

Design and Development of a Web Based User Interface

Magda László

Bachelor's thesis of the Degree Programme in Business Administration
Business Information Technology
Bachelor of Business Administration

TORNIO 2014

ABSTRACT

LAPLAND UNIVERSITY OF APPLIED SCIENCES, Business and Culture

Degree Programme:	Bachelor of Business Administration
Writer:	Magda László
Thesis title:	Design and Development of a Web Based User Interface
Pages (1):	71(2)
Date:	30.05.2014
Thesis instructors:	Juha Meriläinen,
<p>The first objective of the thesis is to study the technological background of application design and more specifically the Unified Modeling Language (hereinafter UML). Due to this, the research provides deeper understanding of technical aspects of the practical part of the thesis work. The second and third objectives of this thesis are to design and develop a web application and more specifically a Web Based User Interface for Multimodal Observation and Analysis System for Social Interactions (hereinafter MORE) project. The aim of the web application is to support video and audio file management and analysis. The video and audio files are related to pedagogical research.</p> <p>The research questions include three questions. The first research question is as follows: What is the concept of UML? UML modeling technique used for application design has to be defined. The second research question studies what the functionalities of MORE web application are. The third research question focuses on what the development details and operation of MORE web application are.</p> <p>The exploratory research methodology is employed to ensure the academic validity of this thesis. Exploratory research is relevant for this research because it enabled studies into application design technology. Constructive method, the most suitable for software engineering researches is used to accomplish the objectives of the practical part of the research work. While this research is largely practical, constructive method is used for suggesting a way in which theoretical and practical solutions can be provided for an existing problem.</p> <p>The result and the expected output of this research are introducing an application design tool and the functions of the design tool during the design process. The outputs of this thesis work also includes development work, which aims to propose ways in which web application's functionality can be designed and implemented. The basic functions of the web application are video and audio annotation, search and report.</p>	
<p>Keywords: MORE project, data management, UML, eXist database, XQuery, web based user interface</p>	

ABBREVIATIONS

XML	Extensible Markup Language
XQUERY	XML Query Language
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
eXist database	Open source Native XML database
UML	Unified Modeling Language
Apache	HTTP Server
MORE	Multimodal Observation and Analysis System for Social Interactions
Ladybug SDK	Software Development Kit for Ladybug camera

CONTENTS

ABSTRACT

ABBREVIATIONS

FIGURES

ABSTRACT.....	2
1 INTRODUCTION	8
1.1 Background information and motivation	8
1.2 The MORE project introduction	9
1.3 Application design tool introduction.....	9
1.4 Structure of thesis.....	9
2 RESEARCH QUESTIONS AND METHODOLOGY.....	11
2.1 Research questions	11
2.2 Research methodology	12
2.3 Development details.....	13
3 REQUIREMENTS OF THE APPLICATION.....	15
3.1 System Overview	15
3.2 Interface Overview.....	17
3.2.1 Video annotation module.....	18
3.2.2 Video annotation filter module and, Video and Audio search module	18
3.2.3 Audio annotation module	19
3.2.4 Web application structure.....	20
4 UML.....	21
4.1 The concept of Unified Modeling Language	21
4.2 Objectives and Features of Unified Modeling Language.....	21
4.2.1 Diagrams of Unified Modeling Language.....	22
4.2.2 USE CASE diagram	23
5 DESIGN.....	24
5.1 USE CASE diagram.....	24
5.2 USE CASE tables.....	25

5.2.1	USE CASE tables of Administrator and User	25
5.2.2	USE CASE tables of Administrator	37
6	DEVELOPMENT	39
6.1	Introduction of development	39
6.2	Video Annotation Module's development in details.....	40
6.2.1	Search function	40
6.2.2	Search function with property filter.....	41
6.2.3	Confirmation function	42
6.2.4	Generating general information of the video.....	44
6.2.5	Load video file.....	45
6.2.6	Add annotation	46
6.2.7	Save annotation.....	46
7	OPERATION OF WEB APPLICATION.....	48
7.1	User Interface	48
7.1.1	User Interface of Home Page.....	48
7.1.2	User Interface of Video Annotation Module.....	50
7.1.3	User Interface of Audio Annotation Module.....	56
7.1.4	User Interface of Audio Annotation Search Module.....	61
8	CONCLUSIONS.....	67
	REFERENCES	68

FIGURES

Figure 1. Lightweight version (Keskinarkaus et al. 2013; Siipo et al. 2014, 3)	15
Figure 2. Rack version (Keskinarkaus et al. 2013; Siipo et al. 2014, 3).....	16
Figure 3. Overview of the MORE system (Keskinarkaus et al. 2013; Siipo et al. 2014, 3) ...	17
Figure 4. Schematic figure of video annotation module	18
Figure 5. Audio annotation module	19
Figure 6. Web application structure	20
Figure 7. Three complementary sets of diagramming tools (Pender 2003).....	22
Figure 8. System use case diagram(Ambler 2014)	23
Figure 9. USE CASE diagram of MORE Web Based User Interface	24
Figure 10. Main functionalities of the Video Annotation Module	40
Figure 11. List video names query function	41
Figure 12. Search page.....	41
Figure 13. Search video with properties query function.....	42
Figure 14. Confirm query function	43
Figure 15. Result of the query function for confirming the selected video	44
Figure 16. General information query function	45
Figure 17. Query function for loading video files	46
Figure 18. Add annotation script.....	46
Figure 19. Query function for saving annotations	47
Figure 20. MORE Web based user interface's components	48
Figure 21. Authentication window of the MORE webpage.....	49
Figure 22. The Home page of MORE Web Based User Interface.....	49
Figure 23. Video Annotation Module's search page	50
Figure 24. Video Annotation Module's video file confirmation web page	51
Figure 25. Video Annotation Module's main web page.....	52
Figure 26. Message about the saving.....	53
Figure 27. Video Annotation Module's property filter web page.....	54
Figure 28. Video Annotation Module's property value selection web page	55
Figure 29. Video Annotation Module's file name selection web page.....	56
Figure 30. Audio Annotation Module's audio file selection web page	57
Figure 31. Audio Annotation Module's selected audio file confirmation web page.....	58
Figure 32. Audio Annotation Module's channel selection web page	59
Figure 33. Audio Annotation Module's main web page.....	60
Figure 34. Audio Annotation Search Module's property value selection web page	62
Figure 35. Audio Annotation Search Module's property selection web page	63
Figure 36. Audio Annotation Search Module's confirmation web page	64
Figure 37. Audio Annotation Search Module's annotation selection page	65
Figure 38. Audio Annotation Search Module's main page	66

TABLES

Table 1. The Login USE CASE	25
Table 2. The Video selection for annotation USE CASE	26
Table 3. The Video selection for annotation with filter USE CASE	27
Table 4. The Add video annotation USE CASE	28
Table 5. The Delete video annotation USE CASE.....	28
Table 6. The Add general comment to the video USE CASE	29
Table 7. The Print and save video annotation USE CASE	30
Table 8. The Save video annotation USE CASE	31
Table 9. The Audio and channel selection for annotation USE CASE.....	31
Table 10. The Add audio annotation USE CASE	32
Table 11. The Delete audio annotation USE CASE.....	33
Table 12. The Add general comment to the audio USE CASE	33
Table 13. The Print and save audio annotation USE CASE	34
Table 14. The Save audio annotation USE CASE	35
Table 15. The Search Video Annotation USE CASE	36
Table 16. The Search Audio Annotation USE CASE.....	37
Table 17. The Upload files USE CASE	38

1 INTRODUCTION

The motivation of the thesis topic is discussed in this chapter. The background information is also described. In addition, the structure of the thesis work is discussed in this chapter.

1.1 Background information and motivation

This thesis focuses on web application design and development for the MORE project. The MORE project is a common effort in cooperation with the Department of computer science and engineering and Faculty of Education, University of Oulu. The work was conducted by the Department of computer science and engineering. The key focuses of the thesis research are MORE web application design with UML and implementation with eXist database. In addition, this research focuses on the concept behind UML application design tool and the tool's operation during the design process.

The background of this thesis work derives from practical placement that was completed at the University of Oulu between 15 July, 2013 and 15 November 2013. Also professional interest towards IT industry and especially for software engineering research lead to choose the topic of the work.

The applications for supporting pedagogical research have been progressively receiving increased attention today due to their effectiveness compared to traditional data collecting methods and their varied field of supporting IT tools. The growing trend of using technology in pedagogical research provides a great opportunity to examine the complexity of real-life learning situations. The interaction of these situations can be related to verbal and non-verbal means. The complexity of these situations requires a system which is capable of capturing the multimodal, multi-seated interactions as rich synchronized. These issues will be considered in a forthcoming article by Siipo, Huttunen, Holappa, Väyrynen, Keskinarkaus, László, Juuso, Laukka and Seppänen (2014, 1).

Several tool have been provided for the reason of observe interactions in the last decades. The aim of these tools focuses on different aspects of situation observation. Some of the best known tools for this purpose are Transana, VideoTrace, The Observer XT and Teachscape. Transana is defined as open source application, which focuses on organizing video materials and specific search based on key words (Transana.org 2005). According to Stevens (2001) VideoTrace is defined as annotation tool, which allows playing video in slow motion and saving comments

for the video. The Observer XT provides opportunity for collecting and analyzing observational data (Noldus Information technology 1989). The Observer XT is a big scale software package, which offers integration with other data acquisition systems. Teachscape is a complex set of tools, which allows developing and sharing video materials (Teachscape.com 2014).

The MORE project aims to create a system for multimodal observation and analysis for social interactions. The real life situations of social interaction can be business meetings or class room scenarios as well. (Siipo et al. 2014, 1.) For the reason of multimodal observation the MORE system provides a multichannel approach for data collection in different situations in various numbers of modalities.

1.2 The MORE project introduction

Based on the meeting with Ms. Keskinarkaus, MORE project leader the MORE project's development focuses on three main aspects. The first aspect is the multimodal research of interaction, which emphasizes the quantitative measurements of the test persons. The second aspect is the study of social interaction, which requires analysis of the measurements using computerized methods to fasten up analysis and extract important quantitative features. This thesis work forms a part of the third aspect, which focuses on supporting technologies for recording, editing, storing, managing data and enable integration of data analysis and data annotation methods. (Keskinarkaus 2013.)

1.3 Application design tool introduction

One of the potential ways for application design is the usage of Unified Modelling Language (hereinafter UML). According to Pender (2003) UML defines three important notations and several modeling techniques. The UML aims to support system development with varied techniques in practice. UML as a tool of system development specifies, visualizes and documents models. These models mainly focus on scalability and high volume implementation (Pender 2003, 3-22).

1.4 Structure of thesis

This thesis is divided into 8 chapters. Chapter 1 includes information about the MORE project, motivation and general background information. Chapter 2 explains the overall process of the research. Chapter 2 describes the selected research methods and techniques for the case

research. Chapter 3 contains the MORE system overview of the reason for describing the requirements of the application, technical specifications and functionalities of the MORE system and the MORE web based user interface. Chapter 4 focuses on the UML. Chapter 4 provides the theoretical knowledge of the application design process. Chapter 5 presents the application design process. Chapter 6 describes the development details. Chapter 7 presents operation of the web application. The final chapter, chapter 6 combines the conclusion of the thesis and future improvement of the web application.

2 RESEARCH QUESTIONS AND METHODOLOGY

The research questions are discussed in this chapter. The research methodology is also discussed. The development details are described in this chapter in order to achieve the objectives of the practical part of the thesis work.

2.1 Research questions

Based on the objectives and the scope of this work, the following research questions are defined.

1. What is the concept of UML?

The definition and concept of UML need to be understood in order to explore and understand the potential ways MORE web application can be designed. Therefore, answers to this research question are discussed in Chapter 4. In this chapter, fundamental concept of UML and diagram classification is defined, followed by detailed descriptions. For the reason of supporting the choice of the modeling technique of the application design process, the concept of USE CASE diagram is studied in details also.

2. What are the functionalities of MORE web application?

The specifications requirement document by Keskinarkaus, Siipo, Huttunen, Holappa, Juuso, Väyrynen and Laukka (2013) of the MORE web application was provided by the Department of Computer Science and Engineering, University of Oulu, in order to meet the aims of the MORE project. The MORE web application focuses on the video and audio annotation, search and report. Moreover, the functionalities required by the MORE project are supporting technologies for recording, editing, storing, managing data and enabling integration of data analysis and data annotation methods. The application focuses on the detailed description of the functionalities and interaction of the application. Due to this the thesis work is suggests an application design with the usage of the UML tool.

3. What are the development details and operation of MORE web application?

Besides the access of video and audio files, the development process focuses on XML data management. The development tool is eXist database. The development process was established with XQuery, HTML, JavaScript and CSS technologies. The MORE web

application suggests a way in which the main functionalities required by the MORE project can be implemented. The thesis work introduces an implementation via user interface functionalities and program code. The thesis work also introduces an established application design and implementation via the representation of the application's operation.

2.2 Research methodology

The research involves exploratory research based on the analysis of the literature and article reviews. According to McDaniel Jr. & Gates (2012, 43) the aim of exploratory research is to create greater understanding of the concept of action. Therefore, the topic of this research is relevant to exploratory research due to the fact that it helps in creating a clear understanding of the technologies used during the design of the application. The research method discovers the basis of application design methodology.

For a reason of established application design, it is important to highlight the way the design tool is operating during the design process. According to Pender (2003, 12-13) the UML aims to describe the modeling elements, not the method for creating tool of software development. Due to this, the research work studies the UML elements, in this case the UML models and notations.

For the reason of achieving successful application design and implementation, the research methodology that was selected for the practical part of the thesis work is constructive methodology. Constructive methodology's research aims to provide solution practical and theoretical problems. Due to the software engineering nature of the thesis work, the most suitable methodology was chosen.

According to Kasanen, Lukka and Siitonen (1993, 246) the essential phases of the construction problem solving are practical relevance, theory connection, practical functioning and theoretical contribution. Firstly, the practical relevance of the problem has to be defined. Secondly, the theory connection has to be represented of the reason for providing general and detailed understanding of the selected topic. Thirdly, the technical solution has to be represented. Finally s the problem solving method has to be supported by theoretical connections and research contribution (Kasanen & Lukka& Siitonen 1993, 244-260)

2.3 Development details

The research of this thesis started on 15 July, 2013, when the topic and aims of thesis were accepted by the Ms. Keskinarkaus, MORE project leader from Department of computer science and engineering, University of Oulu. The researcher had opportunity to work as an intern in MORE project at the Computer Science and Engineering Department of University of Oulu. At this point the basic functionalities of the MORE web application and the development tool were discussed. The major part of the design and development process was managed between 15 July, 2013 and 15 November, 2013. Process continued from January, 2014. The aim of the process was suggesting a thesis work based on the design and development work.

The expected practical outcome of the thesis work is an application design and implementation. This work suggests ways in which the application design can be established by studying the UML modeling techniques. In sum, this work suggests a possible solution for placing the application design methodology into practice, while implementing the first version of the application.

The application design aims to describe the MORE web application requirements and the description of architecture. The requirements of MORE web application are described in the Chapter 3. The USE CASE diagram draws out the description of the architecture.

The implementation includes the basic functionalities required by the MORE project. Thus the video and audio annotation, the annotation report and search functionalities form part of the development work. Chapter 6 includes introduction of the development work.

The following paragraphs describe the techniques, technologies used for the development of the web application.

XML

XML is defined as Extensible Markup Language and is defined as a markup language. XML is developed by World Wide Web Consortium (hereinafter W3C). XML focuses on creating common information format for sharing both the format and the data on the World Wide Web. (W3.org 2003.) XML is a flexible format, which was designed to carry and store data.

XQUERY

XQuery or XML Query Language is defined as a specification provided by W3C. XQuery is a query and functional programming language, which aims to transform collections of structured and unstructured data both locally and on the World Wide Web. (Walmsley2007, 1-3.) XQuery provides interaction between the web layer and the database layer of the web applications.

HTML

HTML is defined as Hypertext Markup Language and is authoring language for displaying data for web pages. HTML defines a structure and layout of documents of the Web. (Kyrnin 2013). Internet browser is the client application for the reason of converting syntax of HTML elements into visible objects.

CSS

CSS is defined as Cascading Style Sheets and is a style sheet used for specifying the display semantics, layout and formatting of a document written in a markup language (Kyrnin 2013). CSS is the style sheet languages that can describe align position and format at any HTML element.

JavaScript

JavaScript is defined as a scripting language developed by Netscape in order to enable design interactive web sites. JavaScript interacts with HTML source code. (Chapman 2013). JavaScript commonly related to a client-side script and focuses on the interactive functions of the web pages.

eXist database

eXist database is defines as Open Source Native XML Database. eXist database provides a platform for development of web applications based on XML format and concerned technologies. (exist-db.org 2014.) eXist database allows the user to develop modular applications which can be applied into any database with standardized packaging format. eXist database includes a set of tools in order to create application packages integrated into XQuery.

3 REQUIREMENTS OF THE APPLICATION

To achieve successful design and implementation of the MORE web based user interface, it is essential to understand the requirements of the application and the technical background of the MORE system. In order to achieve successful web application design process, technical details, which meet the requirements has to be specified. This chapter analyzes the MORE system's component with an in detailed description of the system overview, the relations of the system modules and the web based application requirements.

3.1 System Overview

The basic operation of the MORE system starts with setting up the recording system. The recording system has three essential elements. Figure 3 draws out the basic structure of MORE devices and their functionalities. These elements are sound, video and processing unit. The recording system has two versions. (Keskinarkaus et al. 2013; Siipo et al. 2014, 1.) The MORE recording system has two versions. These versions are Lightweight and Rack version. Figure 1 represents the first version, the Lightweight version.

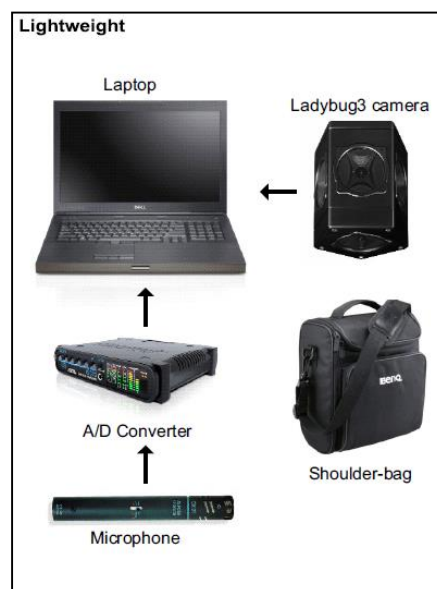


Figure 1. Lightweight version (Keskinarkaus et al. 2013; Siipo et al. 2014, 3)

The figure above includes the technical setup of the Lightweight version of the MORE recording system. According to the meeting with Mr. Huttunen, 2013 the technical setup of the Lightweight version of the MORE recording system contains the laptop, Ladybug3 camera,

A/D converter and microphone. Besides the recording the aim of the Lightweight version of the MORE recording system is to provide an easy to move version. The second version of the MORE recording system is Rack version. Figure 2 depicts the Rack version of the MORE recording system.

