BECOMING

Barberotheca

Software Requirement Specification

Project Management Plan and Software Engineering Specifications for 'Barberotheca'. Curated for the “Project Management and Software Engineering for Cultural Heritage” course.

a.y. 2025/2026

Tommaso Barbato
tommaso.barbato@studio.unibo.it

Martina Uccheddu
martina.uccheddu@studio.unibo.it

Table of contents

About Software Requirement Specification (SRS)

Introduction

Document scope

The scope of this document is to define the functional and non-functional requirements for the Barberotheca project. This includes specification for both the semantic data infrastructure and the dissemination platform. This document outlines the system's scope and features, provides the necessary domain description and architectural diagrams, and details a requirements list. Furthermore, it defines specific sample use cases, user stories and test cases for key data-oriented and process-oriented functionalities. Finally it establishes the verification and validation methodology, detailing the testing strategies.

General Description

Product scope

The Barberotheca system consists of a full-stack software solution designed to manage the complete lifecycle of all *lectio magistralis,* from raw ingestion to public dissemination. The software system is made up of two primary subsystems:

- The semantic data pipeline: a server-side infrastructure responsible for the automated extraction of audio from video sources, normalization of media files, speech to text transcription and semantic enrichment;
- The dissemination platform: a web-based application providing an interface for the "Festival della mente" corpus. It retrieves the data and metadata and offers a visualization and browsing tool.

Features

Semantic data infrastructure

- Data ingestion and management:
    - Audio ingestion pipeline: controlled acquisition of audio tracks from online video sources, supporting batch processing and reproducibility;
    - Semantic file management: consistent semantic naming and directory structure to ensure stable references across audio, transcripts, and metadata;
    - Versioned data organization: separation between raw, processed, and optimized assets to support traceability and rollback.
- Speech-to-text and textual data production:
    - Automated transcription pipeline: generation of full-text transcriptions from audio using configurable speech-to-text models;
    - Multi-format transcript outputs: production of text outputs in multiple formats (plain text, timestamps, structured files) to support downstream uses;
    - Transcription quality control: mechanisms for manual review, correction, and reprocessing when required.

- Metadata extraction and semantic enrichment:
    - Keyword and entity extraction: automated NLP-based identification of relevant terms and named entities from transcripts;
    - Frequency analysis and ranking: statistical evaluation of extracted terms to support selection and relevance assessment;
    - Manual semantic curation support: structured workflows to refine, validate, and override automatically generated metadata;
    - Entity normalization: reconciliation of named entities against authority-controlled references (when applicable).
- Data optimization and provisioning:
    - Scripted and documented workflows: reproducible scripts for each pipeline stage to reduce manual intervention;
    - Audio optimization pipeline: compression of audio assets for efficient storage archival and delivery;
    - Format convertibility: support for exporting data and metadata in alternative representations for reuse and analysis.

Dissemination platform

- Core application structure and navigation. The platform is structured around three pages allowing the user to transversally navigate from a general entry point to specific content items:
    - home page: serves as entry point with search bar and alternative browsing sections;
    - collection page: a faceted catalog view for searching and filtering the lecture corpus;
    - lection page: the detail view for consuming a single lecture, integrating audio, text and metadata and offering semantically relevant browsing opportunity;
- Information retrieval:
    - Global and scoped search: implementation of a search bar on the home page for global queries, and specific search logic within the collection page;
    - Faceted filtering: the collection page includes filters to narrow down the results;
    - Full-text search: specific search bar within the lection page to allow user to locate keywords directly within the transcript text;
- Interactive media and transcription interface:
    - Integrated audio player: audio player embedded in lection page for playback of audio tracks;
    - Synchronized transcript rendering: render of full-text transcription synchronized with the audio timeline;
    - Interactive entities highlighting: recognized entities within the transcript are visually highlighted. User can interact to access contextual information without leaving the player interface;
- Semantic exploration and data visualization:
    - Geospatial navigation (map browsing): interactive map interface allowing user to browse lectures and entities based on their geographical locations;
    - Temporal navigation (timeline): a chronological visualization tool enabling user to explore specific historical events along a timeline;

- People directory: a dedicated visual index for browsing the “person” entities identified across the corpus;
- Contextual recommender: a “related lesson” section in the lection page that suggest content based on shared macro-themes or semantic proximity
- Semantical recommender: an “explore more” section semantically curated that allows to browse further the collection.

