

Project: ArtAI*
Group: AlBeSa
Date: 9th of October
Version: 1

Software Requirements Specification

1. Introduction

This Software Requirements Specification describes the design principles, scope, and intended use of ArtAI - our intelligent art-recognition and explanation system. This document provides a thorough guide of the system functions, operation, target audience, and needs in relation to the projects primary goals. It gives a summary of the functional and non-functional specifications, AI-specific limitations, and user & system requirements. By outlining these elements precisely, the document guarantees that all parties involved have a common understanding of the problem statement and facilitates the software's constant growth and assessment.

1.1. Purpose

ArtAI aims to provide users with the ability to instantaneously recognize artwork and obtain insightful visual descriptions of it. The system automatically identifies an artwork from an image using image recognition and large language models. Then, it generates informative textual and spoken descriptions in a variety of languages. In addition to supporting museum curators to maintain and update digital art collections, this project seeks to improve accessibility and engagement with art for both online and onsite tourists. The SRS defines the scope of the system - from image input to information retrieval and multilingual explanation production.

1.2. Intended audience

This model is intended for software developers and AI engineers who will implement and maintain the system; project managers who will monitor progress and compliance; museum curators and content administrators who will manage the artwork database; and testers who will ensure that all functional and non-functional requirements are met. External stakeholders, such as academic supervisors or clients, may also seek information on the systems architecture and purpose.

1.3. Additional information

Version 1 of the ArtAI specification is represented by this SRS. As the project evolve through the design and testing stages described in the course, it will change. In order to provide transparency, maintainability, and traceability throughout the development lifecycle, the document follows the standard structure covered in the class and links the systems goals to measurable software and AI-specific criteria.

2. User requirements

ID	User Requirement (in natural language)
UR1	The system shall allow users to take or upload a photo of an artwork using their device camera or gallery.
UR2	The system shall automatically recognize the artwork from the provided image without requiring manual input.
UR3	The system shall provide an informative and concise explanation of the recognized artwork, including the title, artist, art style, and creation period.
UR4	The system shall present the explanation both as on-screen text and as spoken audio narration.
UR5	The user should be able to select the preferred method for obtaining the information: text, audio, or both.
UR6	The system should deliver the result within 15 seconds.
UR7	The system should keep all the artwork information accessible and allow for the text and/ or audio to be replayed.
UR8	The system shall be compatible with mobile and web platforms.
UR9	The system shall allow museum curators or administrators to update and manage the collection of artworks displayed within the system.
UR10	The system shall offer an option to share artwork information or audio explanations through social media or messaging platforms.

The user requirements outlined above collectively support the main objectives of the Art Guide system (present in task 1 as sub-goals). Requirements UR1, UR2, and UR6 directly contribute to the objective of providing fast and accurate artwork recognition by allowing users to capture or upload an image and have the system automatically identify quickly the artwork without manual input. UR3 aligns with the goal of delivering concise and contextual explanations beyond basic labels, ensuring that users receive meaningful information about the artwork's title, artist, and style. UR4 and UR5 address the objective of ensuring inclusivity by providing both text and audio explanations and enabling users to choose their preferred mode of interaction. UR7 reinforces the goal of creating an intuitive and user-friendly interface by allowing users to replay or re-read explanations with minimal effort. UR8 and UR10 align with the objective of encouraging adoption among tourists and casual art enthusiasts, ensuring the system is available on both mobile and web platforms. Finally, UR9 supports the objective of scalability and adaptability by allowing curators and administrators to update and manage the underlying artwork database.

3. System requirements

3.1. Functional requirements

Feature/function	Description	Priority	Action/result
Image capturing and upload	User can take a photo of an art piece and send it to the system (app).	High	The system (app) receives the image and sends it to the recognition module (Embedder, Retriever) for processing.
Art piece recognition	System identifies the artwork by comparing the input image's embeddings to embeddings in the vector database	High	Retriever outputs the most similar artwork.
Information retrieval	System retrieves metadata such as artist, year, art period, and style from the database, related to the identified art piece.	High	Retriever outputs the information about the most similar artwork.
Text explanation generation	System generates a cohesive, tour-guide-style explanation of the identified artwork using an LLM.	High	The system outputs a written (correct) explanation about the recognized artwork.
Audio explanation generation	Convert text explanations into audio.	Medium	The system outputs an audio (correct and natural-sounding) explanation about the recognized artwork.
Multilingual output	System generates both text and audio explanations in ≥ 4 languages.	Medium	User can choose preferred language and receive explanations (text and audio) in the preferred language.
Data management	System maintains desired set of art pieces in a vector database	High	Developers can update, add, or remove artworks as exhibitions change.

3.2. Non-functional requirements

Category	Description	Link to Functional Requirements
Product	The system must process recognition and generate explanations within 10 seconds on average.	Art piece recognition +Information retrieval +Text/audio explanation generation
Product	Architecture must support adding new exhibitions or galleries without redesign.	Art piece recognition +Information retrieval +Data management
Product	Uploaded images must be processed securely and deleted after inference. No personal data stored.	Data management
Organizational	Implemented in Python using LangChain, Weaviate, and Gemini API.	All
External	AI explanations must remain factual, unbiased, and culturally appropriate.	Text/Audio explanation generation
External	Respect museum curators' content guidelines and avoid misrepresentation.	Data management

4. AI specific requirements

Reliability: The model shall respond with the appropriate information within a maximum of 12-15 seconds depending on Gemini API traffic (even if the model responds with an unavailability message). On an 100 art piece validation set the model shall provide appropriate information on >80% of the cases.

Maintainability: Changing the embedder should automatically trigger the reindexing of the whole vector database. All combinations of LLM+Embedder+System Prompt shall be stored in versions so a rollback to them can be instantly done.

Adaptability: Changing to a new exhibition (changing the art data in the database) should be a no-code action which only requires the upload of a csv/xls file. Different LLMs and system prompts should be easily swappable and only changed after quality testing.

Scalability: Vector database will maintain only the art pieces in the exhibition. The assumption is that the average amount of art pieces in a given exhibition will not grow exponentially (current range is 50 - 500 art pieces), therefore plans for scaling the vector database are not needed. Since we are working with the APIs of foundational models (like Gemini 2.5 Flash) they are designed to scale well and especially faster models are optimized for low latency.

Sustainability: The LLM shall have a limit on the tokens generated by the “thinking” of the model and also the tokens generated in the response. This shall be adjusted after experimenting with the pipeline. Values should be adjustable according to the budget set by clients.