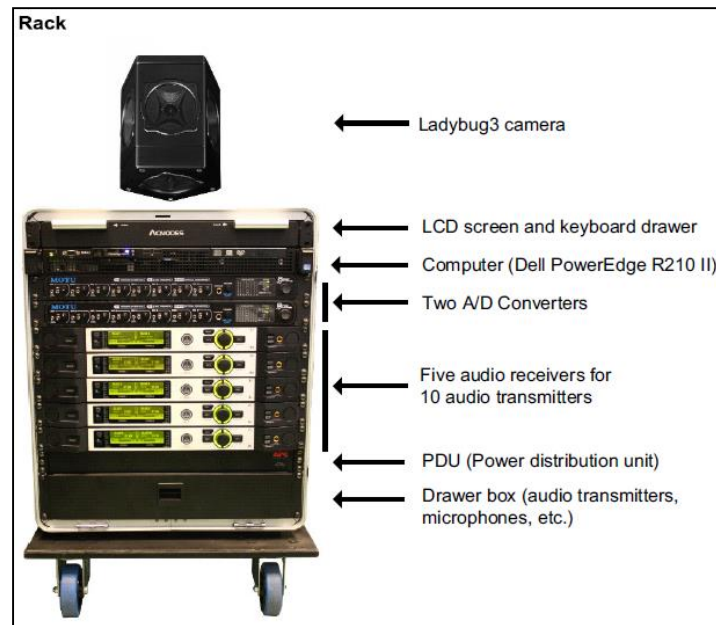


Figure 2. Rack version (Keskinarkaus et al. 2013; Siipo et al. 2014, 3)

The Rack version contains the Ladybug3 camera, LCD screen and keyboard drawer, Dell PowerEdge R210 II computer, A/D converters, audio receivers, power distribution unit and drawer box. (Siipo et al. 2014, 3-4.) All of the elements represented by figure 2 are necessary for appropriate operation of the MORE recording system.

Figure 3 draws out the functionalities, which have been provided by the MORE system. The recording system provides an opportunity for the user to record video, sound and processing unit.

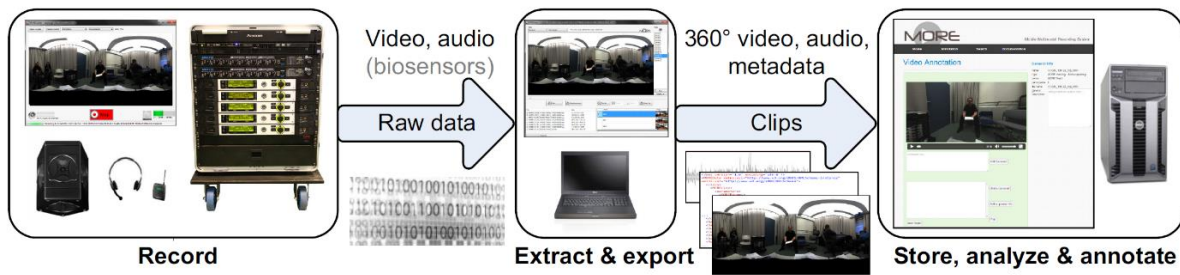


Figure 3. Overview of the MORE system (Keskinarkaus et al. 2013; Siipo et al. 2014, 3)

With these recorded materials, the system creates raw data. These materials need to be transferred via MORE Viewer application to clip format. The aim of the MORE Viewer application is to extract and export the raw material to clip and XML data. (Keskinarkaus et al. 2013; Siipo et al. 2014, 4.) The Web Based User Interface uses these video clips and XML data for video and audio display and annotation.

The MORE Web Based User Interface sub project forms a part of the MORE project. The scope of the sub project is to design and develop a web based user interface for the reason of adding and modifying documentation related to audio and video materials. (Siipo et al. 2014, 6-7.) More specifically, the objective of the MORE Web Based User Interface sub project is to create a web based environment, where the user can search, play and add general comments to the video or audio material. The user also can add comments to the video or audio files at the timestamp based level. The MORE Web Based User Interface also provides an opportunity to the user for saving and printing video report about the annotation.

3.2 Interface Overview

The three main annotation modules can be found in the menu of the webpage. The main modules are as follows: Video annotation module, Video annotation filter module and Audio annotation module.

The two Search modules can be found under the Search main menu. The search modules can help to the user to find the previously saved annotations. The search modules are Video search and Audio search.

3.2.1 Video annotation module

The Video annotation module allows the user to annotate video files in real time. Figure 4 below illustrates the schematic figure of the video annotation.

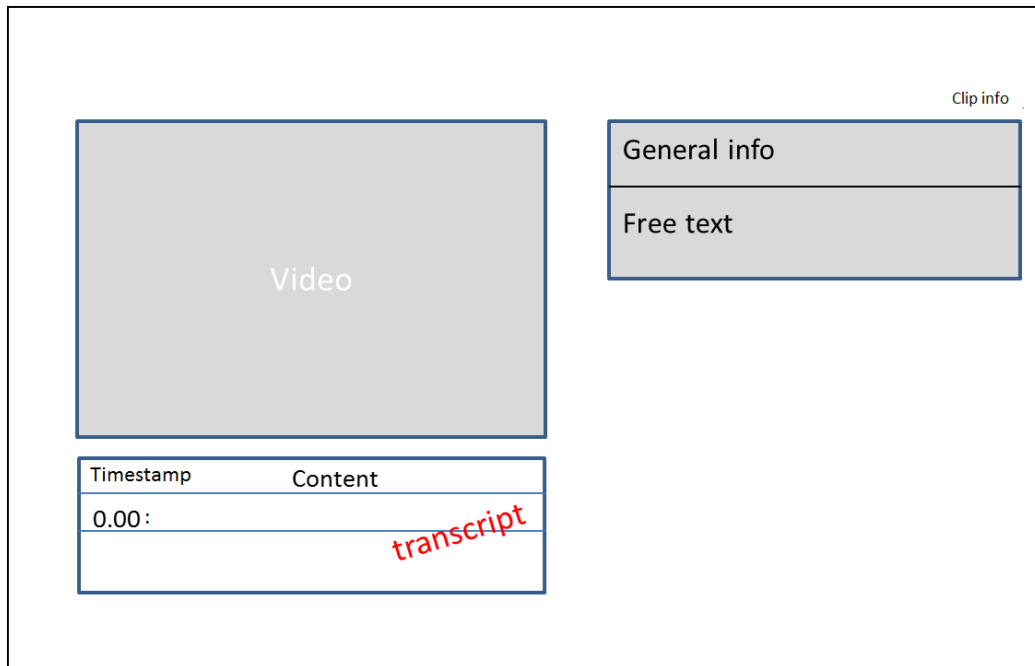


Figure 4. Schematic figure of video annotation module

The Video annotation module contains the video player, general information of the video file and video transcript. The video player is HTML 5 video player and it plays the previously selected video file. The General info field includes the general information relating the video, which has been previously stored into the database in XML file format. The Free text area enables the user to add general comment to the video. The transcript part enables the user to annotate the video at the timestamp based level. The annotations are saved into the database in XML file format. The Video Annotation module's operation starts with a search page. On the search page, the user can search for video based on the video's name. After the user selected the video, the video annotation is available.

3.2.2 Video annotation filter module and, Video and Audio search module

The basic logic of Video and Audio search are the search based on the video's or audio's name. After the search has been finished, the annotation is available.

The Video annotation filter differs from the Video annotation module in the search tool. The user can search for video based on the video's properties, such as the type of the video or owner of the video.

The Video annotation filter supports the user to find the right video. It provides different options for searching, while the Video annotation module search tool manages the search based on the name of the video. Both functions finish with opening the Video annotation module's main page.

3.2.3 Audio annotation module

The objective of the Audio annotation module is to provide an opportunity for the user to annotate different channels of audio files in real time. Figure 5 below introduces the schematic figure of the audio annotation.

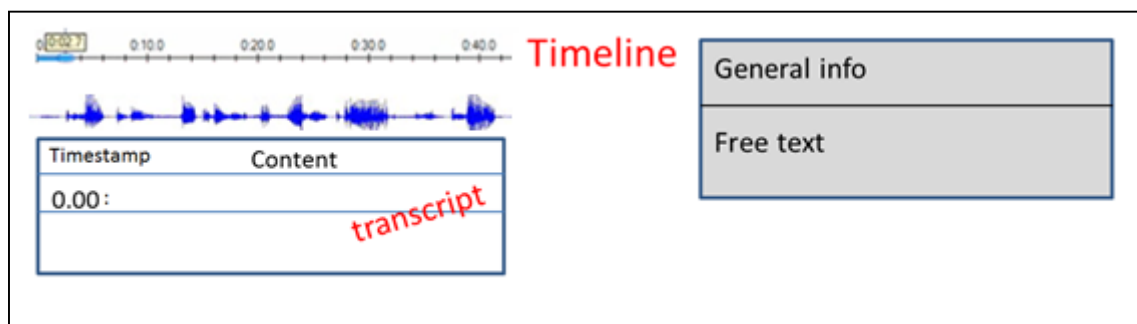


Figure 5. Audio annotation module

Figure 5 includes the most important parts of the Audio annotation module. The Audio annotation module contains the audio player, the audio channel visualization, the general information of the audio file and audio transcript. The audio player is HTML 5 player and it plays the previously filtered audio file. The General info field includes the general information relating the audio file, which has been previously stored into the database in XML file format. The Free text area enables the user to add general comment to the audio. The time line helps the user to control the audio player. The audio visualization highlights those intervallic of the audio track, where the user activity is increased. The transcript part enables the user to annotate the audio channel at the timestamp based level.

The Audio annotation module's operation starts with a search page. On the search page the user can search for audio based on the audio file's name. After selecting the video, the user

gets access to the channel selection tool. With the channel selection tool the user can choose the right audio channel of the video files. After the channel has been selected, the audio annotation module is available. The annotations are saved into the database in XML file format.

3.2.4 Web application structure

The required technologies for the development process are discussed as components in this thesis work. Figure 6 is a graphical representation of the web applications structure.

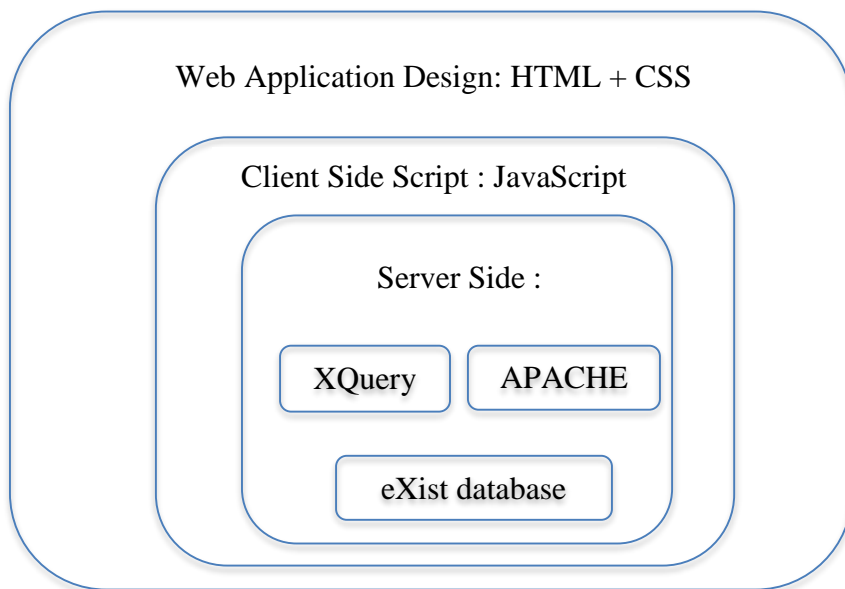


Figure 6. Web application structure

The MORE Web Based User Interface is built from components. The components are the following: The web application design content and semantics are HTML and CSS. JavaScript technology is used for the client-side scripts. The server side technology is XQuery. eXist database provides the XML based database layer on the server side.

4 UML

This thesis aims to suggest a way in which the MORE web application can be designed. To provide theoretical foundations for application design and give reason for the choice of the design tool, the concept of UML and its diagrams are described in this chapter.

4.1 The concept of Unified Modeling Language

UML focuses on the level of abstractness with analysis and design. UML highlights the level of coherence across systems and implementation via components and component libraries. For the reason of customization UML contains several extensions. (Pender 2003, 3-22.)

The high level of customization was the main aspect why UML was chosen for introducing the application design process. The different level of modeling emphasizes the main concept of application design from different aspects.

Extensively diversity of modeling techniques is presented by UML according to Bell (2004, 2). One risk of the design could be the too detailed model. Depending on the complexity of the system, UML tool usage could take more time of the development process than the program coding itself (Bell 2004, 2-3).

4.2 Objectives and Features of Unified Modeling Language

One of the most significant objectives of UML is to find a way in which the widespread real world needs meet with the capabilities of software developers. UML provides an integrated environment where UML diagrams can be built without determination of the programming language or the application runtime environment. Stated otherwise, the application design is independent from the implementation tools. (Pender 2003, 9-12.)

Even though UML do not insist on object oriented programming terms, it aims to support them. The infrastructure of UML allows for creating bases for different kind of development in various programming languages. Due to this, these features support the interoperability of UML between different technologies at a high level.

The diverse functionalities of UML are extensibility mechanisms, threads and processes, patterns and collaborations, activity diagrams, refinement, interfaces and components, constraint language and action semantics. The extensibility mechanisms aims to create bases

for unique design, while the threads and processes focuses on the interactional modeling. Patterns and collaborations focus on standard approaches to solving general problems. USE CASE diagrams are effective tools for highlighting the levels of abstraction. (Pender 2003, 12-14.) Constraint language provides the integrity of the model with specified syntax of the detailed description. Action semantics ensures the precisions of the model.

4.2.1 Diagrams of Unified Modeling Language

The UML provides a wide range of diagrams for specifying the design. One of the most accurate ways to classify the diagrams is to focus on the views. The categorization of the views allows for creating the relation between the diagrams of UML. Figure 7 draws out the relations between the diagrams of UML and their classification.

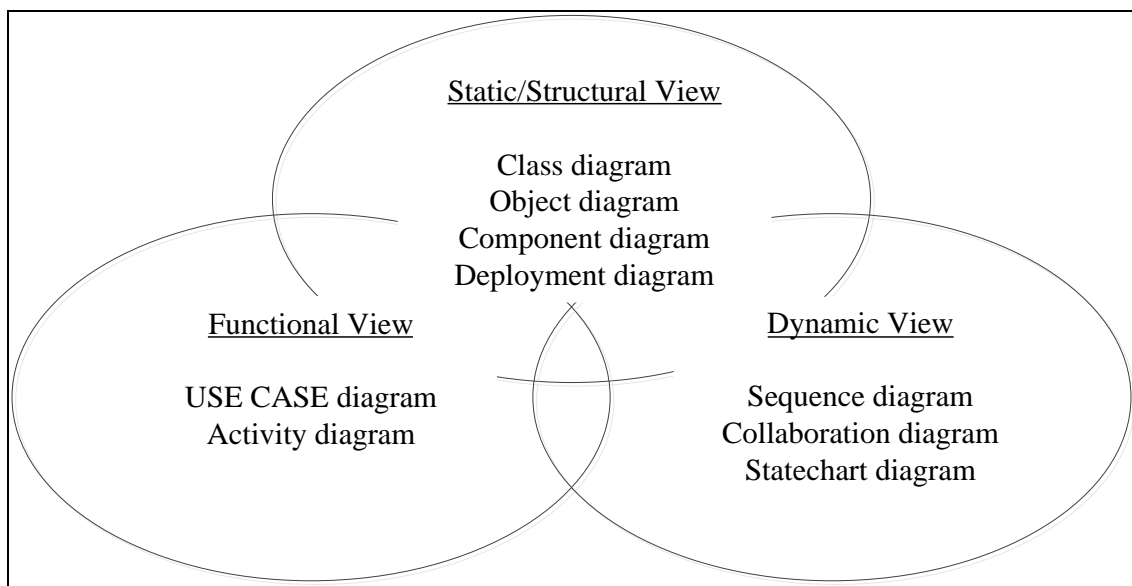


Figure 7. Three complementary sets of diagramming tools (Pender 2003)

Figure 7 represents three related collection of diagramming tools. The collections are categorized by views. These views are static/structural, functional and dynamic view. The functional view focuses on the behavior and models logic. For the reason of highlighting the user-application interaction, in the thesis work an UML tool were used in functional view.

4.2.2 USE CASE diagram

The USE CASE diagram identifies the operation of the application from the user aspect. The USE CASE represents the functionalities that the user expects during the operation of application. The figure below illustrates a sample for USE CASE diagram.

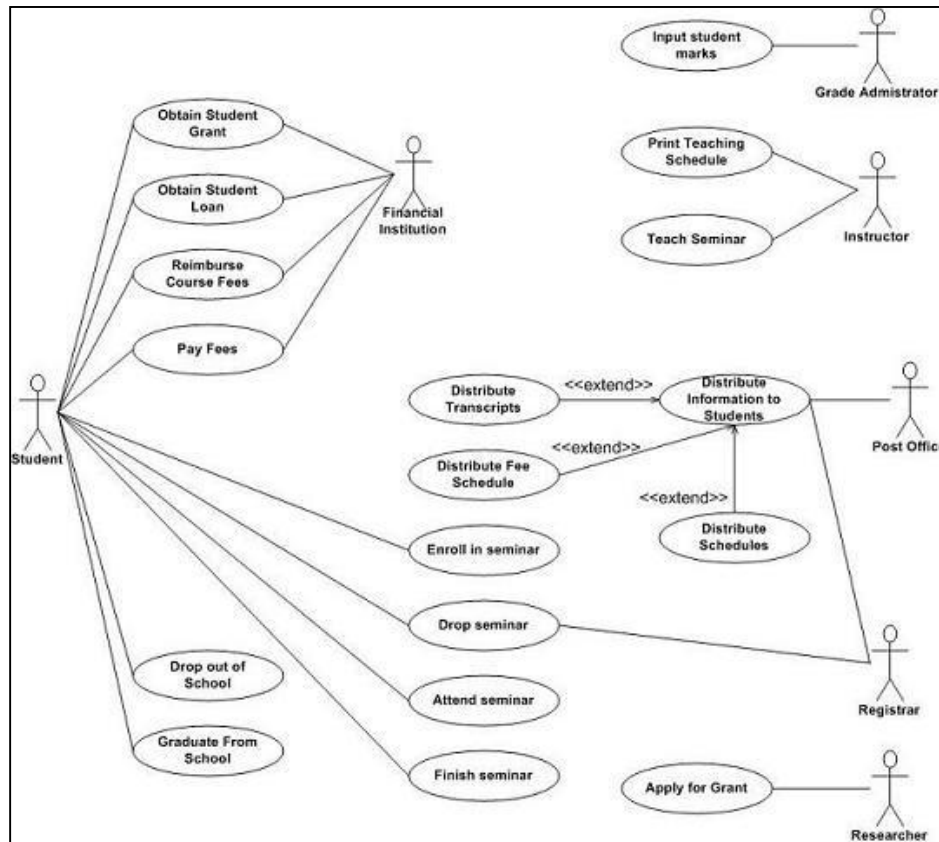


Figure 8. System use case diagram (Ambler 2014)

The USE CASE diagram includes actors and use cases. The USE CASE diagram also aims to identify shared behavior between the application functionalities. (Friedenthal & Moore & Steiner 2012, 303.) The available functionalities operation is defined as use cases. The USE CASE diagram aims to provide a representation of external behavior of the application.

