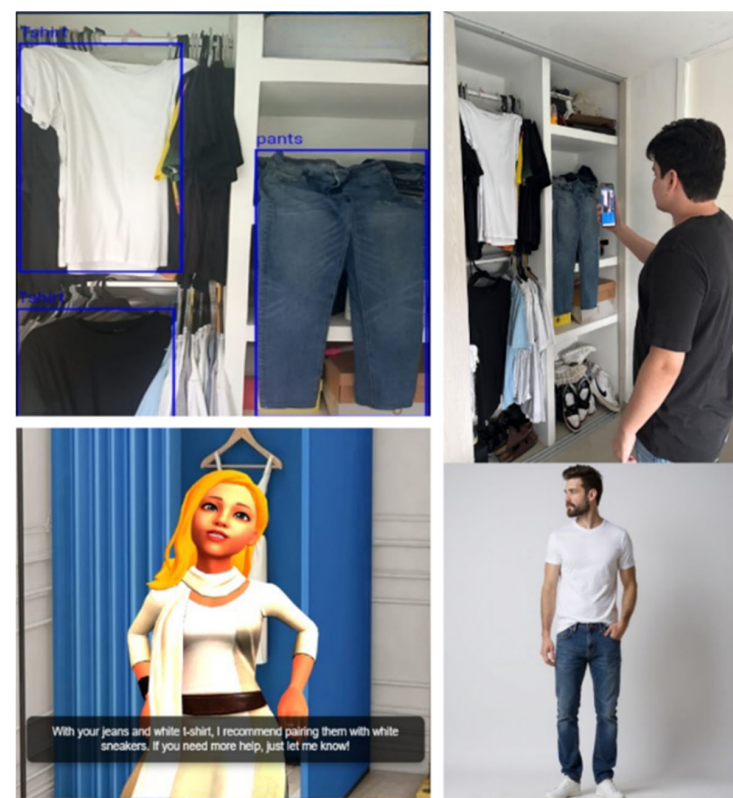


Figure 12. Interaction with fashion assistant.

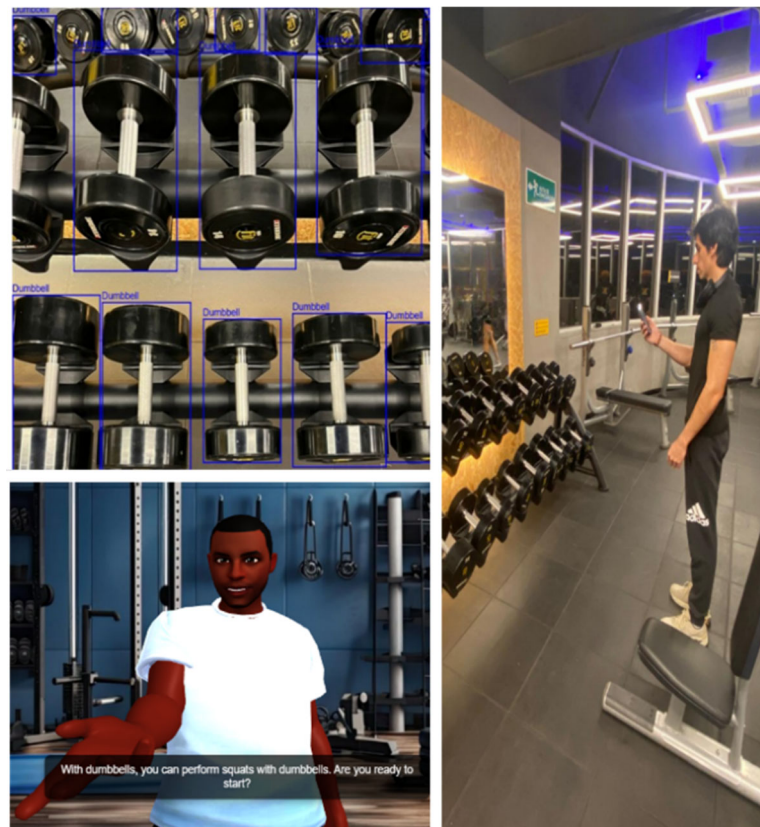


Figure 13. Interaction with fitness assistant.

3.3.9. Web Application Features

EverydAI is deployed as a cloud-hosted, responsive web application on Amazon Web Services (AWS). The backend architecture is implemented using Flask, with the Python microframework selected for its compatibility with AI workflows. The frontend is developed using HTML, CSS, and JavaScript, incorporating the Three.js library to support interactive 3D rendering, particularly for real-time avatar visualization. User authentication and session management are handled through OAuth 2.0, integrated with Google Cloud Identity Services.

3.3.10. Limitations and Considerations

The developed prototype relies on multiple commercial tools, including GPT-4o, ElevenLabs, Ready Player Me, and Roboflow. While these technologies provide advanced capabilities that facilitate integration and rapid deployment, their use entails certain challenges:

Long-term maintenance costs: The costs associated with maintaining the tool over time are primarily tied to three key components: the OpenAI model, the text-to-speech service, and cloud hosting. As the number of users grows, the demand for tokens—both for text generation and for converting that text to speech—will proportionally increase, leading to higher expenses in these services. Additionally, the growth in user traffic will require the implementation of scaling techniques on AWS and the deployment of more powerful virtual machines capable of supporting higher concurrency of connections and simultaneous requests, thereby ensuring the system’s stability and performance.

Long-term sustainability: Dependence on third-party services exposes the system to potential changes in models, APIs, or access policies that could impact its operation or availability. However, these changes can be addressed thanks to the fact that our development is deployed in the cloud and managed through a CI/CD system with a

GitHub (version 2.46.0) Actions pipeline. In the event that modifications in the behavior of certain functionalities are detected, or in an extreme scenario, a global outage of a service occurs, the system can be updated by replacing one technology with another from those considered in the Technology Selection section.

Data privacy and security: The assistant processes multimodal data—including text, voice, images, and contextual information—which presents significant challenges in protecting user information. To address these concerns, the system collects and manages data under strict privacy guidelines. Specifically, the system detects items in the user’s environment to build a personalized inventory that informs EverydAI’s context-aware recommendations. Additionally, user profile data, including names, are obtained securely via Google OAuth during registration and login. This allows the assistant to personalize interactions by addressing users by their names. All user authentication and authorization processes utilize the Google OAuth 2.0 protocol, ensuring secure management of user credentials. Session data is maintained securely within the Flask application, while contextual inventory and user preferences are stored in a protected MySQL database and cached via Redis for optimized performance. Importantly, user data handling complies with Google Cloud’s privacy and security policies, as described in the Web Application Features section. Sensitive data is encrypted upon rest and access is tightly controlled. No personal information is shared beyond the scope of the service’s operational needs. The specific data requirements for each module are:

- **AWS (RDS with MySQL handled with SQLAlchemy and Redis):** Used solely as the hosting server for the web application via Python Flask implementation. User data, including names obtained via Google OAuth2 and any resources provided by users, are stored securely in the RDS database. No user data is shared outside this server beyond what is necessary for the application to function.
- **OpenAI:** Receives the user’s first name (to personalize responses) and a list of resources detected in real time by the detection systems (Roboflow, YOLO APIs, Gemini).
- **Real-time detection systems (Roboflow, YOLO APIs, Gemini):** Receive only images captured by the user to detect the resources they have. No other personal information is sent.
- **Avatar and animation services (Mixamo, Ready Player Me, Blender, Three.js):** No user data is exchanged; these technologies are used solely for rendering avatars and animations.
- **Runware API:** Receives only the text response generated by OpenAI to create an image; no user data is transmitted.
- **SerpApi:** Receives a list of items that the user lacks in their inventory but that appear in the AI recommendation—for example, missing ingredients—to search for relevant information online.
- **ElevenLabs:** Receives the AI-generated response to convert it into audio. The only user-related data involved are the name and the resources they possess.
- **Web Speech API:** Receives only the audio spoken by the user to convert it to text, which is then sent to OpenAI as part of the conversation. The only personal data it may access is what the user verbally provides.

