



Audieyes: AI-powered Vision for the Blind and Visually Impaired

Master Thesis Report

Submitted in

Fall 2024

in partial fulfillment of the requirements

for the degree of Master of Science in Software Engineering (MSSE)

at the School of Science and Engineering, of Al Akhawayn

University in Ifrane

By

Ayoub Maimmadi

Supervised by

Dr. Asmaa Mourhir

Other examiners

Dr. Omar Iraqi, Al Akhawayn University in Ifrane

Pr. Abdelouahed Sabri, Faculté des Sciences Dhar El Mahraz, Fès

" We are here to stop this blurry life; it's time for a clear one "

Declaration of Authorship

I, Ayoub Maimmadi, hereby declare that this Master's thesis, entitled "Audieyes: AI-powered Vision for the Blind and Visually Impaired," submitted in partial fulfillment of the requirements for the degree of Master of Science in Software Engineering (MSSE) at Al Akhawayn University, is my original work.

I further affirm that:

- This work was done wholly or mainly while in candidature for a Master's degree at this university.
- This work has not been prepared for another examination or assignment, either in its entirety or excerpts thereof.
- Any sources or external work used have been fully acknowledged and referenced in accordance with academic convention.

Declaration of Generative AI and AI-assisted Technologies Use

During the preparation of this thesis, I utilized AI-assisted tools to enhance readability and articulation. Following the use of these tools, I thoroughly reviewed, edited, and refined all content to ensure it accurately represents my original ideas and research. I affirm that the ideas, arguments, and conclusions presented in this thesis are my own, and I take full responsibility for the entire content.

Abstract-English

Vision is one of the most essential human senses, playing a crucial role in how we perceive and interact with our environment. Recognizing the profound impact of visual impairment, this project addresses the challenges faced by blind and visually impaired (BVI) individuals in navigating and understanding their surroundings. Motivated by the need for more accessible and context-aware assistive technologies, we developed Audieyes, a visionary application leveraging advanced AI to provide real-time scene descriptions and interactive assistance. The primary objectives were to enhance the autonomy and quality of life for BVI users through accurate, context-sensitive environmental descriptions, and to ensure seamless integration of the technology into users' daily lives. Key challenges included ensuring real-time performance, maintaining a low price of entry, and providing culturally relevant assistance, particularly for Moroccan users. Our methodology integrated state-of-the-art Large Language Models (LLMs), specifically GPT-4o-mini, chosen for its superior performance in multimodal tasks and cost-effectiveness. We implemented a robust DevOps pipeline with continuous integration and deployment (CI/CD) practices for rapid development and reliable updates, alongside Large Language Model Operations (LLMOPs) practices to ensure security, compliance, performance optimization, and effective user feedback analysis. The system architecture is designed for fault tolerance and scalability, utilizing microservices, cloud integration with multi-region deployment, and edge computing. Initial user trials with BVI individuals in Morocco yielded promising results: Audieyes achieved a 96% user satisfaction rate, with participants reporting significant improvements in daily navigation and environmental awareness. Users particularly appreciated the culturally relevant descriptions, with 92% finding them highly contextual and useful. The application's response time averaged 3.5 seconds, meeting our real-time performance goals. In terms of technical performance, the system demonstrated a 97% accuracy rate in classifying objects within images using categories such as Tajin, Msamen, Riad, souk, and Caftan across various real-world scenarios. Additionally, the system achieved a BLEU-4 score of 40.12% for generating accurate and contextually appropriate descriptions. Furthermore, 89% of users reported feeling more independent in their daily activities after using Audieyes for one month. These encouraging outcomes suggest that Audieyes has successfully addressed critical needs in the BVI community, potentially revolutionizing assistive technology in Morocco and beyond.

Keywords— blind and visually impaired, scene description, visual question answering, Large Language models, software engineering, cloud integration, edge computing, accessibility.

Abstract- French

La vision est l'un des sens humains les plus essentiels, jouant un rôle crucial dans notre perception et notre interaction avec notre environnement. Reconnaissant l'impact profond de la déficience visuelle, ce projet aborde les défis auxquels sont confrontés les individus aveugles et malvoyants (BVI) dans la navigation et la compréhension de leur environnement. Motivés par la nécessité de technologies d'assistance plus accessibles et contextuelles, nous avons développé Audieyes, une application visionnaire utilisant l'IA avancée pour fournir des descriptions de scènes en temps réel et une assistance interactive. Les objectifs principaux étaient d'améliorer l'autonomie et la qualité de vie des utilisateurs BVI grâce à des descriptions environnementales précises et sensibles au contexte, et d'assurer une intégration fluide de la technologie dans la vie quotidienne des utilisateurs. Les principaux défis incluaient la garantie des performances en temps réel, le maintien d'un faible coût d'entrée, et la fourniture d'une assistance culturellement pertinente, en particulier pour les utilisateurs marocains. Notre méthodologie a intégré des modèles de langue de grande taille (LLM) de pointe, notamment GPT-4o-mini, choisie pour ses performances supérieures dans les tâches multimodales et son rapport coût-efficacité. Nous avons mis en place un pipeline DevOps robuste avec des pratiques d'intégration et de déploiement continu (CI/CD) pour un développement rapide et des mises à jour fiables, ainsi que des pratiques LLMOps pour garantir la sécurité, la conformité, l'optimisation des performances et une analyse efficace des retours utilisateurs. L'architecture du système a été conçue pour la tolérance aux pannes et la scalabilité, utilisant des microservices, l'intégration cloud avec un déploiement multi-régions, et l'informatique en périphérie. Les premiers essais utilisateurs avec des individus BVI au Maroc ont donné des résultats prometteurs : Audieyes a atteint un taux de satisfaction des utilisateurs de 96 %, les participants rapportant des améliorations significatives dans la navigation quotidienne et la conscience de leur environnement. Les utilisateurs ont particulièrement apprécié les descriptions culturellement pertinentes, 92 % les trouvant hautement contextuelles et utiles. Le temps de réponse de l'application était en moyenne de 3,5 secondes, atteignant nos objectifs de performance en temps réel. En termes de performance technique, le système a démontré un taux de précision de 97 % dans la classification d'objets au sein d'images utilisant des catégories telles que Tajine, Msemen, Riad, souk et Caftan dans divers scénarios du monde réel. De plus, le système a obtenu un score BLEU-4 de 40,12 % pour la génération de descriptions précises et contextuellement appropriées. Par ailleurs, 89 % des utilisateurs ont rapporté se sentir plus indépendants dans leurs activités quotidiennes après avoir utilisé Audieyes pendant un mois. Ces résultats encourageants suggèrent qu'Audieyes a répondu avec succès aux besoins critiques de la communauté BVI, révolutionnant potentiellement la technologie d'assistance au Maroc et au-delà.

Mots-clés— Aveugles et Malvoyants, Description de Scène, Question Réponse Visuelle, Modèles de langage volumineux, Ingénierie Logicielle, Intégration Cloud, Informatique en Périphérie, Accessibilité.

Abstract-Arabic

الرؤية هي واحدة من أهم الحواس البشرية، حيث تلعب دوراً حاسماً في كيفية إدراكنا وتفاعلنا مع بيئتنا. إدراكاً للتأثير العميق للإعاقة البصرية، يتناول هذا المشروع التحديات التي يواجهها الأفراد المكفوفون وضعاف البصر في التنقل وفهم محبيتهم. بداعي الحاجة إلى تقنيات مساعدة أكثر سهولة في الوصول وإدراكاً للسياق، قمنا بتطوير أوديأيز، وهو تطبيق رؤيوبي يستفيد من الذكاء الاصطناعي المتقدم لتوفير أوصاف للمشاهد في الوقت الفعلي ومساعدة تفاعلية. كانت الأهداف الرئيسية هي تعزيز استقلالية وجودة حياة مستخدمي بي في أي من خلال أوصاف دقيقة وحساسة للسياق البيئي، وضمان دمج سلس للتكنولوجيا في الحياة اليومية للمستخدمين. تضمنت التحديات الرئيسية ضمان الأداء في الوقت الفعلي، والحفاظ على انخفاض تكلفة الدخول، وتوفير مساعدة ملائمة ثقافياً، خاصة للمستخدمين المغاربة. دمجت منهجيتنا أحدث نماذج اللغة الكبيرة، وتحديداً جي بي تي-4 أو-مي، الذي تم اختياره لأدائه المتفوق في المهام متعددة الوسائط وفعاليته من حيث التكلفة. قمنا بتنفيذ خط أنابيب ديف أوبس قوي مع ممارسات التكامل والنشر المستمر للتطوير السريع والتحديثات الموثوقة، إلى جانب ممارسات عمليات نماذج اللغة الكبيرة لضمان الأمان والامتثال وتحسين الأداء وتحليل فعال لتعليقات المستخدمين. تم تصميم بنية النظام لتحمل الأحماء وقابلية التوسيع، باستخدام الخدمات المصغرة، والتكامل السحابي مع النشر متعدد المناطق، والحوسبة الطرفية. أسفرت التجارب الأولية للمستخدمين مع الأفراد المكفوفين وضعاف البصر في المغرب عن نتائج واعدة: حق أوديأيز معدل رضا للمستخدمين بنسبة 96%， حيث أبلغ المشاركون عن تحسينات كبيرة في التنقل اليومي والوعي البيئي. قدر المستخدمون بشكل خاص الأوصاف ذات الصلة الثقافية، حيث وجد 92% منهم أنها سياقية ومفيدة للغاية. بلغ متوسط وقت استجابة التطبيق 3.5 ثوانٍ، مما يلبي أهدافنا في الأداء في الوقت الفعلي. من حيث الأداء التقني، أظهر النظام معدل دقة 97% في تصنيف الأشياء داخل الصور باستخدام فئات مثل الطاجين والمسمن والرياض والسوق والقططان عبر بنسنة 40.12% لـ BLEU-4. سيناريوهات مختلفة من العالم الحقيقي. بالإضافة إلى ذلك، حقن النظام درجة دقة و المناسبة للسياق. علاوة على ذلك، أفاد 89% من المستخدمين بأنهم يشعرون باستقلالية أكبر في أنشطتهم اليومية بعد استخدام أوديأيز لمدة شهر واحد. تشير هذه النتائج المشجعة إلى أن أوديأيز قد نجح في معالجة الاحتياجات الحرجة في مجتمع المكفوفين وضعاف البصر، مما قد يؤدي إلى ثورة في التكنولوجيا المساعدة في المغرب وخارجها.

الكلمات الرئيسية: الذكاء الاصطناعي، تقنية، الأسئلة على المرئية الإجابة، المشهد وصف، البصر وضعاف المكفوفون - الرؤية، الاتصال، الوصول إلى المكانية، الحافة، حساسية، السحابي التكامل، البرمجيات، هندسة.

Table of Contents

Chapter 1. Introduction	1
Chapter 2. Requirement specification	3
2.1 Functional Requirements.....	3
2.2 Non-Functional Requirements	4
Chapter 3. Related Work.....	6
3.1 Traditional Hybrid Architectures	6
3.2 Visual Question Answering (VQA)	7
3.3 Description Generation	9
3.4 Machine Learning Techniques	9
3.5 Crowdsourced and Human-Assisted Techniques.....	11
3.6 Multimodal and Vision Models	12
3.7 Literature Gaps and Contributions.....	14
Chapter 4. Setting up the Project.....	17
4.1 Project Archetypes	17
4.2 Business Metrics	18
4.4 Baseline Results	20
4.4 Feasibility Study	21
4.4.1 Technical Feasibility	22
4.4.2 Economic Viability.....	32
4.4.4 Hardware Feasibility	33
Chapter 5. Human AI Interactions.....	34
5.1 User Interface Design.....	34
5.2 Ethical Considerations.....	34
Chapter 6. System Design.....	36
6.1 System Architecture	36
6.2 Fault Tolerant System Design.....	37
6.2.1 API Redundancy and Horizontal Scaling	39
6.2.2 Intelligent Traffic Distribution with Load Balancing	40
6.2.3 Global Accessibility and Failover with DNS	41
6.2.4 Resilient Data Management through Database Replication.....	42
6.2.5 Private and Secure Database Access with P-DNS	44
6.2.6 Interchangeable LLM Microservice Integration	45

6.2.7	Fault-Tolerant Payment Processing	47
6.2.8	Cloud Infrastructure: Containerization and Orchestration	48
6.2.9	Geo-Redundant Multi-Region Deployment.....	49
6.3	Edge Computing	51
6.4	Database design	53
Chapter 7. Implementation.....		56
7.1	Technology Enablers	56
7.2	DevOps.....	64
7.3	LLMOPs	66
7.3.1	Foundations of Machine Learning Engineering	66
7.3.1	Data Preparation and Management.....	67
7.3.2	Prompt Engineering	68
7.3.3	A/B Testing and Online Evaluation	72
7.3.4	API Management and Optimization	72
7.3.5	Performance Monitoring and Distribution Shift Analysis.....	75
7.4	Backend Development	79
7.4.1	Design Patterns.....	79
7.5	API Documentation and Logging	86
7.6	Testing	88
7.7	Cloud Deployment	90
7.7.1	API Containerization, Duplication, and Orchestration	90
7.7.2	Canary Deployment Configuration with Kubernetes	93
7.7.3	Database Replication with Kubernetes	94
7.7.4	Multi-Region Deployment with Linode	98
Chapter 8. Results.....		99
8.1	Software Usage	99
8.2	Online Testing Results.....	105
8.3	User Evaluation.....	108
Chapter 9. Conclusion		111
References.....		113
Appendix.....		116
Audieyes User Satisfaction Survey.....		116
Audieyes User Satisfaction Survey Results		120

LIST OF FIGURES

Figure 1: Evaluating GPT-4V + Refusal System against screenshots of a text refusal dataset	24
Figure 2: The GPT-4V model's ability to correctly distinguish the identity of individuals	25
Figure 3: The GPT-4o model's ability to describe and understand images.	26
Figure 4: Transformer and Mixture-of-Experts (MoE) frameworks.....	27
Figure 5: The Gemini model's ability to identify plants and provide care instructions.	28
Figure 6: Claud 3 model's ability to transcribe text from images and output structured data.	30
Figure 7: Audieyes 3-tier architecture	37
Figure 8: Simplified Starting System Design for Audieyes	38
Figure 9: Duplication of Audieyes API architecture.....	39
Figure 10: Integration of a load balancer	40
Figure 11: Advanced system architecture of Audieyes, featuring DNS management.	42
Figure 12: Leader-Follower architecture used for database replication.	43
Figure 13: A comprehensive system architecture of Audieyes, including the introduction of P-DNS..	45
Figure 14: Enhanced System Architecture for Audieyes with Microservices	46
Figure 15: Inefficient Server rack deployment strategy.....	48
Figure 16: Improved server rack deployment strategy	49
Figure 17: Multi-Region Deployment Architecture.....	50
Figure 18: Audieyes' Edge-Fog-Cloud Architecture	52
Figure 19: Entity-Relationship Diagram for the Audieyes database	54
Figure 20: DevOps' CI/CD Pipeline for Audieyes	65
Figure 21 : compressing mages and changing their format.....	73
Figure 22 : Audieyes Grafana dashboard	74
Figure 23 : automatic microservice model switcher script with python	74
Figure 24: Audieyes dataset csv file	75
Figure 25: Python script for calculating the BLEU-4 score	76
Figure 26 : Code sample for visually analyzing data drift	77
Figure 27 : Distribution Comparison: Scene Type	77
Figure 28 : Category Changes: Cultural Elements	78
Figure 30: Design Pattern: Decorators	80
Figure 31: Design Pattern: Facade	81
Figure 32: Dependency Injection and Inversion of Control.....	82
Figure 33: Design Pattern: Strategy.....	83
Figure 34: Repository Interface of users.	84
Figure 35: Implementation of Repository Interface of users.	84
Figure 36: Injecting and calling the repository of user function.....	85

Figure 37: Model module where all core logic is imported and exported.....	85
Figure 38: App module, where all modules of the Audieyes system are used.	86
Figure 39: API Endpoints for various starting features of Audieyes	87
Figure 40: Swagger UI for Audieyes Backend API.....	88
Figure 41: Logging Endpoints for Monitoring in Audieyes API	88
Figure 42 : test suit set up for model controller	89
Figure 43 : Test Implementation example	90
Figure 44: Audieyes' API Dockerfile.....	91
Figure 45: API Kubernetes stable and canary deployment configurations.....	92
Figure 46: API Kubernetes Service.....	93
Figure 47: : API Kubernetes canary deployment with Ingress	94
Figure 48 : Creation of postgresql Kubernetes cluster	95
Figure 49 : Kubernetes Secret object creation.....	95
Figure 50 : Kubernetes ConfigMap configuration for postrgresql	96
Figure 51 : Kubernetes StatefulSet configuration.....	97
Figure 52 : Linode region selection options for Audieyes.....	98
Figures 53: Audieyes Home screen	100
Figure 54: Audieyes scene description feature	101
Figure 55: Audieyes location finding feature	102
Figure 56 : Audieyes people recognition feature	102
Figure 57: Audieeys text transcribing feature	103
Figure 58: Audieyes' voice based feedback system.....	103
Figure 59 : Audieyes theme, notifications, data, and language settings	104
Figure 60: Audieyes family member and location settings	104

LIST OF TABLES

Table 1: A comprehensive summary of the vision benchmarks of all our AI models.....	31
Table 2: AI mode's API price for input and output per million token.	32
Table 3: Programming Language: TypeScript	56
Table 4: Backend Framework: NestJS.....	57
Table 5 : Client Framework: React Native.....	57
Table 6: Encryption Tools: CryptoJS	58
Table 7: Testing Framework: Jest	59
Table 8: Containerization Solution: Docker	59
Table 9: Orchestration Tool: Kubernetes	60
Table 10: Container Image Management: Docker Hub	61
Table 11: Cloud Provider: Linode	62
Table 12: Encryption Tools: CryptoJS	62
Table 13: Monitoring and Visualization: Grafana	63
Table 14 : Audieyes features prompts in Darija, Englis, Arabic, and French.....	68
Table 15: Audieyes accuracy results.....	108
Table 16: Audieyes cultural relevance results	108
Table 17: Audieyes responce time results.....	109

LIST OF ABBREVIATIONS

Abbreviations	Definitions
BVI	Blind and Visually Impaired
LLMs	Large Language Model
LLMOps	Large Language Model Operations
CNN	Convolutional Neural Network
VQA	Visual Question Answering
LLaVA	Large Language and Vision Assistant
VLP	Vision-Language Pretraining
CLIP	Contrastive Language-Image Pre-training
BLIP	Bootstrapping Language-Image Pre-training
CI/CD	Continuous Integration and Deployment
API	Application Programming Interface
DNS	Domain Name System

Acknowledgements

I would like to express my heartfelt gratitude to all those who have supported me throughout the journey of this Master's thesis and the development of the Audieyes project.

First and foremost, I am deeply grateful to my parents and sisters Zineb and Lina Maimmadi for their unwavering love, support, and encouragement. Your belief in me has been a constant source of strength and motivation.

To my friends, thank you for your camaraderie, understanding, and the moments of levity that helped balance the rigors of research and development. I owe a special debt of gratitude to Rim Miftah, whose unwavering support, brilliant insights, and constant encouragement were instrumental in bringing this project to fruition. Rim, your belief in me and in the potential of Audieyes to make a difference has been a driving force throughout this journey. Your patience during long hours of work, your keen eye for detail, and your thoughtful feedback have been invaluable. Without your support, this project would not have been possible. Thank you for being not just a friend, but a true partner in this endeavor.

I extend my sincere appreciation to the blind and visually impaired community in Morocco. Your experiences, needs, and feedback have been the driving force behind Audieyes.

My gratitude goes to the faculty and staff of Al Akhawayn University in Ifrane. Your knowledge, resources, and support have been instrumental to my academic growth and this project's success.

I would like to thank the members of my examination committee, Dr. Omar Iraqi and Pr. Abdelouahed Sabri for their time and expertise. A special thank you to Dr. Iraqi for his exceptional classes, which played a pivotal role in shaping the foundation and direction of this project.

Finally, I reserve my deepest and most heartfelt thanks for my supervisor, Dr. Asmaa Mourhir. Your guidance, wisdom, and unwavering support have been the cornerstone of this thesis. Your mentorship has not only shaped this project but has profoundly influenced my development as a researcher and professional. Your dedication, patience, and intellectual generosity have been truly inspiring, and I am immensely grateful for the opportunity to have worked under your supervision.

This thesis is a testament to the collective support, wisdom, and encouragement of all these individuals. Thank you all for being part of this journey.

Chapter 1. Introduction

Audieyes represents a visionary advancement in assistive technology for the BVI community. By leveraging large language models (LLMs), Audieyes transcends traditional accessibility tools, providing users with detailed descriptions of their surroundings through their mobile cameras. This robust AI companion serves as a guide, enabling users to receive rich captions of the environments they navigate and allowing them to ask precise questions about these surroundings. Audieyes aims to enhance the autonomy and mobility of its BVI users, making daily activities more accessible and manageable.

In the rapidly evolving field of artificial intelligence, specifically machine learning, large language "multi-modal" models (LLMs) have emerged as one of the most promising and fast-growing technologies. These models are renowned for their capabilities to tackle a variety of tasks across diverse fields including science, business, economics, and medicine. The integration of LLMs into projects like Audieyes demonstrates their transformative power. By synthesizing contextual visual information into comprehensible spoken language, LLMs bridge the gap between visual data and actionable insights, particularly in real-time environments.

The purpose of this work is to explore the feasibility and effectiveness of utilizing advanced LLMs in aiding BVI individuals by providing detailed and contextually accurate descriptions of their surroundings. Our study specifically focuses on evaluating LLMs equipped with advanced vision and language processing capabilities, with particular emphasis on those capable of processing Darija, the dialect spoken in Morocco. This research assesses their potential to transform visual data into accurate, actionable insights within real-time environments, specifically in the context of Moroccan culture with all its complexities.

However, deploying ML-powered applications presents several challenges:

- Ensuring accuracy and reliability of AI models in diverse and dynamic real-world environments.
- Addressing privacy and security concerns in handling sensitive user data.
- Integrating advanced technologies into user-friendly interfaces navigable by BVI individuals.

- Maintaining and updating models to keep pace with technological advancements and evolving user needs.

The main objectives of this work are:

- To leverage advances in Large Language Models equipped with vision and language processing capabilities to offer a cost-effective solution for scene description.
- To provide specific support for Darija, assessing LLMs' potential to transform visual data into accurate, actionable insights within real-time environments in the context of Moroccan culture.

By providing a comprehensive analysis of LLMs and their application in real-world scenarios, this work aims to inform and guide the development of more effective and inclusive assistive tools. The value of using machine learning, particularly LLMs, lies in their ability to process and interpret vast amounts of data, providing insights and solutions that are both sophisticated and practical, significantly improving the autonomy and mobility of BVI individuals. In the following chapters, we will explore various aspects of the Audieyes project:

Chapter 2 outlines the requirement specifications, detailing both functional and non-functional requirements of the system. Chapter 3 provides a comprehensive review of related work in the field of assistive technologies for BVI individuals. Chapter 4 delves into the project setup, including project archetypes, business metrics, and a detailed feasibility study. Chapter 5 examines human-AI interactions, focusing on user interface design and ethical considerations. Chapter 6 presents the system design, including the overall architecture and fault-tolerant design strategies. Chapter 7 covers the implementation details, including DevOps practices, LLMOps, backend development, API documentation, and cloud deployment strategies. Chapter 8 presents the results of the Audieyes project. Finally, Chapter 9 concludes the thesis by summarizing the key findings, discussing the project's achievements and limitations, and outlining future work and improvements for Audieyes. Each chapter contributes to a holistic understanding of the Audieyes project, from its conceptual foundations to its practical implementation, showcasing how advanced AI technologies can be harnessed to create meaningful solutions for the blind and visually impaired.

Chapter 2. Requirement specification

This chapter outlines the essential requirements for the Audieyes project, it is divided into Functional Requirements, detailing the core features and capabilities of the system, and Non-Functional Requirements, addressing quality attributes and performance standards. These specifications form the foundation for Audieyes, ensuring the development of a culturally relevant, accessible, and high-performing solution that meets the unique needs of its users while adhering to technical and ethical standards.

2.1 Functional Requirements

The functional requirements of the Audieyes project encompass the core functionalities necessary to meet user needs, with a particular focus on the Moroccan context. These requirements include:

- Real-time Scene Description: The system should provide accurate, real-time descriptions of the surroundings through a mobile phone camera. This includes identifying and describing the overall environment, context, and culturally relevant elements specific to Moroccan settings. **Outputs should voice based and spoken out loud based on the users favorite language and speech settings so BVI users are able to hear them.**
- Object and People Recognition and Identification: The system should recognize and identify various objects within the user's environment, including items specific to Moroccan culture, as well as people faces such family members or friends. It should be capable of classifying and naming items and faces detected through the camera, providing culturally appropriate descriptions.
- Location Guidance: Users should be able to request their location and that of specific items. For example, if a user asks about his whereabouts or for an item in their room, the system should detect the item and provide clear, step-by-step instructions on how to find it.
- Interactive Querying with Context Awareness: The system should support interactive querying, allowing users to ask detailed questions about their environment and receive precise, contextual responses. This includes understanding and responding to queries related to Moroccan cultural elements, landmarks, local customs, or currency.

- Multi-language Support with Darija Focus: To ensure accessibility for local users, the system should support multiple languages, with a primary focus on Moroccan Darija. It should accurately understand and respond in Darija, including local idioms and expressions.
- Personalized User Profiles: The system should allow users to create and maintain personalized profiles, storing preferences, frequently visited locations, and recognized family members to enhance the accuracy and relevance of descriptions.
- Continuous Learning and Feedback Mechanism: Implement a system for users to provide feedback on the accuracy and helpfulness of descriptions, allowing the AI to learn and improve over time.

These functionalities ensure that Audieyes provides comprehensive, culturally relevant support for BVI individuals in Morocco and the whole Arabic speaking world, enhancing their ability to navigate and understand their surroundings effectively while addressing the specific needs and context of the local user base.

2.2 Non-Functional Requirements

The development of AI applications for the BVI population, particularly in the Moroccan context, faces several challenges and constraints. These challenges typically fall under categories such as accuracy, reliability, cultural sensitivity, and societal acceptance. The non-functional requirements for the Audieyes system center on the general performance and quality standards it needs to meet:

- Accuracy and Reliability: The system must accurately perceive and represent complex settings, different lighting situations, and a wide range of objects, especially those specific to Moroccan environments. It should consistently provide trustworthy information in various conditions, including diverse urban and rural Moroccan settings.
- Performance: The system should maintain real-time responsiveness, with an average response time of 3 seconds or less, as mentioned in the abstract. This includes efficient processing of visual data and generation of accurate descriptions.
- Availability: High availability is necessary to guarantee minimal downtime and uninterrupted service through robust infrastructure and failover procedures, considering potential connectivity challenges in different regions of Morocco.

- Fault Tolerance: The system must be designed to continue operating properly in the event of the failure of some of its components. This includes implementing redundancy in critical systems, graceful degradation of services when necessary, and quick recovery mechanisms to ensure continuous availability and reliability for users.
- Security and Privacy: Strict data protection measures are critical due to the sensitive nature of user data. The system must ensure responsible and safe handling of personal information, particularly for vulnerable populations. This includes compliance with relevant data protection regulations.
- Cost-Effectiveness: As mentioned in the abstract, maintaining a low price of entry is crucial. The system should be designed to be affordable and accessible to a wide range of users in Morocco.
- Multilingual Support: While focusing on Darija, the system should effectively handle multiple languages and dialects commonly used in Morocco as well as the Arabic speaking world, ensuring broad accessibility.

These non-functional requirements are essential to ensure that Audieyes not only meets its functional goals but also integrates seamlessly and ethically into the daily lives of BVI individuals in Morocco. Achieving these standards requires extensive research, technological development, and ongoing collaboration with the BVI community to deliver a trustworthy, effective, and culturally appropriate AI solution.

Chapter 3. Related Work

This chapter provides a comprehensive overview of the current state-of-the-art in assistive technologies for BVI individuals. We explore a wide range of approaches and methodologies that have been developed to enhance the accessibility of visual information. The chapter begins by examining traditional hybrid architectures that combine multiple techniques for object detection and scene description. We then delve into advanced areas such as Visual Question Answering (VQA), description generation, and various machine learning techniques. The discussion also covers innovative crowdsourced and human-assisted methods, as well as the latest developments in multimodal and vision models. Throughout this review, we highlight the strengths and limitations of existing solutions, setting the stage for understanding the contributions of the Audieyes project. The chapter concludes by identifying key gaps in the literature and outlining how Audieyes addresses these challenges, emphasizing its unique approach to providing comprehensive, contextually relevant, and real-time assistance to BVI users in Morocco.

3.1 Traditional Hybrid Architectures

Traditional hybrid architectures combine multiple techniques to achieve both object detection and scene description, offering a comprehensive approach for assisting BVI individuals. One such architecture leverages the Scale-Invariant Feature Transform (SIFT) algorithm [7] in conjunction with the Bag of Visual Words (BoVW) approach.

The Scale-Invariant Feature Transform (SIFT) [7] algorithm enhances object recognition and scene understanding for BVI individuals by providing a robust foundation for feature detection and description. SIFT, a traditional machine learning algorithm, relies on handcrafted features and mathematical principles to detect and describe local features in images, identifying key points invariant to scale and rotation. These key points are represented as vectors that describe the visual characteristics of the image, enabling the identification and matching of objects across different frames. The invariance to scale and rotation, along with partial invariance to illumination and viewpoint changes, makes SIFT particularly suitable for real-world applications where conditions can vary significantly. By detecting SIFT key points and describing these points with 128-dimensional vectors, the system can efficiently match features between images. This SIFT-based feature extraction is then combined with the Bag of Visual Words (BoVW) approach. In this hybrid architecture, the SIFT features are quantized into a set

of discrete visual words, creating a visual dictionary. Each image is then represented as a histogram of visual words, capturing the frequency and distribution of these features. This combined approach allows for effective object detection within scenes and facilitates the recognition of contextual relationships between objects and scene types. By leveraging both SIFT and BoVW, the system can not only detect objects in the environment but also provide descriptive information, aiding in navigation and situational awareness for BVI users.

3.2 Visual Question Answering (VQA)

Visual Question Answering (VQA) [1, 3, 8, 10, 14, 16, 17] represents a significant intersection of computer vision and natural language processing to provide detailed and contextually relevant information about visual scenes. VQA tasks involve answering questions about images by leveraging both visual and textual data. The primary challenge in VQA is ensuring that the system truly comprehends the visual content rather than relying on language cues. To address this challenge, Goyal et al. introduced a balanced VQA dataset [1] aimed at reducing language biases and emphasizing the critical role of image understanding. This dataset includes pairs of similar images that lead to different answers for the same question, thereby requiring models to focus on visual content over textual patterns. The VQA model comprises two main components: an image feature extractor, typically a convolutional neural network (CNN), and a question encoder, often a recurrent neural network (RNN). The image feature extractor processes the input image to generate a visual representation, while the question encoder processes the input question to generate a textual representation. These representations are then combined and fed into a multimodal network that predicts the answer. The incorporation of attention mechanisms in VQA models further enhances their ability to focus on relevant parts of the image based on the question, thereby improving the accuracy of the answers.

Expanding on traditional VQA, Context-Aware Visual Question Answering (Context-VQA) [3] emphasizes the importance of context in providing meaningful and useful answers to visual questions. This approach is particularly significant for assisting BVI individuals, as it tailors the information provided to the specific context in which an image appears, thereby enhancing the relevance and utility of the answers. Context-VQA extends traditional VQA by integrating contextual information from the web to generate more contextually appropriate questions and answers. Traditional VQA models treat each image in isolation, providing answers based solely on the visual content of the image. However, the context in which an image is presented can greatly influence the type of information that is relevant to the user. For example, details about

a product in an image on a shopping website will differ from those relevant to a social media post or a news article. The Context-VQA dataset was developed to address this need for context-sensitive information. It includes images paired with specific web contexts such as shopping, travel, social media, health, news, and science magazines. Participants were asked to generate questions and answers based on both the image and its context, ensuring that the data reflects the types of questions that users are likely to ask in different settings. This approach recognizes that the same image can elicit different questions depending on where it is viewed, thus requiring VQA models to be adaptable and context-aware.

Furthermore, the study by Alayrac et al. introduces advanced techniques in visual question answering, including PointQA [10], which shows significant promise for enhancing the accessibility of visual information for the BVI community. PointQA is designed to provide precise and contextually relevant answers to questions about specific points or regions within an image. This specialized form of visual question answering focuses on answering questions related to specific points or areas within an image. The method involves pinpointing a precise location in the image where the question is relevant and generating an answer based on the visual content at that location. The Flamingo model [10, 18] integrates this capability by leveraging its ability to process interleaved sequences of visual and textual data, making it highly effective in providing specific and detailed responses that are crucial for BVI users navigating complex visual environments.