5 DESIGN

This chapter goes into details concerning the functionalities of MORE web application. The USE CASE UML diagram introduces the functionality of the MORE web application from user interaction point of view.

5.1 USE CASE diagram

Figure 9 below introduces the USE CASE diagram. The USE CASE diagram supports the higher-level view of the MORE Web Based User Interface with deep understanding of the user's behavior. It contains all of the possible function of the MORE Web Based User Interface from user point of view.

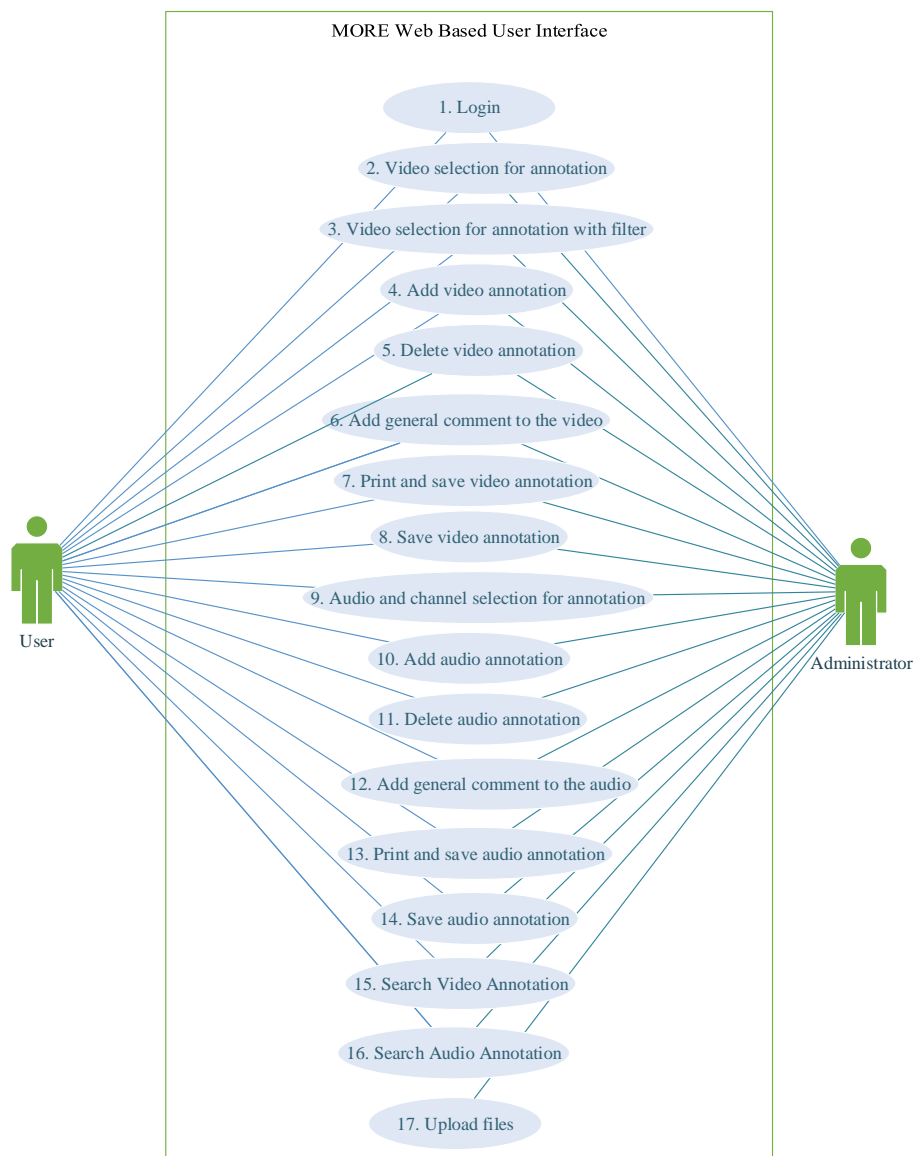


Figure 9. USE CASE diagram of MORE Web Based User Interface

5.2 USE CASE tables

The USE CASE tables aim to describe the USE CASE diagram in details. Each USE CASE table focuses on the different cases of the interface's interaction.

The USE CASE tables include Use Case Name, Trigger, Precondition, Basic Path, Alternative Paths, Post Condition and Exception Paths. The Use Case Name introduces the name of the USE CASE. The Trigger describes the main action of the user. The Precondition focuses on the status of the interface. The Precondition includes the requirement of starting the USE CASE's Basic Path. The Precondition is often a Post condition of a previous USE CASE. The Basic Path draws out the main function of the USE CASE. The Basic Path can include the interface's functions and the user's action. The Alternative Paths includes the non-regular interactions of the interface. The post condition includes the status of the interface after the Basic Path or the Alternative Path has been completed. The Exception Paths draws out the cases when the Basic Path or the Alternative Path cannot be completed.

5.2.1 USE CASE tables of Administrator and User

The table below draws out the Login process of the application. The Login USE CASE does not have precondition, because it forms the start function of the application.

Table 1. The Login USE CASE

1. Login USE CASE	
Trigger	A user opens the MORE Webpage.
Precondition	-
Basic Path	1. The application presents the user name and password field. 2. The user inputs user name and password.
Alternative Paths	2.1 If a user does not have an existing account a new account must be registered.
Post condition	A user is logged in.
Exception Paths	2.1 If user name or password is wrong the user is not logged in and error message is shown.

The Login process ensures safe usage of the application. The user has to be restarted to be able to login to the application.

The webpage provides a selection of options for different operation. In the menu the user can find the Video Annotation menu point. Table 2 below describes the process of Video selection for annotation.

Table 2. The Video selection for annotation USE CASE

2. Video selection for annotation	
Trigger	The user clicks on the Video Annotation menu point.
Precondition	The User is logged into the webpage of the MORE Webpage.
Basic Path	<ol style="list-style-type: none"> 1. The webpage presents the names of the video files in a drop-down menu. 2. The user selects the video file. 3. The webpage presents the selected video file information. 4. The user confirms the selection of the video.
Alternative Paths	-
Post condition	The webpage presents the Video Annotation page.
Exception Paths	-

The USE CASE of Video selection for annotation includes search option for the previously saved videos. If the user can find the right video, the annotation becomes available.

In the menu the user can find the Video selection for annotation with filter function. The table below describes the process of Video selection for annotation.

Table 3. The Video selection for annotation with filter USE CASE

3. Video selection for annotation with filter	
Trigger	The user clicks on the Video Annotation filter menu point.
Precondition	The User is logged into the webpage of the MORE Webpage.
Basic Path	<ol style="list-style-type: none"> 1. The webpage presents all of the filter options. 2. The user chooses which filter should be used for the search. 3. After selecting the filter, the user gets access to the video files' properties. 4. The user selects the video property. 5. After selecting the video property by the user, the application presents the names of the video files in a drop-down menu. 6. The user selects the video file. 7. The webpage presents the selected video file information. 8. The user confirms the selection of the video.
Alternative Paths	-
Post condition	The webpage presents the Video Annotation page.

The Video selection for annotation with filter function has similar search as the Video selection for annotation function, but in detailed opportunities for filtering the searching parameters

After successfully selecting the video file by the user, the webpage displays the Video annotation module. Table 4 below describes the Add video annotation function of Video Annotation module.

Table 4. The Add video annotation USE CASE

4. Add video annotation	
Trigger	The user decided to annotate to the video.
Precondition	The webpage presents the Video Annotation page.
Basic Path	<ol style="list-style-type: none"> 1. The user clicks on Play button of the video player. 2. The user adds comment to the video file with typing into the comment field and pressing the Add Comment button. 2. The webpage adds the comments with the time of the video to the second field.
Alternative Paths	-
Post condition	The video comment has been added.

The video Annotation module contains the video play, general information display and adds general information. The video Annotation module also includes add video annotation, delete video annotation, save and print and, save functions.

The user can remove comments from the comment field. The application provides opportunity for removing the selected annotation from the time stamped list of annotations. Table 5 below draws out the process of the Delete video annotation.

Table 5. The Delete video annotation USE CASE

5. Delete video annotation	
Trigger	The user decided to delete annotations from the comment field.
Precondition	The video comment has been added.
Basic Path	<ol style="list-style-type: none"> 1. The user selects the comment, which should be removed. 2. The user clicks on the Delete Comment button.
Alternative Paths	-
Post condition	The video comment has been deleted.

The Delete video annotation USE CASE describes the function of deleting. The deleting process starts with the decision of the user that an annotation has to be removed. The delete Comment button allows the user to use this function.

After access has been given to the user to the Video annotation module, the application gives opportunity to add general comment to the video file. Table 6 below introduces the process of the Add general comment to the video.

Table 6. The Add general comment to the video USE CASE

6. Add general comment to the video	
Trigger	The user decided to add general comment to the video.
Precondition	The webpage presents the Video Annotation page.
Basic Path	1. The user writes to the general description field on the Video Annotation page.
Alternative Paths	-
Post condition	The general description field has been filled.

The Add general comment to the video functionality allows the user to add comment to the general information of the video. The general information includes name and type of the video among others.

With the Print and save button on the Video Annotation webpage the user can print the report of the annotation and save it into the database. The table below introduces the Print and save video annotation USE CASE.

Table 7. The Print and save video annotation USE CASE

7. Print and save video annotation	
Trigger	The user decided to print and save the video annotation report.
Precondition	The video comment has been added.
Basic Path	<ol style="list-style-type: none"> 1. The user clicks on the Print and save button on the Video Annotation page. 2. The web browser presents the printing options. 3. The webpage saves the XML information and annotations to the database and the saving message appears on the bottom of the webpage.
Alternative Paths	2. If the user doesn't have printer, the webpage still saves the video annotation report.
Post condition	The video annotation report has been printed and saved.

When the user presses the Print and save button the web application saves the information related to the video and the annotations at the time stamped level. The annotation file is saved into the database.

The functionality of the Print and save and, the Save button is similar. With the Save button on the Video Annotation webpage the user can save the video annotation into the database. Table 8 below draws out the USE CASE of save video annotation function.

Table 8. The Save video annotation USE CASE

8. Save video annotation	
Trigger	The user decided to save the video annotation report.
Precondition	The video comment has been added.
Basic Path	1. The user clicks on the Save button on the Video Annotation page. 2. The webpage saves the XML information and annotations to the database and the saving message appears on the bottom of the webpage.
Alternative Paths	-
Post condition	The video annotation report has been saved.

The web application saves the information related to the video and the annotations into the annotation XML file. The annotation is at the time stamped level. The Audio and channel selection for annotation function enables searching for audio files in the database not only based on the video name, but on the channel number as well. The table below describes the process of Audio and channel selection for annotation.

Table 9. The Audio and channel selection for annotation USE CASE

9. Audio and channel selection for annotation	
Trigger	The user clicks on the Audio Annotation menu point.
Precondition	The User is logged into the webpage of the MORE Webpage.
Basic Path	1. The webpage presents the names of the audio files. 2. The user selects the video file which is related to the audio file. 3. The webpage presents the selected audio file information. 4. The user confirms the selection of the video. 5. The webpage presents the channels of the audio files in a drop-down menu and the audio channels' signal pictures with audio player. 6. The user selects the audio channel.
Alternative Paths	-
Post condition	The webpage presents the Audio Annotation page.

Audio and channel selection for annotation process starts with selecting the video, because all of the audio files are related to one video. After selecting the video file and the channel of the audio, the user can enter the Audio annotation module's main page. Table 10 below describes the Add audio annotation USE CASE.

Table 10. The Add audio annotation USE CASE

10. Add audio annotation	
Trigger	The user decided to annotate to the audio.
Precondition	The webpage presents the Audio Annotation page.
Basic Path	<ol style="list-style-type: none"> 1. The user clicks on Play button of the audio player. 2. The user adds comment to the audio file with typing into the comment field and pressing the Add Comment button. 2. The webpage adds the comments with the time of the audio to the second field.
Alternative Paths	-
Post condition	The audio comment has been added.

Add audio annotation function operating similar as the video annotation function. The difference between the displays of these two function is that the Audio annotation webpage includes signal image about audio channel. The image is placed in a way that the audio player and the sound wave's image are parallel, so the user can find the most active part of the audio material.

After the accomplishing to add at least one annotation – comment to the audio channel, the user get access to remove the selected annotation from the time stamped list of annotations. Table 11 draws out the process of the Delete audio annotation.

Table 11. The Delete audio annotation USE CASE

11. Delete audio annotation	
Trigger	The user decided to delete annotations from the second comment field.
Precondition	The audio comment has been added.
Basic Path	1. The user selects the comment, which should be removed. 2. The user clicks on the delete Comment button.
Alternative Paths	-
Post condition	The audio comment has been deleted.

The delete audio annotation function works on the same way as the Delete video annotation. The user can delete the selected annotation. On the Video annotation module includes the Add general comment to the audio function as well. The table below introduces the process of the Add general comment to the video.

Table 12. The Add general comment to the audio USE CASE

12. Add general comment to the audio	
Trigger	The user decided to add general comment to the audio.
Precondition	The webpage presents the Audio Annotation page.
Basic Path	1. The user writes to the general description field on the Audio Annotation page.
Alternative Paths	-
Post condition	The general description field has been filled.

The Audio Annotation Module's Add general comment to the video function operates on the same way as the Video Annotation Module's function. After the access has been given to the user to the Video annotation module, the web application gives opportunity to add general comment to the video file.

With the Print and save button on the Audio Annotation webpage the user can print the report of the audio annotation and save it into the database. The web application saves the information related to the audio channel and the annotations at the time stamped level into the annotation XML file. The XML file includes the selected channel's number as well. The table below introduces the Print and save audio annotation USE CASE.

Table 13. The Print and save audio annotation USE CASE

13. Print and save audio annotation	
Trigger	The user decided to print and save the audio annotation report.
Precondition	The audio comment has been added.
Basic Path	<ol style="list-style-type: none"> 1. The user clicks on the Print and save button on the Audio Annotation page. 2. The web browser presents the printing options. 3. The webpage saves the XML information and annotations to the database and the saving message appears on the bottom of the webpage.
Alternative Paths	2. If the user doesn't have printer, the webpage still saves the audio annotation report.
Post condition	The audio annotation report has been printed and saved.

With the Save button on the Audio Annotation webpage the user can save the audio annotation into the database. The web application saves the information related to the audio and the annotations at the time stamped level into the annotation XML file. The table below draws out the save audio annotation USE CASE.

Table 14. The Save audio annotation USE CASE

14. Save audio annotation	
Trigger	The user decided to save the audio annotation report.
Precondition	The audio comment has been added.
Basic Path	<ol style="list-style-type: none"> 1. The user clicks on the Save button on the Audio Annotation page. 2. The webpage saves the XML information and annotations to the database and the saving message appears on the bottom of the webpage.
Alternative Paths	-
Post condition	The audio annotation report has been saved.

The table below represents the Search Video Annotation USE CASE. The aim of this function is to support the searching of saved annotations in the database. The search process works on a similar way as the Video selection for annotation with filter function. The user can select the filter for the search and start a search based on the selected criteria. If the application can find matched result, the application displays it. With the web application the user also can search for annotations. Table 15 draws out the Search Video Annotation USE CASE.

Table 15. The Search Video Annotation USE CASE

15. Search Video Annotation	
Trigger	The user clicks on the Search Video Annotation menu point.
Precondition	The User is logged into the webpage of the MORE Webpage.
Basic Path	<ol style="list-style-type: none"> 1. The webpage presents all of the filter options. 2. The user chooses which filter should be used for the search. 3. After selecting the filter, the user gets access to the annotation's properties. 4. The user selects the annotation's property. 5. After selecting the annotation's property, the user can enter the webpage, which presents the names of the annotation files. 6. The user selects the annotation file. 7. The webpage presents the selected annotation file information. 8. The user confirms the selection of the annotation.
Alternative Paths	If there is not any annotation file with the searched information, the application not displays the result.
Post condition	The selected annotation has been displayed.

The Search Audio Annotation function works on a similar way as the Search Video Annotation.

Table 16 below represents the Search Audio Annotation USE CASE.

Table 16. The Search Audio Annotation USE CASE

16. Search Audio Annotation	
Trigger	The user clicks on the Search Audio Annotation menu point.
Precondition	The User is logged into the webpage of the MORE Webpage.
Basic Path	<ol style="list-style-type: none"> 1. The webpage presents all of the filter options. 2. The user chooses which filter should be used for the search. 3. After choosing the filter, the user can get access to the webpage, which presents the annotation's properties. 4. The user selects the annotation's property. 5. After the user selected the annotation's property the webpage presents the names of the annotation files. 6. The user select the annotated audio channel. 7. The user selects the annotation file. 8. The webpage presents the selected annotation file information. 9. The user confirms the selection of the annotation.
Alternative Paths	If there is not any annotation file with the searched information, the application not shows the result.
Post condition	The selected annotation has been displayed.

The first difference between these two functions is the file, which can be audio or video file. The second difference between these functions is that the Audio Annotation function includes search for the audio channel as well.

5.2.2 USE CASE tables of Administrator

The Upload files are the only function, which is available only for the Administrator. Table 17 illustrates the Upload files USE CASE.

Table 17. The Upload files USE CASE

17. Upload files	
Trigger	The administrator decides to upload video or audio files.
Precondition	The video clip has to be created by MORE Viewer.
Basic Path	The administrator uploads the file to the server.
Alternative Paths	
Post condition	The file has been uploaded.

The USE CASE's main process is very simple. The administrator uploads the files into a special folder. The application refers to that folder during the operation. On the current level of development it ensures the security of the application and structuring of database. The function of the uploading files is provided by the framework of the web development environment – eXist database.