Validation Scope: The study’s validation is carried out with participants affiliated with Universidad del Norte, including mostly young students from different semesters and academic programs, as well as university staff and personnel in other roles. All participants agreed to take part in the validation knowing that one of EverydAI’s main features involves the integration of multiple technologies provided by external agents, and that using the system requires providing images, audio, and text, as well as signing in. For minors, additional verbal consent is obtained from both the participants and their parents. The

validation process relies on questions adapted from previous state-of-the-art studies on virtual assistants, specifically [39–41], with a primary focus on assessing user satisfaction with EverydAI.

EverydAI is a system that integrates multiple technologies to deliver a seamless, intelligent, and personalized user experience. This approach brings together backend and frontend tools, voice recognition, NLP, object detection, and 3D avatar models. Each component plays a specific role, and when properly implemented and interconnected, they enable the virtual assistant to provide real-time recommendations; this integration is illustrated in Figure 14 and the flow of data is illustrated in Figure 15.

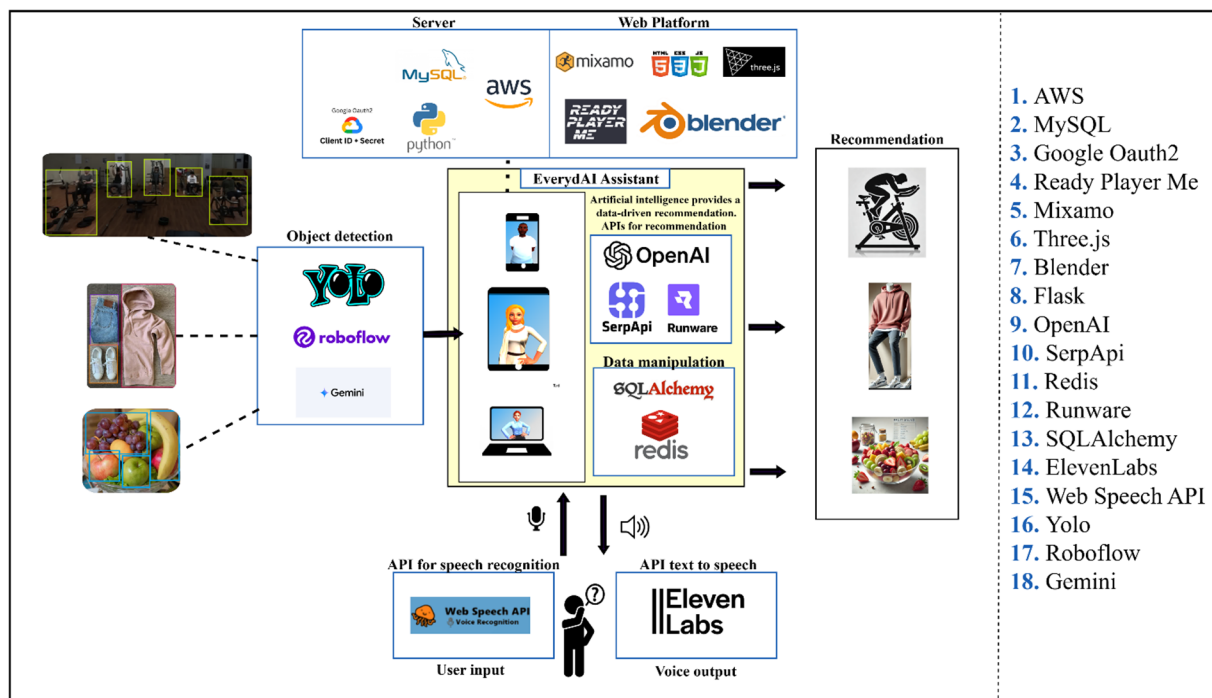


Figure 14. A diagram of the integration of technologies.

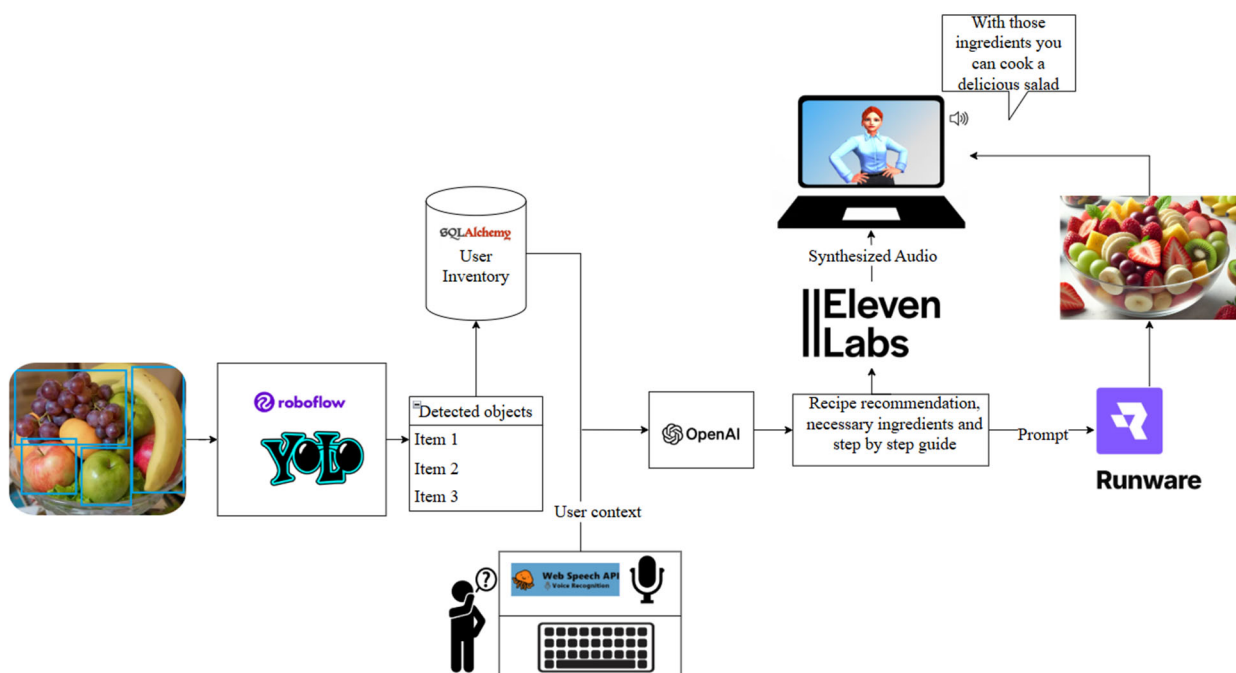


Figure 15. General flow of data in an interaction with EverydAI.

4. Experimental Results

4.1. User Based Validation

To analyze the system's performance based on user validation, several experiments are conducted in which a diverse group of people representing various age groups and backgrounds interact with EverydAI and complete a survey. Each participant is assigned to a specific domain of use, and is given a general explanation of the assistant's purpose and main features of the system. This setup aims to observe how users naturally adapt to the tool over a period of 30 min.

The survey included questions similar to those used in state-of-the-art developments, focusing on participants' demographic characteristics, the effectiveness of the system's core functionalities during the experiment, and how well the system supported users in addressing the targeted problem [39–41]. These questions are adapted to fit our specific approach, ensuring their relevance and applicability to the context of EverydAI. The initial stage, referred to as Phase 1, provided valuable feedback to identify and address the system's most critical issues. After implementing the necessary improvements, a second group of participants is invited to use the updated version of EverydAI and complete the survey again. Phase 2 allowed us to obtain more accurate performance metrics and evaluate the effectiveness of the enhancements.

