

## Task 5 - Implementation and testing

AIBeSa

*Authors: Beloslava Malakova (TU/e: 1923404), Alicja Gwiazda(TU/e:2017830),  
Stanimir Dimitrov(TU/e: 1932217), Aleksandra Nowińska (TU/e: 2008580)*

During the past two Sprints, we focused on generating embeddings for our dataset, which consists of both images and associated text descriptions. Once the embeddings were created, we stored them in our vector database (Qdrant). In parallel, we researched prompt engineering strategies and wrote system prompts according to the found principles for the LLM (Gemini), which will later be integrated into our pipeline. Additionally, we constructed a test dataset to evaluate the retriever's performance within the vector database. For each of the 50 artwork images we had embedded, we collected corresponding non-professional photographs of the same artworks, such as casual photos taken from different angles or distances, to simulate real-world usage scenarios. The Scrum Masters for these sprints were Bela and Alicja, who were responsible for keeping the team organized, maintaining motivation, and reminding about the deadlines.

For our system, the unit tests will focus on verifying the correctness and stability of the core components before integration. First, we will test the data upload and preprocessing module, ensuring that uploaded images are correctly saved, labeled, and converted into the expected format (valid file paths, consistent CSV entries). Second, we will include tests for the embedding generation pipeline, verifying that each image produces a fixed-length vector of the correct dimension and that embeddings are reproducible for identical inputs. Third, we will test the vector database operations, confirming that insertions, deletions, and similarity searches return consistent results and handle edge cases such as empty queries or duplicates. Fourth, the retrieval and ranking logic will be tested using mock embeddings to ensure that top-k results are correctly ordered by similarity score. Finally, we will include prompt-handling and output validation tests for the LLM component, checking that the system prompt is correctly formatted and that responses follow the expected schema. Together, these unit tests will maintain reliability and traceability across the image-to-LLM pipeline before integration testing.