6 DEVELOPMENT

For the reason of answering the third research question of this thesis work, this chapter introduces the MORE web application's development. To create foundation for the development work the first section of this chapter includes a short study about the technological background of the development. Due to the scale of the MORE web application development project and the similarities of Video Annotation Module with the other module's functionalities, the second section of this chapter provides description of the Video Annotation Module's development.

6.1 Introduction of development

Besides the design of the application, the practical part of the thesis work suggests a way in which the application can be implemented. XQuery, JavaScript, HTML, CSS and eXist database technologies has been used for the implementation of the application.

The development work is introduced by some of the main scripts and queries of the application. Firstly, the Video annotation module's functionalities are very similar to the Audio annotation module's functionalities, the basic difference is the channel selection option. Secondly, the Video annotation module's functionalities are very similar to Video annotation filter module, the difference is the detailed search functionality. Finally, the Video search and Audio search module are similar to Video annotation module's functionalities, the difference is that the search module uses the annotation XML files not the video or audio annotation files. Due to these facts the development work is drowned out only with the most important program code parts of the Video annotation module.

MORE web based application is a distributed application. According to Coulouris, Dollimore, Kindberg and Blair (2011) Distributed application's components are located on the networked computers, which communicate and coordinate their operation by passing messages. Distributed application structure partitions tasks or workloads between the providers of a resource and service requesters. The providers of a resource are and the service requesters are the clients. (Oracle.com 2014.)

The web application development is server and client sided. The web application's main components from implementation point of view are the database layer, the queries, scripts and user interface. The on the server side, the database includes the video, audio files, XML files,

which include the metadata for the annotation tool, such as general information of the video. The database is in a predefined structure. The queries act on the server side, while the JavaScript is operating on the client side.

The development environment is eXide, managed eXist database - online open source native XML Database. eXist database allows the usage of complete platform for the development of web applications related to XML technologies.

6.2 Video Annotation Module's development in details

The main queries are in app.xql file. The aim of the queries is to make requests into and modifications in the database. The queries also ensure the connection between the client sides of the application. The main module of the application is the Video Annotation Module from functionality point of view. The figure below represents the main functionalities of the Video Annotation Module.

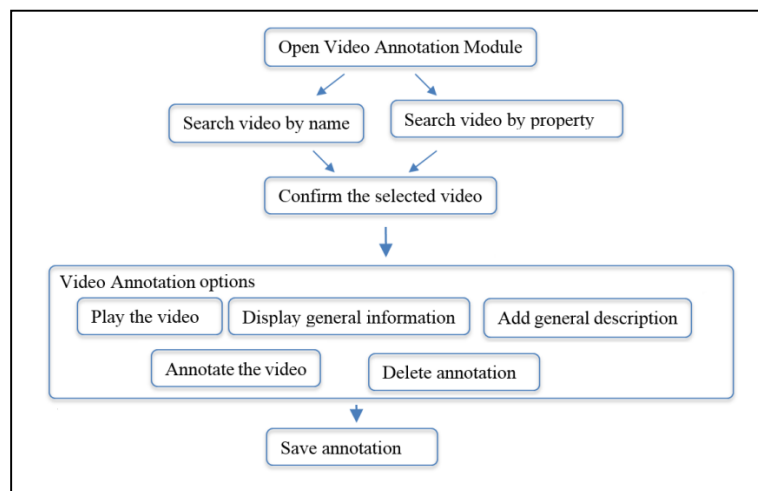


Figure 10. Main functionalities of the Video Annotation Module

The Video Annotation Module's essential functionalities are search, search by parameter, confirm the selected video, display general information, play video, annotate, delete annotation, add general description and save functions.

6.2.1 Search function

The figure below illustrates query function for searching video names. This query function is used for the Video Annotation Module's search page.


```

(:~ generate all clip names into selectbox:)
▼ declare function app:getvideopath($node as node(), $model as map(*) {
    <select name="fileSelection" id="fileSelection" onchange="">
    ▼
    {
        .....
        for $videopath in collection('/db/apps/MORE?select=*.xml')//Name
        return
            <option>{$videopath/text()} </option>
    }
    </select>
}

```

Figure 11. List video names query function

The aim of this query is to generate a list of names into a drop-down box. The \$ videopath variable transfers the list of the names from the datable to the user interface. The list includes all of the videos' names, which has been stored into the database. The output of the query appears in the search page's html code. The figure below represents this search page's html file.

```

<h1>Video annotation</h1>
<div class="row-fluid">
  <div class="span6">
    <p>Please select file:</p>
    <form method="GET" action="ListDataSelectedFile.html?fileSelection=value">
      <select name="fileSelection" id="fileSelection" onchange="">
        <option>TS316 Panorama</option>
        <option>TS 316 Sync test</option>
        <option>TS316 Own test</option>
        <option>ONK 1 Clip 1</option>
        <option>ONK 1 Clip 2</option>
        <option>ONK 1 Clip 0</option>
        <option>ONK 1 Panorama</option>
        <option>MORE Team Meeting Panoramic</option>
        <option>MORE meeting - Jukka speaking</option>
        <option>ONK 2 Panorama</option>
        <option>ONK 2 - Clip 0</option></select><br />
        <input type="submit" id="parameterselectOR" title="Load parameter" value="OR" onclick="" />
      </form>
    </div>
  </div>
</div>

```

Figure 12. Search page

The result of the query for searching video names can be found in the listSelectedFileData.html file. When the name is selected, the fileSelection value forwards the name of the video to the confirmation page.

6.2.2 Search function with property filter

The figure below illustrates the query function for searching video with properties. This query function is used for the search function Video Annotation Module's property value selection web page.

```

{:~ generate all clip ownres into selectbox, no duplicates :)
declare function app:getvideopathowner($node as node(), $model as map{+}) {
  <select name="ownerSelection" id="ownerSelection" onchange="">
  {
    for $videopath in distinct-values(collection('/db/apps/MORE?select=*.xml')//Owner/text())
    return
    <option>{$videopath [1] } </option>
  }
</select>

```

Figure 13. Search video with properties query function

The aim of this query is to generate a list of owners of the videos into a drop-down box. The application includes similar queries for the other property of the video, such as number of participants or name of the video. The \$videopath variable transfers the list of the names from the database to the user interface. The list includes all of the videos' owners, which has been stored into the database. The result of the query function for searching video with properties is represented by the figure above.

6.2.3 Confirmation function

The search video process is followed by the confirmation function. Figure 14 below shows the query function for confirming selected video. This query function is used for the search function Video Annotation Module's property value selection web page.

```

(: list selected data based on video name- in use :)
declare function app:getDataSelectedFile($node as node(), $model as map(*), $fileSelection as xs:string?)
{
  for $name in collection('/db/apps/MORE?select=*.xml')//MOREClip
  where $name/Name = $fileSelection
  return
  <div>
    <table id = "" class="">
      <tr>
        <td>name:</td>
        <td >{$name/Name}</td>
      </tr>
      <tr>
        <td>type:</td>
        <td>{$name/Type}</td>
      </tr>
      <tr>
        <td>owner:</td>
        <td>{$name/Owner}</td>
      </tr>
      <tr>
        <td>participants:</td>
        <td>{$name/Participants}</td>
      </tr>
      <tr>
        <td>file name:</td>
        <td>{$name/FileName}</td>
      </tr>
      <tr>
        <td><input type="hidden" name="filename" id="generalinfo" value="{ $name/FileName}"/></td>
      </tr>
    </table>
    <table>
      <tr>
        <td></td>
      </tr>
    </table>
  </div>
};

```

Figure 14. Confirm query function

The aim of this query is to generate a list of owners of the videos into a drop-down box. The application includes similar queries for the other property of the video, such as number of participants or name of the video. The \$videopath variable transfers the list of the names from the database to the user interface. The list includes all of the videos' owners, which has been stored into the database. The figure below depicts the result of the query

```

<form method="GET" action="videoannotation.html?filename=value">
  <div>
    <table id="" class="">
      <tr>
        <td>name:</td>
        <td>
          <Name>TS316 Panorama</Name>
        </td>
      </tr>
      <tr>
        <td>type:</td>
        <td>
          <Type>Panorama</Type>
        </td>
      </tr>
      <tr>
        <td>owner:</td>
        <td>
          <Owner>SH</Owner>
        </td>
      </tr>
      <tr>
        <td>participants:</td>
        <td>
          <Participants>1</Participants>
        </td>
      </tr>
      <tr>
        <td>file name:</td>
        <td>
          <FileName>130415_144920_clip_0000</FileName>
        </td>
      </tr>
      <tr><input type="hidden" name="filename" id="generalinfo" value="130415_144920_clip_0000" />
      <td></td>
      <td></td>
      </tr>
    </table>
    <table>
      <tr>
      </tr>
    </table>
  </div><br /><input type="submit" id="parametersselectOK" title="Load parameter" value="OK" onclick="" />
</form>

```

Figure 15. Result of the query function for confirming the selected video

The result of the query function for confirming the selected video is represented by the figure below. The result has been generated into ListDataSelectedFile.html file.

6.2.4 Generating general information of the video

One of the most important parts for identifying the selected video is the general description. The figure below represents the query function for generating general information of MORE web application. This query function is used for the Video Annotation Module's main page.

```

{:~ generate all file data on the page side into GENERAL INFO :)
- declare function app:showFolder($node as node(), $model as map(*), $filename as xs:string?) {
  let $name := $filename
  let $v2 := substring($filename,1,13)
  let $pathXML :=concat('/db/apps/MORE/', $v2, '/clips/', $name, '?select=*.xml')
  for $type in collection($pathXML)//Name
  for $owner in collection($pathXML)//Owner
  for $participants in collection($pathXML)//Participants
  for $file in collection($pathXML)//FileName
  for $comment in collection($pathXML)//Comment

  return
  <table id = "infoXML" class="app:showFolder">
    <tr>
      <td>name:</td>
      <td >{$name}</td>
    </tr>
    <tr>
      <td>type:</td>
      <td>{$type}</td>
    </tr>
    <tr>
      <td>owner:</td>
      <td>{$owner}</td>
    </tr>
    <tr>
      <td>participants:</td>
      <td>{$participants}</td>
    </tr>
    <tr>
      <td>file name:</td>
      <td>{$file}</td>
    </tr>
    <tr>
      <td valign="top">general description:</td>
      <td><textarea rows="10" name = "generaldescription"
      id="generaldescription" value="{ $comment}" placeholder="add general description here..."
      onblur="savegeneralinfo()"/></td>
    </tr>
    <td></td>
    <td></td>
  </tr>
</table>

```

Figure 16. General information query function

The aim of this support query is to generate a list of general information for the annotation page. The list of the general information id displayed on the right side of the webpage under General Info header. The general information contains the name, type, owner, number of participants of the video, and video file's name in the database. The general description part calls the `savegeneralinfo()` script of the `page.html` file. The `savegeneralinfo()` script identifies the content of the general description for later saving into the annotation XML file. Appendix 1 includes the result of the query function for generating general information.

6.2.5 Load video file

One of the most important functions of the Video Annotation Module's main page is the video player. The figure below illustrates the query function for loading video files

```

(: load video file- in use :)
declare function app:loadVideo($node as node(), $model as map(*), $filename as xs:string?)
{
  let $filename := request:get-parameter('filename', '')
  let $rootfoldername:= substring($filename,1,13)
  let $filpath:= concat('../MORE/', $rootfoldername, '/clips/', $filename, '/', $filename, '.mp4')
  return
  <div>
  <center><video height="380" width="600" id="ourvideo" type="mp4" controls="controls">
    <source src="{ $filpath }"/></video></center>
  </div>
};

```

Figure 17. Query function for loading video files

The aim of this query is to get the filename from the Video Annotation Module's file confirmation web page for playing the video. The query sets the source of the video for the HTML5 video player and generates it. Appendix 1 presents the result of the query function for loading video files.

6.2.6 Add annotation

The client side JavaScript allows the user to add comment to the video files. The figure below represents the script for adding annotations.

```

function addDateNComment()
{
  var $pop = Popcorn("#ourvideo");
  var textbox = document.getElementById("srt");
  var listBox = document.getElementById('listbox');
  var hiddenlistitems = document.getElementById("hiddenlistitems");
  var newOption = document.createElement('option');
  var string = (new XDate()).clearTime().addSeconds($pop.currentTime()).toString('HH:mm:ss') + " : " + textbox.value;
  hiddenlistitems.value += string + ";";

  textbox.value = string;

  newOption.innerHTML = textbox.value;
  listBox.appendChild(newOption);
  textbox.value=" ";
}

```

Figure 18. Add annotation script

This script is used for the Video Annotation Module's main page. The functionality of this script is to add time stamps to the comment, which has been added to the comment field and replace the comments into the second comment field.

6.2.7 Save annotation

The last step of the annotation is the saving. The figure below depicts the query for saving annotations.

```

(: saves the annotations, in use :)
declare function app:getdataa($node as node(), $model as map(*), $fileSelection as xs:string?, $sracomment as xs:string?,
$gnrcomment as xs:string?)
{
  let $numberofannotations := count(collection('/db/MOREAnootations?select=*.xml')//Filebuffer )
  let $numbering := sum( {1,$numberofannotations} )
  let $my-doc :=
    <MOREAnnotation>
    <Filebuffer>{$fileSelection}</Filebuffer>
    <AnnotationFileName>{concat($fileSelection, '_annotation_', $numbering, '.xml')}</AnnotationFileName>
    <Annotation>{$sracomment}</Annotation>
    <GeneralComment>{$gnrcomment} </GeneralComment>
    </MOREAnnotation>
    (: counts the existing same clip annotations :)

  let $FolderAnn := $my-doc/Filebuffer

  let $collection := 'xmldb:exist:///db/MOREAnootations'

  (: this logs you in; you can also get these variables from your session variables, default:'mylogin', 'my-password' :)
  let $login := xmldb:login($collection, 'admin', '')

  (: replace this with a unique file name with a sequence number :)
  let $file-name := concat($fileSelection, '_annotation_', $numbering, '.xml')

  return
  <results>
  {
    if ($sracomment)
    then {
      let $store-return-status := xmldb:store($collection, $file-name, $my-doc)
      return
      <message>New annotation: {$file-name} has been successfully saved</message>
    }
    else {
      let $store-return-status := xmldb:store($collection, $file-name, $my-doc)
      return
      <message>New annotation: {$file-name} has been successfully saved</message>
    }
  }
  </results>
};

```

Figure 19. Query function for saving annotations

This query function is used for the Video Annotation Module's main page. The aim of this query is to make the annotation saving available. The query gets the filename from the Video Annotation Module's file confirmation web page. The query uses data of the filename, the comments of video filled by the user and the general video description filled also by the user. Based on these data the query generates an annotation XML file. The query also gives message about the result of the saving process on the bottom of the Video Annotation Module's main page. Appendix 1 illustrates the result of the query function for saving annotations.

7 OPERATION OF WEB APPLICATION

One of the expected outcomes of this research is a representation of the established application design and implementation. For the reason of demonstrating the solution with optimal functionality, this chapter introduces the MORE web application's operation.

7.1 User Interface

The user interface form an important part of the web application from interaction point of view. Figure 20 below draws out the web application's main components.

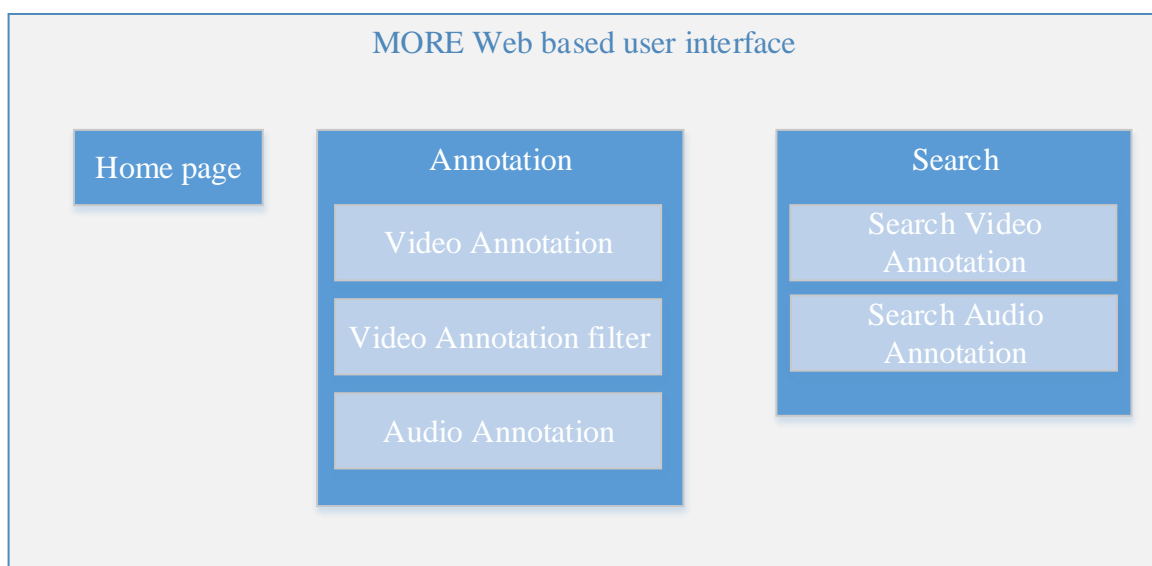


Figure 20. MORE Web based user interface's components

The main components are grouped by the main menu of the application. The groups are Home page, Annotation and Search. The Annotation sub menu includes the Video Annotation, Video Annotation filter and Audio Annotation modules. The Search sub menu includes the Search Video and Audio Annotation modules.

7.1.1 User Interface of Home Page

The application operation process starts with the Authentication window. Figure 21 shows the Authentication window of the application.

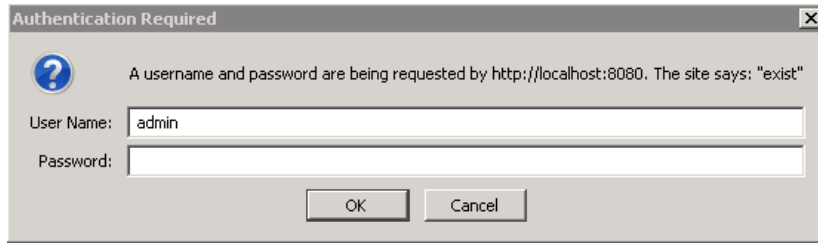


Figure 21. Authentication window of the MORE webpage

The identification of the users is a very important part from security point of view. The application is available only for the employees of the University of Oulu, due to the confidential contents of the research videos. After logging in, the user can access the homepage. The figure below introduces the Home page of the application.