4.1.1. Survey Structure

Thirty individuals participated in each phase, with usage distributed evenly across each domain. Figure 16 shows some participants interacting with EverydAI. The survey is structured as follows:

1. First, the survey asks about the demographic characteristics of the participants, such as the age group.
2. Frequency of use of virtual assistants (daily, weekly, monthly, rarely).
3. User satisfaction. The survey included questions evaluating user satisfaction, such as:
 1. Whether the assistant met their needs.
 2. The ease of interaction with the assistant, rated on a scale from 1 (very difficult) to 5 (very easy).
 3. Willingness to recommend EverydAI to others.
4. Evaluation of functionalities. Participants rated aspects of EverydAI performance on a Likert scale from 1 (strongly disagree) to 7 (strongly agree) based on the following questions:
 1. The voices used by the avatars are suitable for virtual assistants.
 2. EverydAI gives appropriate responses.
 3. When talking to EverydAI, I was able to express everything I wanted to say.
 4. EverydAI seems to understand what I say.
 5. Conversations with EverydAI resemble conversations with real people.
 6. EverydAI detects the resources (ingredients, clothing items, gym equipment) that I show it.
 7. Interactions with EverydAI resemble those with a real person.
5. Task efficiency. Users evaluated the impact of EverydAI on their task efficiency using a 5-point Likert scale for the following statements.
 1. EverydAI is capable of assisting me in my everyday decision-making.
 2. The recommendations I receive from EverydAI surpass those I would normally come up with on my own.
 3. Using EverydAI improves the efficiency of my daily routine.

6. Two open-ended feedback questions aimed at gathering participants' opinions on the most outstanding features of EverydAI and identifying areas where the system can be improved.



Figure 16. Participants interacting with EverydAI.

4.1.2. Demographic Analysis

To begin the analysis, we examined the demographic profiles of the participants from both Phase 1 and Phase 2. Understanding the age and background with the topic distributions of respondents provides context for interpreting the feedback and recommendations gathered in each phase. Both surveys include 30 different participants per phase, with 10 individuals assigned to each domain. Participants come from a variety of age groups and have diverse levels of exposure to technology, as is shown in Figure 17, as a consequence of having people of different backgrounds participating—from students of Universidad del Norte from different academic programs—ensuring a diverse sample that enhances the generalizability of the results. Figure 17 illustrates this demographic diversity; highlighting the representation achieved in both phases of the experiment, it shows that most participants in both phases are aged 18–24, making up the majority of the sample. This age group is slightly more represented in Phase 1 than in Phase 2. The youngest group (under 18) showed a small increase in Phase 2, while participation from the 25–34 and 35+ age ranges remained low and relatively stable. These results suggest that younger participants are the primary users interacting with EverydAI across both phases.

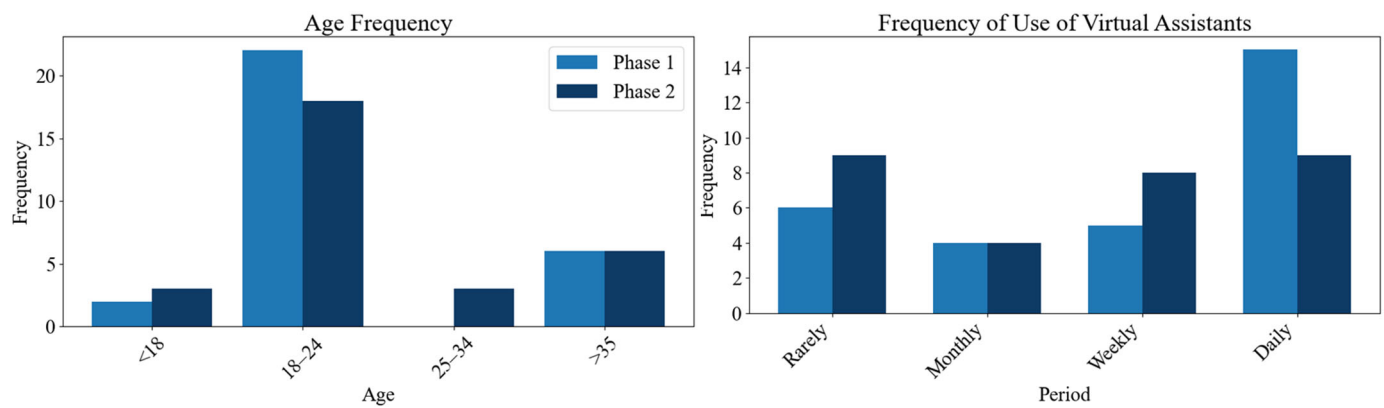


Figure 17. Demographic characteristics results.

Additionally, the figure shows that virtual assistant usage is higher in Phase 1 compared to Phase 2. This indicates that participants in Phase 1 are generally more familiar with this type of technology. Consequently, their feedback likely had greater potential to inform and guide improvements to the system in preparation for Phase 2.

4.1.3. User Satisfaction Results

This section aimed to measure the overall user satisfaction of interactions with EverydAI. In this regard, the results show an improvement in Phase 2 compared to Phase 1. Initially, in both phases, 100% of participants reported that EverydAI met their needs. Secondly, as shown in Figure 18, all participants in Phase 2 indicated they would recommend EverydAI to others, whereas in Phase 1, a couple of individuals responded negatively.

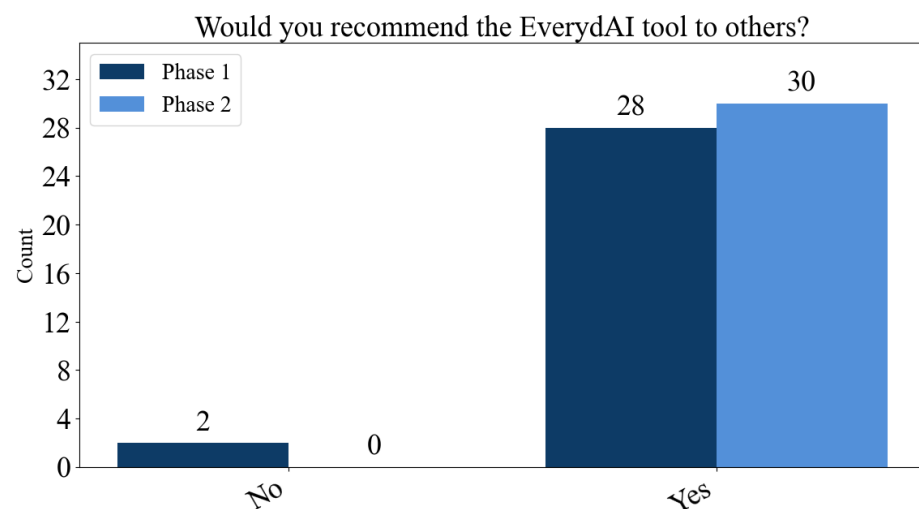


Figure 18. A comparison of the answers given regarding willingness to recommend.

Finally, regarding the ease of interaction and satisfaction with the assistant, EverydAI achieved an average score of 4.53 in Phase 2, compared to 4.37 in Phase 1, on a scale from 1 to 5.

4.1.4. Functionalities Evaluation Results

This section aimed to measure how EverydAI main features performed during participants' interactions. Figures 19 and 20 show the results of this section of the survey for Phase 1 and Phase 2, respectively.