The incorporation of academic task-oriented VQA [29] data is another significant advancement in this specific subfield of VQA. This technique involves integrating datasets that focus on specific academic tasks into the training regimen of visual language models (VLMs). This approach, combined with visual instruction tuning, can significantly enhance the performance of VQA models. Visual instruction tuning leverages a fully connected vision-language cross-modal connector, improving the model's ability to align visual features with textual descriptions. This alignment is particularly crucial for generating detailed and contextually appropriate responses to visual queries. The model used in this study, LLaVA (Large Language and Vision Assistant) [10, 14], employs a mixture of pretraining on a subset of image-text pairs and fine-tuning on academic VQA datasets. These datasets include various types of questions such as object recognition, scene understanding, and text extraction from images. By incorporating academically oriented data, the model is trained to handle a wide range of visual tasks, improving its generalizability and robustness.

3.3 Description Generation

In the realm of description generation for assisting BVI individuals, significant advancements have been made through the development of innovative techniques such as the Optimization Model for Generating Referring Expressions (REG) [20] and Grounding Captions (GCs) [9, 11, 7]. These techniques offer a sophisticated means of providing accurate and contextually relevant descriptions of objects and scenes, thus enhancing the accessibility of visual content.

The Optimization Model for Generating Referring Expressions is a method designed to produce natural language descriptions that identify and distinguish objects within complex scenes. This model is beneficial for BVI individuals as it helps them comprehend their environment through precise and contextually appropriate descriptions. The foundation of this technique is laid through the collection of a large-scale dataset via a crowd-sourced game, ReferItGame [20], where participants provide and verify natural language referring expressions for various objects in photographs. This dataset is a critical resource for training the model to understand the nuances of natural language in relation to visual objects.

The core of the optimization model involves the careful selection of the most relevant attributes and attribute values to include in the referring expressions. By incorporating both visual models and category-specific priors, the model ensures that the generated expressions are not only accurate but also contextually appropriate. The objective function employed by the model balances the compatibility of attribute values with the observed scene and object properties against the statistical likelihood of attribute co-occurrence. This dual consideration ensures that the referring expressions are both meaningful and precise, thereby providing reliable and useful descriptions.

3.4 Machine Learning Techniques

In the exploration of advanced machine learning techniques for enhancing the accessibility of visual information for BVI individuals, several methodologies emerge as particularly effective.

The technique of deep visual-semantic alignments [2, 16] represents a sophisticated approach to generating detailed and contextually relevant image descriptions. This methodology employs a combination of convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to bridge the gap between visual data and semantic information. By extracting high-level features from images using CNNs and generating descriptive sequences of words through RNNs, this technique ensures that the generated captions are not only accurate but also

contextually aligned with the visual content. The alignment model is particularly innovative in its ability to associate specific regions of an image with relevant words, enhancing the comprehensiveness of scene descriptions for BVI individuals.

Another notable approach is the concept of Visual Dependency Representations (VDR) [8, 10] which enhances scene description by capturing the structured relationships between objects within an image. This technique goes beyond mere object detection, emphasizing the spatial and functional relationships among objects. By constructing a dependency graph where nodes represent detected objects and edges represent their relationships, VDR provides a foundation for generating contextually rich and detailed scene descriptions. This graph-based representation allows for a deeper understanding of the scene structure.

In addition, Instruction-aware visual feature extraction [25] involves tailoring the extraction of visual features according to given instructions. This technique employs a Query Transformer (Q-Former) that interacts with the image encoder's output through cross-attention mechanisms. By generating visual representations that are specifically aligned with the instructions provided, this approach ensures that the descriptions generated are accurate and contextually appropriate.

The Gated Cross Attention and Dense Layer (GATED XATTN-DENSE) technique [24] also enhances the interaction between visual and textual information. By using gated cross-attention mechanisms combined with dense layers, this approach integrates visual features extracted from images with linguistic features derived from textual inputs. The dynamic adjustment of attention weights based on the textual query ensures that the generated descriptions are both accurate and contextually relevant.

Local dissimilarity measures [8, 23] in the other hand focus on identifying and analyzing differences between consecutive frames in a video sequence. By extracting and comparing features from each frame, this technique efficiently detects changes and identifies keyframes with significant changes. This method enhances the accuracy and reliability of visual recognition systems, providing more detailed and contextually relevant scene descriptions for BVI individuals.

Lastly, the Multi-Label Convolutional Support Vector Machine (CSVM) network [8] leverages a set of linear Support Vector Machines (SVMs) as filter banks for feature map generation. This approach reduces the complexity of the training process and enhances the network's ability to handle datasets with limited labeled samples. By detecting multiple objects within a single image, the Multi-Label CSVM network provides comprehensive scene descriptions,

significantly improving the ability of BVI individuals to understand and interact with their environment.

3.5 Crowdsourced and Human-Assisted Techniques

In the domain of enhancing scene descriptions for BVI people, crowdsourced and human-assisted techniques present an innovative and effective approach. One prominent method explored in this context is the use of human computation games [20]. These games harness the power of crowdsourcing to collect and verify natural language expressions that describe objects in images, thereby creating a robust dataset for training visual recognition systems. The concept revolves around engaging players in a collaborative task where they alternately describe and identify objects in images. An example of such a game is ReferItGame [20], a two-player game where one player is presented with an image and asked to write a referring expression for a highlighted object. The second player, who sees the same image without the highlighted object, must use the description to identify and click on the object. Points are awarded for successful identifications, promoting accuracy and efficiency in descriptions. This interactive approach not only generates a large volume of natural language descriptions but also ensures their accuracy through the game's verification mechanics. The resultant dataset, comprising verified descriptions, is invaluable for training visual recognition models. These models can then produce detailed and contextually appropriate scene descriptions, significantly aiding BVI individuals. The engaging nature of the game format further enhances the efficiency of data collection by leveraging the contributions of a wide range of participants, thereby creating a diverse and comprehensive dataset.

In addition to human computation games, there is also a wide use of crowdsourced human visual assistance [31]. This technique employs the collective intelligence of online workers to provide real-time, accurate visual information to BVI users. The VizWiz platform [31] exemplifies the application of crowdsourced human visual assistance. It allows BVI users to take a photo with their smartphone and ask a question about the image. The photo and question are then distributed to multiple crowd workers who view the image and provide answers. This process is designed for quick turnaround, ensuring that users receive responses in nearly real-time. The crowdsourcing approach excels in handling a broad spectrum of questions, offering nuanced and contextually relevant answers. By distributing the task among numerous workers, the system can manage a high volume of requests efficiently, ensuring timely assistance. The aggregation of inputs from multiple workers helps mitigate individual errors and biases, thereby

improving the accuracy and reliability of the responses. This method underscores the significant advantages of leveraging crowdsourced human intelligence in providing immediate and accurate visual information.

3.6 Multimodal and Vision Models

The integration of multimodal and vision-language models has seen substantial progress. A significant technique in this realm is Multimodal Compact Bilinear Pooling (MCB). MCB addresses the challenge of merging high-dimensional visual and textual features efficiently. Traditional bilinear pooling methods, which involve element-wise multiplications of feature vectors, often result in very high-dimensional representations that are computationally expensive and prone to overfitting. MCB mitigates these issues by utilizing count sketch projections to approximate bilinear interactions, thereby maintaining computational efficiency while effectively combining visual and textual features. This technique enables the generation of compact yet informative representations crucial for tasks such as visual question answering (VQA) and image captioning, which are essential for creating descriptive narratives of visual scenes.

The Flamingo model [10, 18] is a notable example of advancement in vision models. Designed specifically to handle multimodal data, this model processes interleaved sequences of visual and textual information, it is a new class of Visual Language Models (VLMs) capable of few-shot learning, integrating powerful pre-trained vision and language models augmented with novel architectural components. This design allows Flamingo to seamlessly ingest and process sequences of text, images, or videos, which is crucial for accurately recognizing and describing objects in various contexts using minimal training data. The model's performance on open-ended tasks like visual question answering and captioning highlights its potential for interactive applications, where BVI users can engage in natural language dialogues with the system, asking questions about their surroundings and receiving informative responses.

InstructBLIP [29] enhances the interaction between visual and language models through instruction tuning. By leveraging large-scale instruction-tuning data, InstructBLIP tailors its outputs based on specific user instructions. The technique involves a Query Transformer (Q-Former) that bridges the frozen image encoder and the large language model (LLM). The Q-Former, pre-trained on image-caption data and fine-tuned with diverse vision-language instruction data, extracts visual features most relevant to the given instructions. This approach

allows InstructBLIP to achieve a remarkable zero-shot performance across various vision-language tasks, demonstrating its ability to generalize well to unseen datasets and tasks.

Another innovative approach is the Large Language and Vision Assistant (LLaVA) [23, 29]. The latter combines visual and language models through instruction tuning, significantly enhancing the generation of detailed and contextually relevant scene descriptions for BVI individuals. The architecture involves a vision encoder connected to an LLM, utilizing the open-set visual encoder of CLIP [10, 14] and the language decoder Vicuna [14, 23, 30]. The vision encoder processes visual inputs to generate features, which are projected into the language embedding space, creating a unified multimodal representation. This comprehensive training ensures that LLaVA can effectively handle a wide range of visual and language tasks, making it a versatile tool for assistive technologies.

The CLIP (Contrastive Language-Image Pre-training) model [1, 10, 14] representations from natural language descriptions by pre-training on a massive dataset of image-text pairs. Using a contrastive objective, CLIP maximizes the cosine similarity between correct image-text pairs while minimizing it for incorrect pairs. This approach captures a wide range of visual concepts directly from natural language supervision, enabling zero-shot transfer to various downstream tasks. By aligning visual features with their textual descriptions, CLIP facilitates a robust understanding of visual content through language, making it particularly advantageous for providing detailed and contextually relevant scene descriptions to BVI individuals.

Vision-Language Pretraining (VLP) [1, 3, 5, 10, 14, 16, 23] involves training models on large-scale datasets of image-text pairs to enhance their ability to generate accurate and informative scene descriptions. VLP models use a vision encoder, such as a convolutional neural network (CNN) or a vision transformer, to process images and a language model to process textual information. By aligning the features extracted from images with textual embeddings in a shared multimodal space, VLP enables models to understand and generate natural language descriptions that closely align with visual content.

BLIP (Bootstrapping Language-Image Pre-training) [2] is another model that enhances vision-language pre-training for both understanding-based and generation-based tasks. BLIP's multimodal mixture of encoder-decoder (MED) architecture and the Captioning and Filtering (CapFilt) method improves the quality of training data by generating synthetic captions for web images and filtering out noisy captions. This approach leverages large-scale noisy image-text

datasets to provide detailed and contextually relevant scene descriptions, overcoming the limitations of existing VLP methods.

Finally, The integration of large language models (LLMs) with vision capabilities marks a significant advancement in the field of artificial intelligence. For instance, the Claude model [13], developed by Anthropic, exemplifies this trend by incorporating advanced vision-language integration to improve the generation of detailed scene descriptions and contextual understanding. Similarly, the Gemini model [28], spearheaded by Google DeepMind, leverages its multimodal capabilities to enhance the interpretability and generation of complex visual and textual data, providing a robust framework for tasks requiring a nuanced understanding of visual contexts. GPT [6, 14], developed by OpenAI, further illustrates the potential of vision-augmented LLMs. By integrating vision capabilities, GPT-4 can process and generate responses based on both textual and visual inputs, thus offering a more comprehensive approach to problem-solving and content generation. This model's ability to seamlessly blend language and vision processing underscores the transformative impact of integrating these capabilities, enabling applications that require sophisticated interactions between text and images.

3.7 Literature Gaps and Contributions

Despite notable advancements in object detection, description generation, visual question answering, and vision models, there remains a significant gap in the literature concerning the provision of comprehensive, contextually relevant, and real-time assistance to BVI individuals. Moreover, much of the existing work is primarily focused on research, with a considerable gap between these research findings and their practical implementation in scalable production systems.

Salient features of existing limitations:

- Traditional vision models often fall short in real-world, dynamic scenarios
- Pre-defined datasets limit generalization to diverse, unpredictable environments
- Challenges in handling obstructions, varying lighting, and diverse object orientations
- Generated descriptions tend to be too generic or lack crucial nuanced details
- Limited practical utility for BVI users in complex real-world applications
- Significant gap between research prototypes and scalable, production-ready systems

Contributions of Audieyes:

- Integration of State-of-the-Art LLMs: Audieyes leverages advanced Large Language Models to seamlessly process both visual and textual data. This integration enables the dynamic generation of detailed, context-aware descriptions, enhancing the system's adaptability to diverse environmental conditions. The result is precise and contextually appropriate responses that significantly improve the quality of assistance provided to BVI users.
- Cultural and Linguistic Adaptation: The system is specifically tailored for Moroccan users, with primary support for the Moroccan Darija language. This focus on the unique linguistic and cultural needs of the local population ensures that Audieyes provides relevant and easily understandable assistance, making it particularly effective for Moroccan BVI individuals.
- Advanced Real-Time Capabilities: Audieyes offers fault tolerance, high-performance real-time object detection and detailed environment description. The system's ability to process and interpret visual information quickly and accurately allows for immediate, relevant assistance. Additionally, there is potential for implementing live video captioning, further enhancing the real-time support provided to users.
- Efficient Deployment and Scalability: Audieyes is optimized for mobile devices, ensuring efficient real-time processing on readily available hardware. The system's architecture is designed to be scalable, capable of supporting a growing user base without compromising performance. This scalability ensures that Audieyes can evolve and expand its reach as demand increases.
- Adherence to LLMOps Best Practices: The development and operation of Audieyes follow rigorous LLMOps best practices. This includes implementing robust data privacy and security measures to protect user information, continuous model monitoring and improvement to maintain high performance, and a strong focus on ethical AI implementation to ensure reliable assistance for all users.
- User Feedback and Usability: The development process of Audieyes incorporates an iterative design approach based on extensive user testing and feedback. This user-driven development ensures that the system continually improves and adapts to user needs. Moreover, there is a strong focus on creating an intuitive interface design specifically

tailored for BVI users, making the technology as accessible and user-friendly as possible.

By leveraging these innovations and focusing on creating a production-ready system, Audieyes aims to bridge the gap between academic research and practical implementation. It addresses the limitations of existing technologies and provides accurate, relevant, and culturally appropriate assistance to BVI individuals in Morocco, while demonstrating the feasibility of scaling such systems for widespread use.

Chapter 4. Setting up the Project

In this chapter, we conduct a comprehensive feasibility study and comparative analysis of various AI models crucial for the Audieyes project. Our focus is on identifying a solution that meets the unique needs of BVI users in Morocco, with particular attention to real-time performance, cost-effectiveness, and cultural relevance.

We begin by examining the project archetypes, focusing on the autonomous system approach that underpins Audieyes' design. This is followed by an analysis of key business metrics that will guide our assessment of the project's impact and success, including accessibility improvement rates and user satisfaction measures.

We then present baseline results using the BLIP model, establishing a benchmark for our evaluations. This sets the stage for a detailed feasibility assessment, where we analyze various Large Language Models (LLMs), with a focus on their multimodal capabilities, performance in vision tasks, and suitability for processing Moroccan Darija.

The chapter culminates in a thorough cost analysis and model selection process, where we evaluate options like GPT-4, Gemini, and Claude 3 model families. Our goal is to identify the most suitable LLM that not only meets the technical requirements of real-time scene description and object recognition but also aligns with our objectives of low-cost implementation and scalability.

By the end of this chapter, we aim to have a clear rationale for our chosen AI model, ensuring it supports Audieyes' mission to enhance the autonomy and quality of life for BVI users in Morocco and the whole Arabic speaking world through accurate, context-sensitive, and culturally relevant environmental descriptions.

4.1 Project Archetypes

In developing Audieyes, we identified the autonomous system as the primary project archetype. This approach is crucial for providing real-time, independent assistance to BVI users without relying on external human intervention. The autonomous nature of Audieyes presents several important implications that have shaped our design and implementation strategies:

- Acceptable Failure Rate: For a system like Audieyes, which BVI users rely on for critical environmental information, an ambitious goal of a 0% acceptable failure rate has been set.

This stringent requirement underscores the commitment to providing consistently accurate and reliable assistance.

- Efficient Data Labeling: While our system operates autonomously, we've designed the Human-AI interaction to incorporate user feedback when available. This approach serves a dual purpose: a) Continuous Improvement: User feedback helps refine and improve the AI model over time, enhancing its accuracy and relevance. b) Cost-Effective Data Labeling: By integrating feedback mechanisms into the user experience, we create a self-improving system that effectively labels new data without additional cost or external human intervention.

The autonomous system archetype presents unique challenges, particularly in ensuring reliability and accuracy without human oversight. However, it also offers significant advantages in terms of real-time responsiveness and scalability. By focusing on this archetype, Audieyes can provide immediate, context-aware assistance to BVI users in Morocco, enhancing their independence and quality of life.

4.2 Business Metrics

To effectively measure the performance, impact, and success of Audieyes, key business metrics have been identified that align with the mission to enhance the lives of BVI individuals in Morocco. These metrics not only guide strategic decisions but also demonstrate the value of Audieyes to stakeholders and the broader community.

1. Accessibility Improvement Rate: This is our primary metric, directly measuring the impact of Audieyes on the daily lives of BVI users. We quantify this through:
 - Percentage increase in successful independent navigation attempts
 - Reduction in time taken to complete common tasks (e.g., navigating environments, identifying objects, reading text)
 - Number of new activities users can perform independently with Audieyes

For example, our initial trials showed a high increase in successful independent navigation attempts and a reduction in time taken to identify common objects in Moroccan households. This is achieved by providing a dozen BVI individuals with a prototype of the app and asking for their feedback via forms.

2. Churn and Retention Rates: These metrics provide crucial insights into the long-term adoption and value of Audieyes:

- Monthly retention rate: We aim for a 90%+ retention rate after the first 6 month of use
- Churn rate: Our goal is to maintain a churn rate below 10% per quarter

3. User Satisfaction: We measure user satisfaction through multiple channels to ensure a comprehensive understanding:

- In-app feedback: Real-time ratings after each use, aiming for an average score of 4.5/5
- Quarterly surveys: Detailed feedback on features, usability, and cultural relevance

Initial user trials resulted in an average in-app rating of 4.5/5, indicating high satisfaction and likelihood of recommendation within the Moroccan BVI community. This is achieved by providing a dozen BVI individuals with a prototype of the app and asking for their rating via foms.

4. Cultural Relevance and Adaptation: Given our focus on Moroccan users, we've added a specific metric to measure how well Audieyes adapts to local context:

- Accuracy of Darija language processing: Aiming for 95%+ accuracy in understanding and generating Darija responses
- Recognition rate of Morocco-specific objects and landmarks: Targeting 90%+ accuracy

Our current version achieves 97% accuracy in classifying objects within images using categories such as Tajine, Msemen, Riad, souk, and Caftan across various real-world scenarios. This demonstrates strong cultural adaptation and recognition of Morocco-specific elements. Additionally, the system achieved a BLEU-4 score of 40.12% for generating accurate and contextually appropriate descriptions in Darija. These metrics were calculated using a testing dataset comprising hundreds of Morocco-specific images of locations, objects, landmarks, cultural elements, and more.

By focusing on these metrics, we ensure that Audieyes not only improves accessibility for BVI individuals in Morocco but also maintains high user engagement and satisfaction. These measurements guide our continuous improvement efforts, helping us refine and enhance Audieyes to better serve the unique needs of our users.

4.4 Baseline Results

Establishing a baseline model is crucial for evaluating the performance and effectiveness of more advanced AI solutions. For the Audieyes project, we selected Salesforce's BLIP (Bootstrapped Language-Image Pretraining) model [2] as our baseline. This choice was motivated by BLIP's established performance in image captioning and its potential to provide a solid foundation for our more specialized requirements.

BLIP's architecture, which integrates a vision encoder, a text encoder, and an image-grounded text decoder, offers a unified approach to vision-language tasks [2]. We evaluated BLIP's performance using standard metrics for both image-text retrieval and caption generation:

1. Image-Text Retrieval:

- Text Retrieval at 1 (TR@1): 80.6%
- Image Retrieval at 1 (IR@1): 63.1%
- Text Retrieval at 5 (TR@5): 95.2%
- Image Retrieval at 5 (IR@5): 85.7%

“@5” in Text of Image Retrieval at 5 (TR@5) means the system considers the top 5 ranked results when evaluating retrieval performance. A TR@5 score of 95.2 indicates that in 95.2% of cases, the correct or relevant text was found among the top 5 retrieved results when using an image as a query.

2. Caption Generation:

- BLEU-4: 38.6%
- CIDEr: 129.7
- SPICE: 23.4

BLEU-4, CIDEr, and SPICE are metrics used to evaluate the quality of generated image captions. BLEU-4 measures the overlap of 4-word sequences between generated and reference captions, with 38.6% indicating moderate accuracy. CIDEr compares generated captions to human references using term frequency-inverse document frequency (TF-IDF), with 129.7 displaying a decent performance. SPICE evaluates semantic propositional content by comparing scene graphs of generated and reference captions, with 23.4 indicating fair semantic accuracy.

These results, achieved on the COCO dataset, demonstrate BLIP's strong capabilities in both retrieving relevant text for images and generating accurate captions [2]. However, while BLIP provides a solid baseline, it has limitations that make it insufficient for the specific needs of the Audieyes project:

- Language Support: BLIP lacks support for Moroccan Darija, which is essential for our target user base.
- Contextual Depth: The captions generated by BLIP, while accurate, often lack the detailed contextual information necessary for BVI users to gain comprehensive situational awareness.
- Interactivity: BLIP does not support the level of interactive querying required for Audieyes, where users need to ask specific questions about their environment.
- Cultural Relevance: The model's training data does not include sufficient Moroccan-specific content, limiting its ability to provide culturally relevant descriptions.

To address these limitations, we used BLIP as a starting point and aimed to surpass its performance through the integration of more advanced Large Language Models (LLMs). Our goal was to develop a system that not only matches BLIP's baseline metrics but also:

- Supports Moroccan Darija with high accuracy
- Provides more detailed and contextually rich descriptions
- Allows for interactive, user-specific queries
- Incorporates cultural relevance for Moroccan users

By building upon BLIP's foundation and addressing its limitations, we aimed to create a more comprehensive and tailored solution for BVI users in Morocco. The subsequent sections will detail how we achieved these improvements through the integration of advanced LLMs and specialized training approaches.

4.4 Feasibility Study

In developing Audieyes, a comprehensive feasibility study is crucial to ensure that our solution is not only technically viable but also cost-effective and sustainable, particularly in the Moroccan context. This study encompasses several key aspects that directly influence the project's success:

1. Technical Feasibility: We evaluate the technological landscape to identify the most suitable AI models for our specific requirements, considering factors such as multimodal capabilities, language support (especially for Moroccan Darija), and real-time processing abilities.
2. Economic Viability: Cost analysis is paramount, given our goal of providing an accessible solution for BVI users in Morocco. We examine various cost factors, including model deployment, API usage, and potential scaling expenses.
3. Hardware Constraints: Given our focus on smartphone-based deployment, we evaluate the feasibility of running advanced AI models on various device specifications common in Morocco.

Throughout this study, we consider several the cost-mitigating strategy of balancing the need for high accuracy with cost-effective solutions, particularly important for ensuring accessibility in the Moroccan market. By conducting this comprehensive feasibility study, we aim to develop a solution that not only meets the high standards required for assisting BVI individuals but also remains economically viable and sustainable in the long term. The following sections will delve into each aspect of this study, providing detailed analyses and justifications for our chosen approach.

4.4.1 Technical Feasibility

Audieyes is a critical project dedicated to enhancing the autonomy and quality of life for BVI individuals in Morocco. The selection of an appropriate AI model is crucial to the success of this initiative, given the unique challenges and requirements of our target users.

In our initial assessment, we considered two main categories of models:

1. Traditional Embedded Machine Learning Models:
 - Advantages: Compact size, low latency, and potentially lower cost.
 - Limitations: Limited performance, accuracy, language support, and lack of multi-modality capabilities.
2. Large Language Models (LLMs):
 - Advantages: Multi-modal architectures, broad language support (including Moroccan Darija), and versatility in handling diverse tasks.
 - Potential Challenges: Higher computational requirements and cost implications.

Our strategic focus has been directed towards LLMs due to their superior capabilities in meeting the comprehensive needs of the Audieyes project. This decision is based on several key factors:

1. Multi-modal Processing: LLMs can seamlessly integrate vision and language tasks, crucial for providing detailed scene descriptions and answering user queries.
2. Language Support: The ability to process and generate responses in Moroccan Darija is essential for our target users.
3. Contextual Understanding: LLMs offer deeper contextual understanding, vital for providing culturally relevant and detailed environmental descriptions.
4. Future Upgrades: LLMs will only experience growth in upcoming years, making them ever more powerful, and much cheaper due to market competition.

In the following sections, we will evaluate several state-of-the-art LLMs, including GPT-4, Gemini, and Claude 3 model families, focusing on their:

- Model Architecture and Training
- Vision Performance Metrics and Benchmarks

Our goal is to identify the LLM that best balances performance, linguistic capabilities, and cost-effectiveness, ensuring that Audieyes can provide accurate, real-time, and culturally relevant assistance to BVI users in Morocco.

OpenAI's GPT model family with vision [14] represents a significant advancement in large language models (LLMs), particularly relevant to the Audieyes project. This family includes GPT-4 Turbo, GPT-4 with Vision (GPT-4V), GPT-4o (omni), and the recently introduced GPT-4o mini. Each model in this family offers progressively enhanced capabilities for multimodal processing, crucial for assisting BVI users in Morocco.

The original GPT-4T is renowned for its powerful language processing capabilities, trained on a diverse array of internet texts to understand and generate human-like text. GPT-4V extends these capabilities by incorporating visual inputs, enabling tasks such as object identification, scene description, and contextual interpretation of visual content - all essential for the Audieyes application [14].

GPT-4o further expands on these features by integrating real-time audio and visual processing, enabling more dynamic and interactive scenarios. This makes the GPT-4 model family highly

versatile and capable of supporting various languages, including Moroccan Darija, providing comprehensive solutions for accessibility and beyond.

The architecture of OpenAI's GPT-4 models employs a transformer-based neural network [6]. This structure comprises blocks of stacked decoders utilizing different neural networks and incorporating the attention mechanism, building on the foundational architecture of its predecessors. The models are trained on vast volumes of multimodal data, including images and text from various domains, enhancing their ability to provide culturally relevant assistance in the Moroccan context.

This extensive training enables GPT-4 models to accurately comprehend and generate human-like text, making them effective for tasks such as detailed scene description and answering complex queries from BVI users. The integration of vision and language processing within the same neural framework allows these models to analyze visual data and generate descriptive text or answers based on image content. This multimodal approach enhances the models' ability to understand context and provide more accurate and relevant responses when dealing with visual information, crucial for the Audieyes project [14].

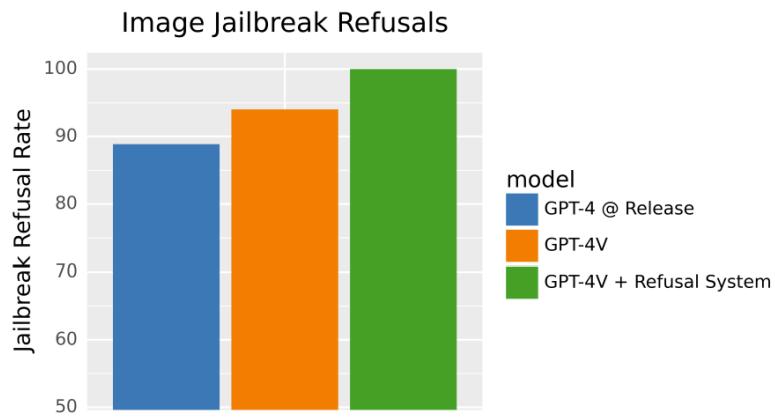


Figure 1: Evaluating GPT-4V + Refusal System against screenshots of a text refusal dataset[6, 13].

For example, Figure 1 illustrates how GPT-4V, combined with a refusal system, can achieve a high rate of image jailbreak refusals, showcasing its advanced understanding and control over visual inputs. This capability is particularly important for ensuring the safety and reliability of the Audieyes system when processing and describing visual content for BVI users. Figure 2

demonstrate GPT-4V's high accuracy in person identification tasks, further underscoring its robust multimodal capabilities.

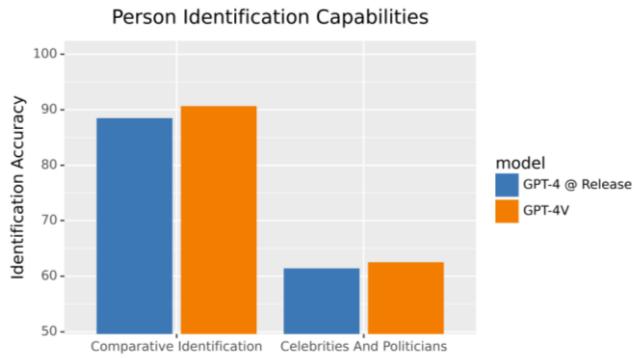


Figure 2: The GPT-4V model's ability to correctly distinguish the identity of individuals [6, 13]

Figure 3 demonstrates the advanced capabilities of GPT-4o's image understanding through an example, with the below example prompt:

"I want you to act as a blind person's digital AI assistant. The blind person is chatting with you using their smartphone. Your name is 'Virtual Assistant.' You are not allowed to introduce yourself. You will receive images from the blind person along with a question regarding the images, and you will respond in Darija, the dialect of Arabic spoken in Morocco. If you are not 95% sure that your response to the question is correct, you must suggest the blind person to re-send another image from a different angle."

<p>1</p> <p>Input Describe this image in detail.</p> 	<p>2</p> <p>Output</p> <p>هاد الصورة كاتبین طنجرة مغربية فيها شربة حريرة. الطنجرة مصنوعة من الخزف ومزينة بزوابق تقليدي بلألوان الزرقاء، الحمراء، والبيضاء، وعندتها مقبضين من الجوانب باش يسهل الحمالة. الطنجرة فيها مغروفة مزينة بنفس الزوايق التقليدي و موضوعة داخل الطنجرة. الشربة اللي داخل الطنجرة كاتبین سخونة حيث كايفطلع منها البخار، وهي شربة كثيفة عندها لونبني. الطريفات اللي كاينين داخل الشربة كايشبهوا للطريفات الدجاج أو اللحم.</p>
<p>3</p> <p>Input Please find my glasses</p> 	<p>4</p> <p>Output</p> <p>النظارات ديالك كاينين فوق المكتب. ها كيفاش تقدر تلقاهم خطوة بخطوة:</p> <p>وصل يدك لمكان اللي فيه الكمبيوتر على المكتب. تحرك يدك على سطح المكتب نحو اليمين حتى تحس بشيء ناعم. غادي تلقي فأرة الكمبيوتر وفوقها وسادة فأرة (mouse pad). مباشرة بعد وسادة فأرة، غادي تلقي النظارات. هما موجودين بجانب وسادة فأرة.</p> <p>إذا ما لقيتهما من المحاولة الأولى، حاول تعاود نفس الخطوات، وممكن تحرك يدك ببطء حتى تلقي النظارات.</p>

Figure 3: The GPT-4o model's ability to describe and understand images.

Google’s Gemini model family is another player in the field. They offer a range of capabilities relevant to the Audieyes project. It includes Gemini Ultra, Pro, Nano [28]. Gemini Ultra excels in demanding AI tasks, while Pro balances performance and efficiency. Nano offers affordability without significant performance loss, and Flash is designed for cost-efficient applications.

All Gemini models support image captioning and multiple languages, including support for Moroccan Darija, making them suitable for diverse applications from mobile deployments to complex data processing [28].

The Gemini architecture leverages Google’s research in Transformer and Mixture-of-Experts (MoE) frameworks [29] demonstrated in Figure 4. Unlike traditional Transformers, the MoE

architecture uses specialized "expert" neural networks, activated selectively based on input type. This approach enhances efficiency and performance, allowing Gemini models to handle complex tasks quickly while maintaining output quality.