MORE Mobile Multimodal Recording System

Home Annotation Search Documentation

About the project

The multitechnological follow up and analyses of emotional activity

Collaborating with:

Professor Tapio Seppänen [e-mail](#) ,
Department of computer science and engineering, Faculty of Technology

Professor Janne Heikkilä [e-mail](#) ,
Center for Machine Vision Research, Department of computer Science and engineering, Faculty of Technology

Professor Matti Lehtihalmes [e-mail](#) ,
Logopedics, Faculty of Humanities

Adunct Professor Seppo J. Laukka [e-mail](#) , Laboratory manager,
LearnLab, Faculty of Education

The MORE recording system has been presented in schools and kindergarten:

- Patamäki
- Oulun normaalkoulut
- Pohjankartano
- Ritaharju
- Pöllönkangas

The MORE recording system has been used in projects:
[Kitkaton kommunikaatio \(KIKO\)](#)

SYSTEM OVERVIEW

The study of social interaction requires analysis of the measurements using computerized methods to fasten up analysis and to extract important quantitative features. The MORE system will be used for multimodal research of interaction, where quantitative measurements of the test persons, measured during the recording, play an important part. MORE development will focus in bringing up technology to store data, to manage data, to enable integration of data analysis and data annotation methods.

Recording event

Web-based multitechnology Observation and analysis system

Server: Analysis

Figure 22. The Home page of MORE Web Based User Interface

The web application's Home page contains the general description of the MORE project, system overview, sample of the system operation, introduction of the system components, further opportunities of the development project and the participants.

7.1.2 User Interface of Video Annotation Module

One of the main components of the MORE web application is the Video Annotation Module. The figure below draws out the video search by name function of the Video Annotation module.

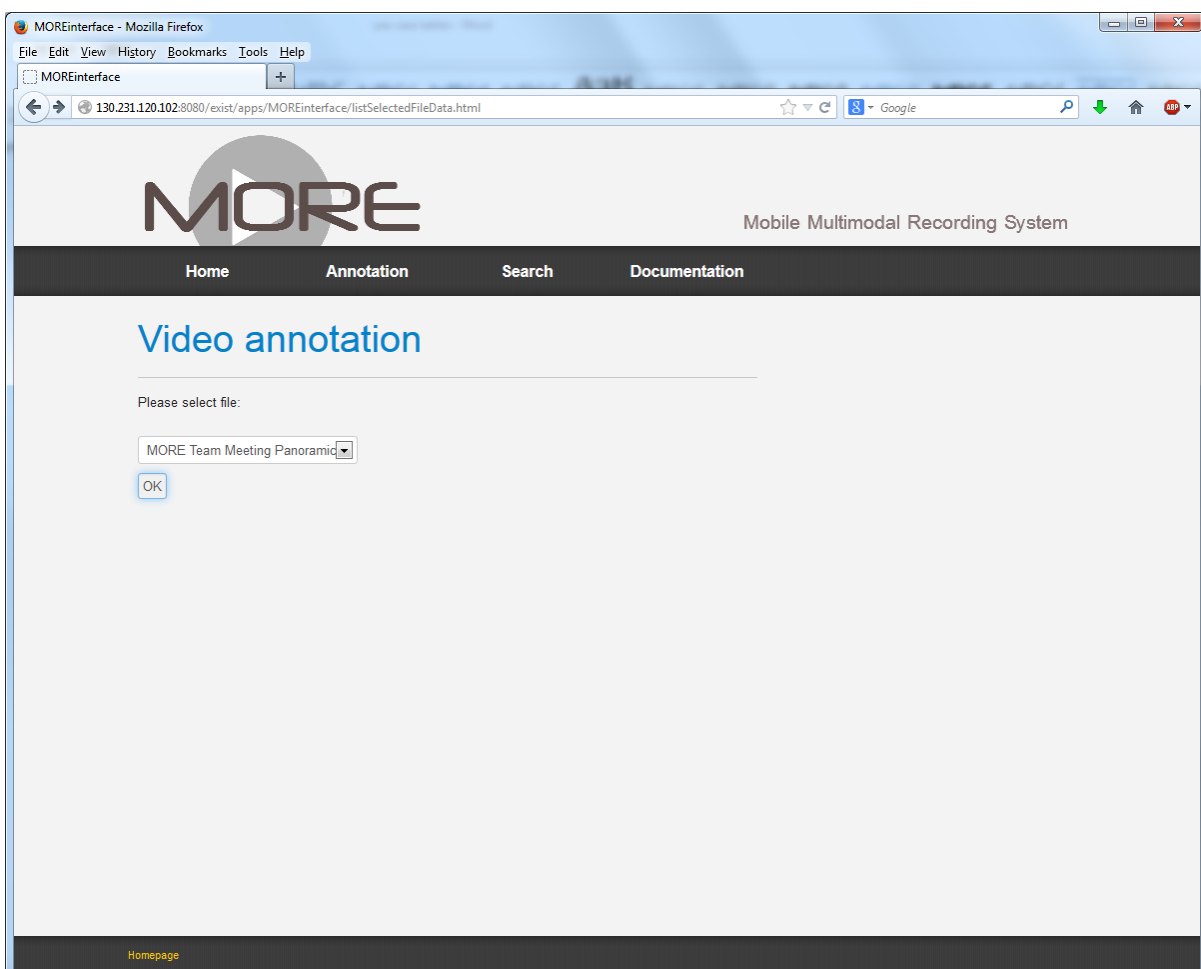


Figure 23. Video Annotation Module's search page

The Video Annotation Module contains several pages webpages. Depending on the user's selection, there are two ways of searching for video files. The first way is to search for the video's name. The second version allows the user to select the property of the video as filter for the search. The functionality of the page includes a drop-down menu from where the user

can select from the names of the video. When the user presses the OK button, the application opens the confirmation web page. Figure 24 illustrates the Video Annotation Module's video file confirmation web page.

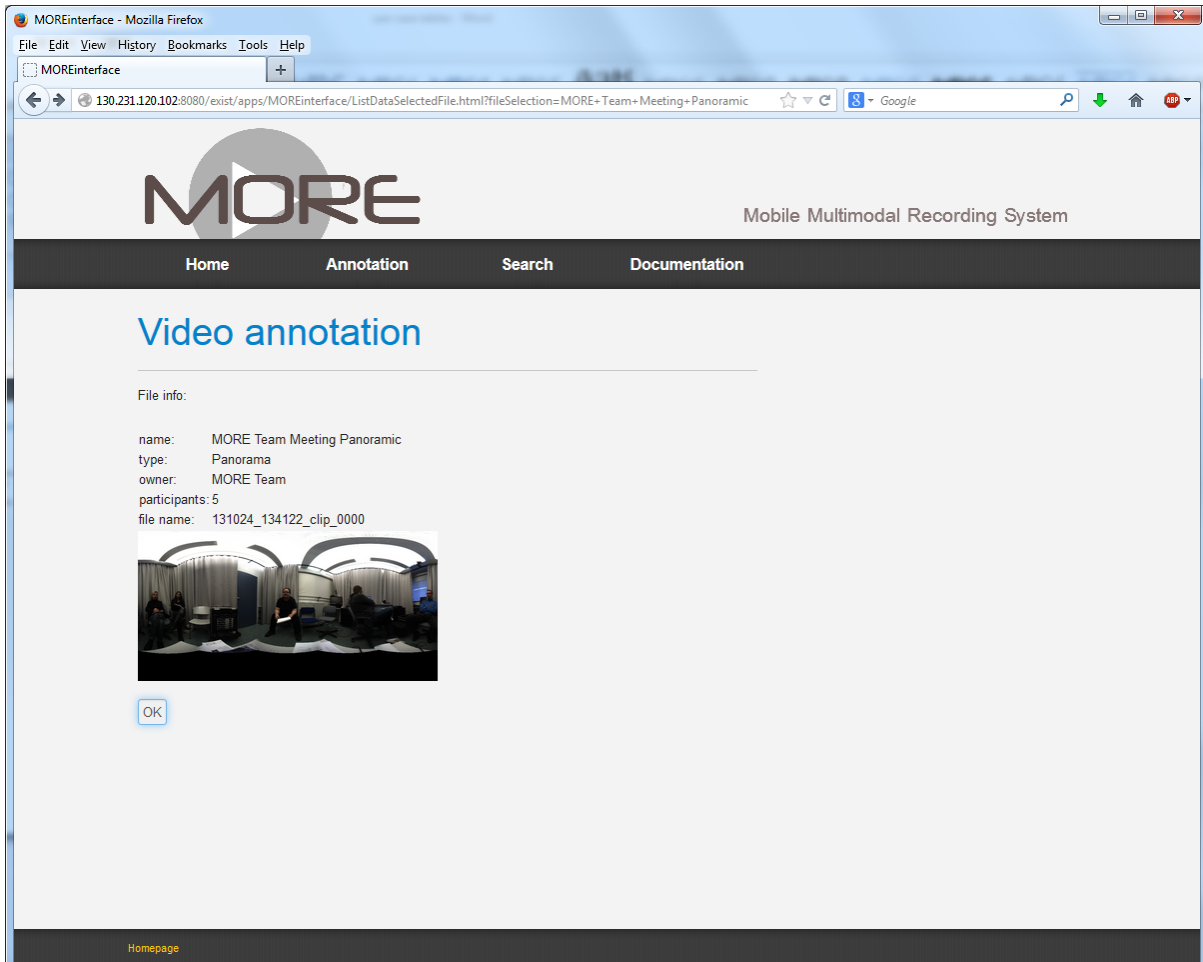


Figure 24. Video Annotation Module's video file confirmation web page

The video file confirmation web page ensures the user about the accuracy of the selected video. The confirmation web page includes information about the video file. This information can be name, type, owner of the video, number of participants and the file name. The page also includes picture of the video. When the user presses the OK button the main web page appears. The figure below includes the Video Annotation Module's main web page.

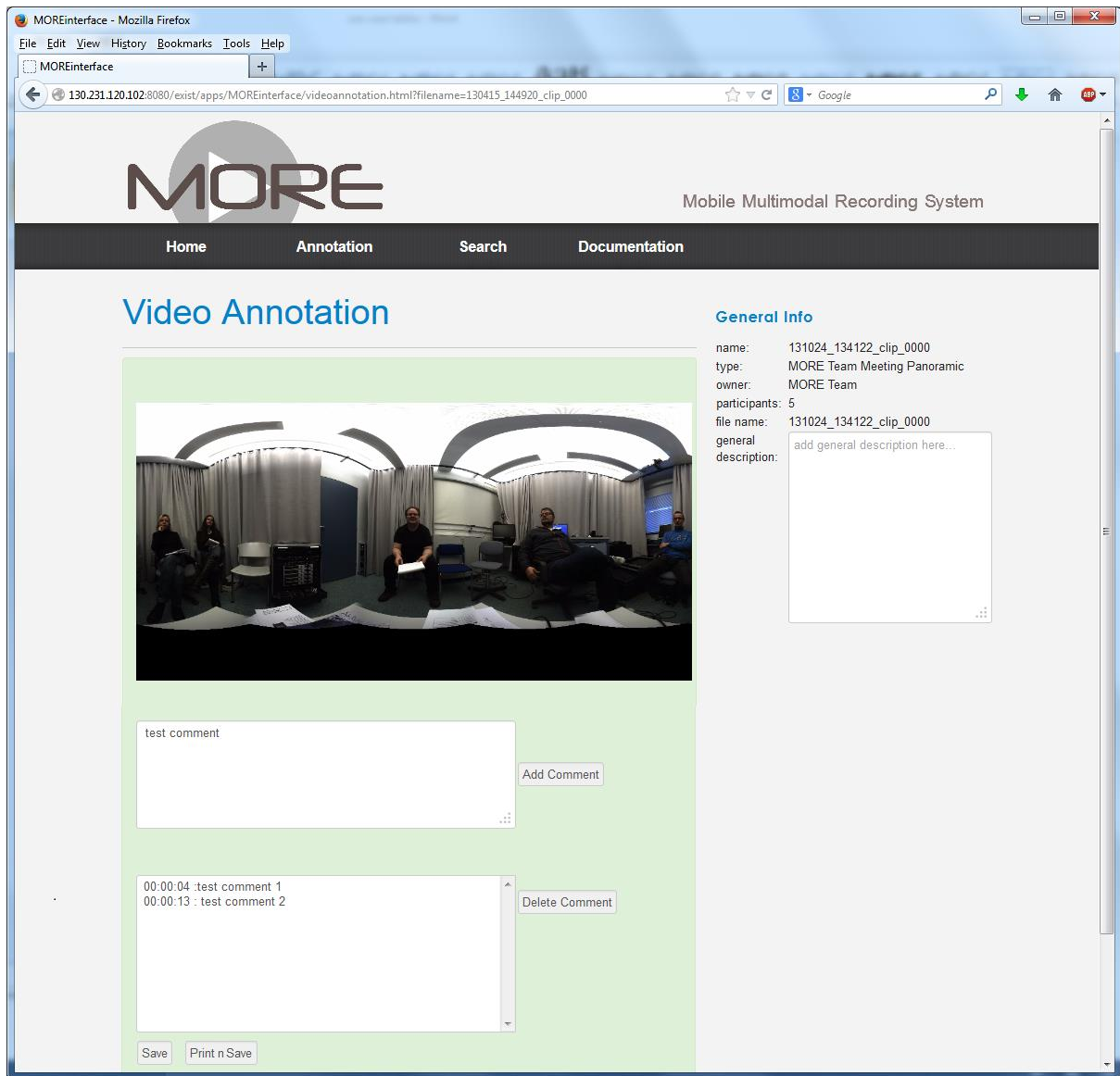
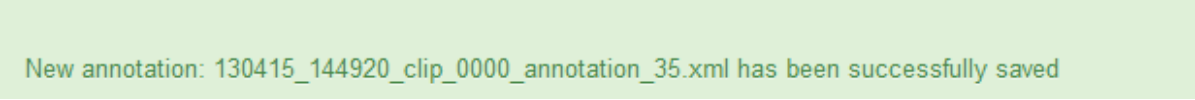


Figure 25. Video Annotation Module's main web page

The Video Annotation Module's main web page includes video player, general information of the video, such as name, type, owner of the video, number of participants and the file name. The general info part of the webpage includes general description field, due to this the user can add general comments to the video file. It also includes comment field for annotations, second comment field for the submitted annotations. The operation of the main webpage starts with playing the video. The user can add annotations with writing comments into the comment field. When the users press the Add comment button, the application replaces the comment to the second comment field with additional time stamp information. When the comment appears in the second comment field, it disappears from the first comment field, with this giving an opportunity for the user to Add Comments. The application has option for deleting comments

from the second comment field with the Delete Comment button. The webpage also includes Save and, Print and Save buttons. With the Save button the user can save the annotation into the database. The annotation are saved in XML file format. The saving function also generates a message about the result of the saving process. Figure 26 represents a sample message about the saving.



New annotation: 130415_144920_clip_0000_annotation_35.xml has been successfully saved

Figure 26. Message about the saving.

The application automatically generates a name for the annotation file, which includes the name of the video file and the number of the annotation. A message on the bottom of the page displays the successful saving. The Print and Save button has similar functionalities, but this function prints and saves the webpage as well.

The previously mentioned second way of searching for video is part of the property filter web page. The figure below draws out the Video Annotation Module's property filter web page.

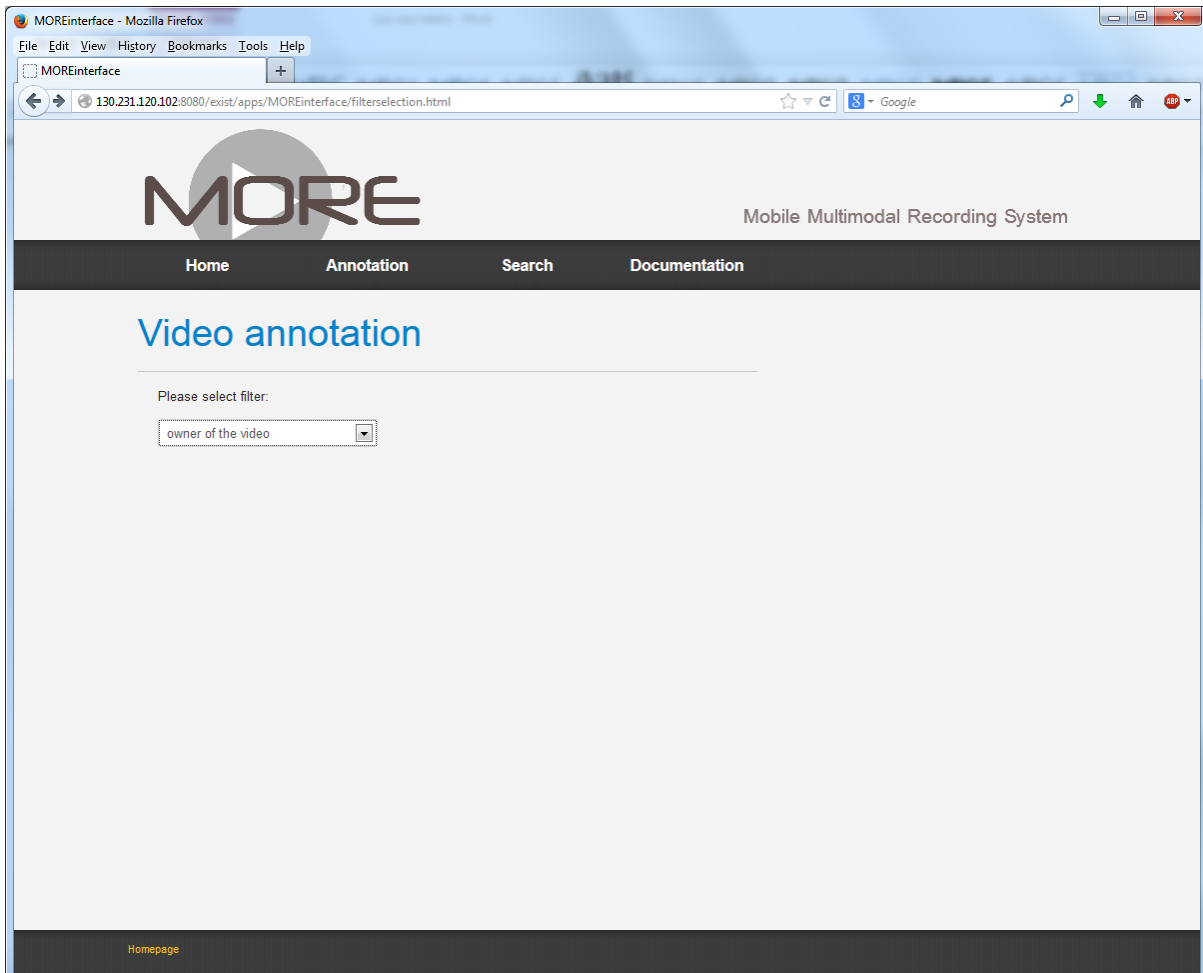


Figure 27. Video Annotation Module's property filter web page

The property filter web page can be found in the menu. Firstly, the user has to select the required property of the video from the drop-down list of the webpage. The properties can be name or type of the video among others. When the user presses the Ok button the application opens the search page. The figure below introduces the Video Annotation Module's property value selection web page.