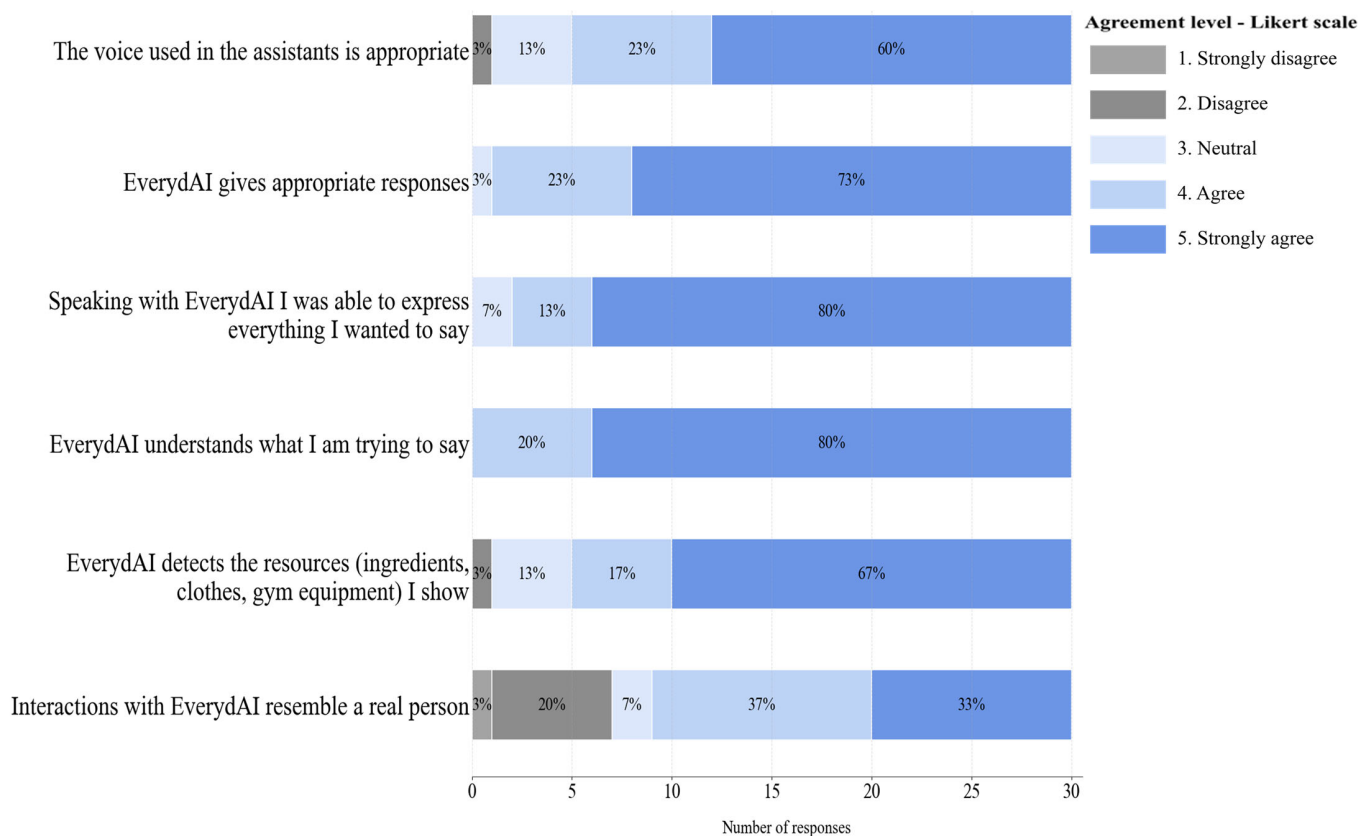


Figure 19. A summary of users' agreement with statements related to technical components for Phase 1.

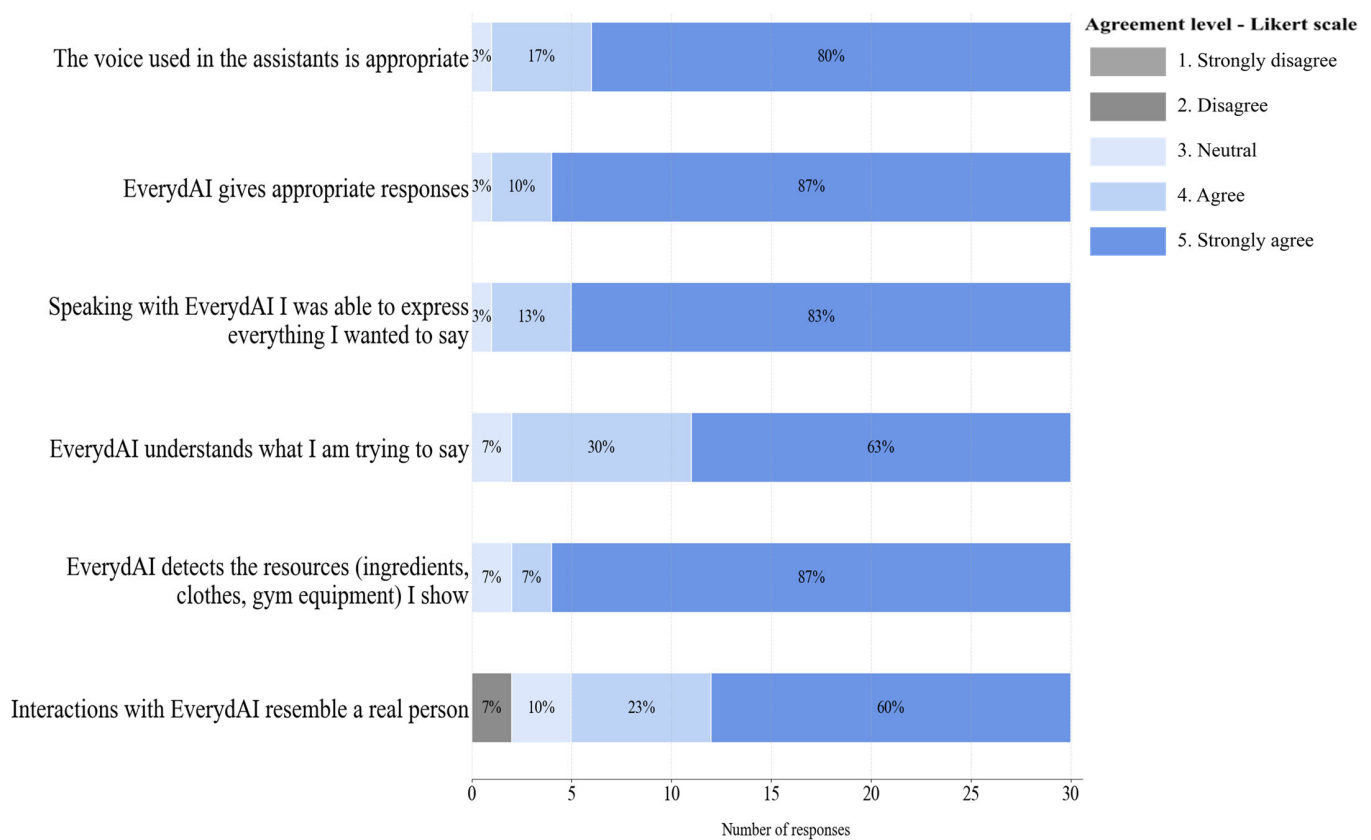


Figure 20. A summary of users' agreement with statements related to technical components for Phase 2.

Comparing the results of Figures 19 and 20, we can observe an overall improvement in agreement with the statements. For instance, agreement with the statement “EverydAI gives appropriate responses” increased from 73% to 87%, likely due to the prompt refinement carried out between Phases 1 and 2. Another example includes the statements “EverydAI detects the resources I show” and “Interactions with EverydAI resemble a real person interaction,” which also saw increased agreement, most likely as a result of frontend changes that made the interface more intuitive. In Phase 1, some participants were unable to show any resources to EverydAI or talk with it, simply because they did not realize they could; they did not see the buttons that allowed them to perform those actions in the first place.

4.1.5. Task Efficiency Evaluation Results

This section aimed to measure participants’ opinions on EverydAI’s usefulness in solving their daily tasks, since that is its primary goal. Figures 21 and 22 summarize their responses in Phases 1 and 2, respectively.

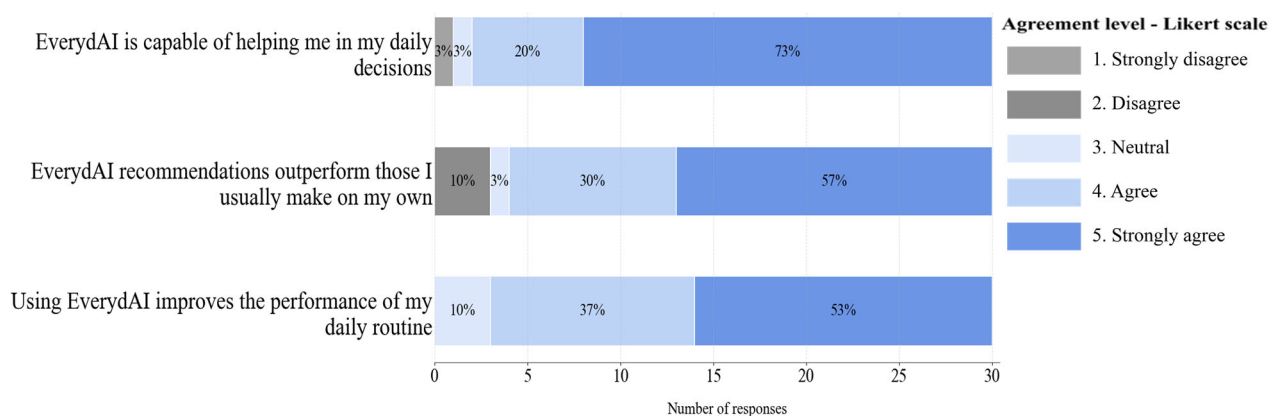


Figure 21. A summary of users’ agreement with statements related to task efficiency for Phase 1.

