

Formal Functional Specification

Grupo 1 - Mafalda Dinis e Martinho Figueiredo

This report consists of a more formalized specification of the functionality of our adaptive streaming multimedia system. In it we will make use of the UML standard to specify the goals for our system in a clear and concise manner. The report will begin by structuring the user requirements for our system from which the use cases are derived, after which we will explain our system's architecture through the use of our component and collaboration diagrams.

User Requirements

By thoroughly understanding the needs and expectations of users, we can ensure that our system will meet those demands. Hence, we started by defining some requirements taking into account the functionality, usability and user interface of our system.

1. Adaptive Streaming

The system should dynamically adapt to the user's environmental sound. When the user is in a noisy environment, video quality should decrease and the audio volume should increase. Finally, in the opposite scenario, the video quality should be increased and the audio volume lowered, to suit a more quiet environment.

2. Real-Time Processing

The system should analyse the user's environmental sounds and make the necessary adjustments in real-time or near real-time.

3. Cross-Platform Compatibility

The system should run on multiple devices and operating systems, providing consistent functionality and performance, which will contribute to an optimal user experience.

4. Seamless User Interface

The system should offer an intuitive and lightweight interface for interaction, using clear buttons to guide users toward desired actions.

Use Cases

Having outlined the requirements that our system needs to meet, we defined all the use cases in order to understand how the user will interact with each specific functionality. Designing a UML use case diagram allowed us to illustrate these interactions between the user and our system.



Figure 1 - UML Use Case Diagram

Component Diagram

In order to have a better understanding of how our system will be structured, we made use of the UML component diagram notation as a high-level representation of our systems architecture.

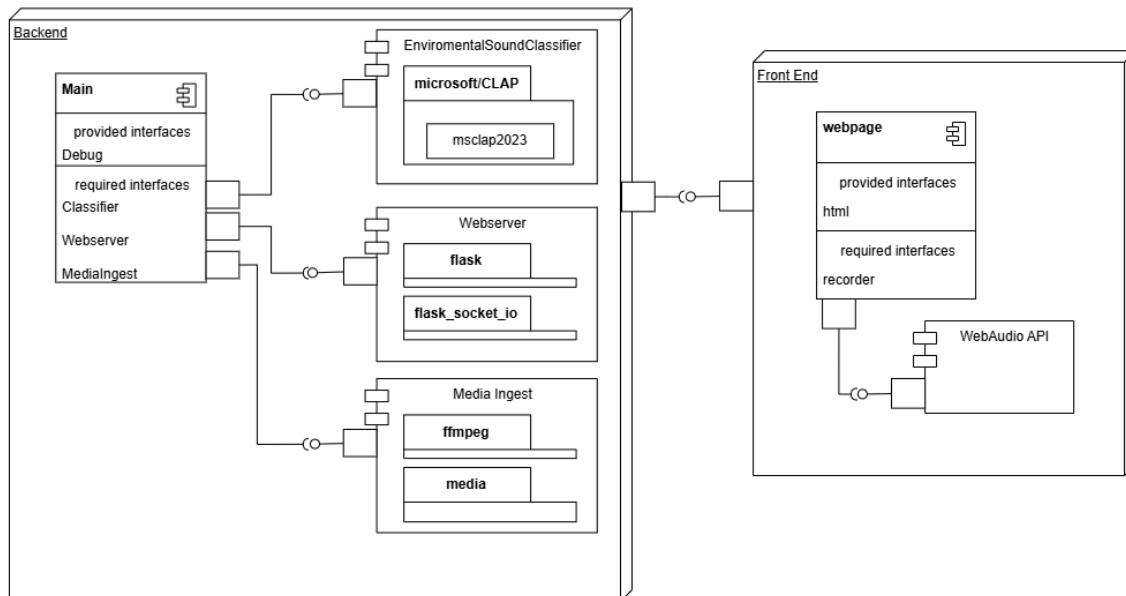


Figure 2 - UML Component Diagram

Collaboration Diagram

Finally, we made a collaboration diagram to help us visualize how the objects in our system collaborate to achieve different tasks, illustrating the flow of messages.

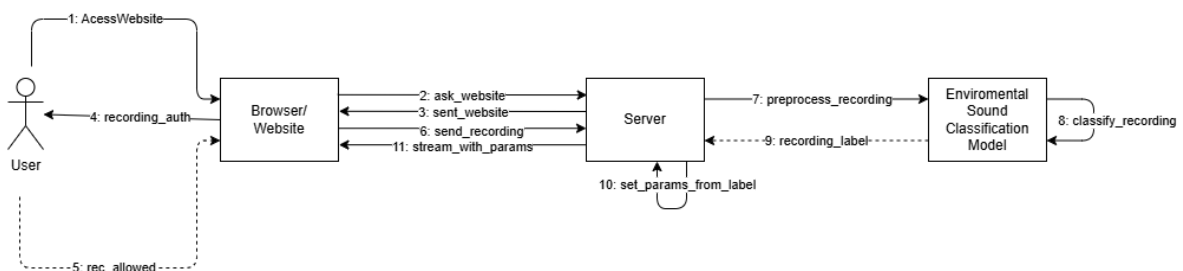


Figure 3 - UML Collaboration Diagram