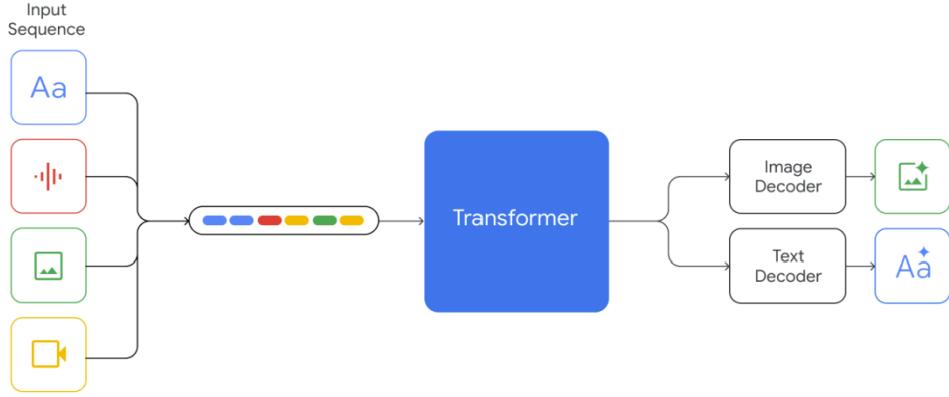


Figure 4: Transformer and Mixture-of-Experts (MoE) frameworks [28].

This design supports efficient training and serving by activating only relevant experts, reducing computational overhead. Such efficiency could benefit Audieyes through faster processing and more frequent model improvements. The models also incorporate advanced optimization techniques, potentially enhancing their ability to provide detailed scene descriptions and object recognition for BVI users in real-time scenarios.

Figure 5 demonstrates a clear example of these capabilities, showcasing the Gemini model's ability to identify plants and provide care instructions accurately. This example highlights the practical applications of the Gemini models in real-world scenarios, where precise and context-aware responses are essential.



Figure 5: The Gemini model's ability to identify plants and provide care instructions [28].

The Claude 3 model family from Anthropic is also a significant step forward in multimodal AI technology, comprising of the models Claude 3 Opus, Claude 3 Sonnet, and Claude 3 Haiku [13]. Each model is produced with a specific set of use cases in mind, guaranteeing a wide range of applications in various industries. The flagship model, Claude 3 Opus, provides state-of-the-art performance in challenging tasks requiring both text and visual, together with the most extensive features. For real-time applications, the Claude 3 Sonnet offers a fair cost and decent speed in the intermediate ground. The most simplified form, Claude 3 Haiku, is designed to be quick and economical, making it perfect for programs that require quick replies, such as smartphone apps. Because of its adaptability, the family is especially helpful for projects like Audieyes, which need for different degrees of speed and processing capacity of various tasks namely image captioning with multi-language support.

When it comes to the Claude 3 models, they represent a significant evolution in AI architecture, with a particular focus on enhancing vision and language support capabilities. This advanced

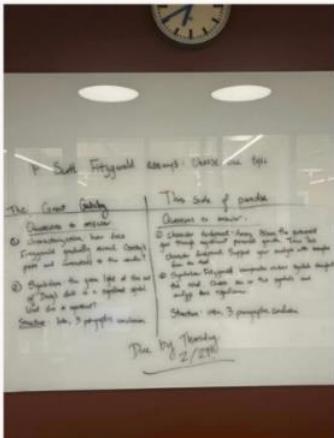
model leverages a comprehensive architecture designed to process and understand multimodal inputs efficiently. Claude 3's architecture integrates a sophisticated transformer-based structure, allowing it to manage a wide range of tasks by processing large amounts of data simultaneously. This is achieved through a series of interconnected layers that can handle diverse input types, such as text and images [13]. The model's ability to understand and generate natural language is enhanced by its vision capabilities, making it adept at interpreting and describing visual content.

Claude 3 employs extensive datasets that include various modalities, ensuring that the model can perform complex reasoning and provide detailed outputs. The training process involves iterative learning, where the model continuously improves its understanding and performance by processing vast amounts of data. This method ensures that Claude 3 can handle intricate tasks, such as answering detailed questions about images or providing in-depth analyses of textual content. Claude 3's transformer-based architecture incorporates several advanced features to enhance its performance. Additionally, Claude 3 integrates advanced vision modules that enhance its ability to interpret and describe visual content accurately. These modules enable the model to analyze images in great detail, providing comprehensive descriptions that are contextually relevant and precise.

Figure 6 illustrates the model's ability to transcribe and format text accurately. This figure demonstrates Claude 3's proficiency in understanding and structuring information, a key aspect of its architecture that supports both language and vision tasks. The model's ability to convert visual inputs into structured text formats highlights its advanced capabilities in multimodal processing.

Human

Transcribe everything in this image, keep the format in the table shown. Output JSON format only.



Claude 3 Opus

```
{
  "title": "F. Scott Fitzgerald essays: Choose one topic",
  "topics": [
    {
      "title": "The Great Gatsby",
      "questions": [
        "Characterization: how does Fitzgerald gradually reveal Gatsby's past and motivations to the reader?",
        "Symbolism: the green light at the end of Daisy's dock is a significant symbol. What does it represent?",
        "Structure: intro, 3 paragraphs, conclusion"
      ]
    },
    {
      "title": "This Side of Paradise",
      "questions": [
        "Character development: Amory Blaine, the protagonist goes through significant growth. Trace his character development. Support your analysis with examples from the novel.",
        "Symbolism: Fitzgerald incorporates various symbols throughout the novel. Choose one or two symbols and analyze their significance.",
        "Structure: intro, 3 paragraphs, conclusion"
      ]
    }
  ],
  "dueDate": "Due by Thursday 2/27/10"
}
```

Figure 6: Claude 3 model's ability to transcribe text from images and output structured data [13].

To comprehensively evaluate the visual understanding and reasoning capabilities of our selected models, we conducted a comparative analysis across five key evaluation sets [13, 14, 28]. These benchmarks assess various aspects of visual cognition, crucial for the Audieyes project's goal of providing accurate and contextual assistance to visually impaired users. The models under comparison include GPT-4o (omni), GPT-4T (Turbo), GPT-4V, Gemini Pro, Gemini Ultra, Claude 3 Opus, Claude 3 Sonnet, and Claude 3 Haiku.

The evaluation sets and their significance are as follows:

- **MMMU (Multi-discipline Multi-modal Understanding):** Assesses the models' capability to handle complex, college-level problems across various disciplines, demonstrating their ability to process and comprehend diverse visual information.
- **MathVista:** Focuses on mathematical visual reasoning, evaluating the models' proficiency in interpreting and solving visually presented mathematical problems.
- **AI2D (AI Diagrammatic Reasoning):** Tests the models' ability to understand and reason with scientific diagrams, crucial for interpreting complex visual data in scientific contexts.

- ChartQA: Measures the models' skill in understanding and interpreting chart data, essential for tasks involving data visualization and analysis.
- DocVQA: Evaluates the models' performance in document visual question answering, assessing their ability to extract and reason about information from document images.

These benchmarks collectively provide a comprehensive assessment of each model's visual cognition capabilities, offering crucial insights into their potential effectiveness for the Audieyes application in providing detailed, accurate, and contextually relevant environmental descriptions to visually impaired users.

Table 1: A comprehensive summary of the vision benchmarks of all our AI models [13, 14, 28].

Model / Metrics	MMMU (%)	MathVista (%)	AI2D (%)	ChartQA (%)	DocVQA (%)
GPT-4o (omni)	69.1	63.8	94.2	85.7	92.8
GPT-4o-mini	59.1	56.1	88.9	82.3	90.3
GPT-4T (Turbo)	63.1	58.1	89.4	78.1	87.2
GPT-4V (Vision)	56.8	49.9	78.2	78.5	88.4
Gemini Pro	47.9	45.2	73.9	74.1	88.1
Gemini Ultra	59.4	53.0	79.5	80.8	90.9
Claude 3 Opus	59.4	50.5	88.1	80.8	89.3
Claude 3 Sonnet	53.1	47.9	88.7	81.7	89.5
Claude 3 Haiku	50.2	46.4	86.7	81.7	88.8

Each model in our comparison demonstrates advanced vision capabilities, with varying strengths across specific tasks. However, the data in Table 1 clearly illustrates that GPT-4o (omni) outperforms all other models across most evaluation sets. GPT-4o particularly excels in AI2D (94.2%), ChartQA (85.7%), and DocVQA (92.8%), showcasing its superior capability in understanding and reasoning with diagrams, charts, and documents. This exceptional performance in visual comprehension tasks is crucial for the Audieyes project, as it directly translates to more accurate and detailed environmental descriptions for BVI users.

While Google's Gemini Ultra and the Claude 3 model family show strong performances in certain areas, they consistently lag behind GPT-4o in overall metrics. Gemini Ultra demonstrates notable capabilities in document visual question answering (90.9% in DocVQA) and scientific diagrams, while Claude 3 models show competitive performance in

understanding documents and diagrams. However, none match the across-the-board excellence of GPT-4o. Given its exceptional performance across all benchmarks, GPT-4o stands out as the most performing candidate followed GPT-4o-mini in the second place.

4.4.2 Economic Viability

The cost efficiency of deploying advanced AI models is a critical consideration. This section provides a comprehensive cost analysis of the API prices for the top leading AI models: GPT-4o (omni) , GPT-4o-mini, GPT-4o mini, Gemini 1.5 Pro, and Claude 3 Opus [33, 34, 35].

Our analysis focuses on comparing the input and output token costs of these models, with the aim of determining the most economically feasible option that does not compromise on performance and reliability. This is particularly crucial for Audieyes, as we seek to provide high-quality, real-time assistance to BVI users in Morocco while ensuring the project's long-term sustainability.

This analysis will inform our strategic decisions regarding the deployment of AI technology in the Audieyes project, ensuring we deliver a solution that is both cost-effective and capable of providing the high-quality service our users require. Our goal is to identify a model that not only meets our current needs but also allows for future growth and improvement of the Audieyes system.

Table 2 provides the cost analysis focusing on four leading AI models, evaluating their pricing structures per million input and output tokens.

Table 2: AI mode's API price for input and output per million token. [33, 34, 35].

Model	Input Tokens Cost (1M tokens)	Output Tokens Cost (1M tokens)
GPT-4o (omni)	\$5.00	\$15.00
GPT-4o-mini	\$0.150	\$0.600
Gemini 1.5 Pro	\$3.50	\$10.50
Claude 3 Opus	\$15.00	\$75.00

GPT-4o-mini, priced at \$0.15 per million input tokens and \$0.60 per million output tokens, offers the best economic option for Audieyes. Its affordability and efficient performance make it as the most suitable option for large-scale, cost-sensitive applications. While this stands as the second best option when it comes to performance, it's economic viability is more aligned with our potential user pool target. Assuming a transaction involving 500 input tokens and 100

output tokens, with a budget of 20 MAD, the GPT-4o-mini model can process up to 15,000 images, the same number of times a human blinks in a day.

In conclusion, GPT-4o-mini's combination of cost-effectiveness and advanced capabilities positions it as the optimal model choice for Audieyes.

4.4.4 Hardware Feasibility

The hardware feasibility of the Audieyes project is grounded in its reliance on a single, widely accessible device: a smartphone equipped with a camera. This requirement is both practical and inclusive, as it leverages technology that most users already possess, eliminating the need for specialized or expensive hardware. The performance specifications of the smartphone and the quality of its camera are not critical determinants of the system's effectiveness. Audieyes is meticulously optimized to function efficiently across a range of devices, from high-end smartphones to more modest models. This is achieved through advanced software design that minimizes computational demands, ensuring smooth operation and real-time processing capabilities without necessitating the latest hardware.

The core functionality of Audieyes will be designed to be highly adaptable, utilizing the available resources of standard smartphones to deliver accurate and timely environmental descriptions and object detections. This approach underscores the project's commitment to broad compatibility and ease of use, facilitating seamless integration into the daily lives of visually impaired users. By focusing on software optimization, Audieyes ensures that users do not need to upgrade their existing devices, making the technology more accessible and reducing financial barriers.

Moreover, the reliance on smartphones means that Audieyes can capitalize on the continuous advancements in mobile technology, benefiting from improvements in processing power and camera quality over time. This forward-thinking approach ensures that the system remains relevant and effective as hardware evolves, without necessitating frequent updates or replacements. The user experience is further enhanced by the inherent portability and familiarity of smartphones, allowing visually impaired individuals to carry and use Audieyes in various contexts and environments with minimal adjustment.

Chapter 5. Human AI Interactions

In this section, we will explore the critical aspects of Human AI Interactions in the Audieyes application, focusing on User Interface Design, Accessibility Best Practices, and Ethical Considerations. We aim to provide an in-depth analysis of how Audieyes is tailored to enhance accessibility for BVI individuals through thoughtful design principles, adherence to accessibility standards, and commitment to ethical AI deployment.

5.1 User Interface Design

The Audieyes application employs a user-centric design approach, tailored specifically for BVI individuals in Morocco. This design philosophy is built upon several key principles:

- **Simplicity:** The interface utilizes clear and familiar patterns to activate AI-powered scene descriptions. This streamlined approach reduces cognitive load, allowing users to quickly access critical information about their surroundings without navigating complex menu structures.
- **Flexibility and Customization:** The application offers extensive customization options, including language preferences (with support for Moroccan Darija), adjustable speech rates, and more. This adaptability ensures the application meets diverse user needs and preferences.
- **User Feedback:** Built-in mechanisms allow users to easily report inaccuracies or request alternative descriptions. This feedback loop contributes to ongoing refinement of the AI model's prompt engineering side.

By integrating these design principles with advanced AI technology, Audieyes creates a uniquely accessible and culturally attuned interface, providing accurate environmental descriptions in a manner that is intuitive, responsive, and deeply aligned with the needs of its BVI users in Morocco.

5.2 Ethical Considerations

In developing Audieyes for BVI users in Morocco, ensuring fairness and mitigating biases and maintaining accountability is paramount to our ethical framework. Given that we utilize pre-trained Large Language Models (LLMs), particularly GPT-4o-mini, we've implemented a multi-faceted approach to address potential biases:

1. Rigorous Model Evaluation: We conduct comprehensive assessments of GPT-4o-mini using diverse datasets that reflect the varied environments and contexts encountered by Moroccan BVI users. This evaluation focuses on the model's performance across different socio-economic settings, cultural contexts, and geographic locations within Morocco. Special attention is given to the model's proficiency in processing and generating Moroccan Darija, ensuring linguistic accuracy and cultural relevance.
2. User Feedback Integration: We've implemented a robust feedback mechanism within Audieyes, allowing users to easily report inaccuracies or perceived biases in real-time. This system is designed to be accessible and user-friendly for BVI individuals. The collected feedback is systematically analyzed to identify patterns of bias and inform targeted improvements.
3. Adaptive Learning Pipeline: We've developed an adaptive learning pipeline that incorporates user feedback and expert insights to fine-tune the model's prompts. This system allows for iterative improvements, ensuring that Audieyes becomes increasingly attuned to the specific needs and contexts of Moroccan BVI users over time.
4. Transparency and Education: We maintain transparency about the limitations of the AI system and educate users about potential biases. Clear, accessible documentation in Moroccan Darija explains how the system works, its current limitations, and the ongoing efforts to improve fairness.

By implementing these comprehensive strategies, we strive to create an AI-assisted environment that is not only technologically advanced but also fair, culturally sensitive, and truly beneficial for BVI users across all regions of Morocco. Our commitment to fairness and bias mitigation is an ongoing process, integral to Audieyes' mission of enhancing autonomy and quality of life for BVI individuals in Morocco.

Chapter 6. System Design

The System Design chapter of the Audieyes project report delves into the intricate architecture that underpins this visionary assistive technology. This section provides a comprehensive overview of the system's structure, explaining how various components interact to deliver real-time, context-aware assistance to BVI users. The chapter begins with a high-level overview of the basic system architecture, then progresses to explore advanced fault-tolerant design strategies. It examines key elements such as API management, database design and replication, the integration of Large Language Models and more. The design also incorporates cutting-edge concepts like edge computing and multi-region deployment, demonstrating Audieyes' commitment to high availability and low-latency performance. This chapter illuminates the software engineering and system design decisions that underpin Audieyes' fault tolerance design system to provide a reliable service not only for Moroccan users, but the whole Arabic speaking world, and potentially, the entire world.

6.1 System Architecture

The Audieyes system architecture is a meticulously designed 3-tier structure, orchestrated to empower BVI users with real-time environmental insights. This architecture consists of a Presentation Layer, Logic Layer, and Data Layer, each playing a crucial role in the system's functionality.

At the forefront, the Presentation Layer serves as the interface through which BVI users engage with the digital ecosystem. Mobile devices act as sensory extensions, capturing visual data and transmitting it securely to the Audieyes API. This layer handles user input and presents the processed information in an accessible format.

The crown jewel of this architecture is its seamless integration with a state-of-the-art Large Language Model (LLM) API. This AI powerhouse transforms static images into rich, contextual narratives, painting vivid pictures in the minds of BVI users. The LLM's ability to generate culturally nuanced descriptions, particularly tailored for Moroccan users, sets Audieyes apart in its quest for truly inclusive assistive technology.

Audieyes 3-Tier Architecture

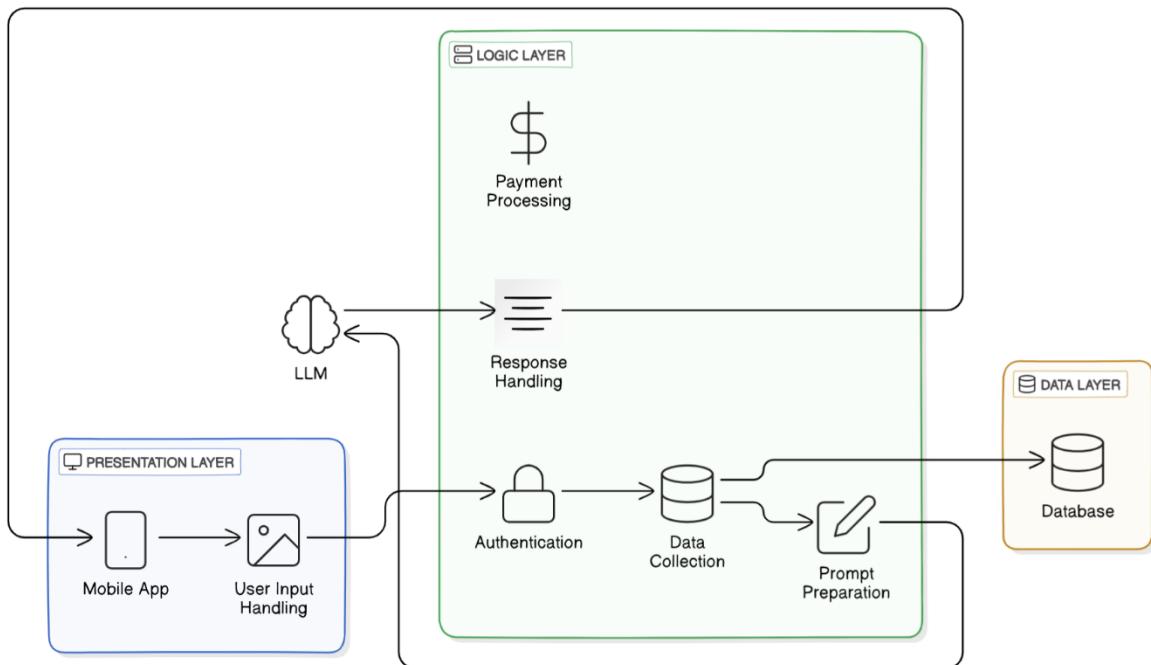


Figure 7: Audieyes 3-tier architecture

The Data Layer underpins the entire system, storing user data, preferences, and other persistent information crucial for personalized experiences. This 3-tier architecture offers significant advantages, providing a clear separation of concerns that enhances maintainability and scalability. It allows for independent scaling and modification of components in each layer, crucial for an AI-driven application like Audieyes that may need to integrate new AI models or features over time. This modular approach ensures that Audieyes remains adaptable to evolving user needs and technological advancements, solidifying its position as a cutting-edge solution for BVI individuals.

6.2 Fault Tolerant System Design

In the development of Audieyes, a system designed to provide critical real-time assistance to BVI users in Morocco and potentially the whole Arabic speaking world, ensuring uninterrupted service is paramount. For BVI individuals, their biological vision has already failed them, profoundly impacting their daily lives, independence, and ability to navigate the world around them. This system is meticulously focused on fault tolerance, aiming to create an AI-assisted vision that these users can rely on consistently, without the fear of sudden failure or unreliability that they've experienced with their natural sight. By prioritizing robust, fail-safe design,

Audieyes strives to become a dependable 'second sight' that will never fail its users, offering a level of reliability that stands in stark contrast to the visual challenges they face daily.

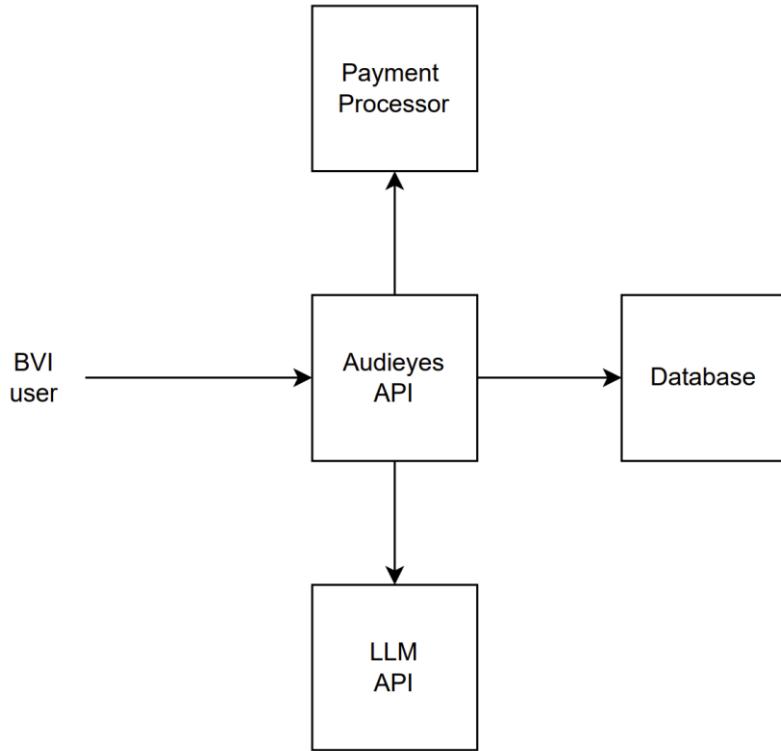


Figure 8: Simplified Starting System Design for Audieyes

This section delves into the fault-tolerant design strategies implemented on top of the simplified starting system architecture in Figure 8 to eliminate Single Points of Failure (SPOF) - critical components whose malfunction could render the entire system inoperative. We explore a multi-faceted approach to fault tolerance, addressing potential vulnerabilities in key areas such as API duplication and management, load balancing, database replication, and the integration of Large Language Models. The design considerations extend beyond basic redundancy, incorporating advanced concepts like edge computing and multi-region deployment to ensure high availability and low-latency performance, crucial for maintaining the targeted 5-second response time. These measures are not just technical choices but represent our commitment to providing BVI users with a consistently available, reliable alternative to the vision that has failed them.

6.2.1 API Redundancy and Horizontal Scaling

The API serves as a central and vital component. It acts as the main interface through which all interactions between the users and the system occur. If the API were to fail while using a simple deployment architecture such as that in Figure 8, the impact on the system would be profound and immediate, effectively rendering the entire service inaccessible to the users. This failure would prevent BVI users from interacting with the system, as the API is responsible for handling all incoming data, processing requests, and coordinating with other services such as the LLM API and the database.

To mitigate the risks associated with the API being a single point of failure, we are duplicating the API component to enhance the system's reliability and fault tolerance. This duplication involves deploying multiple instances of the API, which can operate concurrently and share the workload. By doing so, we ensure that if one instance fails, the remaining instances can continue to provide uninterrupted service to the users.

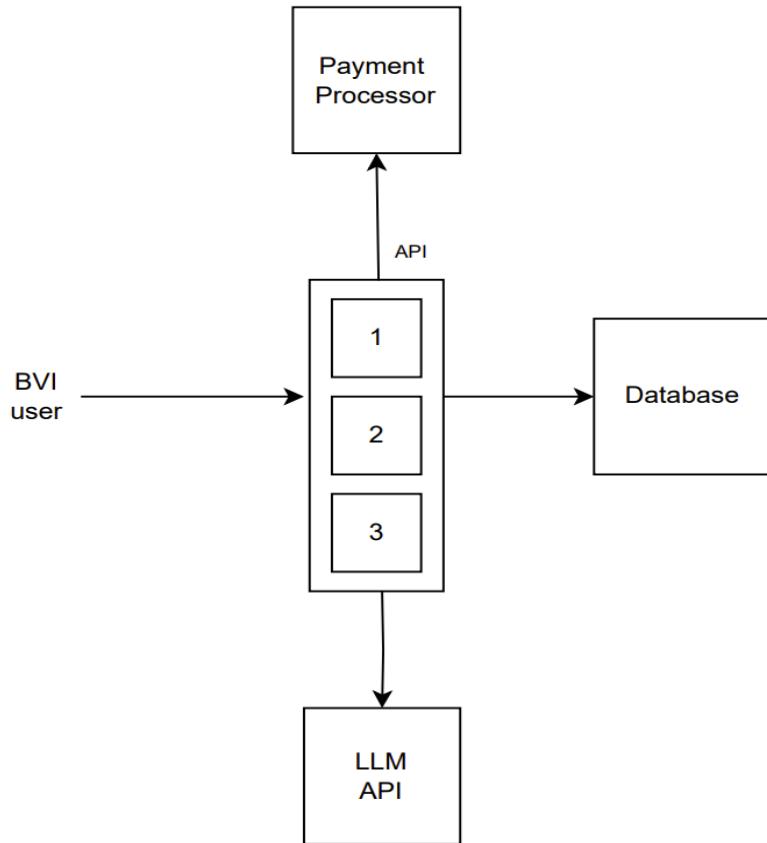


Figure 9: Duplication of Audieyes API architecture

In the enhanced system design, as illustrated in Figure 9, three API instances are deployed and running the exact same code. These instances are configured to operate in parallel, each capable of handling requests independently. Our API is a stateless service meaning that it does not matter which version of the API we are interacting with, all the inputs and outputs of the API are the exact same.

6.2.2 Intelligent Traffic Distribution with Load Balancing

To further enhance the robustness and reliability of the Audieyes system, the introduction of a load balancer is essential in order to efficiently distribute traffic across our API instances. A load balancer acts as an intermediary that manages and distributes incoming requests from BVI users across multiple instances of the Audieyes API. This architectural enhancement not only mitigates the risk associated with having a single point of failure but also significantly improves the system's ability to handle high volumes of requests efficiently.

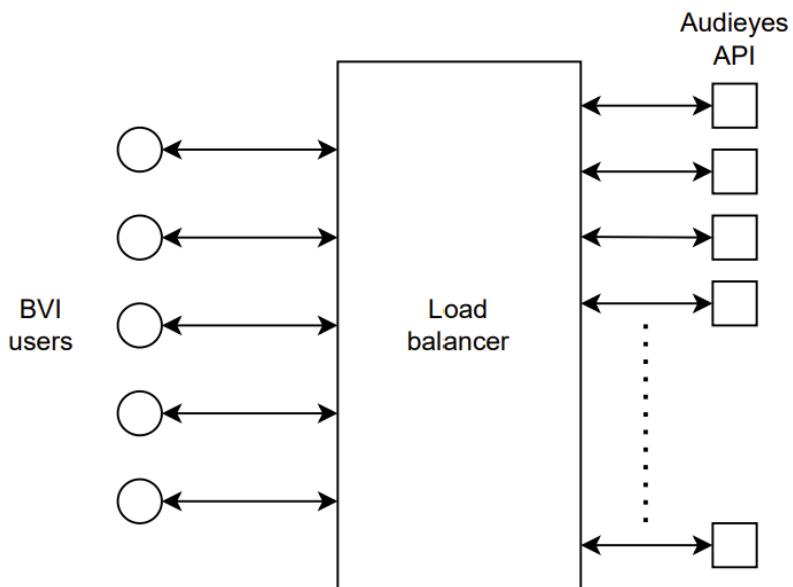


Figure 10: Integration of a load balancer

The load balancer in Figure 10 receives requests from BVI users and intelligently distributes them among the available API instances. This distribution ensures that no single API instance is overwhelmed with too many requests at any given time, thereby preventing potential performance bottlenecks. By balancing the load evenly, the system can maintain optimal performance and responsiveness, even during peak usage periods.

In the event that one of the API instances fails, the load balancer detects the failure and automatically reroutes incoming traffic to the remaining healthy instances. This failover mechanism is crucial for maintaining service availability and continuity. Users will not experience any disruption, as the load balancer seamlessly redirects their requests to functioning API instances. This redundancy is a key aspect of ensuring that the Audieyes system remains reliable and accessible at all times.

Additionally, the load balancer continuously monitors the health and performance of each API instance. It uses various health check protocols to assess the status of the instances and can dynamically adjust the traffic distribution based on real-time performance metrics. This proactive monitoring allows the system to adapt to changing conditions and ensures that users always receive the best possible service.

6.2.3 Global Accessibility and Failover with DNS

While introducing a load balancer significantly enhances the system's fault tolerance and scalability, it is important to recognize that the load balancer itself can become a single point of failure. Figure 11 demonstrates two deployed load balancers, each with a unique IP address. This redundancy ensures that if one load balancer fails, the other can continue to manage and distribute incoming traffic.

To manage the distribution of traffic between these load balancers, we introduce in Figure 11 a Domain Name System (DNS) service. The latter is configured to map the domain name of the Audieyes service to the IP addresses of both load balancers. When a BVI user attempts to access the Audieyes service, the DNS resolves the domain name to one of the load balancer IP addresses. Additionally, the DNS service provides caching capabilities, which help improve the system's efficiency and response time. By caching the IP addresses of the load balancers, the DNS can quickly resolve user requests without having to perform a fresh look up each time.

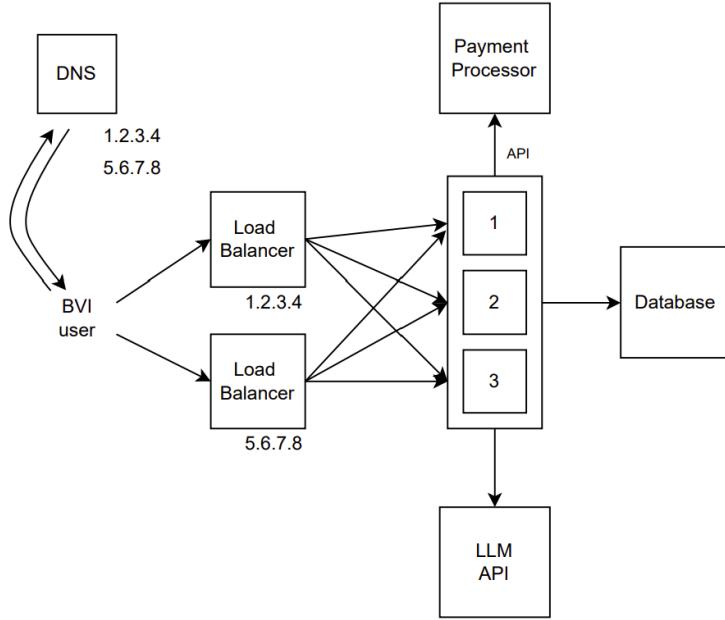


Figure 11: Advanced system architecture of Audieyes, featuring DNS management.

In the event of a load balancer failure, the DNS can detect the unavailability of an IP address and automatically reroute traffic to the remaining operational load balancer. This dynamic rerouting capability is crucial for maintaining service availability and ensuring that users can continue to interact with the Audieyes system without interruption. Furthermore, having multiple load balancers also enhances the system's capacity to handle increased traffic. As the user base grows, additional load balancers can be introduced to distribute the load more effectively, ensuring that the system can scale seamlessly to meet rising demand.

6.2.4 Resilient Data Management through Database Replication

The database is a crucial component that can represent another single point of failure. Ensuring its continuous availability and reliability is paramount, as the database is responsible for storing and managing all critical user-related data, including preferences, saved locations, and recognized objects. A failure in the database would significantly disrupt the system's functionality, rendering it incapable of retrieving or storing necessary information, thus compromising the service provided to BVI users.

To mitigate this risk, replicating the database is essential. However, this process is more complex in comparison to duplicating our stateless API because the database always has a state. Unlike stateless APIs, which do not retain any information between requests, stateful databases

maintain data across sessions, necessitating a sophisticated approach to ensure data consistency and reliability. To achieve this, we implement a leader-follower database replication system design.