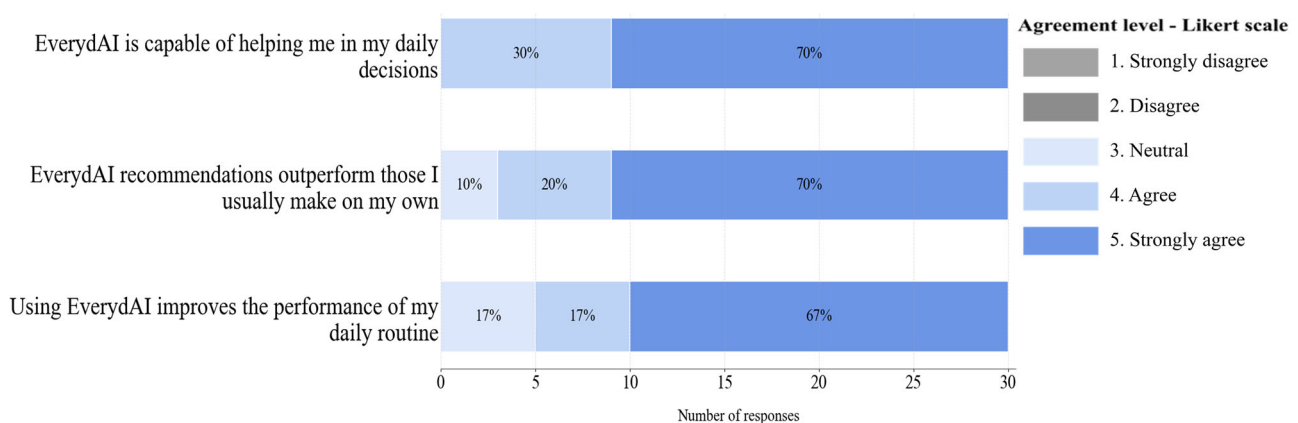


Figure 22. A summary of users’ agreement with statements related to task efficiency for Phase 2.

Figures 21 and 22 demonstrate a clear improvement in user perception of EverydAI’s task efficiency between Phase 1 and Phase 2. For example, in Phase 1, 73% of users “strongly agreed” that EverydAI helps in daily decisions, and while this slightly decreased to 70% in Phase 2, all disagreement disappeared, indicating a consolidation of positive sentiment. More notably, for the statement about EverydAI’s recommendations outperforming users’ own, 10% of responses in Phase 1 are “strongly disagree”, but in Phase 2, 70% of users “strongly agreed” with no negative responses at all. These examples show

that the enhancements carried out in between phases are critical for the improvement of EverydAI interactions.

As a conclusion, the comparison of user satisfaction across both phases of the process highlights a clear improvement in the system's performance following the implemented enhancements. In Phase 1, an average of 88% of users either agreed or strongly agreed with positive statements about EverydAI, while in Phase 2 this average increased to 92.7%. These percentages are calculated by averaging the combined 'agree' and 'strongly agree' responses across all Likert-scale statements. This upward trend reflects not only a consistently high level of user satisfaction but also confirms the positive impact of the enhancements introduced between phases.

4.1.6. Statistical Significance Analysis Between Phases

An initial descriptive comparison between Phases 1 and 2 shows an increase in the proportion of positive responses (those marked as agree or strongly agree), rising from 88% in Phase 1 to 92.7% in Phase 2. This difference suggests an improvement in participants' perceptions regarding the evaluated items.

However, while frequency analysis provides an intuitive overview of the results, it does not determine whether the observed difference is statistically significant. To address this, the descriptive approach is complemented with a Welch's *t*-test conducted using the SciPy library in Python.

To better understand the nature of each participant's responses, Figure 23 shows the distributions of the average Likert score calculated for each participant in both phases. Each Likert response is converted to a numerical scale from 1 (strongly disagree) to 5 (strongly agree). The graphs also include key descriptive statistics such as the mean ($\bar{x}$), standard deviation (std), and skewness (sk) for each group.

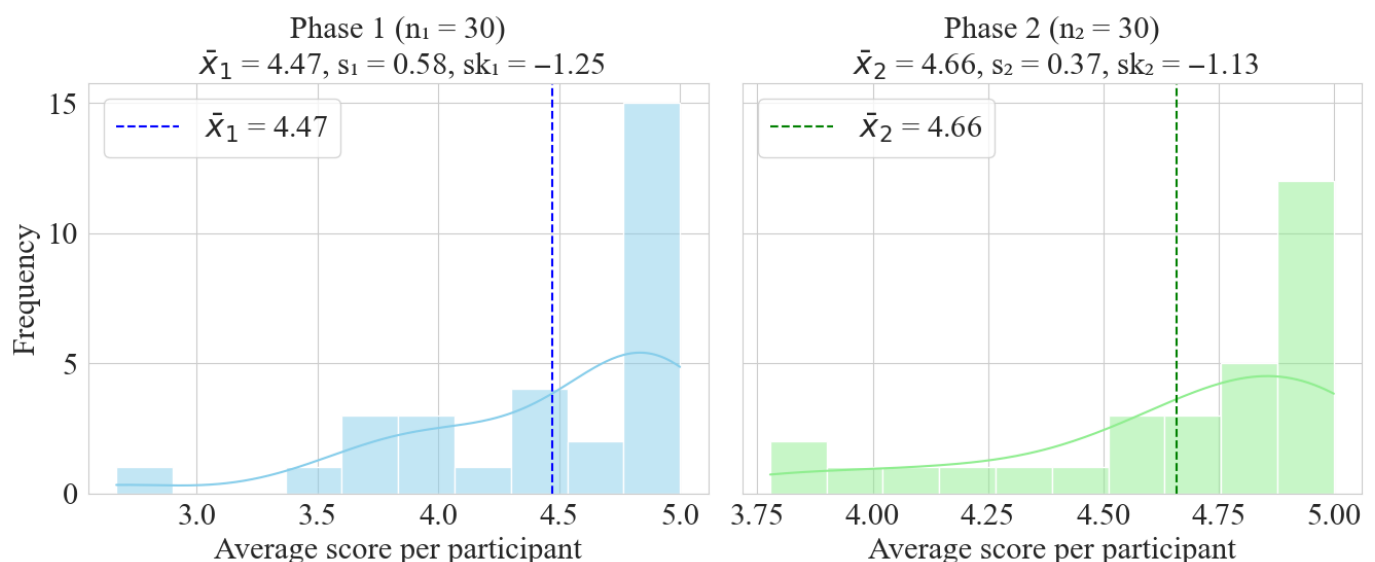


Figure 23. Distribution and statistical parameters of the participants' scores across both phases.

Since different groups of users participated in each phase and we cannot assume equal populational variances, Welch's *t*-test is appropriate [42]. Although the raw data distributions are not symmetric, as indicated by their skewness values of -1.73 and -1.83 for phases 1 and 2, respectively, the Central Limit Theorem (CLT) states that the distribution of the sample mean approaches a normal distribution as the sample size increases, regardless of the original data distribution. This condition is generally considered satisfied when each group has more than 30 or 50 observations, depending on the reference source. In our case, both phases include 30 observations (one per participant). Therefore, we can reasonably

assume approximate normality of the sampling distribution of the mean, and under these conditions, Welch's t -test remains valid.

The null hypothesis H_0 states that there is no difference between the populational means of the two phases, while the alternative hypothesis H_1 proposes that a difference does exist:

$$\begin{cases} H_0 : \mu_1 = \mu_2 \\ H_1 : \mu_1 \neq \mu_2 \end{cases} \quad (1)$$

The test statistic is calculated using the Welch's t -test formula:

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \quad (2)$$

where

- $\bar{x}_1$ and $\bar{x}_2$ are the sample means;
- s_1^2 and s_2^2 are the sample variances;
- n_1 and n_2 are the sample sizes.