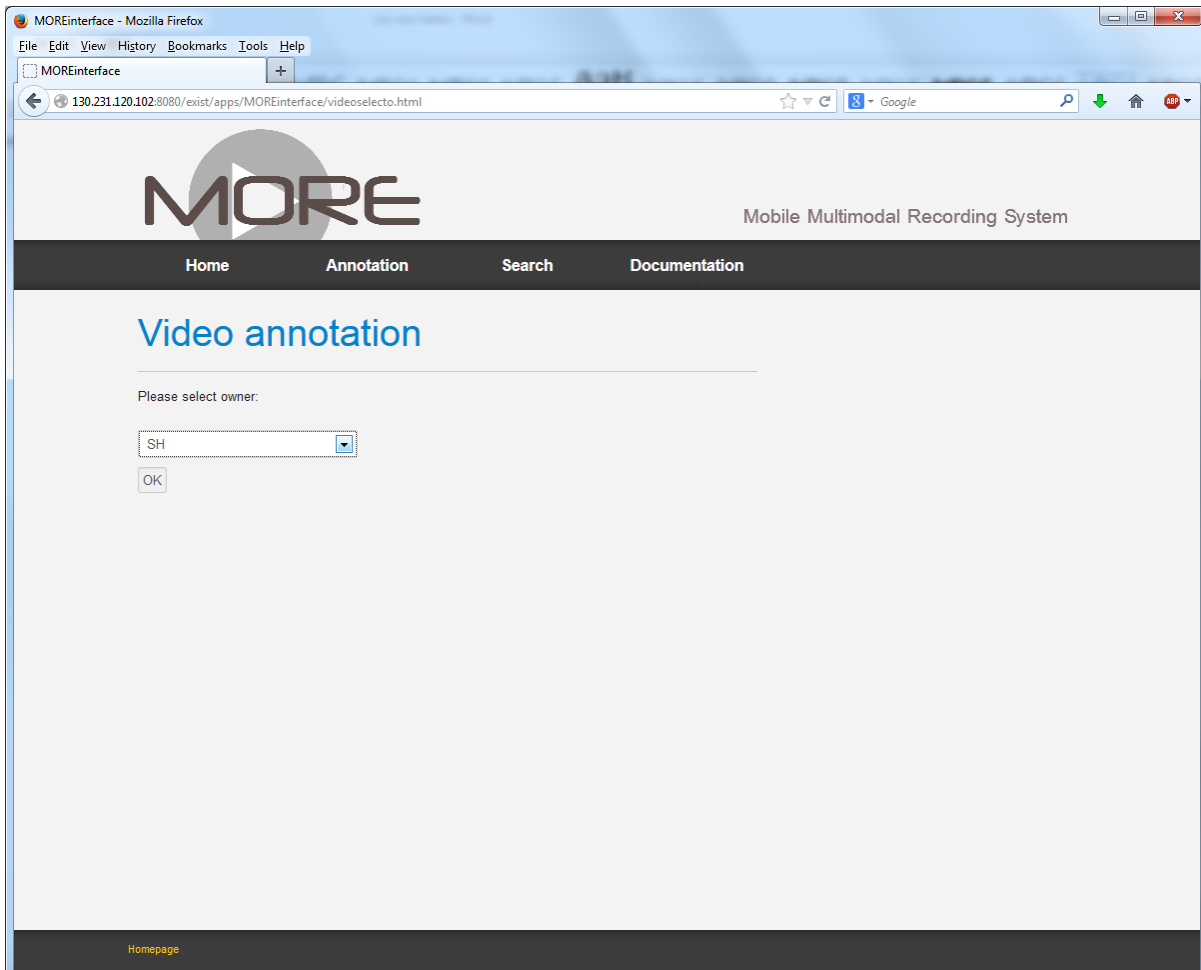


Figure 28. Video Annotation Module's property value selection web page

The search page includes the stored videos' selected parameters. When the user presses the OK button the application opens the video name selection page. Figure 29 below shows the Video Annotation Module's file name selection web page.

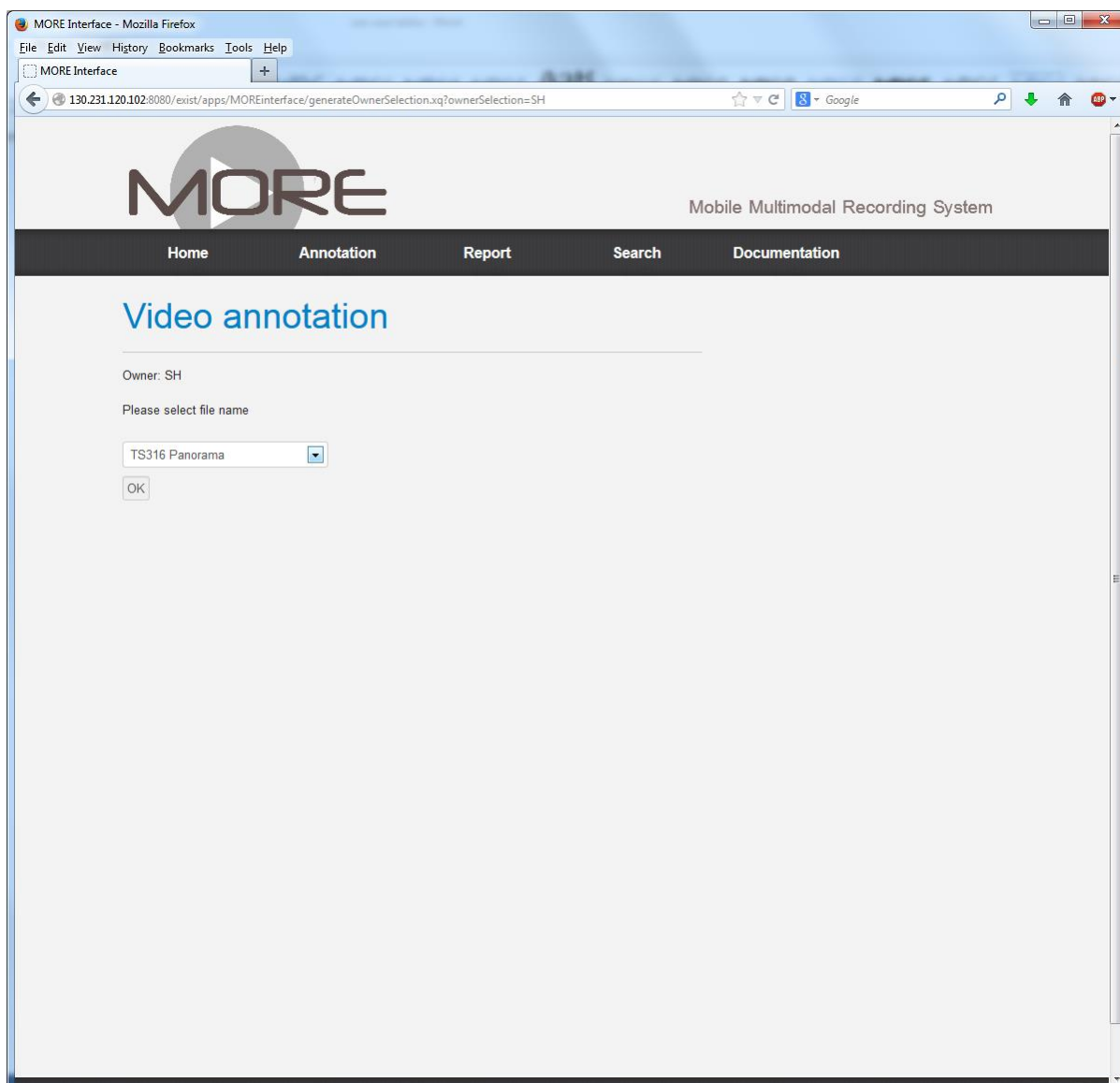


Figure 29. Video Annotation Module's file name selection web page

The user can select the name of the video from the drop-down list and press ok for open the Video Annotation Module's video file confirmation web page. By following the previously discussed method the application opens the Video Annotation Module's main web page.

7.1.3 User Interface of Audio Annotation Module

The Audio Annotation Module's operation is very similar to the Video Annotation Module's. Figure 30 below introduces the Audio Annotation Module's audio file selection web page.

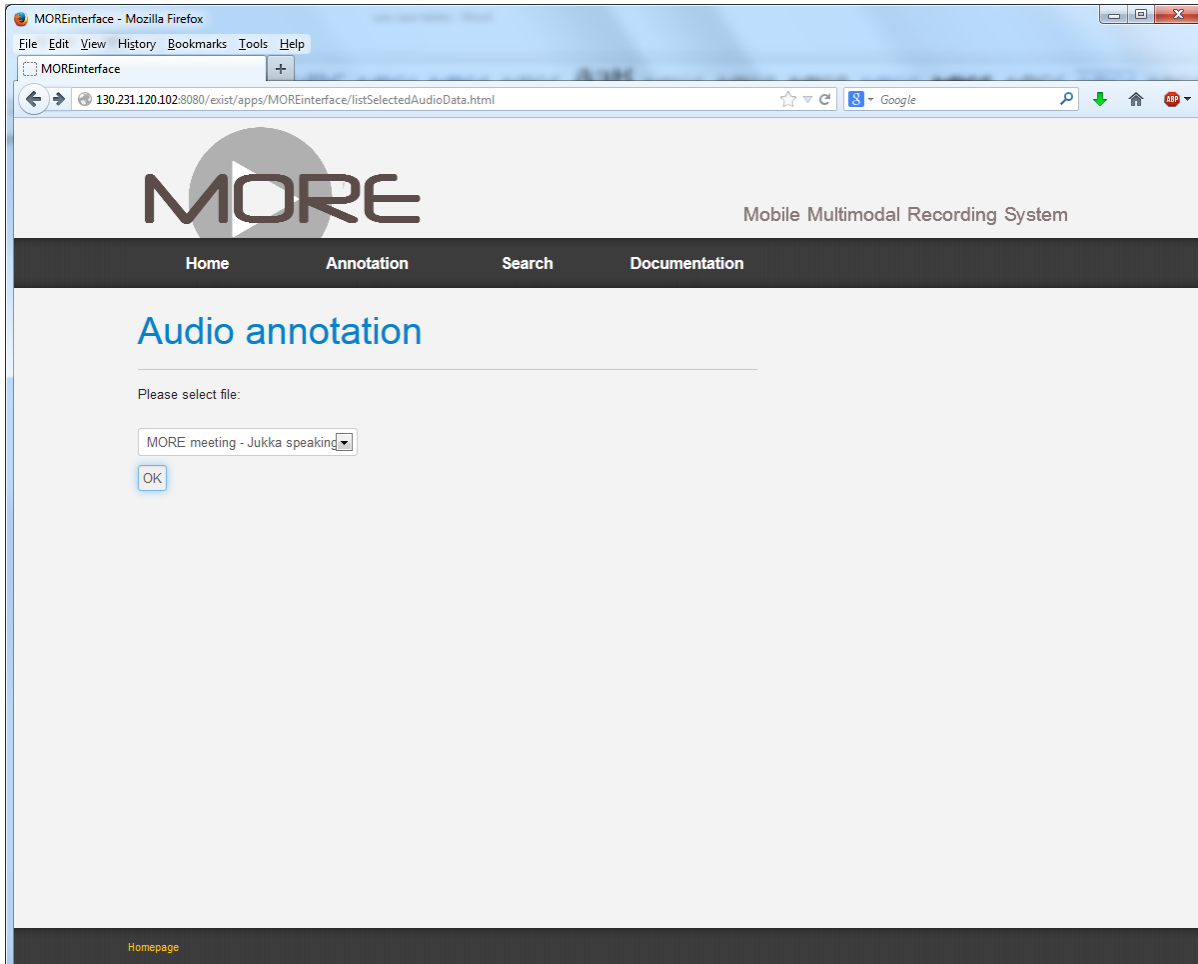


Figure 30. Audio Annotation Module's audio file selection web page

The Audio Annotation Module can be found in the menu. On the first place the user has to select the audio file by name. When the user presses the OK button, the application opens confirmation web page. The figure below introduces the Audio Annotation Module's selected audio file confirmation web page.

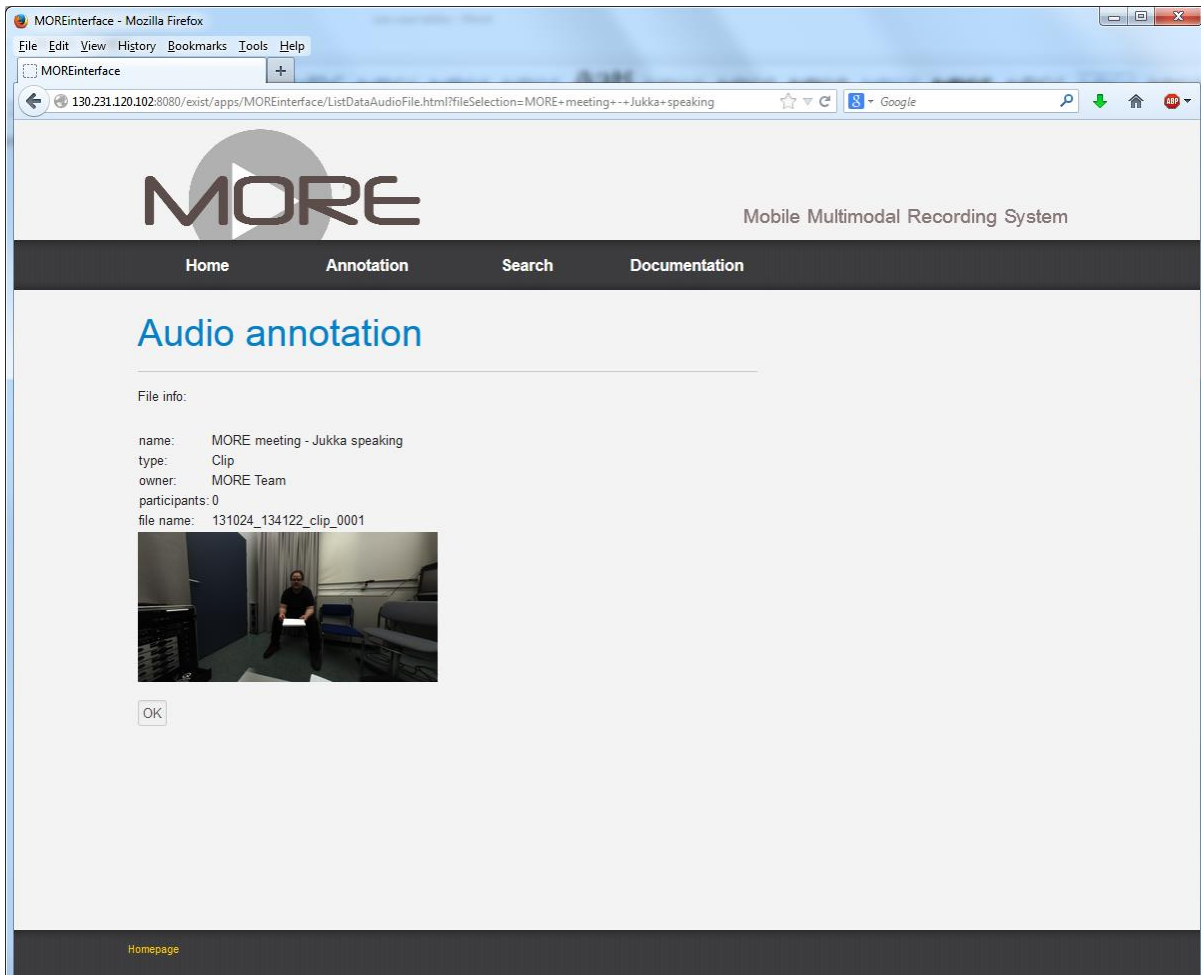


Figure 31. Audio Annotation Module's selected audio file confirmation web page

The selected audio file webpage includes general information of the video which includes the audio file. The general information includes name, type, owner of the video, number of participants and the file name. When the user press the OK button, the application opens the channel selection web page. The figure below depicts the Audio Annotation Module's channel selection web page.

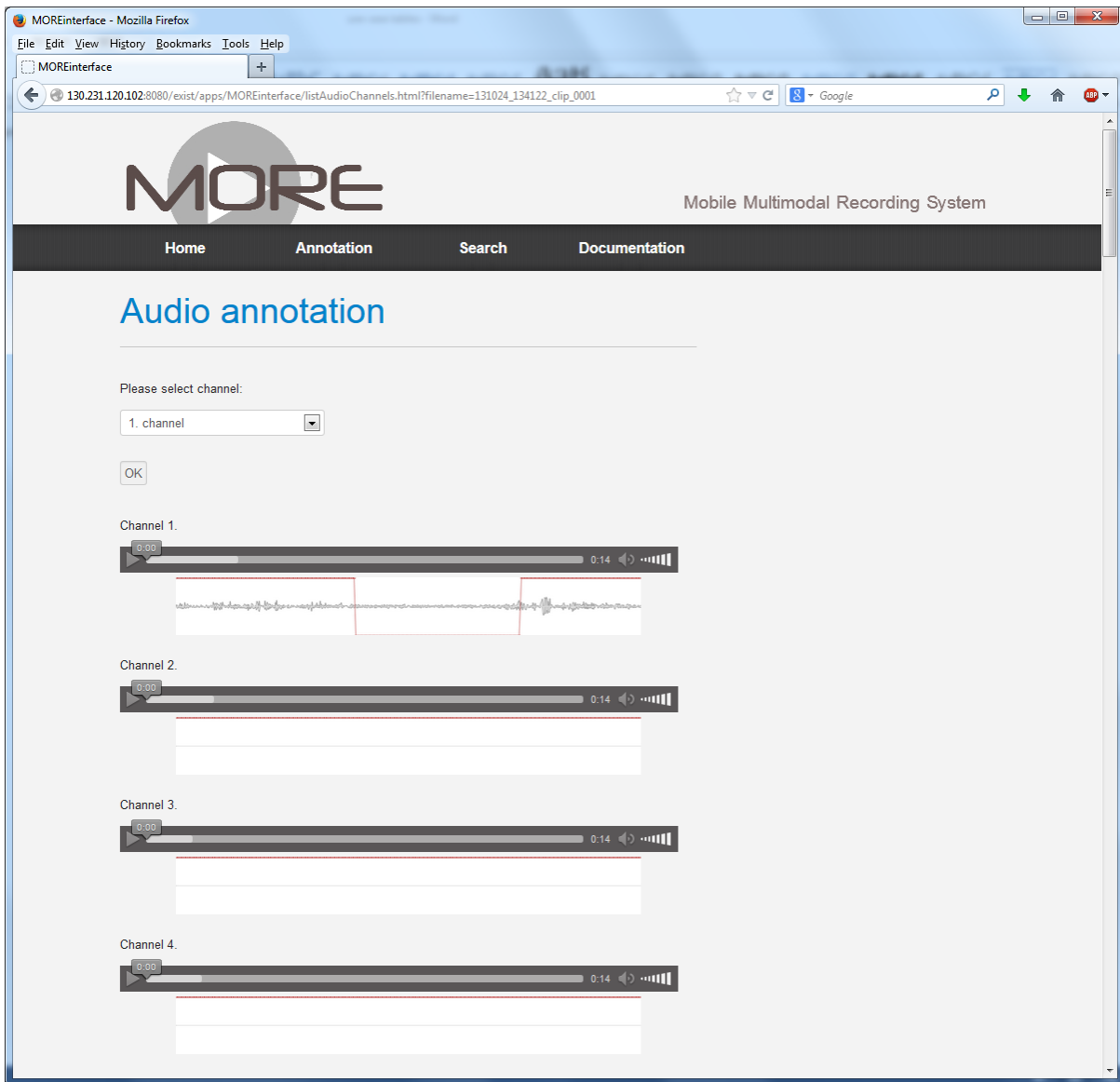


Figure 32. Audio Annotation Module's channel selection web page

The channel selection webpage includes channel selection drop-down box, ok button, image of the audio channels' signals and audio player. With the help of the image of the audio channels' signals and audio player the user can choose the required audio channel easily. When the user press the Ok button the application opens the Audio Annotation Module's main web page.