Domain

Barberotheca: a semantic digital library

The Barberotheca project is situated within the domain of semantic digital libraries. The primary objective is to develop a digital library that leverages Semantic Web technologies not only to optimize browsing and searching but also to facilitate deep exploration of the themes and concepts embedded within the lecture transcriptions.

To achieve this, the project addresses the complete lifecycle of library resources. The underlying **semantic data infrastructure** manages the critical phases of selection, acquisition, description, and preservation. The **dissemination platform** transforms the library into an active research tool, by implementing features such as contextual recommendation and semantic exploration.

The Barberomania: the need for transcriptions

In recent years, professor Alessandro Barbero has transcended the traditional boundaries of academia to become a significant cultural figure. Formerly a professor of Medieval History at the *Università degli Studi del Piemonte Orientale*, Barbero has achieved multimedia fame thanks to his narrative style and storytelling capabilities. His reach spans multiple media formats: from television collaboration, to podcasting world has host, guest and subject.
Barberotheca advances the state of the art by providing a written companion to the existing audio-visual experience. Beyond ensuring accessibility, this resource offers significant cognitive benefits and, most crucially, transforms the content from a diachronic medium, which dictates a linear pace, into a static medium that grants the user full control. This shift is the prerequisite for Barberotheca’s semantic exploration: while a user cannot 'click' a spoken word in an audio file or interact with a video frame, the transcription renders these elements actionable, allowing for deep exploration via a simple tap.

Stakeholders

The primary stakeholder for this project is Festival della Mente, based in Sarzana. Under the direction of Benedetta Marietti and promoted by *Fondazione Carispezia* and the *Municipality of Sarzana*, it holds the distinction of being the first European festival entirely devoted to creativity and thought processes. Held annually at the end of summer, the event is a cornerstone of the Italian cultural landscape, fostering a welcoming community atmosphere. Its mission is to investigate the shifts, energies, and hopes of contemporary society through lectures, workshops, and cultural exchanges aimed at a broad, intergenerational audience.

The Festival currently maintains a robust digital archive hosted on its official web platform. This repository includes a significant collection of podcasts and video recordings from past editions. While this archive ensures the preservation of the events, the content is currently restricted to linear playback (audio/video), limiting the user's ability to search for specific concepts or navigate the dense information contained within the lectures.
The stakeholder's engagement with the *Barberotheca* project is driven by a desire to innovate their digital dissemination strategy. They aim to leverage automated transcription technologies to transform their static media archive into an interactive knowledge base.