The adjusted degrees of freedom are calculated using the Welch–Satterthwaite equation:

$$v = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{\left(\frac{s_1^2}{n_1}\right)^2}{n_1-1} + \frac{\left(\frac{s_2^2}{n_2}\right)^2}{n_2-1}} \quad (3)$$

The results obtained are as follows:

- $t = -1.5042$;
- $p = 0.1390$.

Although the statistical test does not yield a significant difference between the means of the two phases at the 5% level ($p = 0.1390$), the negative value of the t -statistic ($t = -1.5042$) suggests that the mean in Phase 2 is higher. However, this difference is relatively small, most likely because the results in Phase 1 are already notably high, reflecting the strong positive reception EverydAI received from the outset. With such elevated initial performance, the potential for further improvement is naturally limited, yet the data still indicate a positive trend in Phase 2. This upward trend is attributed to improvements in prompt design that better guided users toward the recommendations generated during image generation and suggestion interactions, as well as enhancements to the front-end that elevated the overall user experience. These improvements are described in greater detail in the User Feedback section.

To complement the t -test results, the effect size is calculated using Cohen's d , which is defined as:

$$d = \frac{\bar{x}_1 - \bar{x}_2}{s_p} \quad (4)$$

where

$$s_p = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}} \quad (5)$$

The resulting Cohen's d of 0.39 corresponds to a small-to-medium effect size [43], indicating that, despite the difference in means between the two phases not reaching statistical significance at the 5% level, the magnitude of the change remains practically relevant. This supports the view that Phase 2 showed a meaningful upward trend, which is consistent

with the improvements implemented in prompt design and front-end enhancements that contributed to a better user experience.

In summary, while the statistical test does not show a significant difference between the two phases at the 5% level, both the negative *t*-statistic and the small-to-medium effect size (Cohen's *d* = 0.39) indicate a practically meaningful upward trend in Phase 2. This limited numerical gap is explained by the already high performance achieved in Phase 1, which reflected a strong initial reception of EverydAI and left little room for further gains. Nevertheless, the observed improvement aligns with targeted changes introduced in Phase 2, including refined prompt design to better guide users toward generated recommendations and enhancements to the front-end aimed at improving the overall user experience. Together, these findings suggest that the adjustments implemented have a positive impact.

4.1.7. User Feedback

User feedback collected through open-ended questions provided key insights for the evolution of EverydAI. The first question focused on identifying the most appreciated features of the system. Users highlighted:

1. 3D avatars and animations, which made interactions more engaging.
2. Accurate and appropriate recommendations provided by the assistants.
3. Effective object-detection capabilities integrated into the system.

The second question addressed areas needing improvement. The most frequently mentioned concerns such as high response times, low interface intuitiveness, and limited guidance on accessing recommendations are rigorously analyzed and used as the foundation for the redesign carried out between phases 1 and 2. These issues are systematically addressed through targeted changes to the interface, interaction design, and assistant behavior. Those changes are:

- Front-end changes: The overall interface buttons are redesigned. Previously, buttons contained text indicating their function (e.g., the button to initiate conversation said "Start Conversation"). In the updated version, these buttons are replaced with intuitive icons—for example, a microphone icon now represents the button to speak. This change is made in response to user feedback highlighting low intuitiveness. Some participants reported comments such as "I couldn't do this because I did not know how," indicating a lack of clarity in the original design. Additionally, the buttons are larger than in Phase 1 to further improve usability and overall user experience.
- Prompt refinement: In Phase 1, participants had difficulty accessing the recommendations—such as generated images or web-scraping links—because the recommendations window is initially closed when entering EverydAI. Users had to manually open it by clicking a small button, which many failed to notice. As a result, several participants reported that they could not find the recommendations provided, since EverydAI did not guide them clearly to that section. To address this issue related to low guidance, we refined the prompt given to the GPT model so that it would explicitly mention how to access the recommendations. For example, the assistant would now say: "The images of the recommended recipes can be seen in the recommendations window. Open it by pressing the arrow button."

As a result, these refinements led to the development of the improved version of EverydAI used by participants in Phase 2, which serves as the main version presented in the proposed approach. This updated system represents a direct response to user feedback and significantly improved the user experience. Participants in Phase 2 found it much easier to access both the core functionalities of EverydAI and the recommendations provided by the assistant. The clearer guidance and interface adjustments allowed users to navigate the

platform more intuitively and take full advantage of its capabilities, which is reflected in the more positive feedback and engagement observed during the second phase of testing.

4.2. Latency Testing

Throughout the full testing cycle of EverydAI, data is collected measuring the response time from the moment a user issues a request either via text or voice until the assistant provides an answer. To ensure analytical precision, the dataset is filtered to include only interactions where both the request type and the response type are either text or voice. The type field indicates how the user communicated with EverydAI—via textual input or spoken language—while the response type denotes the mode through which the assistant replied. A response type of text or voice signifies that the assistant's response does not involve any external task execution, such as database queries, web scraping, or API calls.

After applying this filtering, a total of 1082 response time values are considered for analysis. The mean response time is 5656.92 milliseconds with a standard deviation of 1711.58 milliseconds, indicating moderate variability in response durations, as can be seen in Figure 24.

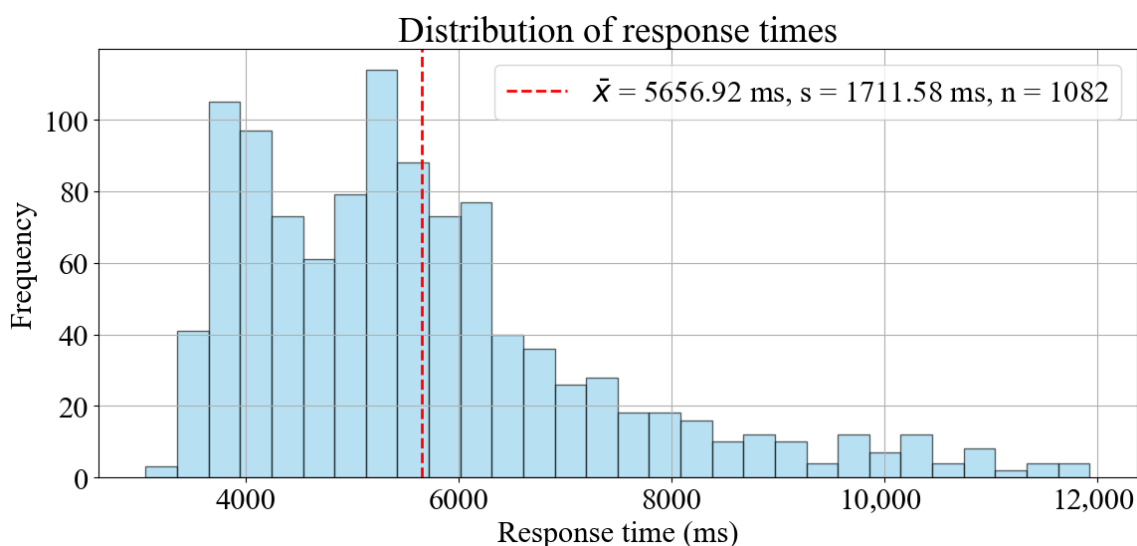


Figure 24. Distribution of the response times.

To statistically evaluate whether the average response time is significantly less than 6000 milliseconds (6 s), a one-tailed t -test is performed. The hypotheses for this test are formally stated as:

$$\begin{cases} H_0 : \mu = 6000 \text{ ms} \\ H_1 : \mu < 6000 \text{ ms} \end{cases} \quad (6)$$

The test statistic is computed as:

$$t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}} \quad (7)$$

where

- $\bar{x}$ is the sample mean;
- μ_0 is the hypothesized population mean;
- s is the sample standard deviation;
- n is the sample size.