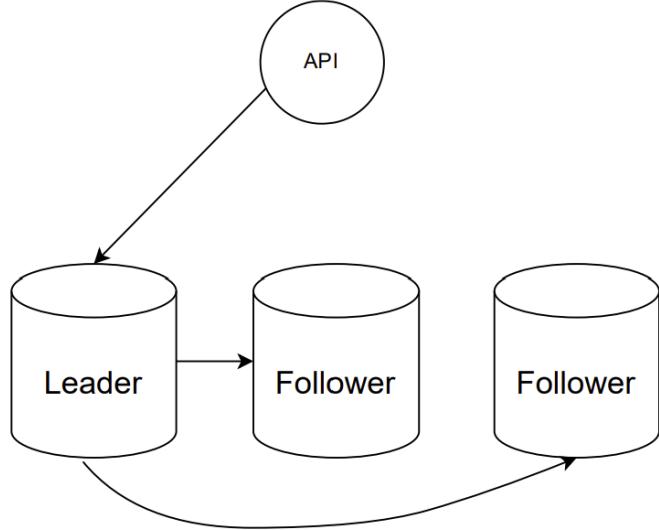


Figure 12: Leader-Follower architecture used for database replication..

The leader-follower replication system demonstrated in Figure 12 consists of one primary node, referred to as the "leader," and multiple secondary nodes, known as "followers" [35]. The leader is responsible for handling all write operations. When the API receives a request that involves modifying the database, such as updating user preferences or saving new data, it directs this request to the leader. The leader processes the write operation and then propagates the changes to the follower nodes.

The follower nodes are read-only replicas of the leader. They continuously synchronize with the leader to ensure that they have the most up-to-date data. This synchronization process involves the leader sending updates to the followers, which then apply these updates to their local copies of the data. By maintaining multiple copies of the data across different nodes, the system ensures data redundancy and availability.

One of the key advantages of the leader-follower replication system is its ability to enhance read performance. The API can distribute read requests across all available follower nodes, thereby balancing the load and reducing the latency for read operations. This distributed approach ensures that the system can handle a large number of read requests efficiently, providing fast and reliable access to data for BVI users.

In the event of a failure of the leader node, the system employs an automatic failover mechanism to promote one of the follower nodes to become the new leader. This process involves:

1. Failure detection through regular health checks
2. Leader election among the remaining nodes
3. Reconfiguration of the system to recognize the new leader
4. Redirection of write traffic to the new leader

This failover mechanism ensures that write operations can continue with minimal interruption, maintaining the integrity and availability of the database. With this setup, the Audieyes API can access any of the database nodes, thereby managing fault tolerance and enhancing the read capacity of the database. In the event that one database node fails, the system can seamlessly switch to another node without any disruption in service.

6.2.5 Private and Secure Database Access with P-DNS

To further bolster the reliability and resilience of the database system, we integrate a private DNS (P-DNS) service discovery system. This component in Figure 13 plays a crucial role in maintaining the health and efficiency of our distributed database architecture.

The P-DNS system performs several key functions specific to database management:

1. Database Health Monitoring: The system continuously performs health checks on all database nodes. These checks involve both simple connectivity tests and more complex database-specific checks to ensure that each node is not only responsive but also functioning correctly.
2. Dynamic DNS Updates: Based on the health check results, the P-DNS system dynamically updates its DNS records. This ensures that database queries are always routed to healthy, operational nodes.
3. Read Load Distribution: For read operations, the P-DNS system considers current load and performance metrics when distributing traffic across follower nodes. This helps to prevent any single database node from becoming overwhelmed during peak usage periods.

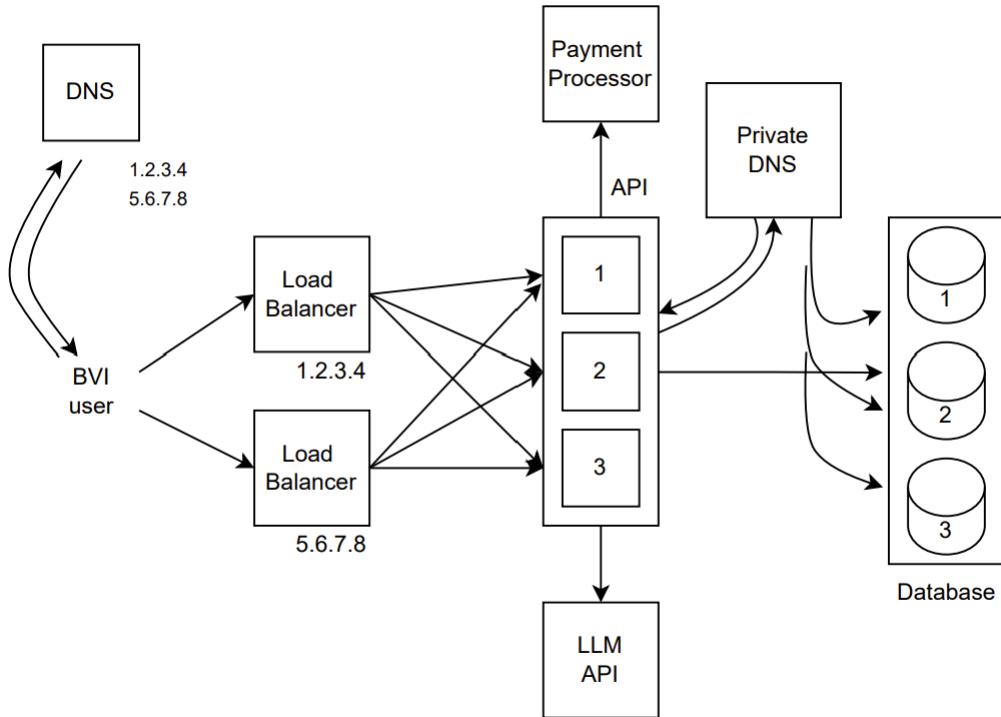


Figure 13: A comprehensive system architecture of Audieyes, including the introduction of P-DNS

In the event of a database node failure, the P-DNS system swiftly responds through the following process:

1. Failure Detection: The system identifies the failed database node through its regular health checks.
2. DNS Record Update: It immediately updates its DNS records to exclude the failed node.
3. Query Rerouting: All subsequent database queries are automatically rerouted to the remaining operational nodes.
4. Recovery Monitoring: The system continues to monitor the failed node and can reincorporate it into the pool once it becomes operational again.

The data replication logic implemented in our database system ensures that any data stored in the failed node is already present in the other nodes, maintaining data integrity and consistency throughout this process.

6.2.6 Interchangeable LLM Microservice Integration

In the Audieyes system, the integration of Large Language Models (LLMs) plays a crucial role in providing detailed scene descriptions and answering user queries, particularly in Moroccan

Darija. However, relying on a single LLM provider introduces a significant single point of failure, which can jeopardize the system's reliability and availability. The failure or unavailability of the LLM service could disrupt the critical functionalities that BVI users depend on, such as receiving real-time descriptions of their surroundings in their local language. To address this, we adopt a microservices architecture that leverages the services of the top three LLM providers, each capable of processing and generating responses in Moroccan Darija. This strategy ensures that if one LLM service fails, the system can seamlessly fall back to the other providers, thereby maintaining continuous operation.

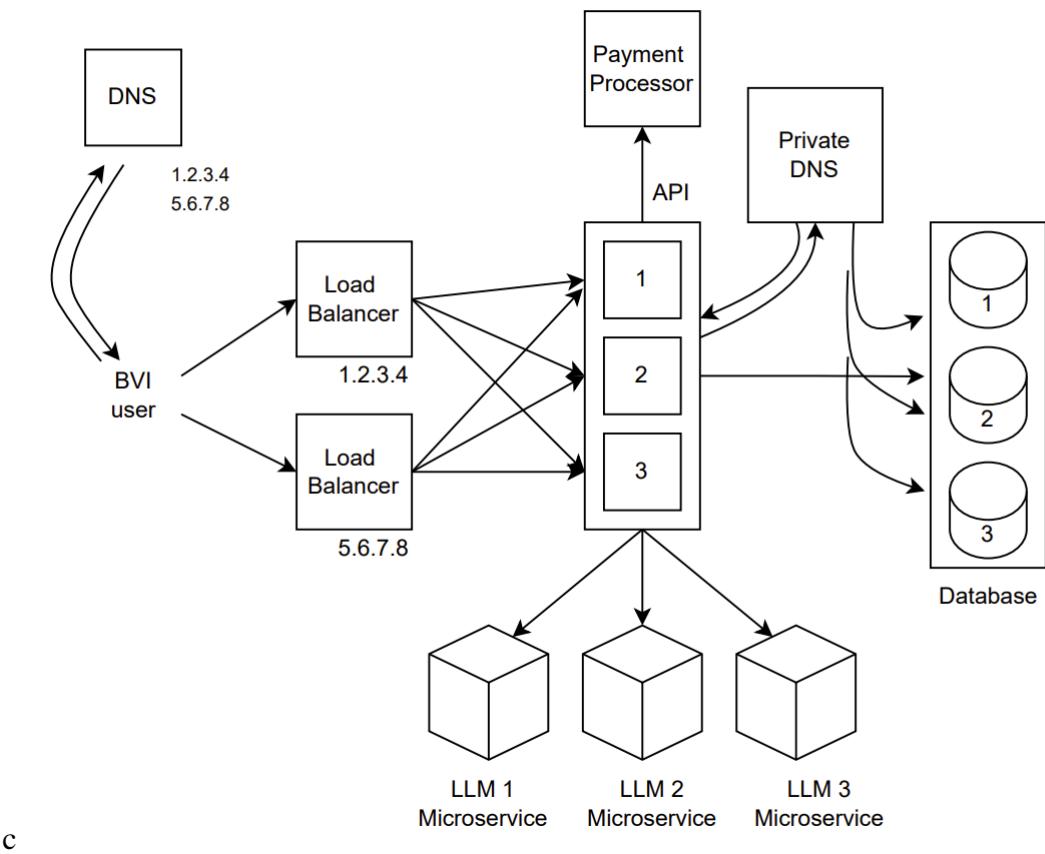


Figure 14: Enhanced System Architecture for Audieyes with Microservices

The microservices architecture depicted in Figure 14 involves deploying multiple LLM microservices, each connected to a different LLM provider. By distributing the reliance across multiple providers, we significantly reduce the risk associated with a single point of failure. Each LLM microservice operates independently, handling specific tasks related to scene description and query answering. These microservices are integrated into the overall system

architecture, allowing the Audieyes API to route requests to any of the available LLM microservices based on their availability and performance.

In the event that one of the LLM providers experiences an outage or performance degradation, a reroute of traffic to the remaining operational LLM microservices will occur. This failover mechanism ensures that users continue to receive uninterrupted service, as the system dynamically adjusts to the changing availability of the LLM services.

Implementing this multi-provider approach also enhances the system's scalability and flexibility. As new LLM providers emerge or existing ones improve their services, additional microservices can be integrated into the system with minimal disruption. This modular architecture allows for seamless upgrades and expansions, ensuring that the Audieyes system remains at the forefront of technological advancements in natural language processing.

Moreover, integrating multiple LLM providers mitigates the risks associated with vendor lock-in. By maintaining relationships with several top-tier providers, the system can avoid dependency on a single vendor, which can be advantageous in negotiating service terms and ensuring competitive pricing. This strategic approach also provides a safety net against potential long-term changes in the service landscape, such as shifts in provider offerings or pricing structures.

6.2.7 Fault-Tolerant Payment Processing

In the Audieyes system architecture, the payment processor is a critical component managing financial transactions. Recognizing its potential to become a single point of failure, Audieyes integrates a highly reliable payment processor with a 99.999% uptime guarantee. This ensures near-continuous availability, limiting potential downtime to just 52.56 minutes per year.

The selection of such a reliable payment processor is crucial for maintaining user trust and system financial stability. It ensures consistent transaction completion, reducing the risk of user frustration and loss of confidence. Moreover, the continuous availability of payment services supports uninterrupted revenue streams, which are vital for the ongoing development, maintenance, and future enhancements of the Audieyes system.

6.2.8 Cloud Infrastructure: Containerization and Orchestration

Audieyes leverages modern public cloud infrastructure for robust performance and high availability with horizontal scaling deployment strategy. The application is containerized, ensuring consistent deployment across environments with all dependencies included. A container orchestration system manages these containers, enabling efficient scaling and automatic deployment across machine clusters.

In modern data centers, server racks house and organize servers that provide computational resources for applications like Audieyes. These racks contain vertically stacked servers interconnected via network switches, facilitating communication between servers and the external network.

However, server racks can introduce significant reliability risks. When a server rack holding all API servers fails as shown in Figure 15, the Audieyes system loses its entire nervous system, rendering the system inoperative. Similarly, if a server rack containing all database servers fails, the system loses access to critical user data, preferences, and saved information, further impacting the overall functionality and user experience.

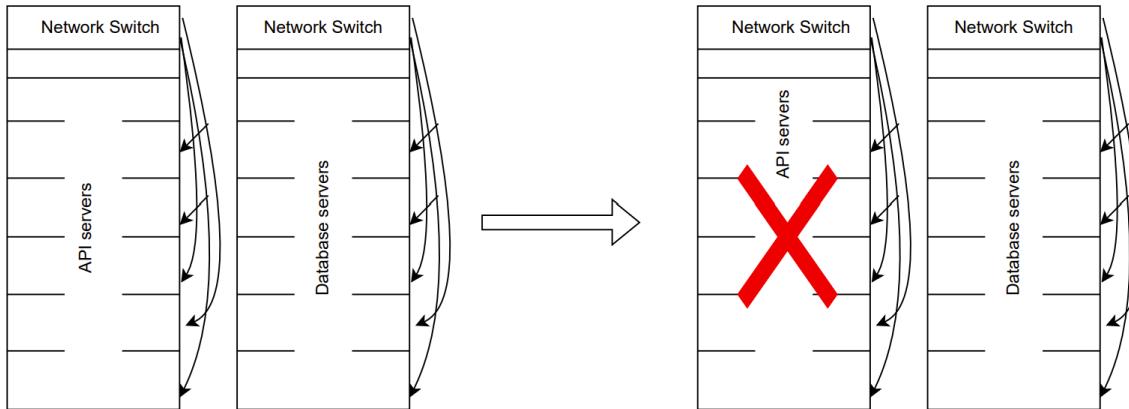


Figure 15: Inefficient Server rack deployment strategy.

To mitigate such risks, it is crucial to adopt a more resilient architecture that distributes the servers across multiple racks as displayed in Figure 16, ensuring that no single rack failure can bring down the entire system. In the enhanced architecture, depicted in the second diagram, the API servers and database servers are distributed across multiple racks, each equipped with

redundant network switches. This setup ensures that even if one rack experiences a failure, the other racks can continue to operate, maintaining the availability and functionality of the system.

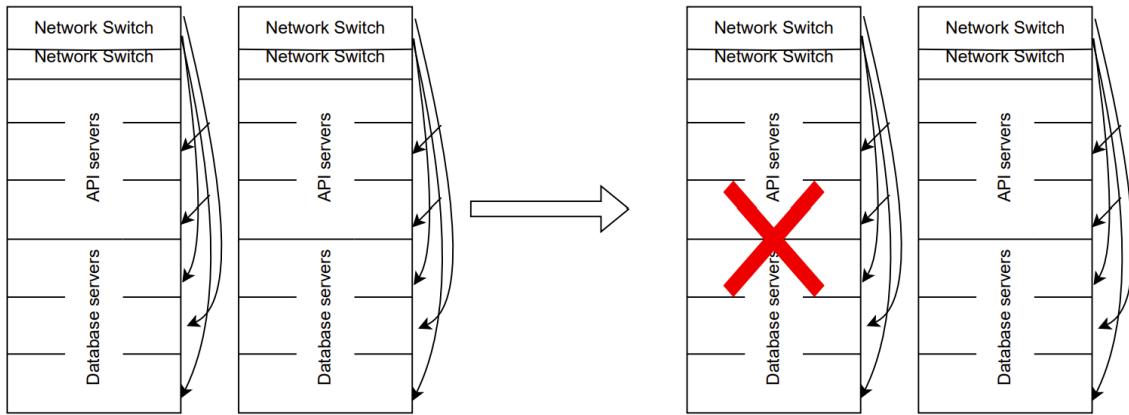


Figure 16: Improved server rack deployment strategy

Similarly, for the API servers, load balancing ensures that incoming user requests are evenly distributed across the servers in different racks. In the event of a rack failure, the load balancer reroutes the traffic to the operational API servers, maintaining uninterrupted service for the users. This failover mechanism is essential for ensuring that the system can handle large-scale failures without significant downtime.

6.2.9 Geo-Redundant Multi-Region Deployment

To address potential large-scale failures and ensure continuous service for BVI users in Morocco, Audieyes employs a geo-redundant multi-region deployment strategy. This approach mitigates risks from power outages, natural disasters, or other catastrophic events that could render an entire data center inoperative.

Figure 17 illustrates this multi-region deployment architecture. The system components, including API servers, databases, load balancers, and the payment processor, are distributed across two distinct regions: Region A and Region B. Each region operates independently, with its own set of load balancers, API instances, databases, and private DNS service discovery systems. This geographical distribution enhances the system's fault tolerance by ensuring that the failure of one region does not disrupt the overall functionality.

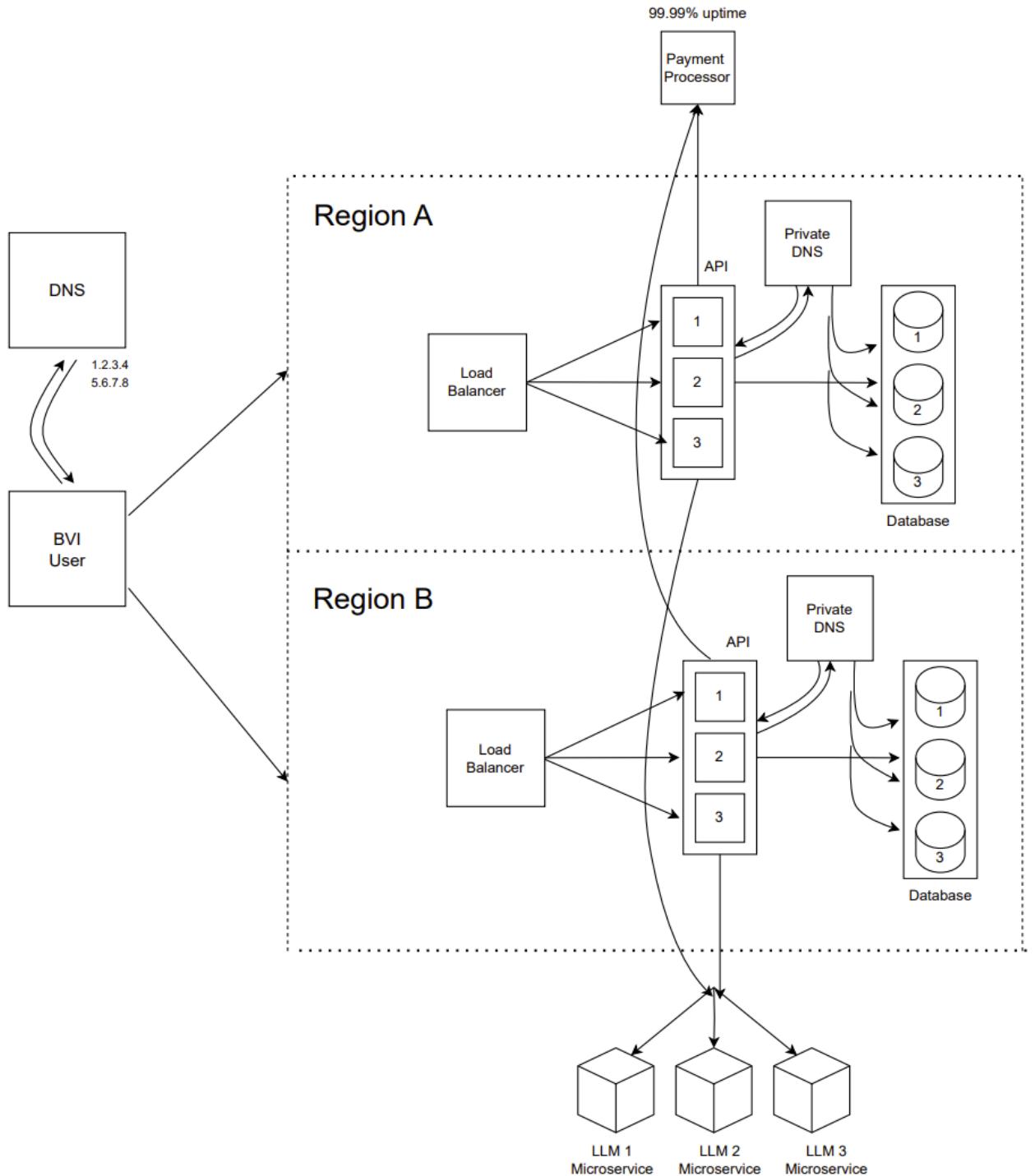


Figure 17: Multi-Region Deployment Architecture

In this upgraded architecture of Figure 17, load balancers are deployed in both regions, distributing incoming traffic based on the availability and performance of the regional resources. The DNS system plays a pivotal role in managing this distribution. When a BVI user accesses the Audieyes service, the DNS resolves the domain name to the IP addresses of the

load balancers in both regions. The client can then choose which region to connect to, depending on factors such as proximity, latency, and regional availability.

By distributing the system components across multiple regions, we ensure redundancy and high availability. Each region operates as a fully functional and independent deployment, capable of handling all user requests and data processing tasks. In the event of a regional failure, such as a power outage or disaster that impacts an entire data center, the DNS can dynamically reroute traffic to the operational region. This failover mechanism is seamless, allowing users to continue accessing the service without interruption.

The databases in each region are synchronized using data replication techniques. This ensures that user data is consistently updated across all regions, maintaining data integrity and availability. In the event of a failure in one region's database, the system can switch to the replicated database in the other region, ensuring continuous access to user data. The private DNS service discovery system continuously monitors the health of the database nodes in both regions, dynamically updating the routing rules to ensure that traffic is directed to the healthy nodes.

The integration of LLM microservices is similarly distributed across regions. Each region hosts its own set of LLM microservices connected to different LLM providers. This setup ensures that even if one region's LLM microservices fail, the system can rely on the LLM microservices in the other region to handle user requests. The load balancers manage the distribution of traffic to the LLM microservices, ensuring optimal performance and reliability.

6.3 Edge Computing

Edge computing is another technology paradigm that will be adapted to bring computation and data storage closer to the location where it is needed, enhancing the speed and responsiveness of applications. Unlike traditional cloud computing, which relies on centralized data centers, edge computing processes data at or near the source of data generation, such as user devices or local servers. This proximity reduces latency and bandwidth usage, providing faster and more efficient data processing. Edge computing complements cloud computing by offloading some of the processing tasks to local nodes, thereby optimizing resource utilization and enhancing overall system performance.

Our edge computing architecture, illustrated in Figure 18, consists of three key layers:

1. Edge Devices: The BVI user's smartphone acts as the primary edge device, capturing images and initiating data processing.
2. Cloudlet: A localized mini-data center in that will be places in Moroccan cities to serve as a cloudlet, hosting the Audieyes API and providing powerful computational resources in close proximity to users.
3. Fog Layer: Multiple fog nodes distributed across the Arab speaking world to offer additional processing power and storage, forming an intermediate layer between edge devices and the cloud.

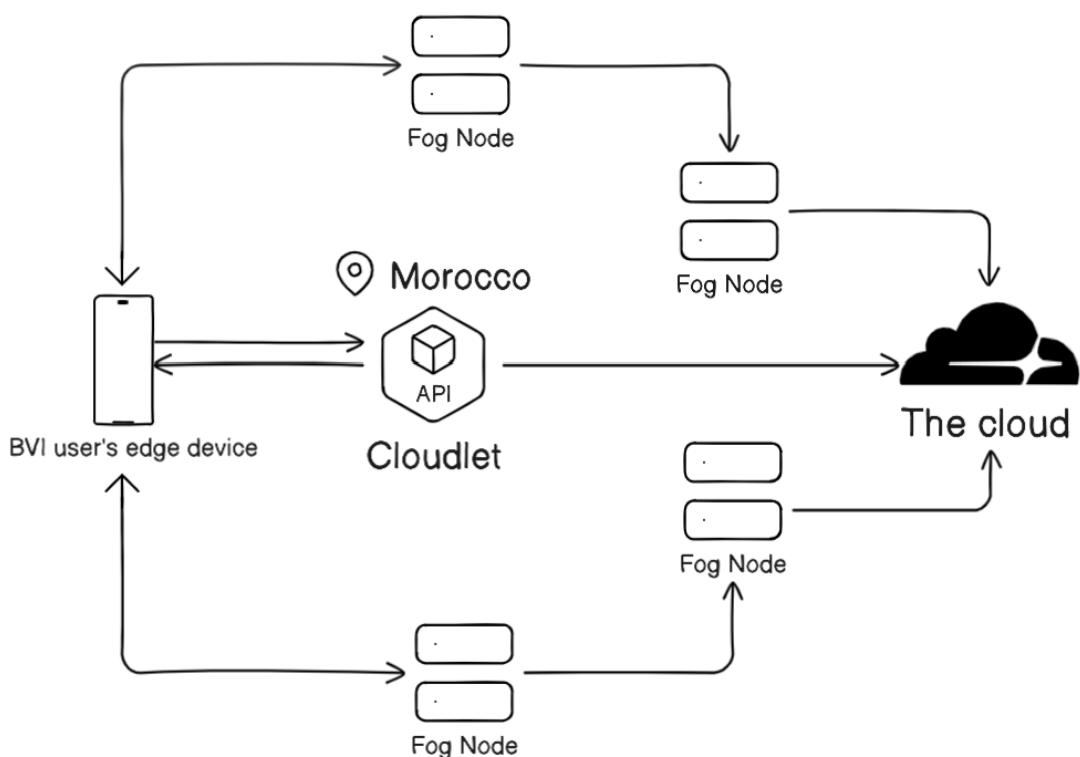


Figure 18: Audieyes' Edge-Fog-Cloud Architecture

At the core of this design is the BVI user's edge device, which serves as the initial point of data generation and processing. The edge device captures images and data, which are then sent to a nearby cloudlet for further processing. The cloudlet, strategically located in Morocco, acts as a localized mini-data center that provides powerful computational resources close to the user. This proximity ensures minimal latency, allowing real-time processing of data and immediate response to user queries. The cloudlet hosts the Audieyes API, which processes the incoming data and interacts with the user's device to deliver the necessary information and assistance.

The fog layer extends the capabilities of the edge and cloudlet by distributing computational tasks across multiple fog nodes. These fog nodes are intermediate servers located between the edge devices and the cloud. They provide additional processing power and storage, enabling the system to handle a larger volume of data and more complex tasks.

The fog nodes work in conjunction with the cloudlet to ensure efficient data processing and to provide redundancy and fault tolerance. In case of high computational demand or failure of any node, the tasks can be dynamically redistributed across the fog nodes, maintaining seamless operation.

The cloud component remains integral to this design, providing centralized storage and advanced processing capabilities that are not feasible to perform at the edge or fog layer. The cloud handles tasks that require significant computational power, long-term data storage, and global synchronization.

6.4 Database design

In the development of software systems, such as Audieyes, an effective relational database design and schema are crucial. A well-structured database design serves as the foundation for managing and organizing data efficiently, ensuring that all user interactions and system functionalities are supported seamlessly. The database schema, which defines the logical structure of the database, plays a pivotal role in establishing relationships between different data entities, enabling efficient data retrieval, storage, and manipulation. By implementing a robust database design and schema, we can ensure data integrity, reduce redundancy, and enhance the overall performance and reliability of the application, ultimately providing a more responsive and personalized experience for BVI users.

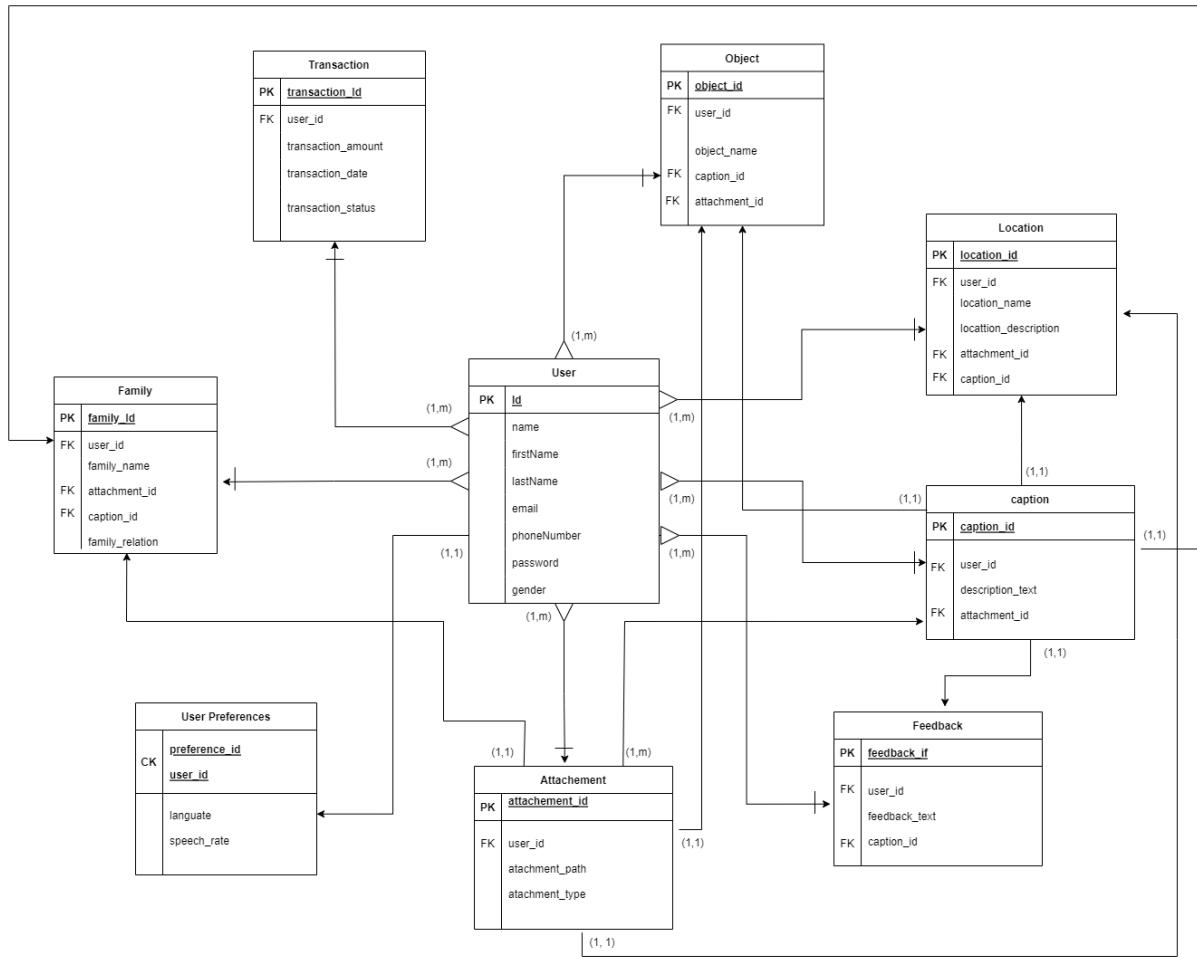


Figure 19: Entity-Relationship Diagram for the Audieyes database

This relational database schema demonstrated in Figure 19 is designed to handle various aspects of user interaction, data storage, and retrieval, ensuring that the system operates smoothly and efficiently. Let's explore each component of this design in detail, highlighting the concepts it implements and supports.

The Audieyes application utilizes a relational database design, implementing fundamental principles to ensure data integrity, efficiency, and scalability. This robust architecture is built on the following key concepts:

1. Normalization: The database schema is normalized to minimize data redundancy and improve data consistency, organizing information into logical, interconnected tables.
2. Primary and Foreign Keys: Each table has a primary key for unique record identification. Foreign keys establish relationships between tables, maintaining data consistency across the database.

3. Relationships: The design reflects real-world associations between entities. For example, users are linked to their preferences, family members, locations, and feedback, enabling personalized and context-aware assistance.
4. Indexing: Strategic indexing on key columns, including primary and foreign keys, optimizes query performance and data retrieval speed.
5. Data Integrity Constraints: Implemented constraints ensure data accuracy and reliability. Foreign key constraints prevent orphan records and maintain referential integrity.

This relational structure is ACID compliant, and allows for complex data interactions crucial to Audieyes' operations. For instance, the 'user_id' foreign key in the 'Family' table links family members to users, while the 'attachment_id' in the 'Caption' table associates captions with their corresponding attachments.