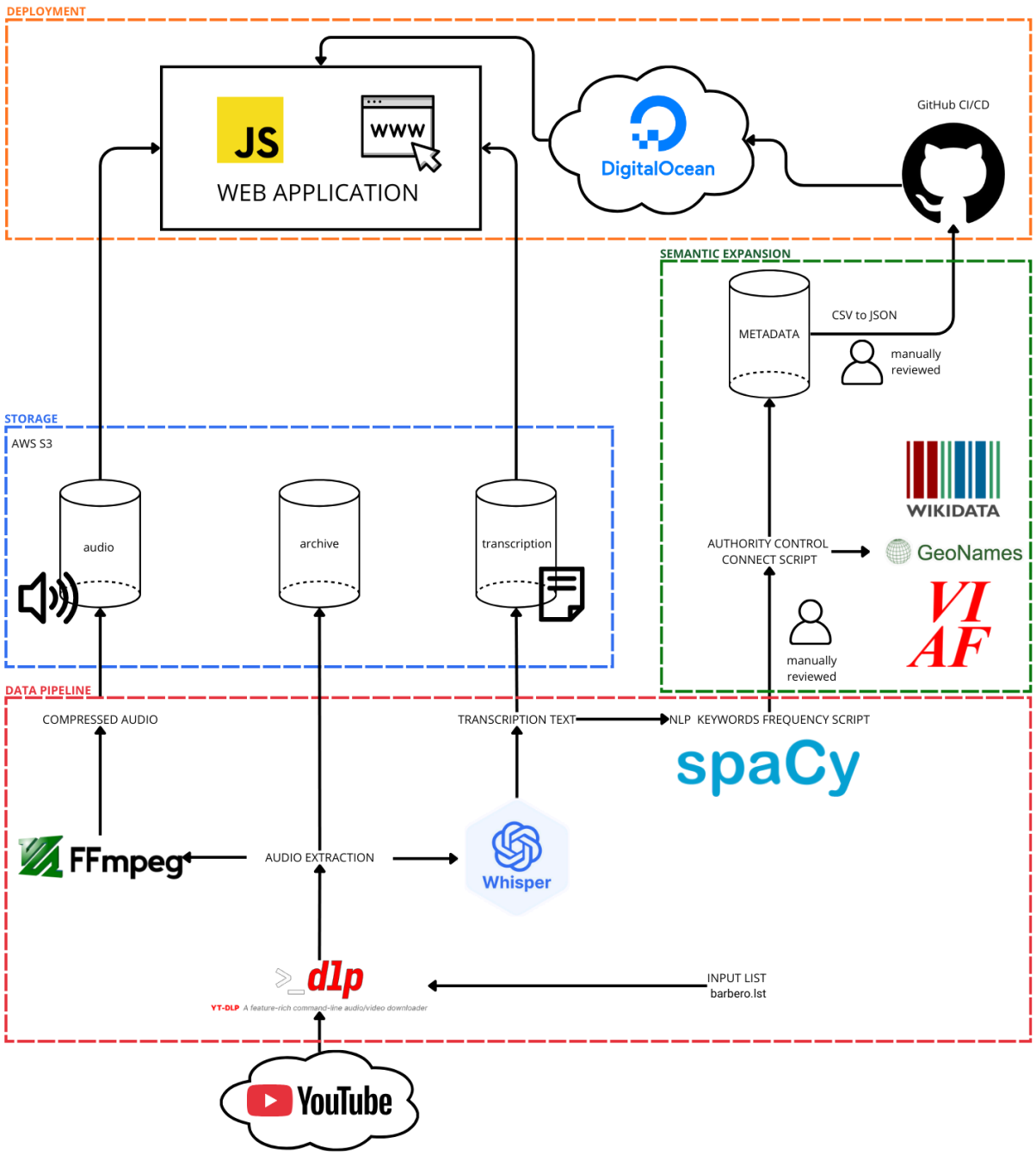
Barberotheca serves as a pilot initiative within this broader strategy. By focusing initially on the lectures of one of their most prominent guests, professor Alessandro Barbero, the project will validate the workflow for converting oral presentations into semantic text. The ultimate goal for the stakeholder is scalability: using the architecture developed for *Barberotheca* to eventually process and index the Festival's entire historical archive, thereby exponentially increasing the accessibility and research value of their cultural heritage.
Furthermore, the replicability extends beyond the *Festival della Mente*'s boundaries. The underlying semantic engine is designed to offer a replicable model that can be adapted to manage audio-visual collections from other cultural institutions.

System architecture

System's architecture diagram

The following diagram illustrates the system architecture design of Barberotheca, structured as a **multi-tier system** to ensure scalability and maintainability. The system is structured into four distinct **functional parts**, each addressing a specific aspect of the solution's workflow:

- **Deployment:** manages the production environment and application delivery. It hosts the client-facing web interface on DigitalOcean, handling user requests and content serving;
- **Data Pipeline:** orchestrates the ingestion and processing workflow. This layer utilizes yt-dlp for media acquisition, FFmpeg for normalization, and Whisper for generating automated transcripts;
- **Semantic Expansion:** dedicated to the logic of content enrichment. It employs components like spaCy to analyze text, extracting entities and generating metadata to support the fuzzy search and filtering features;
- **Storage:** handles the persistence of the system's data assets, creating a decoupling between processing and storage. It utilizes AWS S3 for storing heavy media files and structured JSON repositories for light-weight metadata.