The degrees of freedom for the *t*-test are:

$$v = n - 1 \quad (8)$$

The resulting *t*-statistic is -6.5935 , and the corresponding one-tailed *p*-value is approximately 3.4×10^{-11} . Given that the *p*-value is much smaller than the significance level $\alpha = 0.05$, the null hypothesis is rejected in favor of the alternative. This provides strong statistical evidence that the mean response time is significantly less than 6000 milliseconds.

These results demonstrate that under typical conditions, excluding external service delays and focusing only on text and voice interactions, the assistant provides responses well below the 6000 milliseconds requirement.

While the measured response times satisfy the internal system requirement of 6 s, previous research indicates that response times under approximately 1–2 s are ideal to maintain uninterrupted user flow in human–computer interactions [44,45]. Achieving such rapid responses is challenging in the context of EverydAI, which integrates 18 different technologies operating simultaneously within an advanced architecture. This level of integration is successfully managed to deliver results in real time, and it is worth noting that the wide range of tasks in which EverydAI assists users would take significantly longer if performed manually by the user alone. In this sense, the current latency represents an engineering accomplishment.

Nevertheless, there remains room for improvement. Future design strategies could focus on streamlining module interconnections and exploring alternative architectures, such as automation tools for AI agent workflows, with the goal of reducing latency while preserving the system’s functionality and enhancing the overall user experience.

5. Conclusions and Future Works

The development of EverydAI, a daily task-oriented virtual assistant, is successfully accomplished by integrating 18 different technologies into a single solution. This integration enables EverydAI to support users in their daily routines through intelligent recommendations and contextual assistance. Unlike traditional virtual assistants, EverydAI is specifically designed to enhance task efficiency by providing personalized, context-aware suggestions and step-by-step guidance.

The effectiveness of the system is underscored by the user evaluation metrics: on average, 92.7% of Phase 2 participants agreed or strongly agreed with statements regarding the system’s usefulness, ease of use, and overall performance, reflecting a high level of acceptance and perceived effectiveness of EverydAI. This strong approval suggests growing trust in EverydAI’s capabilities to support decision-making. Complementary statistical analysis, while not showing a significant difference at the 5% level, revealed indicators such as a negative *t*-statistic and a small-to-medium effect size that point to a meaningful upward trend from Phase 1 to Phase 2. This improvement is consistent with the targeted enhancements implemented in Phase 2, including refined prompt design and front-end optimizations, which contributed to a better guided and more engaging user experience.

Future development could explore the implementation of different software architectures to decrease the latency in the responses, the integration of platforms for the development of AI agent workflows to improve the backend functionalities, and the implementation of other recommendation techniques. Additionally, further validation approaches could be explored that focus on decision-making efficiency and time savings, providing a more comprehensive assessment of the system’s practical impact.

Author Contributions: Conceptualization, C.E.P.B., O.I.I.R., M.D.L.A. and C.G.Q.M.; methodology, C.E.P.B., O.I.I.R., M.D.L.A. and C.G.Q.M.; software, C.E.P.B., O.I.I.R. and M.D.L.A.; validation,

C.E.P.B., O.I.I.R., M.D.L.A. and C.G.Q.M.; investigation, C.E.P.B., O.I.I.R., M.D.L.A. and C.G.Q.M.; resources, C.E.P.B., O.I.I.R. and M.D.L.A.; data curation, C.E.P.B., O.I.I.R. and M.D.L.A.; writing—original draft preparation, C.E.P.B.; writing—review and editing, C.E.P.B. and C.G.Q.M.; supervision, C.G.Q.M.; project administration, C.G.Q.M.; funding acquisition, C.G.Q.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: This work was supported by Universidad del Norte, Barranquilla—Colombia. During the preparation of this manuscript, the authors used ChatGPT (GPT-4o) by OpenAI for the purposes of generating Figure 1, which is intended to illustrate the general idea we had before developing EverydAI. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

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

References

- Patel, N.C.; Jain, R.; Patel, N. Som Lalit Institute of Management Studies and Som Lalit Institute of Business Management Abundant Choices: Consumer's Dilemma. 2021.
- Xuan, Z.; Qingwen, Z. Difficult Choices: Exploring Basic Reasons of Difficult Choices. *J. Sociol. Ethnol.* **2021**, *3*, 182–191. [\[CrossRef\]](#)
- Roy, D.; Dutta, M. A Systematic Review and Research Perspective on Recommender Systems. *J. Big Data* **2022**, *9*, 59. [\[CrossRef\]](#)
- Zadeh, E.K.; Alaeifard, M. Adaptive Virtual Assistant Interaction through Real-Time Speech Emotion Analysis Using Hybrid Deep Learning Models and Contextual Awareness. *Int. J. Adv. Hum. Comput. Interact.* **2024**, *1*, 1–15.
- Dong, X.L.; Moon, S.; Xu, Y.E.; Malik, K.; Yu, Z. Towards Next-Generation Intelligent Assistants Leveraging LLM Techniques. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, Long Beach, CA, USA, 6–10 August 2023; pp. 5792–5793.
- Yao, Y.; Duan, J.; Xu, K.; Cai, Y.; Sun, Z.; Zhang, Y. A Survey on Large Language Model (LLM) Security and Privacy: The Good, The Bad, and The Ugly. *High-Confid. Comput.* **2024**, *4*, 100211. [\[CrossRef\]](#)
- Naik, D.; Naik, I.; Naik, N. Decoder-Only Transformers: The Brains Behind Generative AI, Large Language Models and Large Multimodal Models. In Proceedings of the Contributions Presented at the International Conference on Computing, Communication, Cybersecurity and AI, London, UK, 3–4 July 2024.
- Rajan Singh, R.; Singh, P.; Pradesh, U. Chatbot: Chatbot Assistant. *J. Manag. Serv. Sci.* **2024**, *4*, 1–19. [\[CrossRef\]](#)
- Pereira, R.; Lima, C.; Pinto, T.; Reis, A. Virtual Assistants in Industry 4.0: A Systematic Literature Review. *Electronics* **2023**, *12*, 4096. [\[CrossRef\]](#)
- Zou, Z.; Chen, K.; Shi, Z.; Guo, Y.; Ye, J. Object Detection in 20 Years: A Survey. *Proc. IEEE* **2023**, *111*, 257–276. [\[CrossRef\]](#)
- Wu, L.; Zheng, Z.; Qiu, Z.; Wang, H.; Gu, H.; Shen, T.; Qin, C.; Zhu, C.; Zhu, H.; Liu, Q.; et al. A Survey on Large Language Models for Recommendation. *World Wide Web* **2024**, *27*, 60. [\[CrossRef\]](#)
- Schedl, M.; Deldjoo, Y.; Castells, P.; Yilmaz, E. Introduction to the Special Issue on Trustworthy Recommender Systems. *ACM Trans. Recomm. Syst.* **2025**, *3*, 1–8. [\[CrossRef\]](#)
- Zhang, A.; Chen, Y.; Sheng, L.; Wang, X.; Chua, T.S. On Generative Agents in Recommendation. In Proceedings of the SIGIR 2024—Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, Washington, DC, USA, 14–18 July 2024; Association for Computing Machinery: New York, NY, USA, 2024; pp. 1807–1817.
- Altundas, S.; Karaarslan, E. Cross-Platform and Personalized Avatars in the Metaverse: Ready Player Me Case. In *Digital Twin Driven Intelligent Systems and Emerging Metaverse*; Springer Nature: Berlin/Heidelberg, Germany, 2023; pp. 317–330, ISBN 9789819902521.
- Liu, Z.; Wang, D.; Gao, H.; Li, M.; Zhou, H.; Zhang, C. Metasurface-Enabled Augmented Reality Display: A Review. *Adv. Photonics* **2023**, *5*, 034001. [\[CrossRef\]](#)
- Sirisuriya, S.C.M.D.S. Importance of Web Scraping as a Data Source for Machine Learning Algorithms—Review. In Proceedings of the 2023 IEEE 17th International Conference on Industrial and Information Systems, ICIIS 2023—Proceedings, Peradeniya, Sri Lanka, 25–26 August 2023; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 2023; pp. 134–139.
- Vu, M.D.; Wang, H.; Li, Z.; Chen, J.; Zhao, S.; Xing, Z.; Chen, C. GPTVoiceTasker: Advancing Multi-Step Mobile Task Efficiency Through Dynamic Interface Exploration and Learning. In Proceedings of the UIST'24: Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology, Pittsburgh, PA, USA, 13–16 October 2024.