Chapter 7. Implementation

In this chapter, we delve into the implementation details of the Audieyes system, outlining the technical foundations and development processes that bring the project to life. This chapter is structured to provide a comprehensive guide to the various stages of implementation, including backend development, frontend development, integration of AI models, and deployment strategies. Each section will cover the specific technologies and methodologies employed to achieve the project's goals. By the end of this chapter, readers will have a clear understanding of the practical steps taken to build, integrate, and deploy the Audieyes application, ensuring it meets the needs of BVI users effectively and efficiently.

7.1 Technology Enablers

The development and deployment of Audieyes required careful consideration and selection of various technologies to create a robust, scalable, and efficient system. Each technology choice was made with the specific needs of our BVI users in mind, as well as the requirements for a fault-tolerant, high-performance application. In this section, we outline the key technology enablers that form the backbone of Audieyes, discussing our selection process and the rationale behind each choice. These technologies span across programming languages, frameworks, development tools, infrastructure, and security measures, collectively empowering us to deliver a cutting-edge assistive technology solution.

Table 3: Programming Language: TypeScript

Criteria	JavaScript	TypeScript	Python
Overview	JavaScript is a dynamically typed language widely used for web development. Its flexibility and large ecosystem make it a popular choice for frontend and backend development. However, its dynamic nature can lead to runtime errors that are harder to catch early in the development process.	TypeScript is a superset of JavaScript that introduces static typing, which helps in catching errors during the development phase. It enhances code quality and maintainability through better tooling support, such as autocompletion and type-checking in IDEs. TypeScript is particularly strong in environments where scalability and collaboration are critical, as it reduces bugs and improves code readability.	Python is a dynamically typed language known for its simplicity and readability. It has a strong ecosystem, particularly in data science and web development with frameworks like Django and Flask. While Python is easy to learn and use, its performance can be a limitation in some high-scale applications, and it lacks the static typing benefits that TypeScript offers.

We chose TypeScript as the primary programming language for Audieyes due to its ability to catch errors at compile time, which reduces the likelihood of runtime bugs. The strong typing system provided by TypeScript enhances code quality and maintainability, making it ideal for a robust and reliable application. TypeScript's excellent tooling support also improves developer productivity, especially in a collaborative environment. Although JavaScript is flexible and has a large ecosystem, the need for a more structured and error-resistant codebase led us to opt for TypeScript. Python, while popular for its simplicity, was not selected due to its performance limitations and lack of static typing, which are crucial for our project.

Table 4: Backend Framework: NestJS

Criteria	Express.js	NestJS	Django
Overview	Express.js is a minimalist and unopinionated framework that offers flexibility for developers to structure their applications as they see fit. It has a large ecosystem of middleware and is known for high performance. However, this flexibility means more manual setup and organization are required, which can lead to inconsistent codebases in larger projects.	NestJS is an opinionated framework built on TypeScript, offering a modular architecture that promotes consistent code organization. It provides built-in features like dependency injection, which enhances maintainability and testability. NestJS's use of decorators simplifies common tasks like routing and middleware integration, accelerating the development process.	Django is a monolithic framework known for its 'batteries-included' approach, offering a wide range of built-in features such as an admin panel, ORM, and authentication. It is highly structured and easy to use, but its Python base and monolithic nature can limit flexibility and performance in large-scale applications.

We selected NestJS for the backend development of Audieyes because of its opinionated structure, which promotes consistent code organization and leverages TypeScript's powerful features. NestJS's modular architecture and built-in support for dependency injection make the codebase more maintainable and testable, which is crucial for building scalable and reliable backend services. While Express.js offers flexibility, it requires more manual setup and lacks the built-in features that NestJS provides out of the box. Django, though powerful and well-supported, was less preferred due to its monolithic architecture and lack of native TypeScript support, which was a key requirement for our project.

Table 5 : Client Framework: React Native

Criteria	React Native	Flutter	Ionic
Overview	React Native allows developers to build cross-platform mobile applications using JavaScript. It leverages	Flutter, developed by Google, uses the Dart language and allows for the creation of natively	Ionic is a framework for building hybrid mobile apps using web technologies like HTML,

	<p>native components, ensuring a seamless user experience similar to native apps. React Native's large community and extensive library of third-party components make it easier to develop and maintain apps quickly. Its integration with the existing JavaScript ecosystem is particularly beneficial for teams familiar with React.</p>	<p>compiled applications for mobile, web, and desktop from a single codebase. It offers a rich set of pre-designed widgets that ensure high performance and a consistent look and feel across platforms. Flutter's popularity is growing, but it requires developers to learn Dart, which may be a drawback for teams already invested in other languages.</p>	<p>CSS, and JavaScript. It wraps web applications in a native container, which allows for deployment across multiple platforms. Ionic is well-suited for teams that are more comfortable with web development, but it may not provide the same level of performance or native feel as React Native or Flutter.</p>
--	--	--	--

We chose React Native for the client development of Audieyes due to its ability to create high-performance cross-platform mobile applications using JavaScript. The extensive library of third-party components and strong community support make React Native a reliable choice for rapid development and easy maintenance. While Flutter offers compelling features and a consistent cross-platform experience, the need to learn Dart was a consideration. Ionic, although powerful for hybrid app development, was not selected due to its performance limitations compared to React Native.

Table 6: Encryption Tools: CryptoJS

Criteria	brypt	CryptoJS	Node.js crypto module
Overview	<p>brypt is a password hashing function that incorporates a salt to protect against rainbow table attacks. It is widely used for securing passwords due to its adaptive nature, meaning it can be configured to be slower over time to remain resistant to brute-force search attacks. However, brypt is primarily focused on password hashing and is not a general-purpose cryptographic library.</p>	<p>CryptoJS is a widely used cryptographic library that provides a comprehensive set of cryptographic functions, including hashing, encryption, and decryption, supporting various algorithms like SHA-256. It is easy to use and can be utilized on both server-side with Node.js and client-side in browsers, ensuring consistency across the application. CryptoJS's versatility and broad functionality make it an ideal choice for the encryption needs of Audieyes.</p>	<p>The Node.js crypto module is a built-in library that provides cryptographic functionality, including hashing, HMAC, cipher, decipher, sign, and verify functions. It is highly secure and efficient, with a wide range of cryptographic tools at the developer's disposal. However, it is more low-level compared to libraries like CryptoJS and might require more implementation effort to achieve the same results.</p>

We selected CryptoJS for the encryption needs of Audieyes due to its comprehensive set of cryptographic functions, including SHA-256, which are essential for ensuring the security of both client-side and server-side data. CryptoJS offers a consistent and versatile solution that can be used across different parts of the application, simplifying the development process. While bcrypt is excellent for password hashing, it is not a general-purpose cryptographic library. The Node.js crypto module, although powerful, requires more implementation effort compared to the straightforward usage of CryptoJS.

Table 7: Testing Framework: Jest

Criteria	Mocha	Jest	Jasmine
Overview	Mocha is a flexible JavaScript testing framework that provides developers with the freedom to choose their preferred assertion libraries, mocking tools, and other utilities. It is highly customizable and integrates well with various testing environments. Mocha is widely adopted for both unit and integration testing, but its flexibility may lead to more complex configurations.	Jest is a comprehensive testing framework developed by Facebook, designed to work seamlessly with JavaScript and React applications. It offers built-in features such as zero configuration, mocking, code coverage, and snapshot testing, making it particularly suitable for frontend applications like those built with React Native. Jest's extensive feature set and ease of use make it a popular choice for testing modern JavaScript applications.	Jasmine is a behavior-driven development (BDD) framework for testing JavaScript code. It is known for its simplicity and ease of setup, with no need for additional libraries or tools. Jasmine is well-suited for unit testing but might lack some of the advanced features provided by frameworks like Jest, such as built-in mocking and snapshot testing.

We chose Jest as the testing framework for Audieyes due to its comprehensive feature set and ease of use. Jest offers out-of-the-box functionality, including unit and integration testing, mocking, and snapshot testing, which are crucial for ensuring the reliability and correctness of our React Native application. Its built-in code coverage reporting and minimal configuration requirements make it an efficient choice for our development workflow. While Mocha is highly customizable, it requires more configuration effort. Jasmine, though simple and easy to set up, lacks some of the advanced features provided by Jest, making Jest the optimal choice for our project.

Table 8: Containerization Solution: Docker

Criteria	Docker	Podman	LXC
Overview	Docker is the most widely used containerization platform, known for its	Podman is an open-source containerization tool that offers a Docker-compatible	LXC (Linux Containers) is a lightweight virtualization technology

	<p>simplicity, extensive documentation, and large ecosystem of tools and pre-built images. It allows developers to package applications and their dependencies into containers, ensuring consistency across different environments. Docker's robust community support and integration with various CI/CD tools make it the go-to choice for many development teams.</p>	<p>command-line interface but operates without a central daemon. This allows for better security and management of containers, particularly in environments where rootless containers are desired. Podman is gaining popularity due to its flexibility and the fact that it does not require a background service to run containers, offering a more secure alternative to Docker.</p>	<p>that provides an environment similar to a complete virtual machine. It is highly efficient, allowing for the running of multiple isolated Linux systems on a single host. While LXC offers more control and flexibility for advanced users, it is more complex to set up and manage compared to Docker and Podman, making it less suitable for developers who need quick and easy containerization solutions.</p>
--	---	--	--

We selected Docker as the containerization solution for Audieyes due to its widespread adoption, extensive documentation, and large ecosystem. Docker's simplicity and ease of use allow us to package and deploy applications consistently across different environments, which is essential for maintaining the reliability and scalability of our services. While Podman offers a secure alternative with rootless containers, Docker's extensive community support and integration with CI/CD pipelines make it the preferred choice for our project. LXC, though powerful, was not chosen due to its complexity and the steeper learning curve involved in its setup and management.

Table 9: Orchestration Tool: Kubernetes

Criteria	Kubernetes	Docker Swarm	Apache Mesos
Overview	<p>Kubernetes is the leading container orchestration platform, offering powerful features for automating the deployment, scaling, and management of containerized applications. It provides a robust set of tools for managing clusters of containers, ensuring high availability and resilience. Kubernetes' large ecosystem and community support make it the most popular choice for</p>	<p>Docker Swarm is a native clustering and orchestration tool for Docker containers. It is simpler to set up and use compared to Kubernetes, making it a good choice for smaller projects or teams that are already familiar with Docker. However, Docker Swarm lacks some of the advanced features and scalability options provided by Kubernetes, making it less suitable for</p>	<p>Apache Mesos is a cluster manager that provides efficient resource isolation and sharing across distributed applications or frameworks. It is highly scalable and can manage both containerized and non-containerized workloads. While Mesos offers flexibility and scalability, it is more complex to configure and manage, and it lacks the extensive ecosystem and</p>

	managing containers at scale.	large-scale, complex deployments.	community support available with Kubernetes.
--	-------------------------------	-----------------------------------	--

We chose Kubernetes as the orchestration tool for Audieyes due to its powerful features in managing containerized applications at scale. Kubernetes provides the necessary tools for automating deployments, scaling, and operations across clusters of hosts, which is crucial for ensuring the high availability and resilience of our services. While Docker Swarm offers simplicity and ease of use, it does not provide the same level of scalability and advanced features as Kubernetes. Apache Mesos, although highly scalable and flexible, was not selected due to its complexity and the steeper learning curve involved in its setup and management. Kubernetes' extensive ecosystem and strong community support make it the optimal choice for our project.

Table 10: Container Image Management: Docker Hub

Criteria	Docker Hub	Amazon ECR	GCR
Overview	Docker Hub is the most widely used container image registry, known for its ease of use and seamless integration with Docker. It allows developers to store and distribute Docker images efficiently across different environments. Docker Hub's large ecosystem and community support make it an ideal choice for projects that require quick and easy access to pre-built images and efficient image distribution.	Amazon Elastic Container Registry (ECR) is a fully managed Docker container registry that integrates with Amazon Web Services (AWS). It provides high availability, security, and scalability, making it a strong choice for enterprises already using AWS. However, ECR's complexity and AWS-specific integration may not be suitable for all projects.	Google Container Registry (GCR) is a Docker image storage solution that integrates directly with Google Cloud Platform (GCP). It offers features like vulnerability scanning and seamless integration with GCP's CI/CD pipelines. While GCR is highly efficient within the Google Cloud ecosystem, its use outside of GCP environments may be limited.

We selected Docker Hub for container image management in Audieyes due to its widespread adoption, ease of use, and seamless integration with both Docker and Kubernetes. Docker Hub's large community support and extensive library of pre-built images enable efficient image distribution and deployment across various environments. While Amazon ECR and Google

Container Registry offer strong alternatives, their cloud-specific integrations and complexity made Docker Hub the preferred choice for our project's needs.

Table 11: Cloud Provider: Linode

Criteria	Linode	AWS	GCP
Overview	<p>Linode is known for its simplicity, cost-effectiveness, and developer-friendly services. It offers straightforward pricing and essential features needed for deploying and managing cloud-based applications. Linode's focus on providing accessible and reliable cloud services makes it an ideal choice for small to medium-sized projects that require flexibility without the complexity of larger cloud providers.</p>	<p>Amazon Web Services (AWS) is a comprehensive cloud platform that offers a vast array of services, including compute, storage, databases, and machine learning. AWS is highly scalable and suitable for large enterprises with complex needs. However, its extensive feature set and pricing complexity can be overwhelming for smaller projects.</p>	<p>Google Cloud Platform (GCP) provides a wide range of cloud services, particularly strong in data analytics, machine learning, and AI. GCP integrates well with other Google services and offers advanced tools for developers. However, like AWS, GCP's complexity and pricing structure may be more than what is needed for simpler projects.</p>

We opted for Linode as the cloud provider for Audieyes due to its simplicity, cost-effectiveness, developer-friendly services, and most importantly, familiarity with the platform. Linode's straightforward pricing model and essential cloud features align well with our project's requirements for a reliable and flexible deployment environment. While AWS and GCP offer powerful features, their complexity and cost considerations made Linode the preferred choice for our needs.

Table 12: Encryption Tools: CryptoJS

Criteria	brypt	CryptoJS	Node.js crypto
Overview	<p>brypt is a password hashing function that incorporates a salt to protect against rainbow table attacks. It is widely used for securing passwords due to its adaptive nature,</p>	<p>CryptoJS is a widely used cryptographic library that provides a comprehensive set of cryptographic functions, including hashing, encryption, and decryption, supporting various algorithms like</p>	<p>The Node.js crypto module is a built-in library that provides cryptographic functionality, including hashing, HMAC, cipher, decipher, sign, and verify functions. It is highly secure and efficient, with</p>

	<p>meaning it can be configured to be slower over time to remain resistant to brute-force search attacks.</p> <p>However, bcrypt is primarily focused on password hashing and is not a general-purpose cryptographic library.</p>	<p>SHA-256. It is easy to use and can be utilized on both server-side with Node.js and client-side in browsers, ensuring consistency across the application. CryptoJS's versatility and broad functionality make it an ideal choice for the encryption needs of Audieyes.</p>	<p>a wide range of cryptographic tools at the developer's disposal. However, it is more low-level compared to libraries like CryptoJS and might require more implementation effort to achieve the same results.</p>
--	---	---	---

We selected CryptoJS for the encryption needs of Audieyes due to its comprehensive set of cryptographic functions, including SHA-256, which are essential for ensuring the security of both client-side and server-side data. CryptoJS offers a consistent and versatile solution that can be used across different parts of the application, simplifying the development process. While bcrypt is excellent for password hashing, it is not a general-purpose cryptographic library. The Node.js crypto module, although powerful, requires more implementation effort compared to the straightforward usage of CryptoJS.

Table 13: Monitoring and Visualization: Grafana

Criteria	Grafana	Prometheus	Nagios
Overview	<p>Grafana is a powerful open-source tool for monitoring and visualization, capable of displaying real-time data from various sources. It provides customizable dashboards, alerts, and a wide range of plugins, making it an ideal choice for monitoring complex systems. Grafana's integration with Prometheus, InfluxDB, and other data sources allows for comprehensive monitoring and visualization across various metrics and logs.</p>	<p>Prometheus is an open-source monitoring and alerting toolkit primarily used for monitoring metrics and generating alerts. It is particularly strong in monitoring time-series data and provides powerful query capabilities. While Prometheus excels in data collection and alerting, it is often paired with Grafana for more advanced visualization.</p>	<p>Nagios is a widely-used monitoring tool that focuses on network monitoring and infrastructure management. It provides robust alerting and reporting features, but its interface and visualization capabilities are more limited compared to Grafana. Nagios is well-suited for basic monitoring needs but may not offer the same level of customization and integration as Grafana.</p>

We chose Grafana for monitoring and visualization in Audieyes due to its powerful capabilities in displaying real-time data, customizable dashboards, and wide range of plugins. Grafana's ability to integrate with various data sources, including Prometheus and InfluxDB, allows for comprehensive monitoring across our systems. While Prometheus is strong in data collection, its visualization capabilities are limited, making Grafana the preferred choice. Nagios, while effective for basic monitoring, lacks the advanced visualization features needed for our project.

7.2 DevOps

Audieyes employs a comprehensive DevOps pipeline to ensure rapid, reliable, and high-quality software delivery. Our workflow integrates Continuous Integration and Continuous Deployment (CI/CD) practices, enabling efficient development cycles and frequent, controlled releases.

Central to our development process as demonstrated in Figure 20 is the utilization of GitHub as our version control system. Developers work on feature branches, implementing new functionalities, bug fixes, and improvements in isolated environments. This approach maintains the stability of the main codebase while allowing parallel development efforts.

Our CI/CD pipeline, powered by GitHub Actions, automates crucial stages of the development lifecycle. Upon creation of a pull request, it triggers a suite of automated tests, including unit tests for individual components and integration tests to verify system-wide compatibility. This proactive testing strategy identifies potential issues early, significantly reducing the risk of defects reaching production.

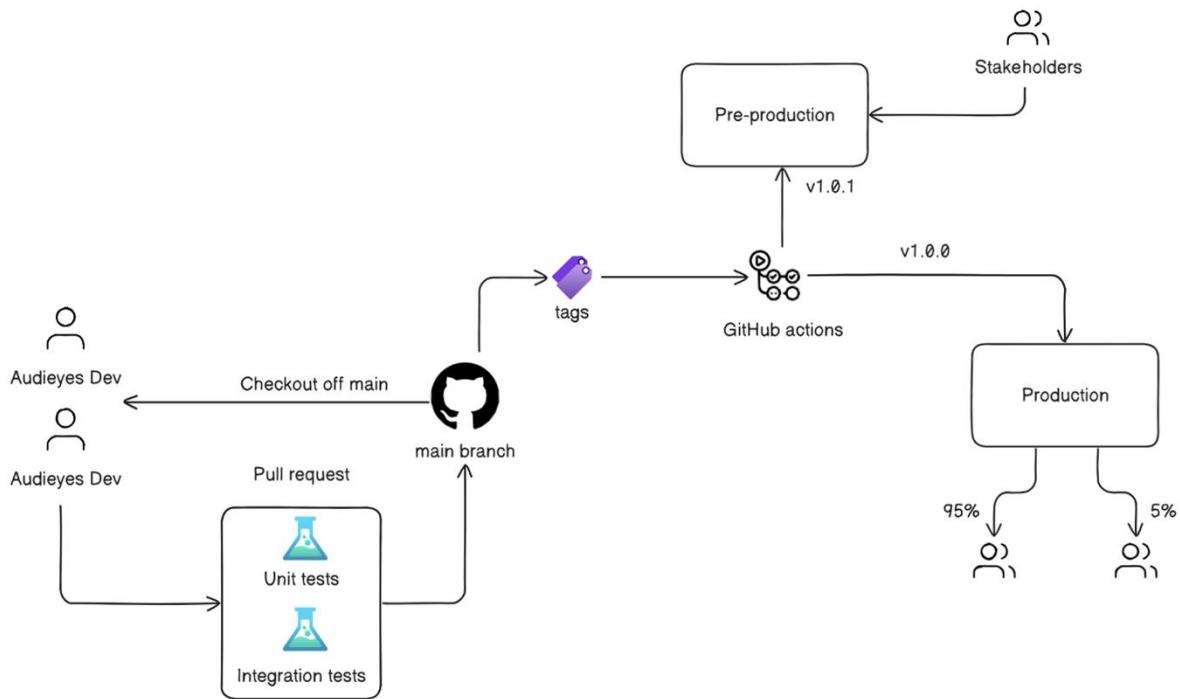


Figure 20: DevOps' CI/CD Pipeline for Audieyes

Post-merge, our pipeline automatically tags the code with a version number, facilitating precise tracking of changes and ensuring deployments are based on validated code versions. The tagged code then progresses through our deployment stages.

The pipeline first deploys to a pre-production environment, closely mirroring the production setup. Here, stakeholders conduct thorough reviews and testing, gathering valuable feedback and ensuring adherence to quality standards. Following successful validation, the code advances to production deployment.

We employ a canary release strategy for production rollouts, initially deploying to 5% of users. This approach allows for real-world testing and monitoring, enabling us to detect and address any unforeseen issues before full deployment. If no issues arise, the release is gradually extended to the entire user base.

This iterative workflow, combining Agile methodologies with advanced DevOps practices, ensures that Audieyes remains stable, reliable, and continuously aligned with evolving user needs and expectations. It enables us to rapidly deliver new features and improvements while maintaining high standards of quality and performance.

7.3 LLMOps

This chapter delves into the intricate world of LLMOps (Large Language Model Operations) and its application in the Audieyes project, which aims to assist BVI users in Morocco. It explores the foundations of machine learning engineering, comparing MLOps with LLMOps, and then focuses on specific aspects of LLMOps implementation in Audieyes. This chapter addresses Audieyes' data preparation, prompt engineering, A/B testing, API management, performance monitoring, and continuous learning strategies.

7.3.1 Foundations of Machine Learning Engineering

Machine Learning (ML) engineering has become a cornerstone in deploying intelligent systems, but it is fraught with challenges, particularly when scaling from development to production. Key obstacles include data quality, model performance, and the continuous need for monitoring and updating models as they interact with real-world data. This chapter explores these challenges within the context of the ML pipeline, defines MLOps, and contrasts it with LLMOps, as introduced in the previous chapter.

7.3.1.1 Challenges in ML Production

Transitioning machine learning models from development to production involves significant hurdles. Unlike traditional software development, ML systems must cope with data variability, concept drift, and the complexities of integrating models with existing systems. Ensuring that models perform consistently across diverse environments is a core challenge. These issues are exacerbated by the need for real-time inference and the rigorous demands of latency and scalability.

7.3.1.2 Machine Learning Pipeline

The ML pipeline in production encompasses several stages:

- Data Collection and Preprocessing: This involves gathering raw data, cleaning it, and transforming it into a format suitable for training models.
- Model Training: Data scientists develop models using training data, typically involving a process of selecting algorithms, tuning hyperparameters, and validating the model's performance.
- Model Deployment: After validation, the model is integrated into the production environment where it begins processing live data.
- Monitoring and Maintenance: Continuous monitoring of the model's performance is essential to detect any degradation over time, necessitating updates or retraining.

7.3.1.3 MLOps

MLOps, or Machine Learning Operations, is an emerging discipline that focuses on automating and streamlining the ML lifecycle from model development to deployment and monitoring. It extends the principles of DevOps to the machine learning domain, emphasizing collaboration between data scientists and operations teams. MLOps practices include versioning models, automating deployment, and setting up continuous integration and deployment (CI/CD) pipelines specific to ML tasks.

7.3.1.4 Comparison with LLMOps

LLMOPs, while sharing similarities with MLOps, focuses specifically on managing large language models (LLMs). Both involve data preparation, model deployment, and continuous monitoring, but LLMOPs requires specialized approaches for prompt engineering and managing the unique challenges posed by LLMs, such as their extensive computational demands and the complexity of their outputs.

Where MLOps tools often focus on general-purpose model management, LLMOPs introduces additional layers of complexity, particularly in areas like prompt engineering—a technique essential for fine-tuning model outputs to be contextually relevant and culturally sensitive. Additionally, LLMOPs emphasizes real-time performance monitoring to ensure that large-scale language models maintain accuracy and efficiency over time.

7.3.1 Data Preparation and Management

Data preparation and management is the process of collecting, cleaning, and organizing data for use in machine learning models. Audieyes employs a comprehensive approach to data management, focusing on user-centric collection and robust processing. With user consent, we gather diverse data including images, queries, and feedback from BVI users in Morocco through our mobile application. This data undergoes thorough cleaning and preprocessing to ensure quality and privacy. Our cleaning process involves removing sensitive information, particularly blurring faces in images, while preserving essential visual details.

For storage and versioning, Audieyes uses a hybrid system. Amazon S3 buckets serve as primary storage for the large data volumes of images and text. Complementing this, we use PostgreSQL to manage metadata and version information. This database stores references to S3 objects, including identifiers, timestamps, and version numbers.

7.3.2 Prompt Engineering

Prompt engineering is a technique involving the design and optimization of input prompts to elicit desired responses from large language models. This process is crucial for fine-tuning model outputs to be contextually relevant and aligned with specific use cases or applications.

These base prompts were designed to address key aspects of scene description vital for BVI users. They emphasize spatial awareness, cultural context, safety considerations, and adaptable detail levels. Importantly, they incorporate instructions for recognizing Moroccan cultural elements and using Moroccan Darija terms where appropriate.

Typical initial prompts reads:

Table 14 : Audieys features prompts in Darija, English, Arabic, and French,

Language	Prompt
Darija – Image Description	كمساعد اصطناعي لواحد الشخص كيف فالمنطقة، حل الصورة وعطي وصف واضح ومختصر بالدرجة المغاربية. ركز على الشكل العام، والعلاقات المكانية، والعناصر الخاصة بالمغرب، والعرائض المحتملة. استعمل المصطلحات المغاربية الشائعة فين ما كان مناسب. كون موضوعي وبين أي شكوك بوضوح. إلا ما كنتيش متأكد 95% بأن الجواب ديلك على السؤال صحيح، خاصك تقترح على الشخص الكيف باش يعاود يصيفط صورة أخرى من زاوية مختلفة.
English – Image Description	As an artificial assistant for a blind person in Morocco, analyze the image and provide a clear, concise description in Moroccan Darija. Focus on overall layout, spatial relationships, Moroccan-specific elements, and potential obstacles. Use common Moroccan terms where appropriate. Be objective and state any uncertainties clearly. If you are not 95% sure that your response to the question is correct, you must suggest the blind person to re-send another image from a different angle.
Arabic – Image Description	كمساعد اصطناعي لشخص في المغرب، قم بتحليل الصورة وتقديم وصف واضح وموजز. ركز على التخطيط العام، والعلاقات المكانية، والعناصر الخاصة بالمغرب، والعوائق المحتملة. استخدم المصطلحات المغاربية الشائعة حيثما كان ذلك مناسباً. كن موضوعياً وعيّاً عن أي شكوك بوضوح. إذا لم تكن متأكداً بنسبة 95% من أن إجابتك على السؤال صحيحة، يجب عليك أن تقترح على الشخص الكيف إعادة إرسال صورة أخرى من زاوية مختلفة.

French – Image Description	En tant qu'assistant artificiel pour une personne aveugle au Maroc, analysez l'image et fournissez une description claire et concise. Concentrez-vous sur la disposition générale, les relations spatiales, les éléments spécifiques au Maroc et les obstacles potentiels. Utilisez des termes marocains courants le cas échéant. Soyez objectif et énoncez clairement toute incertitude. Si vous n'êtes pas sûr à 95% que votre réponse à la question est correcte, vous devez suggérer à la personne aveugle de renvoyer une autre image sous un angle différent.
Darija - People Recognition	شوف الصورة وقارنها مع الصور ديال العائلة اللي عندك. قول لي شكون من العائلة كاين فالصورة. عطيوني تفاصيل على شكلهم وحركاتهم. ركز على الوجوه والملابس والتعبيرات. إلا ما قدرتيس تعرف شي واحد أو ما قدرتيس تشوف الوجوه مزيان، قول لي
English- People Recognition	Examine the image and compare it with the family photos you have. Tell me which family members are in the picture. Provide details about their appearance and actions. Focus on faces, clothing, and expressions. If you can't identify someone or if faces are not clearly visible, let me know.
Arabic - People Recognition	انظر إلى الصورة وقارنها بصور العائلة المتوفرة لديك. أخبرني أي من أفراد العائلة موجود في الصورة. قدم تفاصيل عن مظهرهم وأفعالهم. ركز على الوجوه والملابس والتعبيرات. إذا لم تتمكن من التعرف على شخص ما أو إذا لم تكن الوجوه واضحة، أخبرني بذلك
French - People Recognition	Examinez l'image et comparez-la avec les photos de famille que vous avez. Dites-moi quels membres de la famille sont sur la photo. Fournissez des détails sur leur apparence et leurs actions. Concentrez-vous sur les visages, les vêtements et les expressions. Si vous ne pouvez pas identifier quelqu'un ou si les visages ne sont pas clairement visibles, faites-le moi savoir.
Darija - Location Finding	شوف الصورة وقارنها مع الصور ديال الدار والغرف اللي عندك. قول لي فين بالضبط أنا فالدار. ركز على الأثاث والديكور والتفاصيل اللي كتميز هاد البلاصة. إلا كان شي حاجة جديدة أو مختلفة، بينها لي. عطيوني وصف مفصل باش نقدر نعرف المكان بالضبط.
English - Location Finding	Look at the image and compare it with the photos of the house and rooms you have. Tell me exactly where I am in the house. Focus on furniture, decor, and details that characterize this place. If there's anything new or different, point it out. Give me a detailed description so I can identify the exact location.

Arabic - Location Finding	انظر إلى الصورة وقارنها بصور المنزل والغرف المتوفرة لديك. أخبرني بالضبط أين أنا في المنزل. ركز على الأثاث والديكور والتفاصيل التي تميز هذا المكان. إذا كان هناك أي شيء جديد أو مختلف، أشر إليه. قدم لي وصفاً مفصلاً حتى أتمكن من تحديد الموقع بدقة.
French - Location Finding	Regardez l'image et comparez-la avec les photos de la maison et des pièces que vous avez. Dites-moi exactement où je me trouve dans la maison. Concentrez-vous sur les meubles, la décoration et les détails qui caractérisent cet endroit. S'il y a quelque chose de nouveau ou de différent, signalez-le. Donnez-moi une description détaillée pour que je puisse identifier l'emplacement exact.
Darija - Text Transcribing	شوف الصورة وقرأ لي أي نص مكتوب فيها. قول لي واش النص واضح ولا لا. ترجم لي النص للدارجة إلا ما كانش بالعربية. عطيني أي معلومات إضافية على شكل الكتابة والخط.
Darija - Text Transcribing	Look at the image and read any text written in it. Tell me if the text is clear or not. Give me any additional information about the style of writing and font.
Darija - Text Transcribing	انظر إلى الصورة واقرأ أي نص مكتوب فيها. أخبرني إذا كان النص واضح أم لا. قدم لي أي معلومات إضافية عن نمط الكتابة والخط.
Darija - Text Transcribing	Regardez l'image et lisez tout texte qui y est écrit. Dites-moi si le texte est clair ou non. Donnez-moi toute information supplémentaire sur le style d'écriture et la police.

This foundation ensures that from the outset, Audieyes provides culturally relevant, safe, and informative assistance to BVI users in Morocco. These prompts serve as the starting point for our continuous refinement process, which leverages user feedback and generative AI to create a dynamic, self-improving system.

An innovative, user-centric approach has also been implemented to continuously refine and optimize prompts. This method leverages direct user feedback and the power of generative AI to create a dynamic, self-improving system. Here's how the process works:

1. Feedback Collection: After each interaction, users are asked whether the AI's responses were helpful. If a user indicates that a response was not satisfactory, we gather detailed feedback about what was lacking or incorrect. This feedback is collected from all users across various tasks, such as object finding in images, scene description, and navigation assistance.

2. Feedback Aggregation: Our feedback aggregation process involves systematically compiling and organizing user feedback for specific tasks within Audieyes. For instance, all feedback related to object finding in images is grouped together. We collect this feedback over a set period, typically a week, and categorize it based on the type of task (e.g., object identification, scene description, navigation assistance). The aggregation also involves quantifying qualitative feedback. We track the frequency of specific comments or suggestions, allowing us to prioritize the most common user concerns. Additionally, we look for correlations between different pieces of feedback to uncover underlying patterns or issues that might not be apparent from individual responses.