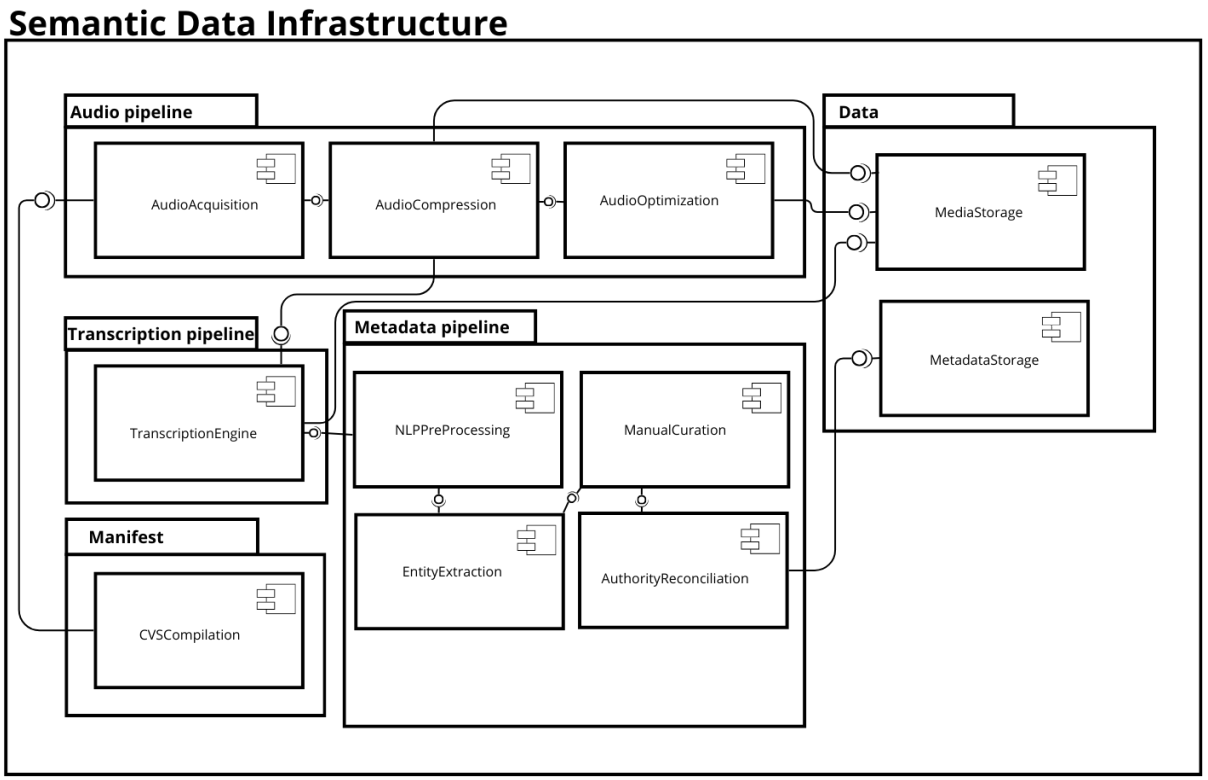


Semantic Data Infrastructure application diagram

The following diagram illustrates the application design of the semantic data infrastructure. The system is structured into five distinct **packages**: Audio pipeline, Transcription pipeline, Metadata pipeline, Manifest, and Data, which serve as groups of functionally integrated **components** designed to execute the ingestion and enrichment workflow.

- **Audio pipeline:** contains **components** responsible for the acquisition and normalization of media, specifically AudioAcquisition, AudioCompression, and AudioOptimization;
- **Transcription pipeline:** encapsulates the core speech-to-text logic through the TranscriptionEngine component;