18. Zheng, J.; Fischer, M. BIM-GPT: A Prompt-Based Virtual Assistant Framework for BIM Information Retrieval. *arXiv* **2023**, arXiv:2304.09333.
19. Manojkumar, P.K.; Patil, A.; Shinde, S.; Patra, S.; Patil, S. AI-Based Virtual Assistant Using Python: A Systematic Review. *Int. J. Res. Appl. Sci. Eng. Technol.* **2023**, *11*, 814–818. [CrossRef]
20. Anand, A.; Subha, R.; Rajan, S.; Bharathi, N.; Srivastava, A.K. An Efficient, Precise and User Friendly AI Based Virtual Assistant. In Proceedings of the 2023 International Conference on the Confluence of Advancements in Robotics, Vision and Interdisciplinary Technology Management, IC-RVITM 2023, Bangalore, India, 28–29 November 2023; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 2023.
21. Babu, G.J.; Safrinfathima, S.; Reddy, K.D.; Sen, S.K. Transforming Human-Machine Interaction: Generative AI Virtual Asst. In Proceedings of the 2024 International Conference on Knowledge Engineering and Communication Systems, ICKECS 2024, Chikkaballapur, India, 18–19 April 2024; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 2024.
22. Li, C.; Wong, C.; Zhang, S.; Usuyama, N.; Liu, H.; Yang, J.; Naumann, T.; Poon, H.; Gao, J. LLaVA-Med: Training a Large Language-and-Vision Assistant for Biomedicine in One Day. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 28541–28564.
23. Dhelia, A.; Chordia, S.; Kanisha, B. YOLO-Based Food Damage Detection: An Automated Approach for Quality Control in Food Industry. In Proceedings of the 8th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), I-SMAC 2024—Proceedings, Kirtipur, Nepal, 3–5 October 2024; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 2024; pp. 1444–1449.
24. Nitiavintokana, F.; Rouf, M.A.; Yu, X.; Wu, H.; Wang, A.; Iwahori, Y. Intelligent Food Identification System Using Improved YOLOv8. In Proceedings of the ISCSET 2024—13th International Symposium on Computer Science and Educational Technology, Lauterbach, Germany, 17–19 July 2024; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 2024.
25. Jiang, H.; Zhang, Y.; Li, W. Construction of a Food Packaging Safety Closure Detection System Based on Improved Yolo Algorithm. In Proceedings of the 2nd IEEE International Conference on Integrated Intelligence and Communication Systems, ICIICS 2024, Kalaburagi, India, 22–23 November 2024; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 2024.
26. Deeksha; Meenpal, T.; Khan, M.S.; Sahu, M. Automatic Food Billing System of Indian Food Using YOLOv8 Model. In Proceedings of the 2024 15th International Conference on Computing Communication and Networking Technologies, ICCCNT 2024, Kamand, India, 24–28 June 2024; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 2024.
27. Han, X.; Zheng, D.; Wang, D. An Enhanced Clothing Detection Model for E-Commerce Applications. In Proceedings of the Proceedings—2024 39th Youth Academic Annual Conference of Chinese Association of Automation, YAC 2024, Dalian, China, 7–9 June 2024; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 2024; pp. 1181–1186.
28. Jain, D.; Thazhathu, E.M.; Adiraju, I.; Bhattacharya, J.; Singh, D. FashionAI: Image-Based Clothing Detection and Shopping Recommendation. In Proceedings of the 2024 3rd International Conference for Innovation in Technology, INOCON 2024, Bangalore, India, 1–3 March 2024; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 2024.
29. Almabruk, S.; Abdalhamid, S.; Almabruk, T. Comparative Reliability Analysis of Selenium and Playwright: Evaluating Automated Software Testing Tools. *Asian J. Res. Comput. Sci.* **2025**, *18*, 34–44. [CrossRef]
30. AI Model & API Providers Analysis | Artificial Analysis Intelligence, Performance & Price Analysis | Artificial Analysis. Available online: <https://artificialanalysis.ai/models/o4-mini> (accessed on 1 July 2025).
31. Yaman, E. FOOD INGREDIENTS DETECTION Dataset. In *Roboflow Universe*. 2025. Available online: <https://universe.roboflow.com/yaman-e/food-ingredients-detection-qf7> (accessed on 1 March 2025).
32. Python, Ingredients Dataset. In *Roboflow Universe*. 2024. Available online: <https://universe.roboflow.com/python-wfiwe/ingredients-mqhx> (accessed on 1 July 2025).
33. DelfinPE3 Clothes Detection V1 Dataset. In *Roboflow Universe*. 2025. Available online: <https://universe.roboflow.com/delfinpe3/clothes-detection-v1-loczc> (accessed on 1 March 2025).
34. cnn with Fashion Detection cnn Fashion Detion in Ecommerce Dataset. In *Roboflow Universe*. 2024. Available online: <https://universe.roboflow.com/cnn-with-fashion-detection/cnn-fashion-detion-in-ecommerce> (accessed on 1 March 2025).
35. Main Fashion Dataset. In *Roboflow Universe*. 2023. Available online: <https://universe.roboflow.com/dataset-yn4f8/main-fashion> (accessed on 1 March 2025).
36. FitFuel All Gym Equipment Dataset. In *Roboflow Universe*. 2024. Available online: <https://universe.roboflow.com/bruuj/main-fashion-wmyfk> (accessed on 1 March 2025).
37. RizzLabzz GymBro Dataset. In *Roboflow Universe*. 2023. Available online: <https://universe.roboflow.com/rizzlabzz/gymbro> (accessed on 1 March 2025).
38. Pruzek, F.E.R.C. Fitness Equipment Recognition Dataset. In *Roboflow Universe*. 2024. Available online: <https://universe.roboflow.com/fitness-equipment-recognition-colin-pruzek/fitness-equipment-recognition-wlluo> (accessed on 1 March 2025).
39. Dean, T.B.; Seecheran, R.; Badgett, R.G.; Zackula, R.; Symons, J. Perceptions and Attitudes toward Artificial Intelligence among Frontline Physicians and Physicians' Assistants in Kansas: A Cross-Sectional Survey. *JAMIA Open* **2024**, *7*, ooae100. [CrossRef] [PubMed]

40. Nam, K.H.; Kim, D.Y.; Kim, D.H.; Lee, J.H.; Lee, J.I.; Kim, M.J.; Park, J.Y.; Hwang, J.H.; Yun, S.S.; Choi, B.K.; et al. Conversational Artificial Intelligence for Spinal Pain Questionnaire: Validation and User Satisfaction. *Neurospine* **2022**, *19*, 348–356. [[CrossRef](#)] [[PubMed](#)]
41. Gusti, M.A.; Satrianto, A.; Candrianto; Juniardi, E.; Fitra, H. Artificial intelligence for employee engagement and productivity. *Probl. Perspect. Manag.* **2024**, *22*, 174–184. [[CrossRef](#)]
42. West, R.M. Best Practice in Statistics: Use the Welch *t*-Test When Testing the Difference between Two Groups. *Ann. Clin. Biochem.* **2021**, *58*, 267–269. [[CrossRef](#)] [[PubMed](#)]
43. Jané, M.B.; Xiao, Q.; Yeung, S.K.; Ben-Shachar, M.S.; Caldwell, A.R.; Cousineau, D.; Dunleavy, D.J.; Elsherif, M.; Johnson, B.T.; Moreau, D.; et al. Guide to Effect Sizes and Confidence Intervals. 2024. Available online: <https://matthewbjane.quarto.pub/> (accessed on 1 July 2025). [[CrossRef](#)]
44. Miller, R.B. Response Time in Man-Computer Conversational Transactions. In Proceedings of the AFIPS'68 Fall Joint Computer Conference, San Francisco, CA, USA, 9–11 December 1968.
45. Card, S.K.; Robertson, G.G.; Mackinlay, J.D. The information visualizer, an information workspace. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, New Orleans, LA, USA, 27 April–2 May 1991.

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