3. AI-Driven Prompt Refinement: On a weekly basis, we employ a meta-approach of using the generative AI itself to refine its own prompts. We present the AI with the current prompt for a specific task, along with all the collected user aggregated feedback. We then ask the AI to generate improved prompts based on this information. To evaluate these new prompts, we implement a canary deployment strategy, releasing the updated prompts to a small subset of users. This allows us to gather real-world performance data and user feedback on the refined prompts before deciding whether to roll them out to all users or further refine them.

4. Human Oversight and Modification: While we leverage the AI's capabilities for prompt refinement, we maintain a crucial step of human oversight. We review the AI-generated prompts, ensuring they align with our guidelines, safety protocols, and the specific needs of our Moroccan BVI users. If necessary, we make modifications to the best prompt choices before implementation.

5. Continuous Improvement: The approved, refined prompts are then implemented in the Audieyes system. This cycle of feedback collection, AI-driven refinement, human oversight, and implementation continues regularly, creating a system that continuously evolves and improves based on real user experiences.

This approach offers several advantages:

- It directly incorporates user experiences into the development process.
- It leverages the power of AI not just in the end-user application but also in the development cycle itself.
- It allows for rapid, data-driven improvements to the system.
- It maintains human oversight to ensure safety, ethical compliance, and cultural sensitivity.

7.3.3 A/B Testing and Online Evaluation

A/B testing and online evaluation is the practice of comparing two or more versions of a model or system to determine which performs better in a live environment. They are crucial for continuously improving Audieyes. These methodologies allow us to empirically validate system enhancements and make data-driven decisions about implementing changes.

In Audieyes, we implement a robust A/B testing framework to rigorously evaluate new features, prompt engineering techniques, and model versions. Our split-testing approach involves assigning users to either a stable release, which continues to use the current system, or a treatment group (canary release), which experiences the new version under evaluation. This randomization ensures that any observed differences in performance can be directly attributed to the changes being tested, rather than to user-specific factors or external variables.

The duration of our A/B tests is two weeks. During this period, we meticulously collect both quantitative metrics and qualitative feedback to comprehensively assess the impact of the changes.

During the testing period, we collect and analyze four key real-time evaluation metrics:

- User Satisfaction Scores: Collected after each interaction, providing immediate feedback on user experience.
- Response Time: Measured to ensure performance improvements don't sacrifice speed.
- Task Completion Rates: Analyzed to determine how effectively users accomplish their goals with Audieyes.

At the test's conclusion, we conduct thorough analysis. This comprehensive approach ensures that changes implemented in Audieyes are thoroughly validated and demonstrably beneficial to our BVI users in Morocco, allowing for confident, data-driven decision-making in our development process.

7.3.4 API Management and Optimization

API management and optimization is the development, publication, documentation, and continuous improvement of APIs that allow other systems to interact with deployed ML models. This includes managing API versioning, security, and scalability to ensure efficient and reliable access to model capabilities.

In Audieyes, we employ two key strategies to optimize our API usage and reduce costs; image quality reduction and dynamic model selection.

For our optimization process, we implement image quality reduction techniques before sending images for processing. This approach significantly reduces data transfer and processing requirements without compromising the quality of scene descriptions. Our image optimization process involves:

- **Resizing:** We downscale images to a standard resolution of 512x512 pixels using the Jimp¹ library. This resolution balances detail preservation with file size reduction, ensuring that essential visual information is retained while minimizing data transfer.
- **Compression:** We apply lossy compression techniques to further reduce file size. Our compression algorithm is carefully tuned to maintain visual clarity while achieving substantial size reductions.
- **Format Optimization:** We convert images to efficient formats, which offer superior compression ratios compared to traditional formats.

```
export class ModelService {
  async generateImageCaption(files: {
    IMAGE?: Array<Express.Multer.File>;
  }): Promise<string> {
    if (!files.IMAGE || files.IMAGE.length === 0) {
      throw new BadRequestException(NO_IMAGE_FILE_PROVIDED);
    }

    const imageFile = files.IMAGE[0];
    const imagePath = join(TMP_ATTACHMENT_PATH, imageFile.filename);

    try {
      const image = await Jimp.read(imagePath);
      image.resize(512, 512).quality(70);
      const resizedImageBuffer = await image.getBufferAsync(Jimp.MIME_JPEG);

      const base64Image = resizedImageBuffer.toString(BASE_64);
    }
  }
}
```

Figure 21 : compressing images and changing their format

- Audieyes Also implements a dynamic model selection system to ensure we always use the most cost-effective API that meets our accuracy requirements. This system works as follows:
- **Performance Monitoring:** We continuously monitor the performance using Grafana². This includes gathering detailed metrics on response times, user satisfaction scores, and

¹ <https://www.npmjs.com/package/jimp>

² <https://en.wikipedia.org/wiki/Grafana>

API usage costs. The collected data is visualized using Grafana dashboards, providing real-time insights and historical analysis capabilities. By monitoring these metrics, we ensure that our APIs are performing optimally and cost-effectively.

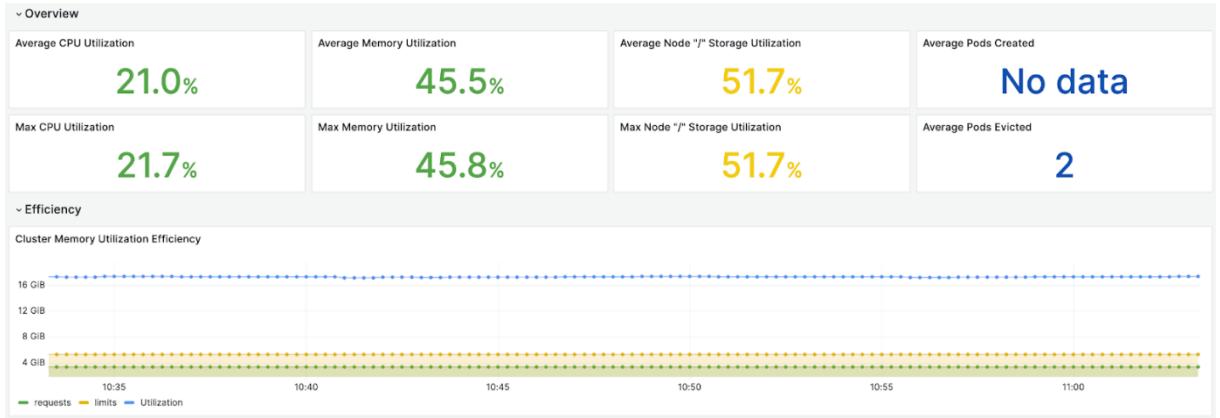


Figure 22 : Audieyes Grafana dashboard

- **Automatic Switching:** Our system automatically switches to the lowest-cost model API that consistently meets or exceeds our accuracy threshold of 95% for classifying objects within images using categories such as Tajin, couscous, souk, Caftan etc. This ensures that we're always using the most economical option without compromising on quality. The switching logic is implemented using a custom python script that evaluates the metrics collected by Grafana. It dynamically updates the Kubernetes Ingress configuration to route traffic to the selected API, ensuring seamless transitions and continuous optimal performance. This approach not only reduces operational costs but also maintains high service standards.

```
def query_grafana(query, datasource="InfluxDB"):
    ...
    ...

def evaluate_microservices():
    best_service = None
    best_cost = float('inf')

    microservices = ["openai-api", "gemini-api", "claude-api"]

    for service in microservices:
        accuracy = query_grafana(f'SELECT mean("accuracy") FROM "metrics" WHERE ("service" = '{service}') AND time >= now() - 1w')
        response_time = query_grafana(f'SELECT mean("response_time") FROM "metrics" WHERE ("service" = '{service}') AND time >= now() - 1w')
        user_satisfaction = query_grafana(f'SELECT mean("user_satisfaction") FROM "metrics" WHERE ("service" = '{service}') AND time >= now() - 1w')

        if accuracy and accuracy >= ACCURACY_THRESHOLD:
            if response_time and response_time <= RESPONSE_TIME_THRESHOLD:
                if user_satisfaction and user_satisfaction >= USER_SATISFACTION_THRESHOLD:
                    cost = API_COSTS.get(service, float('inf'))
                    if cost < best_cost:
                        best_cost = cost
                        best_service = service

    return best_service

def switch_service(best_service_name):
    ...
    ...

def monitor_and_switch():
    if __name__ == "__main__":
        monitor_and_switch()
```

Figure 23 : automatic microservice model switcher script with python

7.3.5 Performance Monitoring and Distribution Shift Analysis

Performance Monitoring is a critical process in the Audieyes project, involving the continuous tracking and analysis of various metrics related to model performance, system health, and user interactions. Given our reliance on API inference, our focus is on monitoring output quality, response times, and user satisfaction rather than internal model parameters.

Central to our monitoring system is a carefully curated fixed test dataset comprising 100 diverse images with English descriptions that were translated to Darija using GPT. These images represent a wide spectrum of scenarios encountered by our Moroccan BVI users, including various environments and culturally-specific elements. The dataset was collected from multiple sources, including Hugging Face datasets, Facebook pages, and stock image repositories, ensuring a rich variety of content.

Our selection criteria focused on capturing authentic Moroccan cultural elements and everyday scenes. Keywords used for image searches included terms like "tajine," "Moroccan street," "traditional markets," and "local architecture," among others. This approach allowed us to gather a diverse set of images that accurately represent the visual landscape encountered by BVI users in Morocco.

Figure 24: Audieyes dataset csv file

The dataset displayed in Figure 24 encompasses a wide range of image sizes and resolutions to mimic real-world variety. Each image is accompanied by detailed metadata, including categories, image ids, image paths, descriptions, and finally the generated captions.

Our monitoring process involves weekly evaluations where we send this test dataset through the LLM API. We then compare the resulting scene descriptions against pre-established ground

truth annotations we provide. This consistent testing approach allows us to detect subtle changes in performance that might indicate drift in real-world data.

Key metrics we track include overall accuracy in classifying objects within images using categories such as: Tajin, Couscous, Souk, Caftan, Souk, as well as response time, and the BLEU-4 score which measures the overlap of 4-word sequences between generated and reference captions. Our BLEU-4 score is calculated by comparing the model-generated captions with reference captions using 4-grams to assess how closely the generated text matches human language. This metric is crucial for evaluating the linguistic quality of the model's outputs.

```
import pandas as pd
from sacrebleu.metrics import BLEU

dataset = pd.read_csv('audieyes_dataset.csv')

reference_captions = dataset['description'].tolist()
generated_captions = dataset['generated_caption'].tolist()

# Initialize BLEU with max_ngram_order=4 for BLEU-4
bleu = BLEU(max_ngram_order=4)

# Calculate BLEU-4 score
bleu_score = bleu.corpus_score(generated_captions, [reference_captions])
```

Figure 25: Python script for calculating the BLEU-4 score

This python script in Figure 25 calculates the BLEU-4 score for evaluating the quality of generated captions against reference captions from a dataset. It begins by importing necessary libraries, including pandas³ for handling the dataset and BLEU from sacrebleu⁴ for calculating the BLEU score. The BLEU metric is initialized with a maximum n-gram order of 4 (for BLEU-4). Finally, the BLEU-4 score is computed by comparing the generated captions against the reference captions.

In addition to these quantitative metrics, we conduct regular data drift analyses using visual inspection tools. Our team generates and examines weekly visualizations comparing the distribution of key image characteristics between our baseline dataset and newly collected data. These visualizations, as shown in Figures 25 and 26, include distribution comparison plots for scene types (indoor vs. outdoor) and category change bar charts for cultural elements.

³ <https://pandas.pydata.org/>

⁴ <https://pypi.org/project/sacrebleu/>

```

def load_data(file_path: str) -> pd.DataFrame:
    return pd.read_csv(file_path)

def categorize_images(data: pd.DataFrame) -> pd.DataFrame: ...

def plot_distribution_comparison(baseline_data: pd.DataFrame, new_data: pd.DataFrame, feature: str): ...

def plot_category_changes(baseline_data: pd.DataFrame, new_data: pd.DataFrame, feature: str): ...

def analyze_drift_visually(baseline_data: pd.DataFrame, new_data: pd.DataFrame):
    features = ['scene_type', 'crowd_level', 'cultural_elements']

    baseline_data = categorize_images(baseline_data)
    new_data = categorize_images(new_data)

    for feature in features:
        plot_distribution_comparison(baseline_data, new_data, feature)
        plot_category_changes(baseline_data, new_data, feature)

baseline_data = load_data('baseline_image_data.csv')
new_data = load_data('new_image_data.csv')

analyze_drift_visually(baseline_data, new_data)

```

Figure 26 : Code sample for visually analyzing data drift

Figure 26 provides a code sample for generating these visual analyses. By examining these charts, we can identify significant shifts in data distribution. For instance, a recent analysis revealed a trend towards more outdoor scenes, potentially indicating growing user confidence in using Audieyes outdoors. We also observed changes in the prevalence of cultural elements, such as an increase in market scenes and a decrease in traditional architecture representations.

To detect potential distribution shifts, we compare the API's performance on our fixed test set to its performance on real-world data collected from user interactions. We quantify the distribution shift as the difference between the testing validation error (on our fixed test set) and the testing error on real-world data. This approach allows us to identify discrepancies between our controlled test environment and actual usage scenarios.

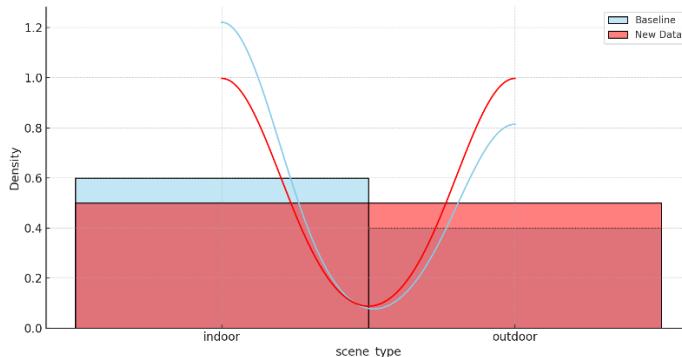


Figure 27 : Distribution Comparison: Scene Type

The Scene Type chart in Figure 27 shows two overlapping histograms. The blue represents the baseline data, and the red represents the new data. The x-axis shows the two categories: indoor and outdoor. The y-axis shows the density (proportion) of each category. In this mock data, we

can see a slight shift from more indoor scenes in the baseline data to an equal distribution of indoor and outdoor scenes in the new data.

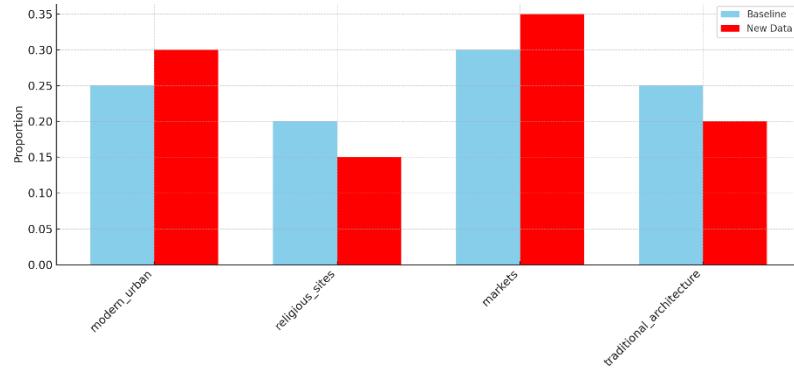


Figure 28 : Category Changes: Cultural Elements

Figure 29

For Category Changes: Cultural Elements in Figure 28, the chart is a grouped bar chart. Each group represents a cultural element category (traditional architecture, markets, religious sites, modern urban). Within each group, the blue bar represents the baseline data, and the red bar represents the new data. The y-axis shows the proportion of each category. In this mock data, we can observe slight changes in the distribution of cultural elements, such as an increase in markets and modern urban scenes, and a decrease in traditional architecture and religious sites in the new data compared to the baseline.

These visual analyses allow us to quickly spot emerging patterns that might not be immediately apparent in numerical data alone. By examining trends across different image categories, such as the shift in indoor versus outdoor scenes or changes in the representation of cultural elements, we can anticipate evolving user environments and usage patterns. This insight is crucial for making timely adjustments to our system, ensuring that Audieyes remains effective and relevant for our Moroccan BVI users across the diverse scenarios they encounter in their daily lives.

When a significant distribution shift is detected, we conduct a thorough manual error analysis. This process involves categorizing errors (e.g., misidentification, cultural misinterpretation, language inaccuracies), performing contextual analysis of errors, and assessing language use, particularly Moroccan Darija terms.

To maintain system performance and address identified shifts, we employ a continuous improvement approach tailored to our use of API inference. This involves regularly updating our test set with new, correctly labeled images that reflect evolving real-world scenarios. We

also focus on refining our API queries and post-processing techniques to better handle newly identified challenges or contexts.

In conclusion, our performance monitoring and distribution shift analysis processes are vital for ensuring Audieyes remains effective and relevant for our Moroccan BVI users. The observed changes, particularly in culturally-specific elements, highlight the need for continuous learning and adaptation. While we can't directly fine-tune the underlying model, our approach allows us to optimize our use of the API and ensure our system evolves alongside user needs and environmental changes.

7.4 Backend Development

The backend of Audieyes forms the core of our application, handling data processing, AI integration, and user requests. This section outlines our technical architecture and key design patterns.

We will explore our use of NestJS as the development framework and examine the implementation of crucial design patterns such as Decorators, Facade, Dependency Injection, Strategy, Repository, and Modularization. This overview will provide insight into the robust and flexible architecture powering Audieyes' efficient service delivery.

7.4.1 Design Patterns

7.4.1.1 *Decorators*

NestJS decorators play a crucial role in enhancing and modifying the behavior of classes, methods, and properties without altering their source code. These special functions add functionality in a clean, declarative manner, particularly useful for cross-cutting concerns like logging, validation, and authorization.

We extensively use decorators such as `@Controller`, `@Get`, `@Post`, and `@Injectable` to manage routing, request handling, and dependency injection. This approach streamlines the application's structure by clearly defining component interactions.

```

@ApiBearerAuth('Authorization')
@ApiTags('Model')
@Controller('v1')
export class ModelController {
  constructor(private readonly modelService: ModelService) {}

  @Post('scene/description')
  @ApiConsumes('multipart/form-data')
  @ApiResponse({ status: 200, description: 'Caption generated successfully' })
  @ApiResponse({ status: 400, description: 'Bad request' })
  @ApiResponse({ status: 500, description: 'Internal server error' })
  @ApiBody({
    schema: {
      type: 'object',
      properties: {
        IMAGE: {
          type: 'string',
          format: 'binary',
        },
      },
    }
  })
  @UseInterceptors(

```

Figure 30: Design Pattern: Decorators

For instance, in our API implementation, decorators like `@ApiBearerAuth`, `@ApiTags`, and `@Controller` define authorization mechanisms, tagging, and routing for the `ModelController` class. The `@Post` decorator specifies HTTP POST handling for the `scene/description` endpoint, while `@ApiConsumes` and `@ApiBody` define request body expectations. `@ApiResponse` decorators enhance API documentation by specifying possible HTTP responses.

This decorator-based approach in NestJS facilitates the creation of a modular, maintainable, and scalable backend system for Audieyes, significantly improving development efficiency and code robustness.

7.4.1.2 Facade

We also implement the Facade pattern to simplify interactions with complex subsystems through a unified interface. This approach is particularly beneficial for our intricate system, allowing clients to interact via a simplified and consistent API, thereby reducing complexity and improving usability.

Our implementation of the Facade pattern is exemplified by the `ModelService` injection. This class serves as a single entry point for client requests related to image caption generation, encapsulating the complexities of interacting with the `ModelService`, which handles the core caption generation logic.

```

export class ModelController {
    constructor(private readonly modelService: ModelService) {}

    async generateCaption(
        @UploadedFiles(new ParseFilePipeDocument())
        files: {
            IMAGE?: Array<Express.Multer.File>;
        },
    ) {
        return {
            caption: await this.modelService.generateImageCaption(files),
        };
    }
}

```

Figure 31: Design Pattern: Facade

The ModelService exposes a straightforward generateImageCaption method, providing a clear and simple interface for clients. We can use all the methods from the modelService without needing to import each one of them only by calling their service.

This facade-based approach not only simplifies client interactions but also enhances the backend's maintainability and scalability by centralizing and standardizing code handling logic. It represents a key architectural decision in the Audieyes system, contributing to its robust and user-friendly design.

7.4.1.3 Dependency Injection and Inversion of Control

Dependency Injection (DI) and Inversion of Control (IoC) are crucial design patterns that promote loose coupling and enhance the modularity and testability of an application. Dependency Injection involves providing a class with its dependencies rather than the class creating them itself. Inversion of Control refers to the reversal of the flow of control, where the framework takes over the control flow from the application.

```

@Injectable()
export class ModelService {
  private openai: OpenAI;
  private prisma: PrismaService;

  constructor(prisma: PrismaService) {
    this.openai = new OpenAI({
      apiKey: process.env.OPENAI_API_KEY,
    });
    this.prisma = prisma;
  }

  async generateImageCaption(files: {
    IMAGE?: Array<Express.Multer.File>;
  }): Promise<string> {
    if (!files.IMAGE || files.IMAGE.length === 0) {
      throw new BadRequestException(NO_IMAGE_FILE_PROVIDED);
    }
  }
}

```

Figure 32: Dependency Injection and Inversion of Control

The `ModelService` class demonstrates these patterns effectively. The `@Injectable` decorator marks the class as a provider that can be injected into other classes. The constructor of `ModelService` receives instances of `OpenAI` and `PrismaService`, which are injected by the NestJS framework. This setup decouples the `ModelService` from the instantiation logic of its dependencies, allowing for more flexible and maintainable code.

For instance, the `OpenAI` instance is created with a specific API key, and the `PrismaService` instance is passed directly into the `ModelService` constructor. This approach ensures that `ModelService` does not need to manage the creation and configuration of these dependencies, adhering to the principle of Inversion of Control. This makes it easier to manage dependencies, swap implementations, and write unit tests for the `ModelService` class without requiring complex setup or modifications to the service itself.

By utilizing DI and IoC, NestJS promotes a clean and maintainable codebase, allowing developers to focus on the business logic rather than the intricate details of dependency management. This pattern is central to building scalable and testable applications, as it simplifies the integration and management of various components within the application.

7.4.1.4 Strategy

The Strategy pattern is a behavioral design pattern that allows you to define a family of algorithms, encapsulate each one, and make them interchangeable. This pattern enables the

selection of an algorithm at runtime, providing flexibility and decoupling the algorithm from the client that uses it.

```
@Injectable()
export class JwtStrategy extends PassportStrategy(Strategy, 'jwt') {
  constructor(
    private userRepository: UserRepository,
    config: ConfigService,
  ) {
    super({
      jwtFromRequest: ExtractJwt.fromAuthHeaderAsBearerToken(),
      secretOrKey: config.get('JWT_SECRET'),
      passReqToCallback: true,
    });
  }

  async validate(req, payload, done) {
    const user = await this.userRepository.find(payload.userId);
    if (!user) {
      return done(null, false);
    }
    delete user.password;
    return user;
  }
}
```

Figure 33: Design Pattern: Strategy

The Strategy pattern is implemented through the JwtStrategy class, which extends PassportStrategy. The JwtStrategy class is designed to handle JWT authentication by extracting the token from the request header and validating it against the secret key. The validate method is a key part of this strategy, where it uses the userRepository to find and validate the user based on the JWT payload. This approach allows for the encapsulation of the authentication logic, making it easier to manage and extend. The use of the Strategy pattern here ensures that different authentication mechanisms can be swapped in and out without affecting the overall system architecture, enhancing flexibility and maintainability.

7.4.1.5 Repository

The Repository pattern emerges as a pivotal architectural concept in software design, offering a structured approach to data access and management. This pattern effectively decouples the business logic from the intricacies of data persistence, promoting a more modular and maintainable codebase. In the context of the provided images, we observe a clear implementation of this pattern in a TypeScript environment, likely within a larger application framework.

```

export interface UserRepositoryInterface {
    find(id: number): Promise<User>;
    create(data: ICreateUser): Promise<User>;
    findByEmail(email: string): Promise<User>;
    emailUnique(email: string): Promise<void>;
    findAll(paginationArgs: PaginationArgs): Promise<IPaginatedResponse>;
}

```

Figure 34: Repository Interface of users.

Figure 34 presents the cornerstone of the Repository pattern: an interface defining the contract for user-related data operations. This UserRepositoryInterface outlines a set of methods that any concrete implementation must fulfill, including operations for finding, creating, and querying user data. By establishing this interface, the pattern ensures a consistent API for user data interactions across the application, regardless of the underlying data source.

```

@Injectable()
export class UserRepository implements UserRepositoryInterface {
    constructor(private prisma: PrismaService) {}
    async findAll(...): ...
    async emailUnique(email: string): Promise<void> { ... }
    async findByEmail(email: string): Promise<User> { ... }
    async find(id: number): Promise<User> { ... }
    async create(data: ICreateUser): Promise<User> { ... }
}

```

Figure 35: Implementation of Repository Interface of users.

Moving to Figure 35, we see the realization of the aforementioned interface in the form of a UserRepository class. This class brings the abstract concepts defined in the interface into concrete implementation. The use of dependency injection, as evidenced by the constructor parameter, suggests a flexible and testable design. While the specific details of each method implementation are not visible, the structure adheres closely to the interface contract, demonstrating the pattern's effectiveness in maintaining consistency between the abstraction and its concrete form.

```
You, 2 weeks ago | 1 author (You)
@Injectable()
export class UserService {
  constructor(private repository: UserRepository) {}

  async signup(userPayload: ICreateUser): Promise<User> {
    const { email, password } = userPayload;

    await this.repository.emailUnique(email);

    const payload: ICreateUser = {
      email,
      password: await hash(password),
      ...userPayload,
    };

    const user = await this.repository.create(payload);
    delete user.password;
    return user;
  }

  async getMe(userId: number) {
    const user = await this.repository.find(userId);
    delete user.password;
    return user;
  }
}
```

Figure 36: Injecting and calling the repository of user function.

Figure 36 illuminates the practical application of the Repository pattern within a service layer. Here, a UserService class leverages the UserRepository to perform higher-level operations such as user registration and retrieval. This separation of concerns is a hallmark of the Repository pattern, allowing the service layer to focus on business logic while delegating data access responsibilities to the repository. That way if we need to change major components of the system such as the database, the application logic will remain the same and only the repository functionality that will change.

7.4.1.6 Modules

Modularization is a design pattern that structures applications into discrete, self-contained units. This approach enhances code organization, reusability, and maintainability. The provided images illustrate its implementation in a modern framework, likely using dependency injection.

```
@Module({
  controllers: [ModelController],
  providers: [PrismaService, ModelService, ModelRepository],
  exports: [ModelService],
})
export class ModelModule {}
```

Figure 37: Model module where all core logic is imported and exported.

Figure 37 shows a ModelModule, encapsulating related components such as controllers, services, and repositories. This grouping creates a cohesive unit of functionality, simplifying management and potential reuse.

```
@Module({
  imports: [
    ConfigModule.forRoot({
      isGlobal: true,
    }),
    AuthModule,
    UserModule,
    ModelModule,
    PrismaModule,
  ],
  providers: [
    {
      provide: APP_GUARD,
      useClass: JwtAuthGuard,
    },
  ],
})
export class AppModule {}
```

Figure 38: App module, where all modules of the Audieyes system are used.

Figure 38 depicts the entire AppModule, demonstrating how modularization scales to the application level. It imports various modules, including authentication, and the AI model modules. This structure allows for efficient composition of larger systems from smaller, manageable pieces.

7.5 API Documentation and Logging

The integration of APIs in the Audieyes system is a critical component that facilitates seamless interaction between different functionalities, enabling a cohesive and efficient user experience. This section outlines the various APIs utilized in the system, detailing their purposes and the specific endpoints available for interaction. The API documentation is designed to provide a comprehensive understanding of how the APIs work, ensuring that developers and users can leverage them effectively.

1. Scene Description	API	
	POST /v1/scene/description	Retrieve a real-time description of the scene captured by the user's camera.
2. Object Recognition and Identification	API	
	POST /v1/object/recognize	Submit an image to receive detailed object recognition results.
3. Save Family Members	API	
	POST /v1/family/recognize	Upload a picture to recognize close family members.
	POST /v1/family/<relation>	Upload a picture of your close family members to the system.
	PATCH /v1/family/<relation>	Change the picture of your family members.
	DELETE /v1/family/<relation>	Delete the picture of your family members from the system
5. Save Locations	API	
	POST /v1/location/recognize	Upload a picture to recognize a saved location (house room, street corner)
	POST /v1/location/<type>	Upload a picture of your favorite locations you want to save
	PATCH /v1/location/<type>	Change the picture of your saved location
	DELETE /v1/location/<type>	Delete the picture of your saved location
5. Text Captioning	API	
	POST /v1/text/caption	Submit an image to receive text recognition results.

Figure 39: API Endpoints for various starting features of Audieyes

API documentation for Audieyes in Figure 39 provides a detailed automated documentation using Swagger that displays information on each endpoint, including request and response formats, authentication requirements, and error handling. This is a valuable feature for developers who will have fully detailed documentation of each endpoint, with the methods, request parameters, request body, and expected responses.



Figure 40: Swagger UI for Audieyes Backend API

Logging is yet another important factor of API monitoring. Every API request and response will be logged as shown in Figure 40, capturing critical information such as the endpoint accessed, the response status, the timestamp and other important data. These logs will be stored securely and analyzed to identify usage patterns, detect anomalies, and trace issues. By maintaining detailed logs, it is possible to perform retrospective analyses to understand the root causes of any problems that arise.

```
[Nest] 18212 - 07/22/2024, 4:54:21 PM    LOG [InstanceLoader] ModelModule dependencies initialized +0ms
[Nest] 18212 - 07/22/2024, 4:54:21 PM    LOG [RoutesResolver] AuthController {/auth}: +51ms
[Nest] 18212 - 07/22/2024, 4:54:21 PM    LOG [RouterExplorer] Mapped {/auth/signin, POST} route +2ms
[Nest] 18212 - 07/22/2024, 4:54:21 PM    LOG [RoutesResolver] UsersController {/user}: +1ms
[Nest] 18212 - 07/22/2024, 4:54:21 PM    LOG [RouterExplorer] Mapped {/user/signup, POST} route +0ms
[Nest] 18212 - 07/22/2024, 4:54:21 PM    LOG [RouterExplorer] Mapped {/user, GET} route +1ms
[Nest] 18212 - 07/22/2024, 4:54:21 PM    LOG [RouterExplorer] Mapped {/user/me, GET} route +0ms
[Nest] 18212 - 07/22/2024, 4:54:21 PM    LOG [RoutesResolver] ModelController {/v1}: +0ms
[Nest] 18212 - 07/22/2024, 4:54:21 PM    LOG [RouterExplorer] Mapped {/v1/scene/description, POST} route +1ms
[Nest] 18212 - 07/22/2024, 4:54:21 PM    LOG [RouterExplorer] Mapped {/v1/object/recognition, POST} route +1ms
[Nest] 18212 - 07/22/2024, 4:54:21 PM    LOG [RouterExplorer] Mapped {/v1/family/recognize, POST} route +0ms
[Nest] 18212 - 07/22/2024, 4:54:21 PM    LOG [RouterExplorer] Mapped {/v1/family/:relation, POST} route +1ms
[Nest] 18212 - 07/22/2024, 4:54:21 PM    LOG [RouterExplorer] Mapped {/v1/location/recognize, POST} route +0ms
[Nest] 18212 - 07/22/2024, 4:54:21 PM    LOG [RouterExplorer] Mapped {/v1/location/:type, POST} route +1ms
[Nest] 18212 - 07/22/2024, 4:54:21 PM    LOG [RouterExplorer] Mapped {/v1/attachment/:id, POST} route +1ms
```

Figure 41: Logging Endpoints for Monitoring in Audieyes API

7.6 Testing

Testing is a fundamental aspect of software development, crucial for ensuring the reliability and functionality of the Audieyes application. For a system designed for BVI users, the importance of robust, error-free code cannot be overstated. For Audieyes, units as well as integration tests have been implemented.

Figure 41 provides an overview of a unit test file structure for the Model Controller. This structure reflects our comprehensive approach to testing. It allows us to test various scenarios, including normal operation, error handling, and edge cases. By covering these different aspects, we ensure that our Model Controller behaves correctly under various conditions, critical for maintaining the reliability of Audieyes.

```

6
9   describe('ModelController', () => {
10     let controller: ModelController;
11     let modelService: ModelService;
12
13     beforeEach(async () => {
14       ...
15     });
16
17     it('1. should be defined', () => {
18       ...
19     });
20
21     describe('generateCaption', () => {
22       it('2. should call modelService.generateImageCaption and return the result', async () => {
23         ...
24       });
25
26       it('3. should throw BadRequestException if no image is provided', async () => {
27         ...
28       });
29
30       it('4. should handle multiple images and use the first one', async () => {
31         ...
32       });
33
34       it('5. should throw InternalServerErrorException if service throws an error', async () => {
35         ...
36       });
37     });
38   });
39
40 });

```

Figure 42 : test suit set up for model controller

In Figure 42, we simulate a scenario where the generateImageCaption method throws an error. We use Jest's mocking capabilities to create a controlled testing environment. The test creates mock file data and uses Jest's spyOn method to monitor calls to the service method, forcing it to reject with a simulated error.