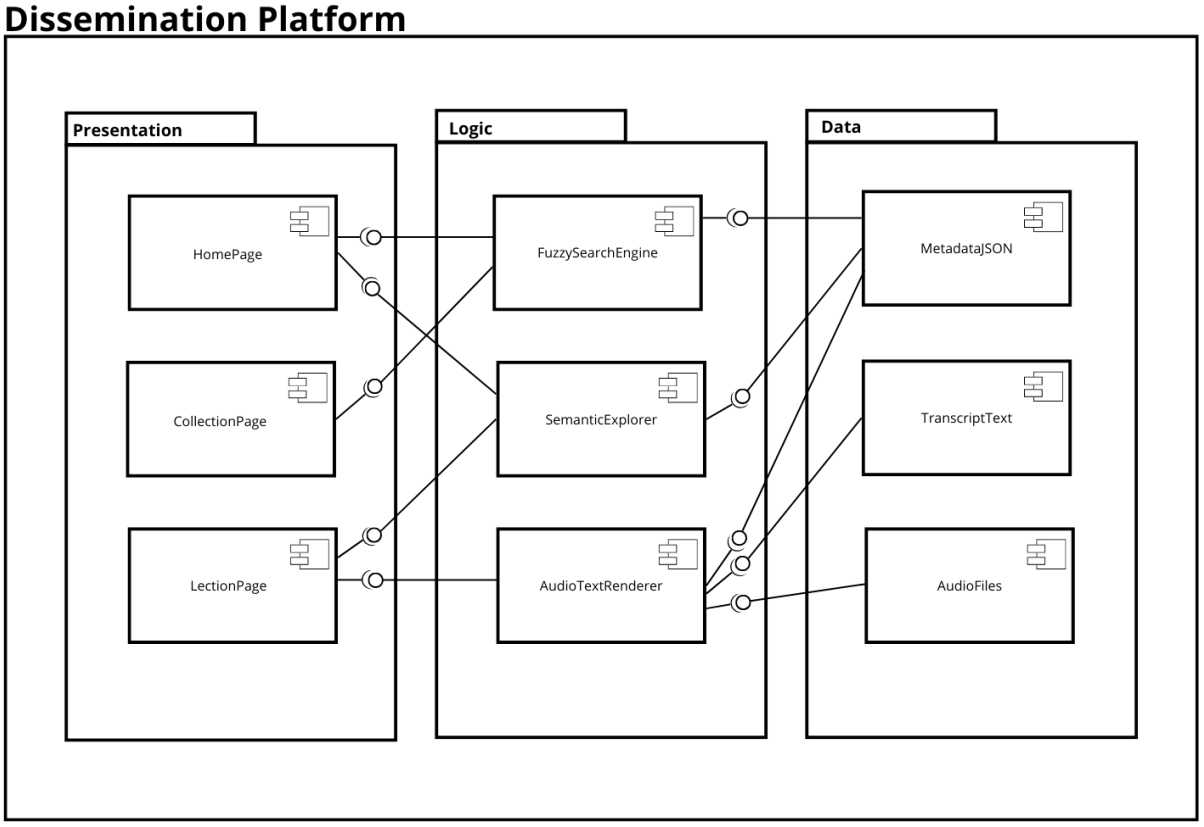
- **Metadata pipeline:** groups the components dedicated to semantic enrichment, including NLPPreProcessing, EntityExtraction, and AuthorityReconciliation, as well as ManualCuration for human validation;
- **Manifest & Data:** manage the persistence and final output generation via components like CVSCompilation, MediaStorage, and MetadataStorage.



Dissemination platform application diagram

The following diagram illustrates the application design of the dissemination platform. The **system** is arranged in three primary **packages**, presentation, logic, and data, which serve as groups of functionally integrated **components** deployed as a unit.

- **Presentation package**: contains the user interaction elements, specifically the Homepage, CollectionPage, and LectionPage components;
- **Logic package**: encapsulates the core processing functions via components such as the FuzzySearchEngine, SemanticExplorer, and AudioTextRenderer;
- **Data package**: manages the storage assets through components like MetadataJSON, TranscriptText, and AudioFiles.



General constraints and assumptions

The Barberotheca project operates within a framework of strict operational and technical constraints. The primary constraint is the timeline, which restricts the project lifecycle to **six weeks**. This necessitates the adoption of an agile development methodology to ensure deliverables are met on schedule. Financially, the project is bound by a **fixed budget** heavily weighted towards personnel costs, requiring precise resource allocation to prevent cost overruns.

The project's success relies on several critical assumptions regarding technology and resources. A fundamental technical assumption is the reliability and accuracy of the OpenAI Whisper models; it is presumed that automated **transcription quality** will be high enough to minimize the need for "human-in-the-loop" validation. The architecture also assumes the stability and **availability of external services**, specifically authority control APIs (Wikidata, VIAF, GeoNames) and source video streams accessible via yt-dlp. Finally, regarding human resources, it is assumed that the **agile methodology** will provide the necessary flexibility to manage partial requirements, and that the **team's composition** will effectively bridge the gap between technical implementation and domain expertise.

Requirements

The following tables list the identified requirements for the software, classifying them as either **functional** or **non-functional** and ordering them by **relevance** (from Mandatory to Nice to have). The requirements analysis adopts a dual perspective: the documented requirements related to the user interface primarily address the specific needs of the

end-users. On the other hand, the requirements listed for the data infrastructure largely reflect the team and stakeholders demands, specifically targeting the data re-use, interoperability, and the easy maintenance of the data environment over time.

Furthermore a detailed analysis was conducted on two sample use cases from which specific requirements were deduced. To maintain a dual perspective, one use case was selected to address the dissemination platform, while the other focuses on the data infrastructure. For each instance, the analysis defines the requirement scope, identifies potential user profiles, and outlines a corresponding user story and test case.