Audio Annotation Module's main web page is very similar in functionality to the Video Annotation Module's main web page. The Audio Annotation Module's main web page includes audio player, general information of the video which includes the audio channel. The general information contains the name, type, and owner of the video, number of participants, the file name and number of the channel. The webpage also illustrates the picture of the video.

The signal image of the audio channel and the audio player placed parallel to support the timing of the annotation. The general info part of the webpage includes general description field, due to this the user can add general comments to the audio channel. It also includes comment field for annotations, second comment field for the submitted annotations. Figure 33 below includes the Audio Annotation Module's main web page.

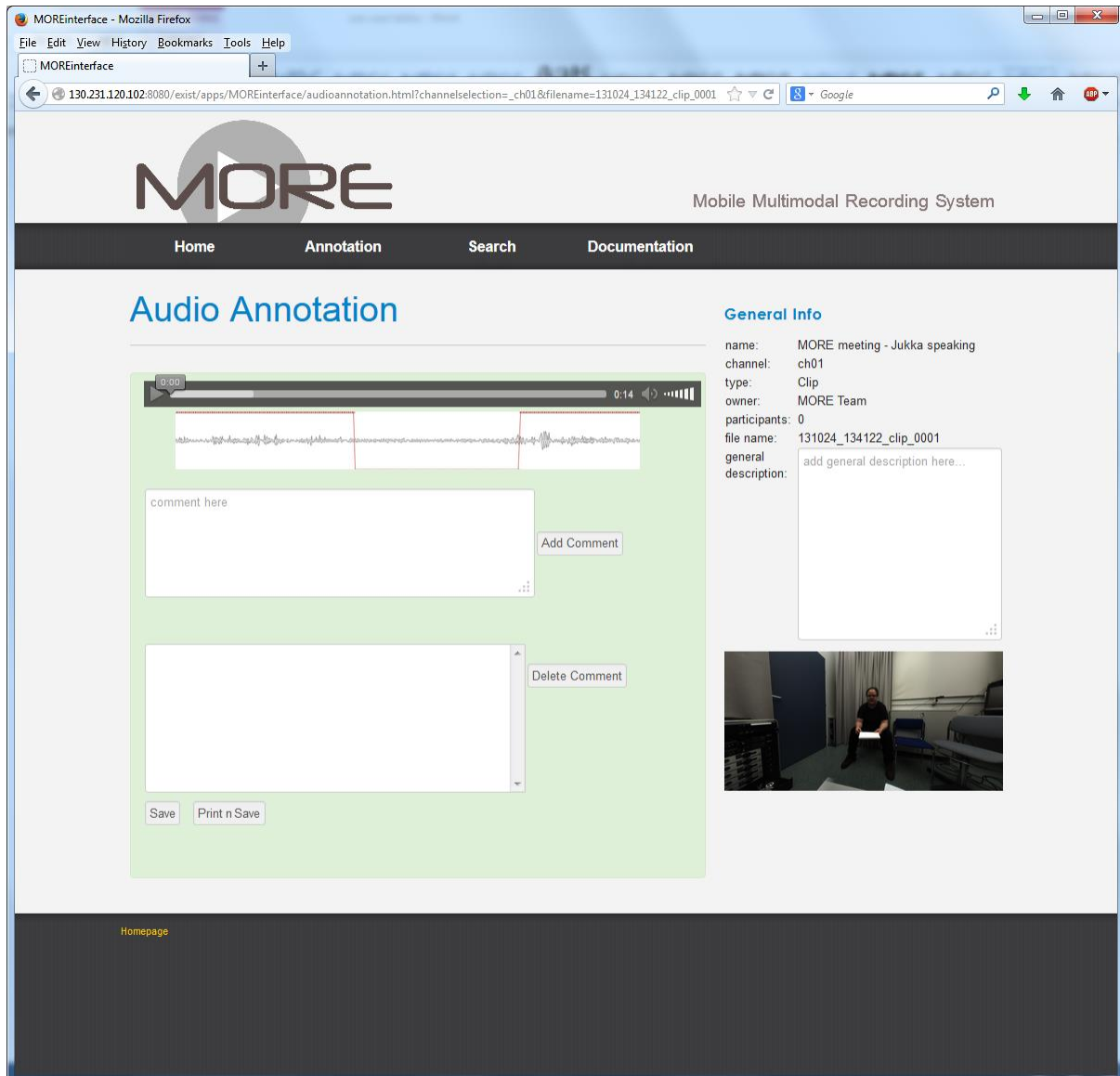


Figure 33. Audio Annotation Module's main web page

The operation of the main webpage starts with playing the audio channel. The user can add annotations with writing comments into the comment field. When the users press the Add Comment button, the application replaces the comment to the second comment field with

additional time stamp information. When the comment appears in the second comment field, it disappears from the first comment field, with this giving an opportunity for the user to add new comments. The application has option for deleting comments from the second comment field with the Delete Comment button. The webpage also includes Save and, Print and Save buttons. With the Save button the user can save the annotation into the database. The annotation is saved in XML file format.

7.1.4 User Interface of Audio Annotation Search Module

Since the Video Search Module's operation is very similar to the Audio Search Module's operation, this work includes only the description of Audio Annotation Search Module. Figure 34 below introduces the Audio Annotation Search Module's property value selection web page.

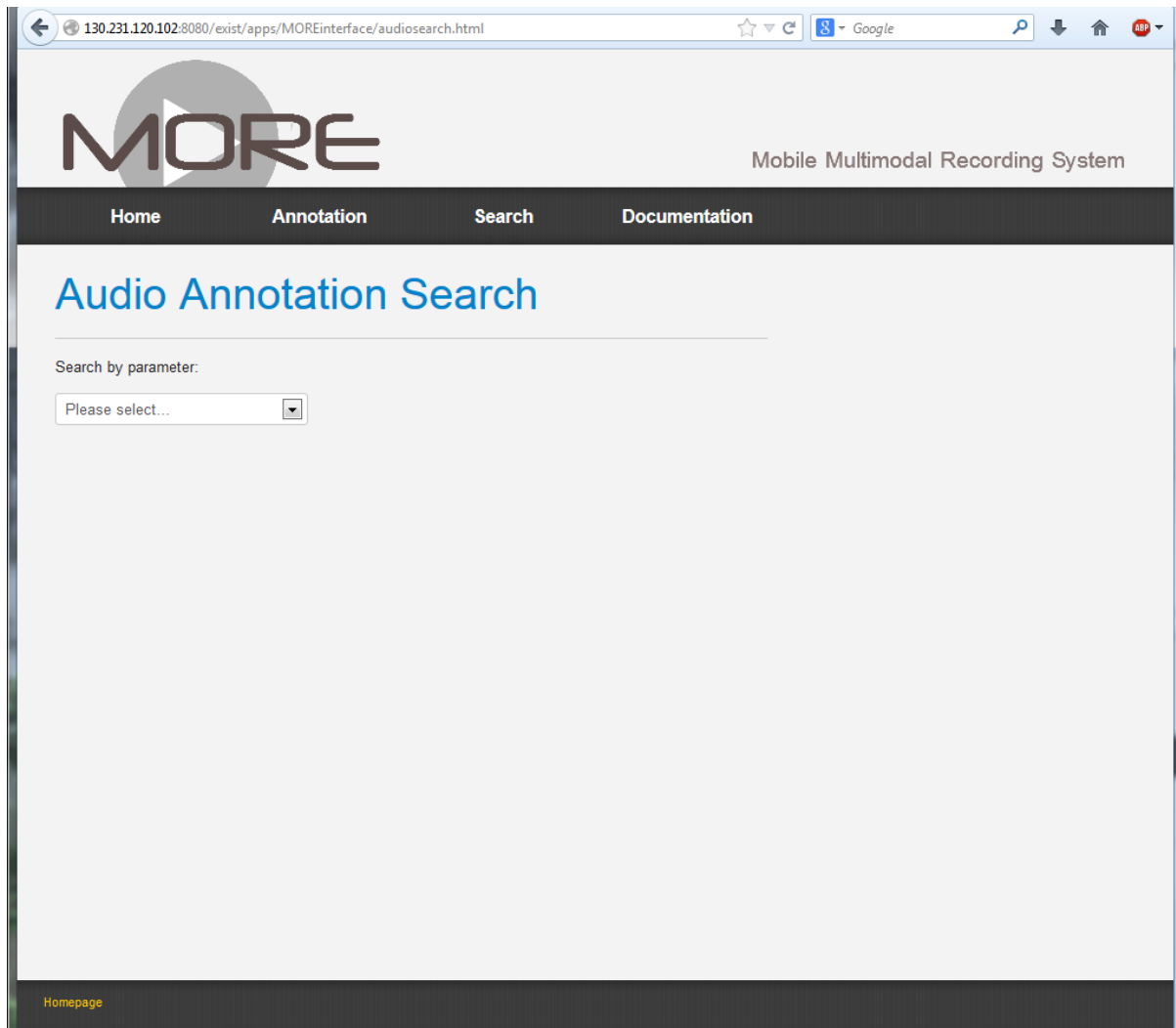


Figure 34. Audio Annotation Search Module's property value selection web page

The Audio Annotation Search function can be found in the menu. The Audio Annotation Search module includes several webpages. The operation starts with the selection of the parameter for the search. The property selection follows the search function. Figure 35 below introduces the Audio Annotation Search Module's property selection web page.

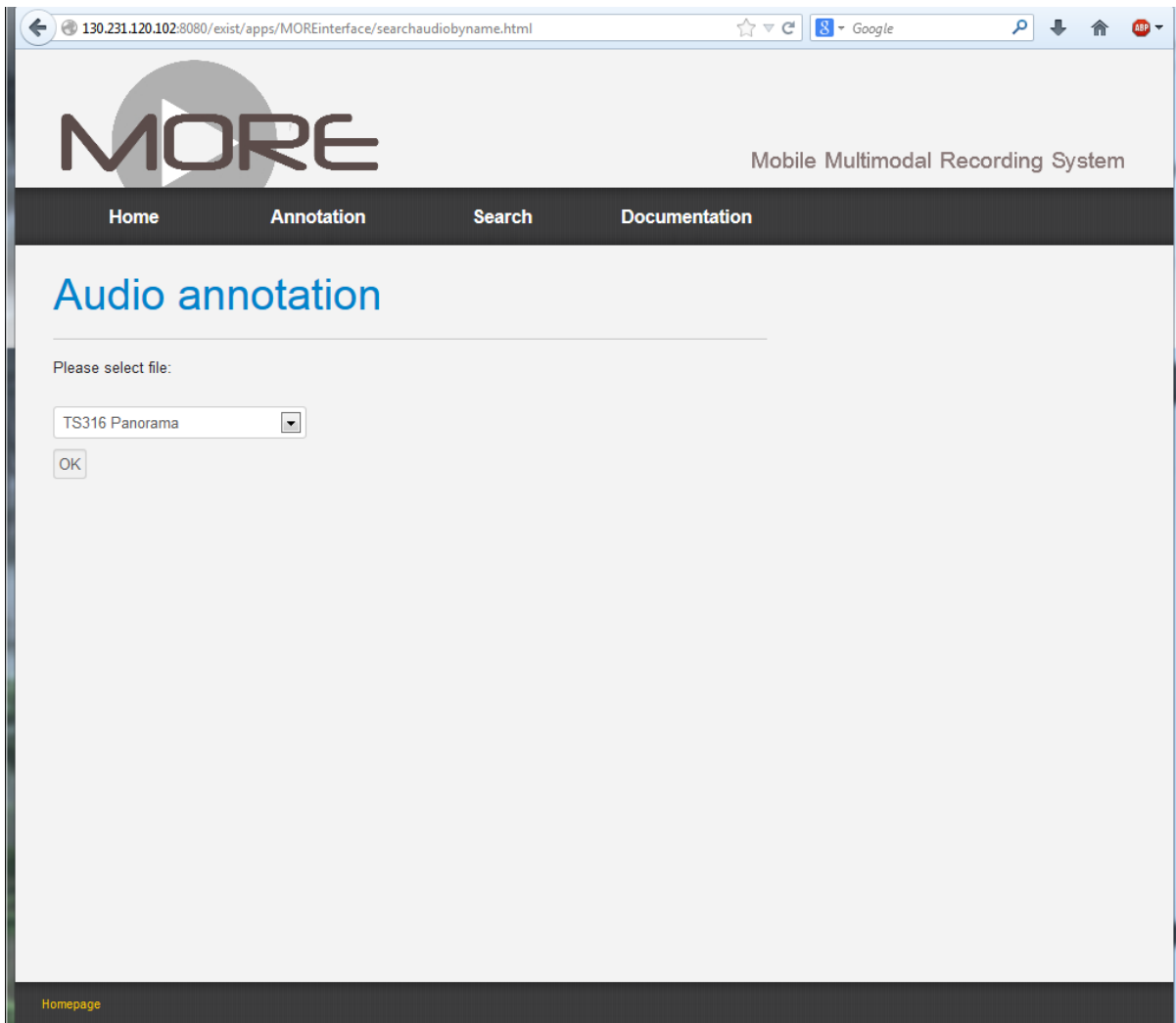


Figure 35. Audio Annotation Search Module's property selection web page

The property selection web page works on the same way as the Video Annotation Module's property value selection web page. After the user selected the property the application opens the confirmation page. Figure 36 below draws out Audio Annotation Search Module's confirmation web page.

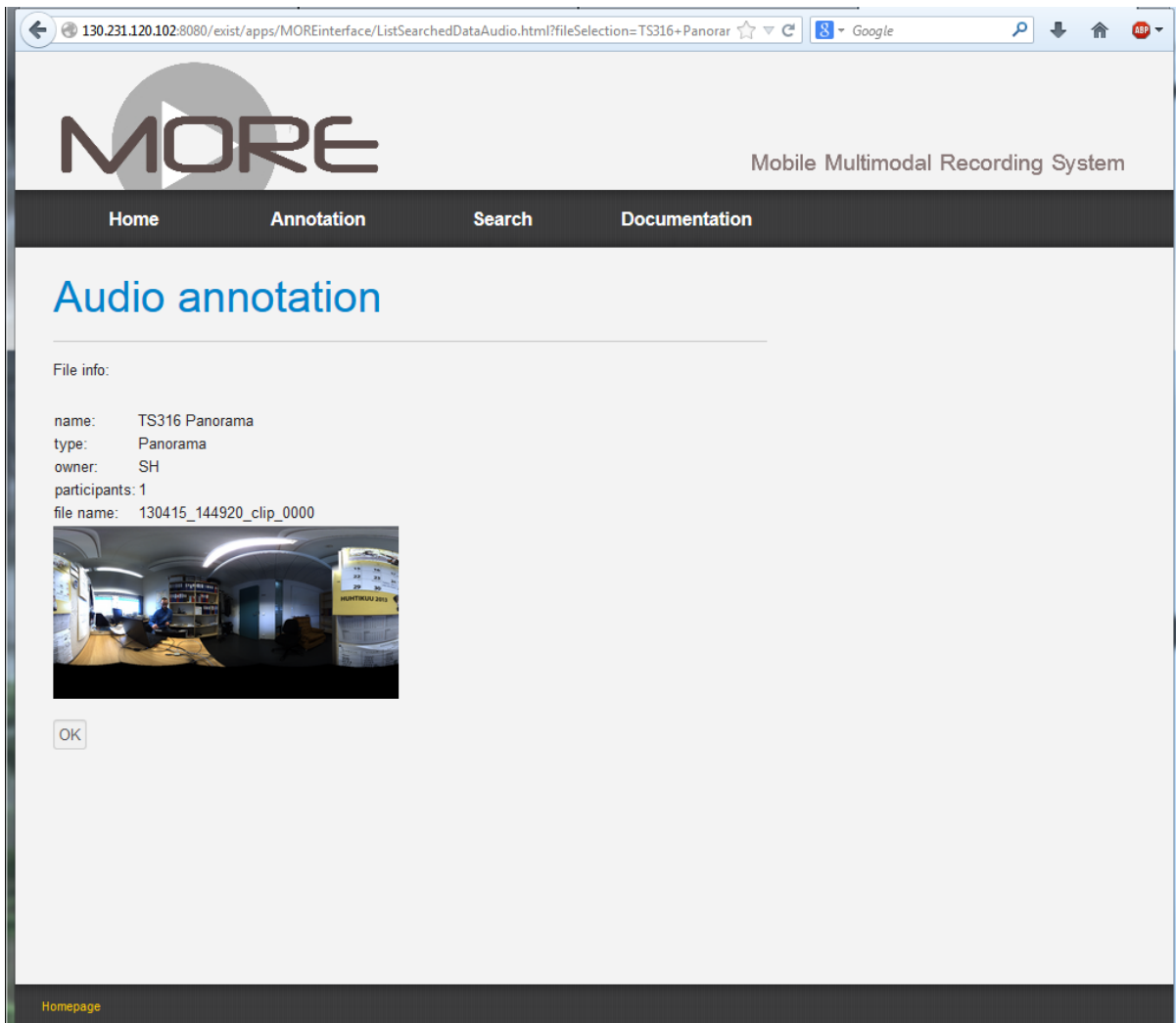


Figure 36. Audio Annotation Search Module's confirmation web page

The Audio Annotation Search Module's confirm web page operates on the same way as the previously discussed confirmation pages. After pressing the OK button, the user can get to the the confirmation page, the application opens the main annotation web page. The figure below represents the Audio Annotation Search Module's annotation selection page.

