Collaborative Audio System

By Jack Holmes

Abstract

A system that allows users to interact with each others recordings via a client/ server environment.

Essentially a client application that allows users to interact with each others audio clips by overdubbing. The server stores and provides the content to the client application.

Research

The research stage comprised of two main tasks:

- Researching core aspects of the proposed system.
- Carrying out experiments with possible solutions to test their suitability.

The research and experimentation stage provided invaluable insight into certain features that would have to be implemented, but also how they would be implemented.

Some of the topics that were covered by research and experimentation:

Data Interchange Formats	**Open-source Audio Tools**	**Server-side Languages**	**Audio File Formats**	**Concurrency Issues**
Metadata	**File Transfer**	**Data Storage**	**GUI Develop-ment**	**Internet Protocols**

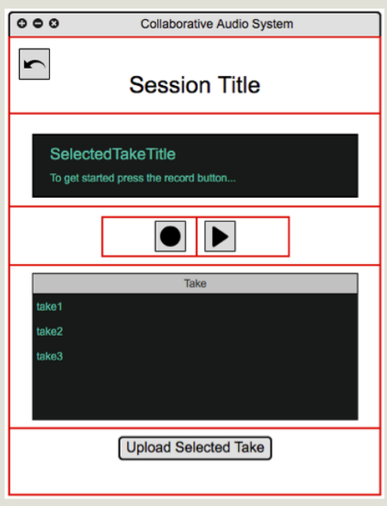
Once the majority of initial research had been completed, an experimental system was developed. This experimentation better informed the design and development stages.

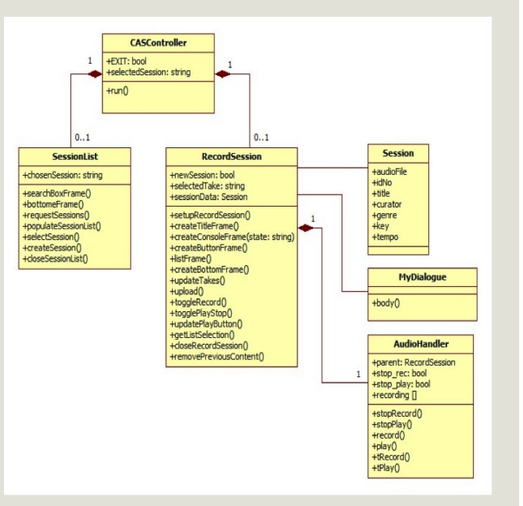
Design

Following the main research and experimentation phase an informed design could be made.

The first step was do define the system requirements, following this a use case diagram could be developed. The GUI was accomplished by taking into account the use cases, user scenarios and GUI design guidelines.

Once both the functional requirements and the system use was defined, the inner workings of the system could be designed by class diagrams and pseudo -code.

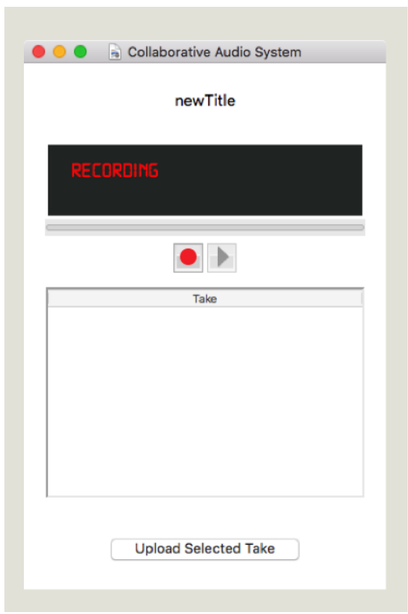




Development

The experimentation that had been carried out to this point provided a good starting point for development.

The first thing to be developed was the GUI. By developing this first, all backend features could be added through extension.



The second thing that needed to be implemented was the server. This required implementing a RESTful API that provided access to HTTP 'GET' and 'POST' requests.

Finally audio manipulation functionality could be implemented that allowed recording and overdubbing.

Testing & Evaluation

Requirements Testing

The first test carried out was requirements testing, essentially a checklist of features that had been successfully implemented. Every single requirement had been implemented.

Statement Testing

Statement testing was carried out to check crucial point in the code and input data that would potentially cause an error. It is debatable how effective this method of testing is, however, statement testing serves as good way of removing unstable areas of code. The statement testing actually uncovered that there were large amounts of code that was highly unstable and unable to handle exceptions.

User Testing

Finally, user tests were carried out by eight individuals to further test the requirements of the software. To measure whether the results obtained truly reflected the functionality of the system, a second user test measuring the effectiveness of the GUI was carried out.

The test results showed that all of the requirements had been successfully executed by the users. The GUI tests also showed that the users ability to carry out tasks was unhindered by the GUI.

Conclusion

Since the objectives were set at the beginning of the project , these objectives have steered the entire development of the project. Every stage of the project has fed-forward into following stages, creating a more robust system.

However, the project has suffered some small issues. Even when using research to find potential implementation issues, it is impossible to accommodate for every single problem that may arise. By using an incremental development lifecycle (carrying out experimentation with various implementations) a successful prototype was developed.