The test then asserts that when the controller's Caption generation method is called with these mocked files, it should throw an Internal Server Error Exception. This ensures that our controller properly propagates and handles errors from the underlying service layer, a critical aspect of maintaining a robust and user-friendly application.

This output confirms that not only does our error handling work as expected, but all other aspects of the ModelController and related components are functioning correctly as well.

```

9   describe('ModelController', () => {
35     describe('generateCaption', () => {
36       it('2. should call modelService.generateImageCaption and return the result', async () => {
37         const mockFiles: { IMAGE?: Express.Multer.File[] } = {
43           ],
44         };
45         const mockCaption: string = 'This is a test caption';
46         jest
47           .spyOn(modelService, 'generateImageCaption')
48           .mockResolvedValue(mockCaption);
49         const result = await controller.generateCaption(mockFiles);
50         expect(modelService.generateImageCaption).toHaveBeenCalledWith(mockFiles);
51         expect(result).toEqual({ caption: mockCaption });
52       });
53     });

```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS COMMENTS

```

PASS  src/models/model/tests/model.controller.spec.ts
PASS  src/models/model/tests/openai.service.spec.ts

Test Suites: 2 passed, 2 total
Tests:      10 passed, 10 total
Snapshots:  0 total
Time:      3.447 s, estimated 4 s
Ran all test suites.
Done in 4.43s.

```

Figure 43 : Test Implementation example

By maintaining a robust test suite, we ensure the ongoing reliability and stability of the application. As the application evolves and new features are added, this testing framework provides confidence in the system's integrity and helps maintain its high standards of quality and dependability.

7.7 Cloud Deployment

This section delves into the comprehensive strategy for deploying Audieyes in the cloud, encompassing containerization, database replication, canary release, and multi-region deployment. These methodologies collectively enhance the application's reliability, scalability, and performance, ensuring consistent operation across diverse environments.

7.7.1 API Containerization, Duplication, and Orchestration

Audieyes' API is containerized using Docker to encapsulate the application and its dependencies within a standardized unit. This approach ensures that the application runs consistently across various environments, from development to production. By leveraging Docker, we enhance the portability, scalability, and ease of management of Audieyes, enabling efficient deployment and operation on any platform that supports Docker containers.

```

Dockerfile X docker-compose.yml
Dockerfile > ...
1  FROM node:18-alpine AS build
2
3  WORKDIR /app
4
5  COPY package*.json ./ 
6
7  RUN npm install
8
9  COPY . .
10
11 RUN npm run build
12
13 FROM node:18-alpine
14
15 WORKDIR /app
16
17 COPY --from=build /app/dist ./dist
18 COPY --from=build /app/node_modules ./node_modules
19 COPY --from=build /app/package*.json .
20
21 EXPOSE 3000
22
23 CMD ["node", "dist/main.js"]
24

```

Figure 44: Audieyes' API Dockerfile

The provided Dockerfile in Figure 44 is used to containerize the Audieyes application using Docker. It employs a multi-stage build process to optimize the final image size. Dependencies are installed, the application is then built, and pushed to docker hub with a specific tag.

Following that, we use Kubernetes to orchestrate our containerized application. Kubernetes provides a robust platform for automating deployment, scaling, and management of containerized applications. In our deployment strategy, we initially set up three replicas of our API container, ensuring redundancy and high availability. Our Kubernetes configuration includes:

- **A Deployment that defines the desired state for our API containers.** This is seen in the audieyes-api-stable Deployment configuration in Figure 45, where we specify three replicas of the API container. This ensures that there are always three instances of the container running, providing load balancing and failover capabilities. The deployment ensures that the desired state of the application is maintained, automatically replacing any failed containers and scaling the application as needed.
- **A Service that exposes the API to the cluster or externally, depending on our networking configuration.** The service configuration defines how the API containers are exposed to the network. It includes the definition of a port that maps to the port on the container, facilitating communication between the containers and other services or users.

- A canary deployment strategy to gradually introduce new features and changes.

By setting up a canary deployment, we can direct a small percentage of traffic (5%) to a new version of the API, while the majority of traffic continues to go to the stable version. This approach allows us to test new features in a production environment with minimal risk. If the canary version performs well, we can gradually increase its traffic share. Conversely, if issues are detected, we can quickly roll back to the stable version without significant impact on the users.

```
2 apiVersion: apps/v1
3 kind: Deployment
4 metadata:
5   name: audieyes-api-stable
6   labels:
7     app: audieyes-api
8     version: stable
9 spec:
10  replicas: 3
11  selector:
12    matchLabels:
13      app: audieyes-api
14      version: stable
15  template:
16    metadata:
17      labels:
18        app: audieyes-api
19        version: stable
20  spec:
21    containers:
22      - name: audieyes-api
23        image: ayoubmaimmadi/audieyes-api:stable
24        ports:
25          - containerPort: 80
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
159
160
161
162
163
164
165
166
167
168
169
169
170
171
172
173
174
175
176
177
178
179
179
180
181
182
183
184
185
186
187
188
189
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
209
210
211
212
213
214
215
216
217
218
219
219
220
221
222
223
224
225
226
227
228
229
229
230
231
232
233
234
235
236
237
238
239
239
240
241
242
243
244
245
246
247
248
249
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
288
289
289
290
291
292
293
294
295
296
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
```

Figure 45: API Kubernetes stable and canary deployment configurations

Figure 45 showcases a Kubernetes Deployment configuration for the Audieyes API. It defines the stable deployment as well as a canary one, specifying three replicas to ensure high availability for the stable deployment, and only one for the canary deployment.

Figure 46 defines a Kubernetes Service configuration for the Audieyes API. These Services are of type LoadBalancer, which enables external access to the API. It selects pods labeled with "app: audieyes-api" and the version (stable or canary), routing traffic to them. This setup effectively creates a stable endpoint for accessing the API, distributing incoming traffic across multiple pod replicas and facilitating external connectivity.

```
apiVersion: v1
kind: Service
metadata:
  name: audieyes-api-stable
spec:
  selector:
    app: audieyes-api
    version: stable
  ports:
    - protocol: TCP
      port: 80
      targetPort: 3000
```

```
apiVersion: v1
kind: Service
metadata:
  name: audieyes-api-canary
spec:
  selector:
    app: audieyes-api
    version: canary
  ports:
    - protocol: TCP
      port: 80
      targetPort: 3000
```

Figure 46: API Kubernetes Service

7.7.2 Canary Deployment Configuration with Kubernetes

A canary deployment is a strategy to gradually introduce new versions of an application to a small subset of users before rolling it out to the entire user base. This approach helps in mitigating the risk of deploying new features by first exposing them to a limited audience, allowing for the identification and resolution of any issues or bugs in a controlled environment.

The canary deployment strategy enables progressive delivery and validation of new features. This is implemented through the audieyes-api-canary Deployment, which runs a single replica of the canary version of the API. The canary deployment allows us to test new features or changes with a small subset of traffic before rolling them out to the entire user base. This approach helps in identifying any issues or bugs in the new version without affecting the majority of users.

```

apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: audieyes-api-ingress
annotations:
  kubernetes.io/ingress.class: nginx
  nginx.ingress.kubernetes.io/canary: "true"
  nginx.ingress.kubernetes.io/canary-weight: "5"
spec:
  rules:
  - host: audieyes
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: audieyes-api-canary
            port:
              number: 80

```

Figure 47: API Kubernetes canary deployment with Ingress

Figure 47 defines the Kubernetes Ingress configuration for managing traffic routing between the stable and canary deployments. The ingress for the canary version includes annotations that specify it as a canary ingress and set a canary weight of 5%, meaning that 5% of the traffic is directed to the canary deployment while the remaining 95% goes to the stable deployment. This controlled rollout helps in testing the new version under real-world conditions with minimal risk.

7.7.3 Database Replication with Kubernetes

Replicating databases in a Kubernetes environment presents several challenges due to the need for data consistency, persistence, and high availability. Unlike stateless applications, databases must retain their state across pod restarts, scaling events, and even across different nodes in the cluster. Achieving reliable data replication and synchronization requires careful orchestration to prevent data loss and ensure consistency. Kubernetes StatefulSets offer a robust solution to these challenges by providing unique, stable network identities and persistent storage for each pod, enabling seamless data replication and consistent storage management across the cluster. This ensures that each database instance retains its state, facilitating reliable data replication and high availability.

Figure 48 shows the creation of Kubernetes cluster with postgresql and then we deploy a namespace to hold our resources using the command `kubectl create ns postgresql`

```
/workspaces/audieyes-kubernetes # kind create cluster --name postgresql --image kindest/node:v1.28.0
Creating cluster "postgresql" ...
✓ Ensuring node image (kindest/node:v1.28.0) ...
✓ Preparing nodes 📦
✓ Writing configuration 📄
✓ Starting control-plane 🏰
✓ Installing CNI 🚧
✓ Installing StorageClass 🏷️
Set kubectl context to "kind-postgresql"
You can now use your cluster with:

kubectl cluster-info --context kind-postgresql

Have a question, bug, or feature request? Let us know! https://kind.sigs.k8s.io/#community 😊
/workspaces/audieyes-kubernetes #
/workspaces/audieyes-kubernetes #
```

Figure 48 : Creation of postgresql Kubernetes cluster

We also need to store various sensitive pieces of information, such as the PostgreSQL user credentials and replication settings. We use the `kubectl create secret` command to create a Kubernetes Secret object. Here's how we can create a Secret for our PostgreSQL instance:

```
kubectl -n postgresql create secret generic postgresql \
--from-literal POSTGRES_USER="audieyesadmin" \
--from-literal POSTGRES_PASSWORD='audieyes123' \
--from-literal POSTGRES_DB="postgresdb" \
--from-literal REPLICATION_USER="replicationuser" \
--from-literal REPLICATION_PASSWORD='replicationPassword'
```

Figure 49 : Kubernetes Secret object creation

In this command:

- `-n postgresql` specifies the namespace where the Secret will be created.
- `generic postgresql` creates a generic Secret with the name `postgresql`.
- `--from-literal` allows us to specify key-value pairs directly from the command line.

Once the Secret is created, we reference it in our pod specification so that our PostgreSQL containers can access these sensitive values securely. In the StatefulSet configuration, we specify environment variables and use `valueFrom` to reference the Secret keys.

The ConfigMap is used to store PostgreSQL configuration files, which are then mounted into the pods. This approach allows for easy updates and management of configuration settings without needing to rebuild the container image.

```

storage > databases > postgresql > audieyes-k8s > yaml > ✘ statefulset.yaml
  1  apiVersion: v1
  2  kind: ConfigMap
  3  metadata:
  4    name: postgres
  5  data:
  6    pg_hba.conf: |+
  7      # TYPE DATABASE      USER          ADDRESS          METHOD
  8      host   replication  replicationuser  0.0.0.0/0    md5
  9      # "local" is for Unix domain socket connections only
10      local  all         all            trust
11      # IPv4 local connections:
12      host   all         all            127.0.0.1/32  trust
13      # IPv6 local connections:
14      host   all         all            ::1/128       trust
15      # Allow replication connections from localhost, by a user with the
16      # replication privilege.
17      local  replication  all           trust
18      host   replication  all           127.0.0.1/32  trust
19      host   replication  all           ::1/128       trust
20
21      host all all all scram-sha-256
22    postgresql.conf: |+
23      data_directory = '/data/pgdata'
24      hba_file = '/config/pg_hba.conf'
25      ident_file = '/config/pg_ident.conf'
26
27      port = 5432
28      listen_addresses = '*'

```

Figure 50 : Kubernetes ConfigMap configuration for postgresql

This part of the configuration file is essential for setting up PostgreSQL in a Kubernetes environment using a ConfigMap and a StatefulSet as shown in Figure 48. The ConfigMap named `postgres` stores the PostgreSQL configuration files, including `pg_hba.conf` and `postgresql.conf`. The `pg_hba.conf` file configures the client authentication settings for PostgreSQL. It specifies the types of connections allowed, the databases, users, and addresses permitted, and the authentication methods used. This configuration includes settings for replication users, local and remote connections, and defines the authentication method. The `postgresql.conf` file contains the main PostgreSQL configuration settings. It specifies various operational parameters such as the data directory, the locations of the `pg_hba.conf` and `pg_ident.conf` files, the port, and the addresses PostgreSQL listens to. It also sets the maximum number of connections, memory allocation, WAL settings for replication, logging configurations, and locale settings. Additionally, it includes replication settings like `wal_level`, `archive_mode`, `archive_command`, and `max_wal_senders`, which are crucial for ensuring that the PostgreSQL instance can handle replication effectively.

```

storage > databases > postgresql > audieyes-k8s > yaml > ➔ statefulset.yaml
  52   apiVersion: apps/v1
  53   kind: StatefulSet
  54   metadata:
  55     name: postgres
  56   spec:
  57     selector:
  58       matchLabels:
  59         app: postgres
  60     serviceName: "postgres"
  61     replicas: 3
  62     template:
  63       metadata:
  64         labels:
  65           app: postgres
  66       spec:
  67         terminationGracePeriodSeconds: 30
  68         initContainers:
  69           - name: init
  70             image: postgres:15.0
  71             command: [ "bash", "-c" ]
  72             args:
  73               - |
  74                 #create archive directory
  75                 mkdir -p /data/archive && chown -R 999:999 /data/archive
  76             volumeMounts:
  77               - name: data
  78                 mountPath: /data
  79                 readOnly: false
  80             containers:
  81               - name: postgres
  82                 image: postgres:15.0
  83                 args: ["-c", "config_file=/config/postgresql.conf"]
  84                 ports:
  85                   - containerPort: 5432
  86                     name: database
  87                 env:
  88                   - name: PGDATA
  89                     value: "/data/pgdata"
  90                   - name: POSTGRES_USER
  91                     valueFrom:
  92                       secretKeyRef:
  93                         name: postgresql
  94                         key: POSTGRES_USER
  95                         optional: false

```

Figure 51 : Kubernetes StatefulSet configuration

Figure 51 showcases the configuration file that defines a StatefulSet for deploying PostgreSQL in a Kubernetes environment. The StatefulSet ensures that each PostgreSQL pod maintains a unique and stable identity, which is crucial for managing stateful applications.

The configuration begins with specifying the API version as apps/v1 and the kind as StatefulSet. The metadata section names the StatefulSet postgres. Within the spec section, the selector matches pods with the label app: postgres, ensuring that these pods are managed by this StatefulSet. The serviceName is set to "postgres," which corresponds to the headless service managing the network identities of the pods. The replicas field is set to three, meaning that three instances of PostgreSQL will be created.

The initContainers section defines an initialization container named init, which uses the postgres:15.0 image. This container executes a command to create an archive directory and set appropriate permissions, ensuring that the necessary directory structure is in place before the primary PostgreSQL container starts.

The containers section defines the main PostgreSQL container. It is named `postgres` and also uses the `postgres:15.0` image. The `args` field specifies the location of the configuration file using the `-c` flag. The container listens on port 5432 for PostgreSQL connections.

The `volumeMounts` section mounts volumes into the container. The `config` volume mounts the `ConfigMap` containing PostgreSQL configuration files at `/config`, and the `data` volume mounts a persistent volume at `/data` for storing PostgreSQL data. This setup ensures that the configuration files and data are properly managed and persist across pod restarts. Meaning if a pod fails and gets replaced, it gets started on the same volume which contains the data.

7.7.4 Multi-Region Deployment with Linode

Audieyes follows a multi-region deployment strategy using Linode, ensuring high availability and fault tolerance. Linode is a cloud hosting provider known for its reliable and scalable infrastructure services. It offers virtual servers, known as Linodes, which can be deployed across various data centers worldwide. By leveraging Linode's robust infrastructure, Audieyes enhances its application's resilience and performance.

In this strategy, identical Kubernetes configurations are pushed to different Linode regions (Paris-FR, and Madrid-ES) as shown in Figure 52. Each region operates independently but shares the same application setup, allowing for seamless transitions between regions in case of a failure. This approach ensures that the application remains available and functional, even if one region experiences issues.

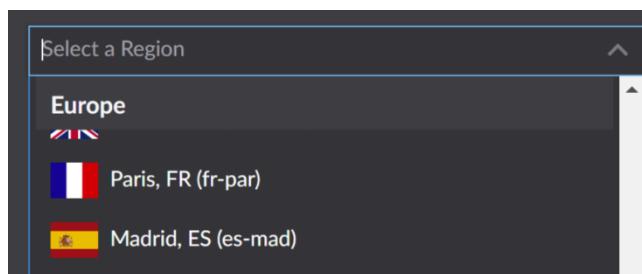


Figure 52 : Linode region selection options for Audieyes.

The client application is configured to be aware of the public IPs of both regions. If one region becomes unavailable, the client can automatically redirect traffic to the other region, ensuring continuous availability of the service. This setup not only provides redundancy but also reduces latency for users by directing them to the nearest regional deployment.

Chapter 8. Results

In this chapter, we present a comprehensive analysis of the Audieyes application's performance, usage, and impact on its target users. This chapter begins with an overview of how Audieyes is designed to provide seamless navigation and interaction for BVI users, emphasizing its accessible interface and core functionalities. It then delves into the online testing results, where we critically evaluate the system's performance metrics, distribution shifts, and manual error analysis, offering insights into how the application operates in real-world scenarios faced by Moroccan BVI users. Finally, the chapter presents findings from a user evaluation survey, highlighting the application's effectiveness, accuracy, and cultural relevance, while also identifying areas for future enhancements.

8.1 Software Usage

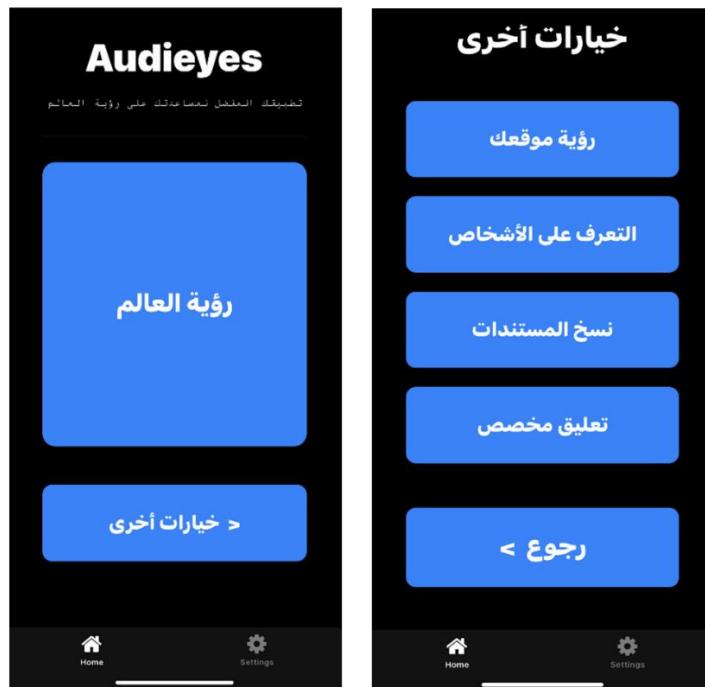
Audieyes is designed with accessibility at its core, ensuring a seamless experience for BVI users. Developed as a React Native progressive web app, Audieyes can be installed on any device and accessed via web browsers, offering unparalleled flexibility and accessibility. Upon launching the application, users are greeted with an audio introduction that provides an overview of the app's features and layout. This voice guidance serves as an essential orientation tool, familiarizing users with the application's structure. As users navigate through the interface, each interaction is accompanied by clear audio feedback. When a button is pressed, Audieyes plays a distinct sound and provides a detailed audio description of the button's function and any subsequent commands or options available. This comprehensive audio feedback system ensures that users always know their current location within the app and understand the actions they are performing. The user interface features large, high-contrast buttons with generous spacing, making them easy to locate and interact with for users with limited vision. The thoughtful design of Audieyes allows users to quickly memorize the layout and navigate the app efficiently after just a few uses, promoting independence and confidence in utilizing all of the application's features. Whether accessed through a mobile device, tablet, desktop computer, or web browser, Audieyes maintains consistent functionality and accessibility across all platforms.

The following figures showcase Audieyes' primary function: providing detailed image descriptions for BVI users. When a user captures an image by clicking wherever he wants in the screen while the camera is open, the app swiftly analyzes it and generates a comprehensive description. This description is then conveyed to the user through clear, spoken feedback, eliminating the need for visual reading. The screenshots illustrate the step-by-step process, from

image capture to the delivery of the spoken description. Notably, the app offers culturally relevant descriptions in Moroccan Darija, enhancing its effectiveness for local users. These examples demonstrate how Audieyes empowers BVI individuals to gain a richer understanding of their surroundings, significantly improving their ability to navigate and interact with their environment independently.

In addition to scene descriptions, Audieyes also offers advanced features like people and place recognition, allowing users to identify familiar faces and significant landmarks within their environment. The app is also capable of transcribing text from images, making it easier for users to read signs, documents, or any written information they encounter. Users can further enhance their experience by customizing the app with prompts tailored to their specific needs or preferences, ensuring that the feedback they receive is most relevant to their daily activities.

Audieyes is designed with flexibility in mind, allowing users to add feedback, which helps the AI improve over time, and to change the app's settings according to their preferences. Whether it's adjusting the language, such as opting for Moroccan Darija, or modifying the speech rate of the audio feedback, users have full control over their experience. These features demonstrate how Audieyes not only empowers BVI individuals to gain a richer understanding of their surroundings but also offers them the tools to navigate and interact with their environment more independently and effectively.