Semantic data infrastructure

Requirement	Type	Class	Relevance
The system must support the acquisition of audio data from online video sources in a reproducible way	System	Non-functional	Mandatory
The system must generate full-text transcriptions from audio files in Italian using automated speech-to-text tools	System	Non-functional	Mandatory
The system must support automated extraction of keywords and named entities from transcriptions	System	Non-functional	Mandatory
System pipelines must be portable in order to be executed on cloud or on-premises powerful machines	Domain	Non-functional	Mandatory
The system must support incremental extension of the corpus without reprocessing existing data	User	Functional	Mandatory
The system should enforce a consistent semantic naming and file structure across audio, transcripts, and metadata	User	Functional	Useful
Domain experts should be able to review, refine, and override metadata	User	Functional	Useful
The system should support metadata format conversion to enable interoperability and reuse	User	Functional	Useful
The system should support reconciliation of entities against authority-controlled references	User	Functional	Nice to have
The system should provide validation checks to detect misalignment between	User	Functional	Nice to have

audio, transcriptions, and metadata			

Sample requirement description and verification & validation

Use case	Requirement	Type	Class	Relevance
User wants to extend the corpus	The system must support incremental extension of the corpus without reprocessing existing data	User	Functional	Mandatory

Requirement scope

This requirement defines the process for extending and maintaining the semantic data corpus. The semantic data infrastructure must provide a reproducible workflow to ingest new lecture sources, generate audio and textual data, enrich them with semantic metadata, and expose consistent, structured outputs for downstream consumption. The focus is on data integrity, semantic coherence, and repeatability, rather than on user-facing interaction.

User profiles

Maintainer

A technical user responsible for running and maintaining the data pipelines. This user adds new lecture sources, executes processing scripts, and ensures that outputs comply with existing conventions.

Domain expert

A subject-matter expert responsible for reviewing and correcting semantic metadata, such as selected keywords and named entities, and for reconciling entities with authority-controlled references. This user validates semantic accuracy and promotes corrected metadata to production.

Stakeholder

A non-technical user interested in the outcomes and long-term impact of the project rather than its implementation details. This profile represents cultural or institutional stakeholders who aim to promote the project (e.g. to the *Festival della Filosofia* of Modena-Carpi-Sassuolo) and therefore require confidence in the cultural value of the corpus and in the reproducibility of the semantic data infrastructure in different contexts or institutional settings.

User story

As a Maintainer who needs to extend the corpus over time, I want to add a new lecture source to the semantic data infrastructure and process it through the existing pipeline, so that audio, transcription, and metadata are generated in a consistent and reusable form without impacting previously processed items.

Acceptance criteria

Given a new lecture source and a correctly configured environment, when the processing workflow is executed, then the system produces a correctly named audio file, a complete transcription, and a structured metadata entry aligned under a single semantic identifier and ready to be reviewed by a domain expert and then consumed by the dissemination platform.

Testing methodology

The testing methodology follows an end-to-end functional approach applied to the semantic data infrastructure, focusing on the correctness and coherence of the data production workflow as a whole. The pipeline is exercised using a real input source and evaluated through its tangible outputs, verifying that audio files, transcriptions, and metadata are correctly generated and consistently aligned, while also ensuring that processing occurs under realistic conditions in terms of data complexity, duration and variability, which is essential to assess the robustness of the infrastructure when extending the corpus.

Test case

Test Case T001	Maintainer adds a new lecture to the semantic data corpus
Pre conditions	Processing environment is configured and a new lecture source is available
Scenario	System processes a new lecture from audio extraction to metadata generation
Expected output	Audio, transcription and metadata are produced and aligned under a single semantic identifier

Prerequisite

A new lecture video source has been identified and is not yet present in the corpus. The processing environment (python, libraries and tools) is correctly configured and accessible.

Steps

1. Add the new lecture source and its basic descriptive metadata to the metadata input file
2. Run the audio acquisition process to extract the audio track from the source
3. Execute the speech-to-text transcription pipeline on the extracted audio
4. Run the keyword and entity extraction process on the generated transcript
5. Review and finalize the metadata entry, ensuring alignment between audio, transcript, and metadata outputs

Success

If steps 1–5 complete without errors and the system produces:

A. a new lecture audio source
B. a complete transcription of said audio source
C. a structured metadata entry linked to the same semantic identifier based on the textual content of the transcription

Fails

If any step from 2 to 5 fails, produces inconsistent filenames, incomplete or wrong outputs, or misaligned metadata.