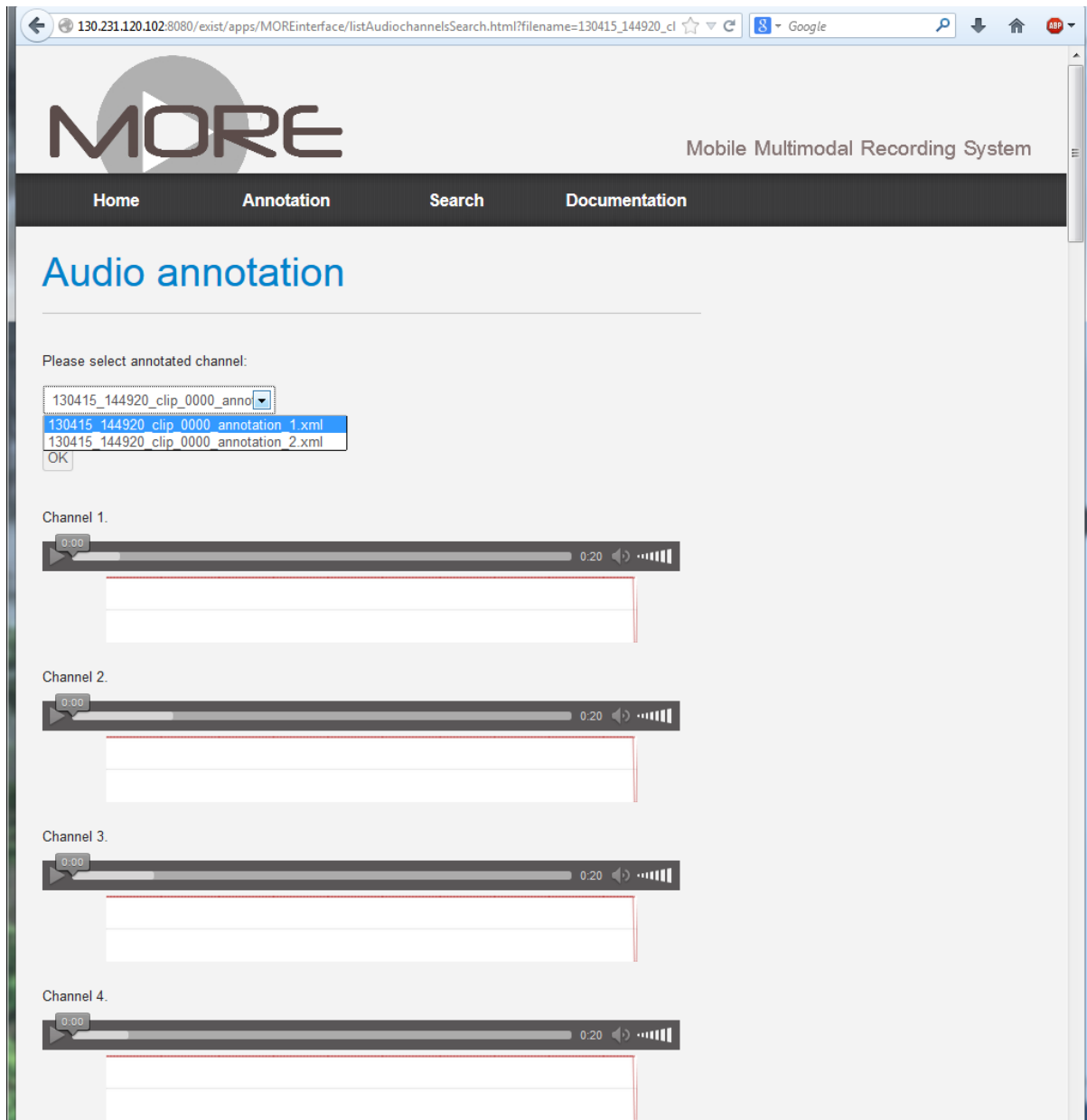


Figure 37. Audio Annotation Search Module's annotation selection page

Audio Annotation Search Module's annotation selection page allows the user to select the annotation regarding the previously selected file. After the user selected the annotation file the application presents the annotation web page. The figure below shows the Audio Annotation Search Module's main page.

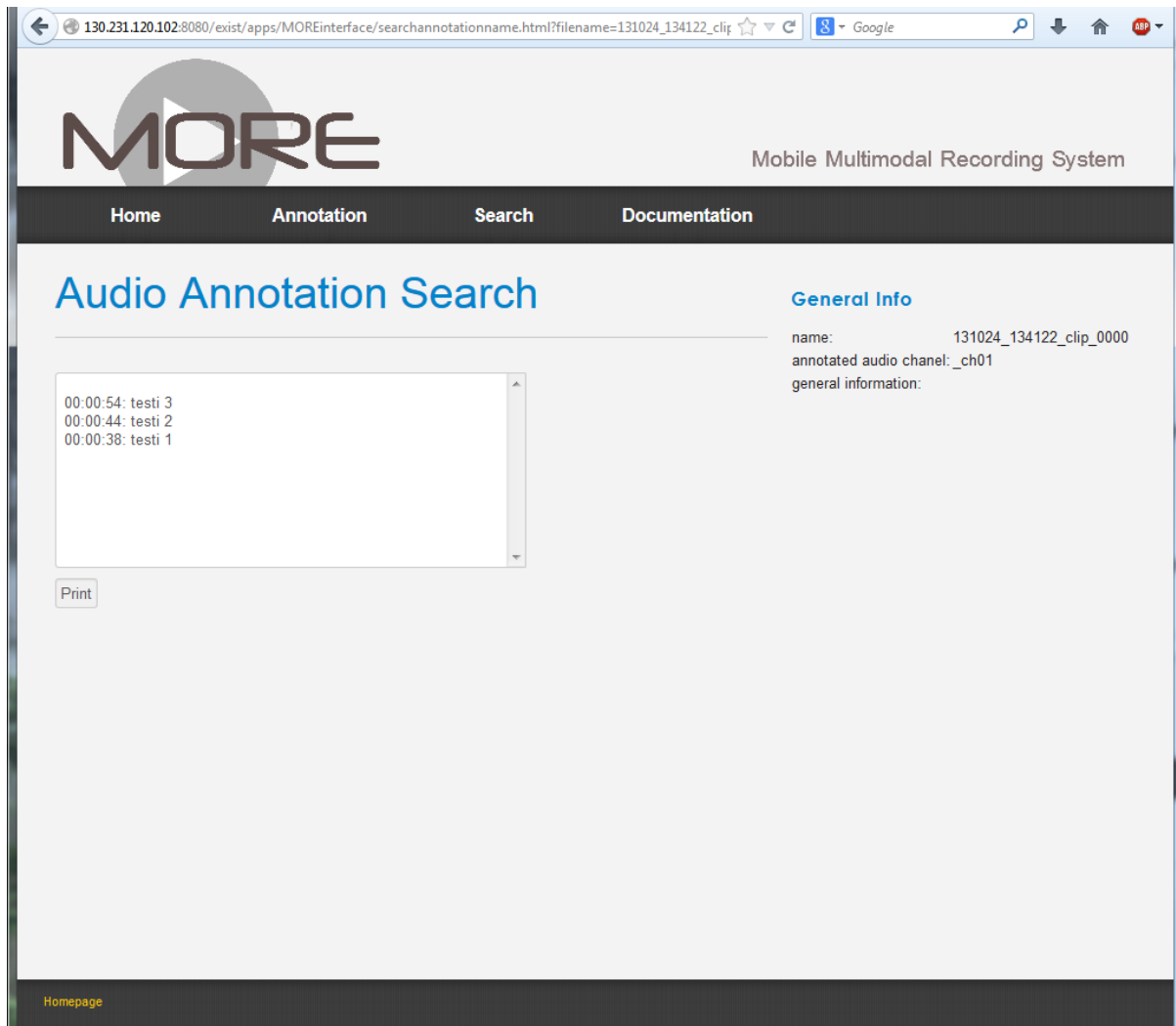


Figure 38. Audio Annotation Search Module's main page

The main page of the Annotation Search Module includes the annotations in a list box. The page also includes the general information of the file such as name of the file. The General Info contains the annotated channel's number and the general information in case the user added general description for the annotation.

8 CONCLUSIONS

Deriving from the objectives of this research, the expected outcome of the research was design and development of MORE web application for the Department of computer science and engineering, University of Oulu. The aim of the MORE web application sub project was to design and develop a web based environment, where the user can search, play, and add general comments to the video or audio material and create the report of the annotation. Firstly, the design specification of the MORE project was studied and extended with additional functionalities of the web application in order to identify all the major components of the MORE project and provide a solid foundation of the solution to develop. The technologies of the development were specified by the supervisor. Secondly, exploratory research for UML application design tool was carried out. Following the exploratory research, the suitable research methodology was selected for the development. Thirdly, the application design with UML diagram and descriptions formed an important part of the thesis work in order to ensure the logical structure of the web application functionalities. Fourthly, the application development was conducted in eXist database environment with XQuery, JavaScript, HTML and CSS technologies. Finally, the accomplished application development was represented by the description of the application's operation.

Due to the time constrains of the internship, only the essential functions of the MORE web application were designed and developed. However, there still is a wide range of opportunities regarding the additional functionalities and other aspects of development, such as security or higher level usability. The system can include advanced features in the future, such as facial recognition or activity level analysis. The opportunity of continuing internship at Oulu University's research group during the upcoming summer ensures the satisfactory quality of the MORE Web Based Application's design and development work. In addition, the current functionalities, the web application's interaction with the other components of the MORE system provide an opportunity for further development, such as emotional recognition annotation.

REFERENCES

- Ambler, Scott W. 2014. Agile Modeling. UML 2 Use Case Diagrams: An Agile Introduction. Downloaded 28 May, 2014.
<<http://www.agilemodeling.com/artifacts/useCaseDiagram.htm>>
- Bell, E. Alex 2004. Death by UML Fever. The Boeing Company 3.2004, Vol. 2, No. 1, 2-3.
- Chapman, Stephen 2013. What is JavaScript? Downloaded February 20, 2014.
<<http://javascript.about.com/od/reference/p/javascript.htm>>
- Coulouris, George & Dollimore, Jean & Kindberg, Tim & Blair, Gordon 2011. Distributed Systems: Concepts and Design .5th Ed. United States of America: Addison-Wesley Inc.
- Friedenthal, Sanford & Moore, Alan & Steiner, Rick 2012. A practical Guide to SysML. The Systems Modeling Language. 2nd Ed. Waltham: Elsevier Inc.
- Huttunen, Sami 2013. Meeting, MORE project developer, Department of computer science and engineering, University of Oulu. Held on 12 August, 2013.
- Keskinarkaus, Anja 2013. Meeting, MORE project leader, Department of computer science and engineering, University of Oulu. Held on 20 July, 2013.
- Keskinarkaus, Anja & Siipo, Antti & Huttunen, Sami & Holappa, Jukka & Juuso, Ilkka & Väyrynen, Eero & Laukka, Seppo. 2013. Requirement specification MORE. MORE documentation, 2013.
- Kyrnin Jennifer 2013. What are Markup Languages? Downloaded April 20, 2014.
<<http://webdesign.about.com/od/htmlxhtmltutorials/p/what-are-markup-languages.htm>>
- Kyrnin Jennifer 2013. What is CSS? Downloaded April 20, 2014.
<<http://webdesign.about.com/od/beginningcss/a/aa021607.htm>>
- Lukka, E. K. Kasanen & Siitonen, A. 1993. The constructive approach. Downloaded 20 May, 2014.
<<http://maaw.info/ArticleSummaries/ArtSumKasanenetal93.htm>>
- McDaniel, Carl Jr. & Gates Roger 2011. Marketing Research 9th Ed. United States of America: John Wiley & Sons Inc.
- Noldus.com 1989. Software and labs for behavioral research and video tracking. Noldus Information Technology Downloaded May 19, 2014
<<http://www.noldus.com>>
- Pender, Tom 2003. UML Bible, 1st Ed. United States of America: Wiley Publishing.
- Research Unit of Psychology. Mobile Multimodal Recording System. Downloaded 1.3.2014
<<http://www.oulu.fi/psych/node/24211>>

- Siipo, Antti & Huttunen, Sami & Keskinarkaus, Anja & Laszlo, Magda & Holappa, Jukka & Väyrynen, Eero & Juuso, Ilkka & Laukka, Seppo & Seppänen, Tapio, Lehtihalmes, Matti. 2013. Pedapäivät slides, public presentation of the MORE project, 7.11.2013, Pedapäivät, Oulu University.
- Siipo, Antti & Huttunen Sami & Jukka, Holappa & Väyrynen, Eero & Keskinarkaus, Anja & László, Magda & Juuso, Ilkka & Laukka, Seppo & Seppänen, Tapio Forthcoming in 2014. MORE – a Multimodal Observation and Analysis System for Social Interactions.
- Stevens, Reed & Macklin, Scott 2001. Representing, exchanging, and assessing ideas in an almost natural way: The Traces digital media annotation systems. University of Washington. Downloaded May 20, 2014.
<http://depts.washington.edu/pettt/presentations/edmedia_03.pdf>
- Teachscape.com 2014. Teachscape. Teachscape, Inc. Downloaded 20 May, 2014.
<<http://www.teachscape.com>>
- Transana.org 2005. Transana. Qualitative analysis of software for video and audio data. Downloaded 1.5.2014
<<http://www.transana.org>>
- Oracle.com 2014. Distributed Application Architecture. Sun Microsystem. Downloaded 20 April, 2014.
<<http://www.oracle.com/technetwork/java/index.html>>
- Walmsley, Priscilla 2007. XQuery. Search across XML Data. 1st Ed. United States of America: O'Reilly Media, Inc.
- W3.org 2003. Extensible Markup Language (XML). Downloaded 12.4.2014
<<http://www.w3.org/XML/>>

VIDEO ANNOTATION MODULE'S MAIN PAGE HTML FILE

```

1 <html xmlns="http://www.w3.org/1999/xhtml">
2 <head>
3 <title class="config:app-title">MORE Interface</title>
4 <meta class="config:app-meta"/>
5 <link rel="shortcut icon" href="resources/images/favicon.ico"/>
6 <link rel="stylesheet" type="text/css" href="resources/css/exist.css"/>
7 <script type="text/javascript" src="$shared/resources/scripts/jquery/jquery-1.7.1.min.js"/>
8 <script type="text/javascript" src="$shared/resources/scripts/loadsource.js"/>
9 <script src="http://popcorn.js.org/code/dist/popcorn-complete.js"/>
10 <script src="videoannotation.js"/>
11 </script>
12 document.addEventListener( "DOMContentLoaded", function() { var popcorn = Popcorn( "#ourvideo" ); popcorn.footnote({
13     start: 2,
14     end: 5,
15     target: "footnote2",
16     text: "Pop!"
17
18 });pop.play();}, false );
19
20 var $pop = Popcorn("#ourvideo");
21 var textBox = document.getElementById("srt");
22 $pop.listen("timeupdate", function(){textBox.value = Math.round(this.currentTime());});
23 $pop.play();
24
25 function run() {
26     document.getElementById("ourvideo").src = document.getElementById("box").value;
27 }
28
29 function setcheckbox() {
30     document.getElementById("checkCh1").checked = true;
31 }
32
33 function audioComment() {
34     var textbox = document.getElementById("srt1");
35     var listbox = document.getElementById('listbox');
36     var newOption = document.createElement('option');
37     var audiotest= document.getElementById("ouraudio");
38
39
40     textbox.value = audiotest.currentTime()+ " " + textbox.value;
41     newOption.innerHTML = textbox.value;
42     listbox.appendChild(newOption);
43     textbox.value=" ";
44 }
45
46 function removeItem(selectbox)
47 {
48     var i;
49     for(i=selectbox.options.length-1;i>=0;i--)
50     {
51         if(selectbox.options[i].selected)
52             selectbox.remove(i);
53     }
54 }
55
56 function gettVideo()
57 {
58     var textbox = document.getElementById("srt1");
59     var selectw = document.getElementById("fileSelection");
60     var folder = selectw.options[selectw.selectedIndex].text;
61     var splittedfolder = folder.slice(0,-10);
62     var textt = "./MORE/" + splittedfolder + "/Clips/" + selectw.options[selectw.selectedIndex].text + "/" + selectw.o
63     ".mp4";
64     textbox.value = textt ;
65 }
66
67 function addDateNComment()
68 {
69     var $pop = Popcorn("#ourvideo");
70     var textbox = document.getElementById("srt");
71     var listbox = document.getElementById('listbox');
72     var hiddenlistitems = document.getElementById("hiddenlistitems");
73     var newOption = document.createElement('option');
74     var string = (new XDate().clearTime().addSeconds($pop.currentTime()).toString('HH:mm:ss') + " ." + textbox.value
75     hiddenlistitems.value += string + ";";
76
77
78     textbox.value = string;
79
80     newOption.innerHTML = textbox.value;
81     listbox.appendChild(newOption);
82     textbox.value=" ";
83 }

```

```

94
95 </script>
96 </head>
97 <body id="body" onload="clearhiddentxt()">
98 <div class="templates:surround?with=templates/page.html&amp;at=content">
99 <div class="row">
100 <div class="span8" id="content">
101 <h1>Video Annotation</h1>
102 <div class="row-fluid">
103 <div class="alert alert-success">
104 <div class="app:loadVideo"/>
105 <table>
106 <tr>
107 <td>
108 <textarea rows="5" id="srt" placeholder="comment here..." style="width: 400px;"/>
109 </td>
110 <td>
111 <input type="submit" value="Add Comment" id="commentbutton" onclick="addDateNComment()"/>
112 </td>
113 </tr>
114 </table>
115 <p/>
116 <table>
117 <tr>
118 <td>
119 <form>
120 <select id="listbox" name="listbox" size="10" style="width: 410px;"/>
121 <p/>
122 <p/>
123 </select>
124 <div class="app:savefilename"/>
125 <input type="hidden" id="readytosave" name="readytosave"/>
126 <input type="submit" value="Save Report" id="createbutton" onclick=""/>
127 </form>
128 </td>
129 <td>
130 <input type="button" value="Delete Comment" id="commentbutton" onclick="removeItem(listbo
131 <p/>
132 <p/>
133 <input type="submit" value="Add to general info" id="commentbutton" onclick=""/>
134 <p/>
135 <p/>
136 <input type="submit" value="Print" id="createbutton" onclick="window.print()"/>
137 </td>
138 </tr>
139 </table>
140 <div id="footnote2"/>
141 <div class="app:getdataa"/>
142 </div>
143 </div>
144 <div class="span4">
145 <h2>General Info</h2>
146 <table class="app:showFolder"/>
147 </div>
148 </div>
149 </div>
150 </body>
151 </html>

```