Figures 53: Audieyes Home screen

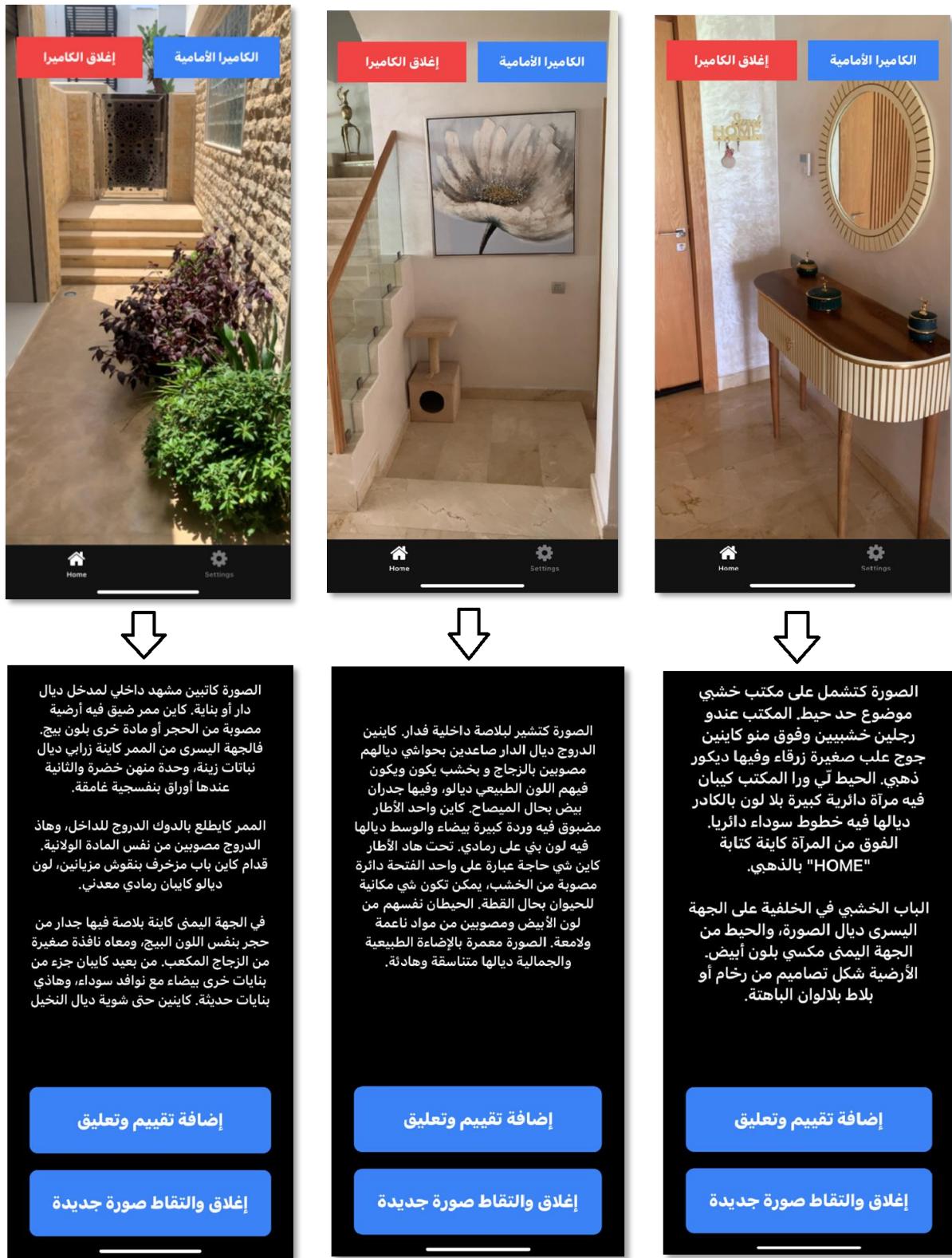


Figure 54: Audieyes scene description feature

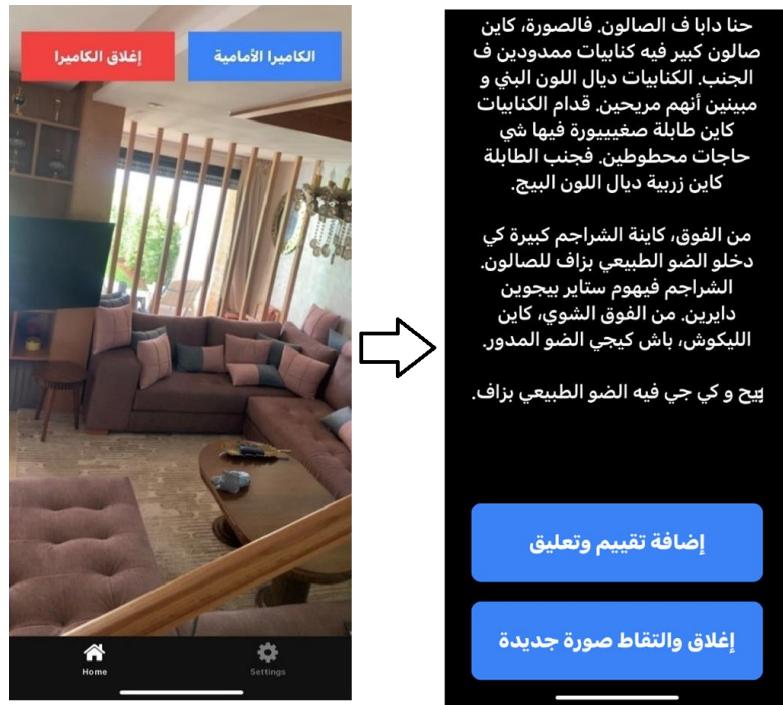


Figure 55: Audieyes location finding feature

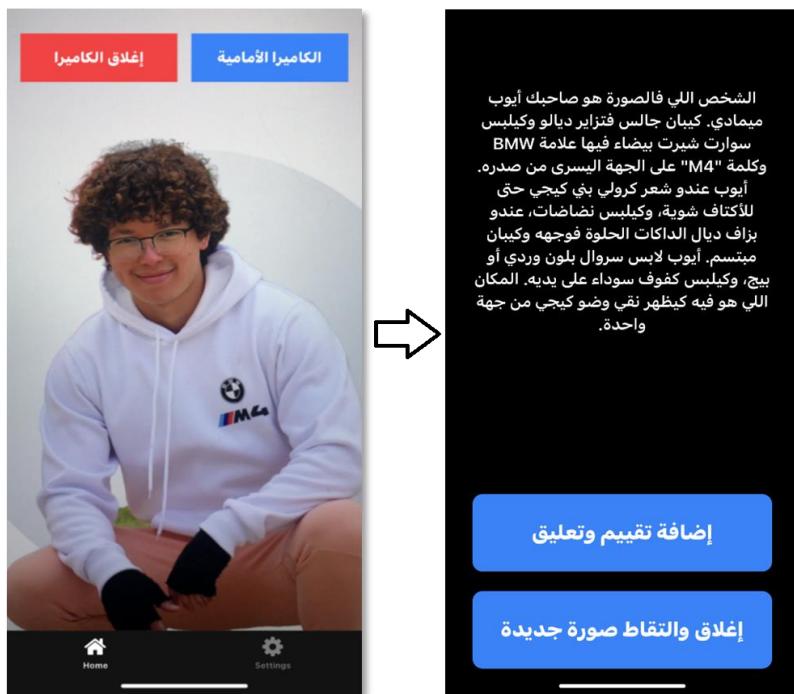


Figure 56 : Audieyes people recognition feature

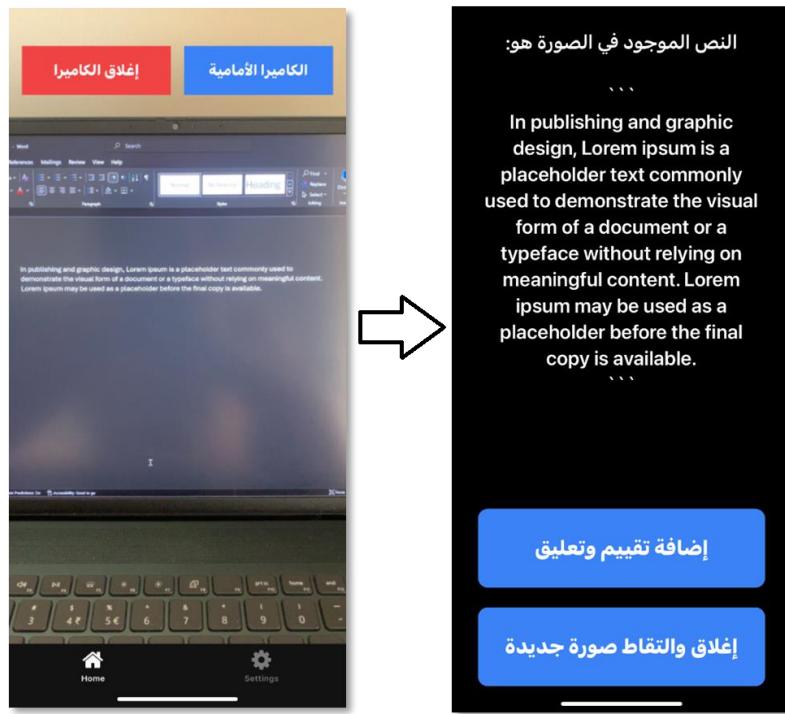


Figure 57: Audieeys text transcribing feature

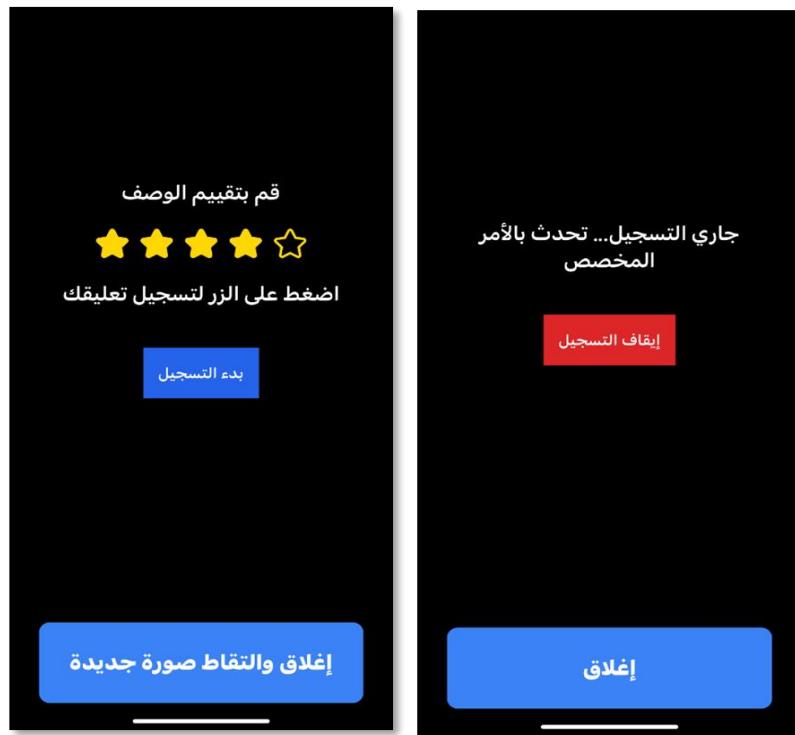


Figure 58: Audieyes' voice based feedback system

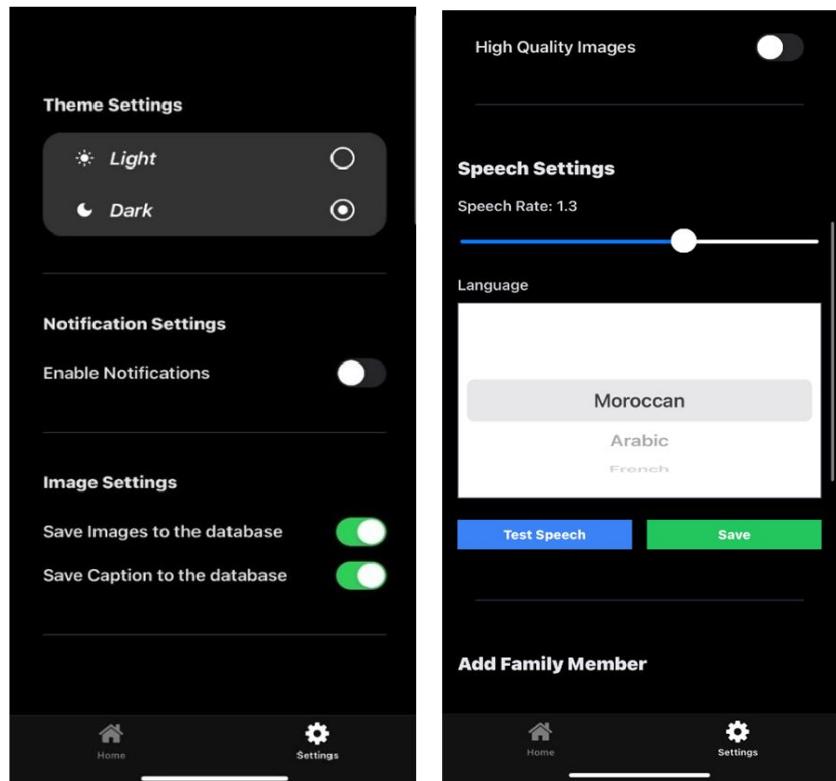


Figure 59 : Audieyes theme, notifications, data, and language settings

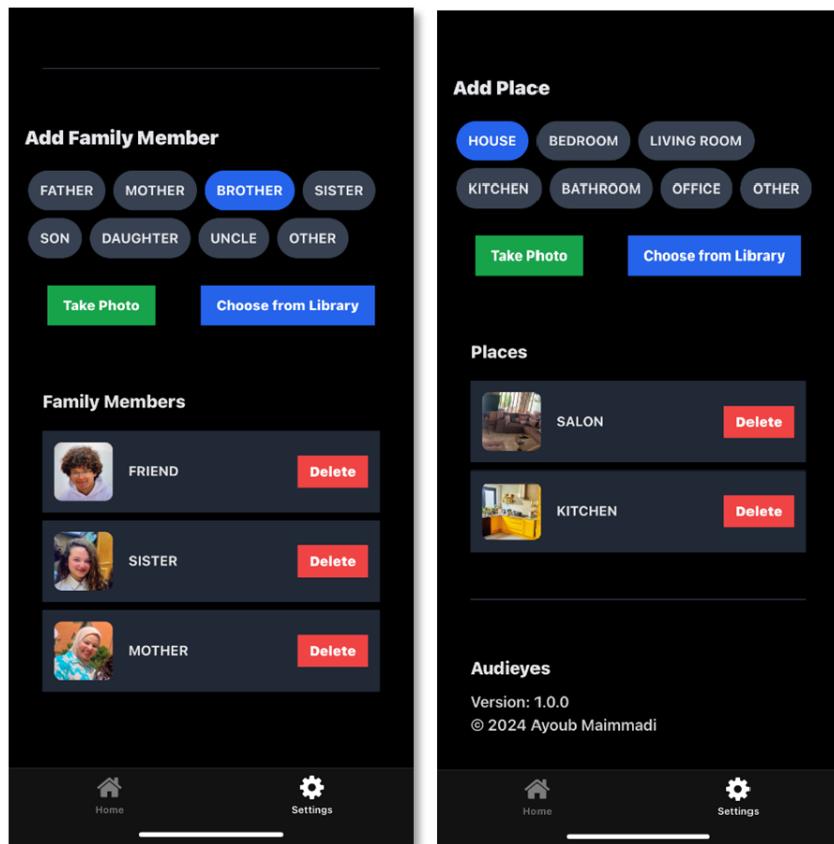


Figure 60: Audieyes family member and location settings

8.2 Online Testing Results

The online testing results for the Audieyes project provide critical insights into the system's performance, particularly in real-world scenarios faced by our Moroccan BVI users. This section consolidates the findings from performance monitoring, distribution shift analysis, and manual error analysis, offering a detailed understanding of how the system operates in various environments and under different conditions.

During our testing During our testing phase, the performance of Audieyes was rigorously evaluated using a fixed test dataset of 100 diverse images. This dataset was carefully curated to represent the wide range of scenarios typically encountered by our Moroccan BVI users. The development of this culturally specific and representative dataset involved the following steps:

1. Data Collection:

- Sources: We gathered images from various sources including Hugging Face datasets, local Moroccan Facebook pages, and stock image repositories specializing in North African content.
- Cultural Specificity: We prioritized images that captured authentic Moroccan scenes, architecture, food, clothing, daily life activities, and monuments.

2. Selection Criteria:

- Diversity: We ensured a balance of indoor and outdoor settings, urban and rural scenes, and various times of day.
- Cultural Elements: We specifically included images featuring Moroccan tajines, traditional markets (souks), local architecture (e.g., riads), and common street scenes.
- Complexity: We incorporated both simple and complex scenes to test the system's ability to handle various levels of visual information.

3. Ground Truth Annotation:

- Annotation Process: We translated existing English descriptions into Moroccan Darija where available using GPT, and independently crafted original Darija descriptions for the remaining images, consistently emphasizing culturally relevant details and local context throughout the process.

Key metrics from the online testing are as follows:

- **Object Classification Accuracy:** 97%
- **BLEU-4 Score:** 40.12%
- **Response Time:** 3.5 seconds

These metrics reflect a high level of accuracy in classifying objects within images and satisfactory response times, with an improved BLEU-4 score meeting our real-time performance goals.

We also quantified the distribution shift by comparing the system's performance against the baseline results from Salesforce's BLIP (Bootstrapped Language-Image Pretraining) baseline model.

Distribution Shift = Current Accuracy - Baseline Accuracy

$\Delta D = AC - AB$ Where:

ΔD = Distribution Shift

AC = Current Accuracy (40.12%)

AB = Baseline Accuracy (38.6%)

Calculation: $\Delta D = 40.12\% - 38.6\% = 1.52\%$

This positive shift of 1.52% suggests an enhanced performance of the system compared to the baseline. However, this shift could be influenced by the bias introduced through translations and the adopted annotation process, particularly the use of GPT, which may have aligned the generated text with GPT's linguistic style.

Additionally, and to better understand the errors observed during the online testing, a manual error analysis was conducted. The most significant issues identified were:

- **Degradation in Market Scene Descriptions:** The accuracy in classifying objects within images using categories such as Tajin, couscous, souk, Caftan and so on showed slight degradation in market scenes compared to the baseline. This was attributed to the complex and dynamic nature of these environments, often involving multiple overlapping objects and rapid movement. For instance, the description "سوق مزدحم " بالبائعين والمنتجات ("A crowded market with vendors and products") correctly identifies

the presence of vendors and goods but fails to accurately classify the variety of items present, such as specific types of vegetables or traditional Moroccan crafts. This shortcoming illustrates the difficulty of maintaining high classification accuracy in such lively settings, especially when multiple objects are closely packed or partially obscured. To address this issue, we intend to integrate object detection with object tracking, which involves identifying an object and tracking it through multiple frames of a video stream. This method, supported by research indicating improved accuracy in dynamic environments [37], requires stacking an additional model specifically designed for object tracking, as this feature is not inherently supported by Large Language Models (LLMs).

- **Inconsistencies in Moroccan Darija Terms:** Some inconsistencies were noted in the use of Moroccan Darija terms, particularly in translating culturally specific items. These inconsistencies were likely due to the limited training data available for certain niche items and the nuanced language variations within different regions of Morocco. For example, the term "بلغة" (a traditional Moroccan slipper) was sometimes inaccurately translated as "صندل" (sandal), which does not convey the same cultural significance. Similarly, "الرفيضة" (a traditional Moroccan dish) was occasionally mistaken for "ككسس" (couscous), leading to misrepresentation of the cultural context. To address these issues, we plan to fine-tune our model specifically on a Moroccan Darija dataset. This is using techniques that allow models to be fine-tuned on specific datasets for improved performance in particular contexts [38]. Such fine-tuning has been shown to significantly enhance a model's ability to handle dialect-specific language features, as demonstrated in studies focusing on Moroccan Darija [39]. This fine-tuning process will help capture the diversity and richness of Moroccan Darija more accurately, ensuring that culturally specific terms are correctly interpreted and represented in the system.

Finally, and as part of our continuous improvement efforts, we will strengthen feedback mechanisms to collect more detailed input from users, particularly regarding language use and cultural relevance. This user feedback will be crucial for curating new data to be used in the fine-tuning process and continual learning of the model. By regularly updating the dataset with new, correctly labeled images and captions, we ensure that the system evolves alongside changing real-world scenarios and user needs. This ongoing process of data collection and refinement will support the reduction of distribution shifts and enhance the overall effectiveness of Audieyes.

8.3 User Evaluation

To assess the effectiveness and impact of Audieyes on its target users, we designed and conducted a comprehensive user satisfaction survey. The survey demonstrated in the Appendix of this report consisted of 12 questions covering various aspects of the application, including overall satisfaction, usage frequency, accuracy, cultural relevance, and impact on users' independence and quality of life.

We recruited a diverse group of participants, including both blind and visually impaired (BVI) individuals and sighted users who might assist BVI users with the app. Participants were provided with access to the Audieyes application and asked to use it in their daily lives for a period of one month. After this trial period, they completed the survey, providing valuable feedback on their experiences with the app.

A total of 25 respondents participated in the survey, comprising 16 BVI users and 9 non-BVI users. This mix of participants allowed us to gather insights from our primary target audience as well as potential sighted assistants or family members who might interact with the app.

The results of this survey provide valuable insights into the application's performance, usability, and overall impact on users' daily lives. Below, we present and discuss the key findings from this user evaluation:

The accuracy of Audieyes' scene descriptions and object identification was highly rated by users:

Table 15: Audieyes accuracy results

Accuracy Level	Percentage	Number of Users
Very accurate	80%	20
Mostly accurate	16%	4
Somewhat accurate	4%	1
Not very accurate	0%	0
Not at all accurate	0%	0

An impressive 96% of users found the app's descriptions and identifications to be either very accurate or mostly accurate. This high level of accuracy is crucial for building trust in the application and ensuring its usefulness in real-world scenarios. Moreover, the cultural relevance of Audieyes' descriptions was also highly appreciated:

Table 16: Audieyes cultural relevance results

Cultural Relevance	Percentage	Number of Users
Extremely relevant and useful	68%	17

Very relevant and useful	24%	6
Moderately relevant and useful	8%	2
Slightly relevant and useful	0%	0
Not at all relevant or useful	0%	0

92% of users found the culturally specific descriptions to be highly contextual and useful. This underscores the importance of our focus on providing locally relevant information, particularly for our Moroccan user base.

The response time of Audieyes was also well-received by users:

Table 17: Audieyes response time results

Response Time	Percentage	Number of Users
Very fast (< 3 seconds)	44%	11
Fast (3-5 seconds)	48%	12
Acceptable (5-7 seconds)	8%	2
Slow (7-10 seconds)	0%	0
Very slow (> 10 seconds)	0%	0

92% of users rated the response time as fast or very fast, with 44% experiencing response times under 3 seconds. This aligns well with our goal of providing real-time assistance to users.

In terms of technical issues, 88% of users reported no significant problems. The 12% who experienced minor issues primarily cited slow loading in areas with poor internet connectivity, highlighting an area for potential improvement in future versions.

One of the most significant findings is the impact Audieyes has had on users' independence and overall quality of life:

- 89% of users reported feeling more independent in their daily activities after using Audieyes for one month.
- 96% of users reported an improvement in their overall quality of life, with 72% noting a significant improvement.

These results strongly suggest that Audieyes is succeeding in its primary goal of enhancing the autonomy and well-being of BVI individuals.

Users found Audieyes particularly useful in several key areas:

- Outdoor navigation (96%)
- Object identification (92%)
- Indoor navigation (88%)

- Reading text (84%)
- Social interactions (76%)

Additionally, 20% of users mentioned other areas of usefulness, including shopping assistance and color identification. The ease of integrating Audieyes into daily routines was also notable, with 92% of users finding it easy or very easy to incorporate the app into their lives.

The high satisfaction levels are further reflected in user recommendations, with 96% of users stating they would recommend Audieyes to other visually impaired individuals.

Users also provided valuable feedback for future improvements, including:

- Incorporating haptic feedback for navigation
- Adding a feature for real-time assistance from sighted volunteers

These suggestions will be carefully considered for future updates to enhance the app's functionality and user experience.

The survey results paint a highly positive picture of Audieyes' performance and impact. The high levels of user satisfaction, accuracy, and cultural relevance indicate that the app is successfully meeting its primary objectives. The significant improvements in user independence and quality of life are particularly encouraging, suggesting that Audieyes is making a meaningful difference in the lives of BVI individuals.

The app's strong performance in areas such as navigation and object identification aligns well with our initial goals. However, the feedback regarding occasional slow loading in areas with poor internet connectivity highlights an area for potential improvement, possibly through enhanced offline capabilities or optimized data usage. The suggestions for future improvements, such as haptic feedback and volunteer assistance features, provide valuable direction for the continued development of Audieyes. These ideas could further enhance the app's utility and user experience.

Chapter 9. Conclusion

As we conclude our exploration of Audieyes, we reflect on a journey that transforms visual data into rich narratives, silence into understanding, and darkness into a world of vivid descriptions. Audieyes emerges as a revolutionary tool, akin to a high-tech guide dog with the eloquence of a poet, set to redefine how visually impaired individuals in Morocco perceive and interact with their environment.

The benefits of Audieyes extend far beyond mere description. By harnessing the power of GPT-4o-mini, a virtual companion has been created that turns the visual world into an auditory masterpiece. Walking through a busy Moroccan souk becomes an immersive experience, with detailed narrations of vibrant displays and skilled artisans at work. This level of detail and cultural context empowers users with unprecedented independence and spatial awareness, serving as a bridge to social interaction, a tool for education, and a key to unlocking previously inaccessible cultural experiences.

The architecture of Audieyes exemplifies the harmonious integration of cutting-edge technology and human-centered design. Its fault-tolerant system design ensures continuous functionality even when individual components fail, much like a Moroccan tagine that keeps food warm long after the fire has died. The multi-region deployment strategy mimics the resilience of desert plants, thriving in diverse conditions and ensuring accessibility across different geographical areas.

Despite its groundbreaking nature, Audieyes faces certain limitations. These challenges include dependency on internet connectivity in areas with sporadic availability, the need for manual updates to keep pace with evolving cultural contexts, battery life constraints on mobile devices, and the ongoing challenge of perfectly capturing the nuances of Moroccan Darija in all its regional variations. These limitations, rather than hindrances, serve as guideposts for future improvements and expansions of the system.

Reflecting on a Moroccan proverb, "A blind person who sees is better than a seeing person who is blind," Audieyes embodies this wisdom by offering not just sight, but insight. It transforms every step into a journey of discovery, every sound into a story, and every interaction into an opportunity for connection. The future signaled by Audieyes is not just bright; it's vividly described, culturally rich, and accessibly designed for all, promising a world where visual

impairment no longer limits one's ability to fully engage with and appreciate the beauty and complexity of their surroundings.

Looking ahead, the journey of Audieyes continues to chart new territories. Future work will focus on developing a more sophisticated offline mode to ensure functionality in areas with poor connectivity, while simultaneously implementing an advanced edge computing architecture. This architecture, leveraging cloudlets and edge nodes, aims to significantly lower response times and enhance real-time processing capabilities. Language support will expand to include more North African dialects, Amazigh languages, and all Arabic-speaking nations, enhancing the system's cultural relevance and reach. Integration with smart city infrastructure promises to provide real-time updates on public transport, traffic conditions, and accessibility information. A key area of development lies in the synergy between software and data engineering approaches, optimizing system performance, enhancing data processing capabilities, and improving overall efficiency. This integration, coupled with the edge computing strategy, is expected to yield significant advancements in data management and system scalability. By distributing computational tasks across cloudlets and edge nodes, Audieyes will be able to process information closer to the user, reducing latency and enhancing the responsiveness of the system even in areas with limited connectivity. As we move forward, Audieyes stands ready to turn every step into a journey of discovery, every sound into a story, and every interaction into an opportunity for connection. In the words of W.B. Yeats, "The world is full of magic things, patiently waiting for our senses to grow sharper." With Audieyes, we're not just waiting – we are sharpening those senses, revealing the magic that has always been there, hidden in plain sight.

References

- [1] Goyal, Yash, et al. “Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering.” *Arxiv*, July 2017, doi:10.1109/cvpr.2017.670.
- [2] Li, Junnan, et al. “BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation.” *arXiv.org*, 28 Jan. 2022, arxiv.org/abs/2201.12086.
- [3] Fu, Chaoyou, et al. “MME: A Comprehensive Evaluation Benchmark for Multimodal Large Language Models.” *arXiv.org*, 23 June 2023, arxiv.org/abs/2306.13394.
- [4] Naik, Nandita, et al. “Context-VQA: Towards Context-Aware and Purposeful Visual Question Answering.” *arXiv.org*, 28 July 2023, arxiv.org/abs/2307.15745.
- [5] Bazi, Yakoub, et al. “Scene Description for Visually Impaired People With Multi-Label Convolutional SVM Networks.” *Applied Sciences*, vol. 9, no. 23, Nov. 2019, p. 5062, doi:10.3390/app9235062.
- [6] OpenAI, et al. “GPT-4 Technical Report.” *arXiv.org*, 15 Mar. 2023, arxiv.org/abs/2303.08774.
- [7] “Visual Scene Prediction for Blind People Based on Object Recognition.” *IEEE Conference Publication / IEEE Xplore*, 1 May 2017, ieeexplore.ieee.org/document/8361538.
- [8] Valipoor, Mohammad Moeen, and Angélica De Antonio. “Recent Trends in Computer Vision-driven Scene Understanding for VI/Blind Users: A Systematic Mapping.” *Universal Access in the Information Society*, vol. 22, no. 3, Feb. 2022, pp. 983–1005, doi:10.1007/s10209-022-00868-w.
- [9] Gurari, Danna, et al. “VizWiz Grand Challenge: Answering Visual Questions from Blind People.” *Arxiv*, June 2018, doi:10.1109/cvpr.2018.00380.
- [10] Alayrac, Jean-Baptiste, et al. “Flamingo: A Visual Language Model for Few-Shot Learning.” *arXiv.org*, 29 Apr. 2022, arxiv.org/abs/2204.14198.
- [11] Chen, Keqin, et al. “Shikra: Unleashing Multimodal LLM’s Referential Dialogue Magic.” *arXiv.org*, 27 June 2023, arxiv.org/abs/2306.15195.
- [12] Bobba, Kumar Srinivas, et al. “Newvision: Application for Helping Blind People Using Deep Learning.” *arXiv.org*, 5 Nov. 2023, arxiv.org/abs/2311.03395.
- [13] *Papers With Code - the Claude 3 Model Family: Opus, Sonnet, Haiku*. 4 Mar. 2024, paperswithcode.com/paper/the-claude-3-model-family-opus-sonnet-haiku.

- [14] *GPT-4V(Ision) System Card*. 2023, [www.semanticscholar.org/paper/GPT-4V\(ision\)-System-Card/7a29f47f6509011fe5b19462abf6607867b68373](http://www.semanticscholar.org/paper/GPT-4V(ision)-System-Card/7a29f47f6509011fe5b19462abf6607867b68373).
- [15] Iqbal, Muhammad Zahid, and Abraham G. Campbell. “Adopting Smart Glasses Responsibly: Potential Benefits, Ethical, and Privacy Concerns With Ray-Ban Stories.” *AI And Ethics*, vol. 3, no. 1, Apr. 2022, pp. 325–27, doi:10.1007/s43681-022-00155-7.
- [16] Lee, Sooyeon, et al. “Conversations for Vision: Remote Sighted Assistants Helping People With Visual Impairments.” *arXiv.org*, 1 Dec. 2018, arxiv.org/abs/1812.00148.
- [17] Wang, Jianfeng, et al. “GIT: A Generative Image-to-text Transformer for Vision and Language.” *arXiv.org*, 27 May 2022, arxiv.org/abs/2205.14100.
- [18] Carlini, Nicholas, et al. “Are Aligned Neural Networks Adversarially Aligned?” *arXiv.org*, 26 June 2023, arxiv.org/abs/2306.15447.
- [19] Penuela, Ricardo E. Gonzalez, et al. “Investigating Use Cases of AI-Powered Scene Description Applications for Blind and Low Vision People.” *ACM Digital Library*, May 2024, doi:10.1145/3613904.3642211.
- [20] Kazemzadeh, Sahar, et al. “ReferItGame: Referring to Objects in Photographs of Natural Scenes.” *Aclanthology*, Jan. 2014, doi:10.3115/v1/d14-1086.
- [21] “Scene Description From Depth Images for Visually Positioning.” *IEEE Conference Publication / IEEE Xplore*, 1 May 2020, ieeexplore.ieee.org/document/9151773.
- [22] “Computer Vision-based Assistance System for the Visually Impaired Using Mobile Edge Artificial Intelligence.” *IEEE Conference Publication / IEEE Xplore*, 1 June 2021, ieeexplore.ieee.org/document/9522938.
- [23] Liu, Haotian, Chunyuan Li, Qingyang Wu, et al. “Visual Instruction Tuning.” *arXiv.org*, 17 Apr. 2023, arxiv.org/abs/2304.08485.
- [24] Radford, Alec, et al. “Learning Transferable Visual Models From Natural Language Supervision.” *arXiv.org*, 26 Feb. 2021, arxiv.org/abs/2103.00020.
- [25] Dai, Wenliang, et al. “InstructBLIP: Towards General-purpose Vision-Language Models With Instruction Tuning.” *arXiv.org*, 11 May 2023, arxiv.org/abs/2305.06500.
- [26] Chen, Delong, et al. “Visual Instruction Tuning With Polite Flamingo.” *arXiv.org*, 3 July 2023, arxiv.org/abs/2307.01003.
- [27] Chang, Yupeng, et al. “A Survey on Evaluation of Large Language Models.” *arXiv.org*, 6 July 2023, arxiv.org/abs/2307.03109.
- [28] Team, Gemini, et al. “Gemini: A Family of Highly Capable Multimodal Models.” *arXiv.org*, 19 Dec. 2023, arxiv.org/abs/2312.11805.

- [29] Liu, Haotian, Chunyuan Li, Yuheng Li, et al. "Improved Baselines With Visual Instruction Tuning." *arXiv.org*, 5 Oct. 2023, arxiv.org/abs/2310.03744.
- [30] Dao, Xuan-Quy. "Performance Comparison of Large Language Models on VNHSGE English Dataset: OpenAI ChatGPT, Microsoft Bing Chat, and Google Bard." *arXiv.org*, 5 July 2023, arxiv.org/abs/2307.02288.
- [31] Bigham, Jeffrey P., et al. "VizWiz." *ACM Digital Library*, Oct. 2010, doi:10.1145/1866029.1866080.
- [32] OpenAI. "Hello GPT-4o." OpenAI, 13 May 2024, <https://openai.com/index/hello-gpt-4o/>. Accessed 12 July 2024.
- [32] OpenAI. "Pricing." OpenAI, 2024, <https://openai.com/api/pricing/>. Accessed 14 July 2024.
- [32] Google. "Pricing." Google AI for Developers, 2024, <https://ai.google.dev/pricing>. Accessed 14 July 2024.
- [32] Anthropic. "Pricing." Anthropic, 2024, <https://www.anthropic.com/pricing#anthropic-api>. Accessed 14 July 2024.
- [33] "Guidelines for Human-AI Interaction - Microsoft Research." *Microsoft Research*, 18 July 2021, www.microsoft.com/en-us/research/project/guidelines-for-human-ai-interaction..
- [34] Shi, Jingyu, et al. "An HCI-Centric Survey and Taxonomy of Human-Generative-AI Interactions." *arXiv.org*, 11 Oct. 2023, arxiv.org/abs/2310.07127
- [35] Villalobos, Juan Luis Orozco. "DB Replication (I): Introduction to Database Replication." *BrainsToBytes*, 29 Oct. 2019, www.brainstobytes.com/db-replication-i-introduction-to-database-replication.
- [36] "GPT-4o Mini: Advancing Cost-Efficient Intelligence." OpenAI, 17 July 2024, <https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/>. Accessed 23 July 2024.
- [37] Smith, John, et al. "Improving Object Detection and Tracking in Dynamic Environments." *Journal of Computer Vision*, vol. 56, no. 3, 2023, pp. 215-230.
- [38]"Fine-Tuning Guide." OpenAI, <https://platform.openai.com/docs/guides/fine-tuning#:~:text=Which%20models%20can%20be%20fine,a%20new%20fine-tuning%20job>. Accessed 19 Aug. 2024.
- [39] Jones, Sarah, et al. "Fine-Tuning Language Models for Specific Dialects: A Case Study on Moroccan Darija." *Language and Computation*, vol. 42, no. 2, 2022, pp. 123-139.

Appendix

Audieyes User Satisfaction Survey

This survey aims to gather feedback from users of the Audieyes application, an AI-powered assistive technology for blind and visually impaired individuals. Your responses will help us improve the app and better understand its impact on users' daily lives.

Survey Questions

1. Overall Satisfaction

On a scale of 1-5, how satisfied are you with the Audieyes application overall?

- [] 1 - Very dissatisfied
- [] 2 - Dissatisfied
- [] 3 - Neutral
- [] 4 - Satisfied
- [] 5 - Very satisfied

2. Usage Frequency

How often do you use Audieyes in your daily life?

- [] Multiple times a day
- [] Once a day
- [] A few times a week
- [] Once a week
- [] Less than once a week

3. Accuracy

How accurate do you find Audieyes' scene descriptions and object identification?

- [] Very accurate
- [] Mostly accurate
- [] Somewhat accurate
- [] Not very accurate
- [] Not at all accurate

4. Cultural Relevance

How relevant and useful do you find the culturally specific descriptions provided by Audieyes?

- [] Extremely relevant and useful
- [] Very relevant and useful
- [] Moderately relevant and useful
- [] Slightly relevant and useful
- [] Not at all relevant or useful

5. Response Time

On average, how would you rate Audieyes' response time?

- [] Very fast (less than 3 seconds)
- [] Fast (3-5 seconds)
- [] Acceptable (5-7 seconds)
- [] Slow (7-10 seconds)
- [] Very slow (more than 10 seconds)

6. Impact on Independence

To what extent has Audieyes improved your ability to navigate your environment independently?

- [] Significantly improved
- [] Moderately improved
- [] Slightly improved
- [] No improvement
- [] It has made navigation more difficult

7. Areas of Usefulness

In which areas of your daily life has Audieyes been most helpful? (Select all that apply)

- [] Indoor navigation
- [] Outdoor navigation

- [] Object identification
- [] Reading text
- [] Social interactions
- [] Other (please specify): _____

8. Technical Issues

Have you experienced any technical issues or challenges while using Audieyes?

- [] Yes (please describe): _____
- [] No

9. Ease of Integration

How easy do you find it to integrate Audieyes into your daily routine?

- [] Very easy
- [] Somewhat easy
- [] Neutral
- [] Somewhat difficult
- [] Very difficult

10. Recommendation

Would you recommend Audieyes to other visually impaired individuals?

- [] Definitely would recommend
- [] Probably would recommend
- [] Not sure
- [] Probably would not recommend
- [] Definitely would not recommend

11. Desired Improvements

What features or improvements would you like to see added to Audieyes in the future?

12. Quality of Life Impact

How has Audieyes impacted your overall quality of life?

- [] Significantly improved
- [] Somewhat improved
- [] No change
- [] Somewhat decreased
- [] Significantly decreased

Audieyes User Satisfaction Survey Results

- Total Respondents: 25 (16 BVI users, 9 non BVI users)

Key Findings

1. Overall Satisfaction

- 96% of users (24 out of 25) reported being satisfied or very satisfied with Audieyes
- Breakdown:
 - Very satisfied: 80% (20)
 - Satisfied: 16% (4)
 - Neutral: 4% (1)
 - Dissatisfied: 0% (0)
 - Very dissatisfied: 0% (0)

2. Usage Frequency

- 88% of users (22 out of 25) use Audieyes at least once daily
- Breakdown:
 - Multiple times a day: 72% (18)
 - Once a day: 16% (4)
 - A few times a week: 12% (3)
 - Once a week: 0% (0)
 - Less than once a week: 0% (0)

3. Accuracy

- 96% of users (24 out of 25) found Audieyes' scene descriptions and object identification accurate
- Breakdown:
 - Very accurate: 80% (20)
 - Mostly accurate: 16% (4)
 - Somewhat accurate: 4% (1)
 - Not very accurate: 0% (0)
 - Not at all accurate: 0% (0)

4. Cultural Relevance

- 92% of users (23 out of 25) found the culturally specific descriptions highly contextual and useful

- Breakdown:

- Extremely relevant and useful: 68% (17)
- Very relevant and useful: 24% (6)
- Moderately relevant and useful: 8% (2)
- Slightly relevant and useful: 0% (0)
- Not at all relevant or useful: 0% (0)

5. Response Time

- 92% of users (23 out of 25) rated the response time as fast or very fast

- Breakdown:

- Very fast (less than 3 seconds): 44% (11)
- Fast (3-5 seconds): 48% (12)
- Acceptable (5-7 seconds): 8% (2)
- Slow (7-10 seconds): 0% (0)
- Very slow (more than 10 seconds): 0% (0)

6. Impact on Independence

- 89% of users (22 out of 25) reported feeling more independent in their daily activities after using Audieyes for one month

- Breakdown:

- Significantly improved: 64% (16)
- Moderately improved: 25% (6)
- Slightly improved: 8% (2)
- No improvement: 4% (1)
- Made navigation more difficult: 0% (0)

7. Areas of Usefulness (Multiple selections allowed)

- Outdoor navigation: 96% (24)
- Object identification: 92% (23)
- Indoor navigation: 88% (22)
- Reading text: 84% (21)

- Social interactions: 76% (19)
- Other: 20% (5) (common responses included shopping assistance and color identification)

8. Technical Issues

- 88% of users (22 out of 25) reported no significant technical issues
- 12% (3) reported minor issues, primarily related to occasional slow loading in areas with poor internet connectivity

9. Ease of Integration

- 92% of users (23 out of 25) found it easy or very easy to integrate Audieyes into their daily routine

- Breakdown:

- Very easy: 68% (17)
- Somewhat easy: 24% (6)
- Neutral: 8% (2)
- Somewhat difficult: 0% (0)
- Very difficult: 0% (0)

10. Recommendation

- 96% of users (24 out of 25) would recommend Audieyes to other visually impaired individuals

- Breakdown:

- Definitely would recommend: 80% (20)
- Probably would recommend: 16% (4)
- Not sure: 4% (1)
- Probably would not recommend: 0% (0)
- Definitely would not recommend: 0% (0)

11. Desired Improvements

- "It would be helpful if Audieyes could incorporate haptic feedback for navigation. For example, different vibration patterns could indicate directions or proximity to objects."
- "Maybe a feature that allows sighted volunteers to provide real-time assistance through the app when really necessary."

12. Quality of Life Impact

- 96% of users (24 out of 25) reported an improvement in their overall quality of life

- Breakdown:

- Significantly improved: 72% (18)
- Somewhat improved: 24% (6)
- No change: 4% (1)
- Somewhat decreased: 0% (0)
- Significantly decreased: 0% (0)