Dissemination platform

Requirement	Type	Class	Relevance
The user must be able to search for a *lection* by text using a search bar widget in the home page	User	Functional	Mandatory
The user must be able to refine search results using specific filtering options	User	Functional	Mandatory
The user must be able to listen to the audio track of the lecture	User	Functional	Mandatory
The user must be able to read the full transcription of the lecture	User	Functional	Mandatory
The user must be able to browse the collection through visualization tools for entities (names, places, and events)	User	Functional	Mandatory
The system must provide a three-layered structure allowing the user to move from a general view to a content-specific view	System	Non-functional	Mandatory
The web application must meet WCAG 2.1 Level AA accessibility standards to ensure inclusivity	System	Non-functional	Mandatory
The application must display an access right disclaimer regarding the content	Domain	Non-functional	Mandatory
The user should be able to view a list of related lessons based on macro-themes	User	Functional	Useful
The user should be able to explore further the lecture theme through a semantically curated graph	User	Functional	Useful
The application interface should be responsive and functional across different browsers and devices	System	Non-functional	Useful
The user should be able to see the transcription synchronized with the audio track	User	Functional	Nice to have

The user should be able to interact with highlighted entities in the transcription to access context without leaving the page	User	Functional	Nice to have
The application UI should adhere to the "Festival della mente" brand guidelines regarding color palette, font choice and logo usage	System	Non-functional	Nice to have
Metadata should be available for exportation or download	Domain	Non-functional	Nice to have

Sample requirement description and verification & validation

Use case	Requirement	Type	Class	Relevance
User searches for a transcription	The user must be able to search a *lection* by metadata using a search bar widget in the home page	User	Functional	Mandatory

Requirement scope

This requirement defines the primary entry point for data retrieval on the platform. The search widget must be displayed on the home page, allowing users to perform metadata queries (title, file name, keywords, entities) without navigating to a specific catalogue page first.

User profiles

General visitor

A casual user exploring the website for the first time, likely to use broad keywords.

“Festival della mente” visitor

An enthusiast of the festival seeking to access specific lecture content. This user utilizes the platform to either experience a session they were unable to attend live or to revisit a talk they previously attended for deeper engagement. They are likely to search by the lecture title.

Researcher / Student

A targeted user looking for specific entities, expecting precise relevance in results.

User story

As a **"Festival della mente" visitor** (who wants to revisit or catch up on an event), I want to search for a specific lecture by its title (e.g. “Caterina da Siena”) on the home page, so that I can quickly find and read the transcription of that exact session.

Acceptance criteria

Given the lecture exists in the database, when I enter the exact or partial title of it, then the system displays the specific lecture card in the collection page.

Testing methodology

To validate the dissemination platform, a usability testing approach is adopted. Aligned with the project’s agile iterative timeline, testing sessions are scheduled at the conclusion of key development sprint.
The methodology consists of the following steps:

- **Participant selection**: a small group of 3-5 representative users is selected, matching the profiles previously defined (general visitors, festival enthusiasts and researchers);
- **Task design**: based on realistic **use cases**. Tasks are prioritized according to **operational profiles**;
- **Execution and metrics:** during the sessions, user interactions with the interface are observed to verify the **test cases**. Both quantitative data (task success rate, time-on-task) and qualitative feedback (user satisfaction, verbalized confusion) are collected to identify friction points in the UX/UI;
- **Iterative improvement:** results are analyzed to classify failure severity and impact. Critical issues are addressed prior to the final release.

Test case

Test Case T002	“Festival della mente” visitor searches for a specific lesson by title in the homepage via search widget
Pre conditions	Homepage search widget is implemented and specific lesson exists
Scenario	User request “Caterina da Siena” lesson transcription
Expected output	The system present the result list page showing the matching lesson card

Prerequisite

The lecture titled “Caterina da Siena” exists in the database.

Steps

1. Open the website home page
2. Locate the search widget
3. Enter the text “Caterina da Siena” into the search bar

4. Click the search button or press enter key
5. If lecture title is:
    a. VALID, then the system returns a result list containing the lecture “Caterina da Siena”.
    b. NOT VALID the system informs the user no matches were found.

Success

If exits step 5 in A or B.

Fails

If exits at step 1, 2, 3, 4. If exits at step 5 is different from A or B.