

Simple and Secure Video Sharing

Petter Haglund and Kristoffer Mårtensson

DEPARTMENT OF DESIGN SCIENCES
FACULTY OF ENGINEERING LTH | LUND UNIVERSITY
2019

MASTER'S THESIS



Simple and Secure Video Sharing

Petter Haglund and Kristoffer Mårtensson

June 18, 2019



LUND
UNIVERSITY

Simple and Secure Video Sharing

Copyright © 2019 Petter Haglund & Kristoffer Mårtensson

Published by

Department of Design Sciences
Faculty of Engineering LTH, Lund University
P.O. Box 118, SE-221 00 Lund, Sweden

Subject: Interaction Design (MAMM01)
Division: Ergonomics and Aerosol Technology
Supervisor: Joakim Eriksson (LTH) and Jon Lindeheim (Axis Communications)
Examiner: Günter Alce

Abstract

Hard disk drives, USB flash drives and local storage servers are all storage solutions of the past. Cloud services are completely taking over and the major companies are adjusting to the new standard. Axis Communications provides complete surveillance systems and are currently transitioning towards incorporating cloud services in their software products.

This master's thesis aimed to research the potential for using cloud services to share video recordings, as well as developing a prototype for this purpose. Verification tests were performed on the prototype to assess its usability. Cloud-based video sharing has during this thesis proven to be a highly viable contender to current solutions and may very well be the new standard in the future.

Keywords: Video Sharing, Cloud Services, CCTV, Surveillance, Security Systems

Sammanfattning

Hårddiskar, USB-minnen och lokala lagringsservrar är alla exempel på gårdagens lagringslösningar. Molntjänster håller på att ta över fullständigt och majoriteten av väletablerade företag anpassar sig till den nya standarden. Axis Communications tillhandahåller fullständiga lösningar för övervakningssystem och håller just nu på att inkorporera molntjänster i sina produkter och system.

Detta examensarbete ämnade undersöka potentialen för att använda molntjänster i syfte att dela inspelat videomaterial, samt utveckla en prototyp som utför detta. Tester genomfördes på prototypen för att utvärdera dess användbarhet. Molnbaserad videodelning har under detta examensarbete visat sig vara en stark utmanare till dagens lösningar och kan mycket väl visa sig vara den nya standarden i framtiden.

Nyckelord: Videodeling, Molntjänster, CCTV, Övervakning

Acknowledgements

This master's thesis was conducted during the spring of 2019 in collaboration with Axis Communications and the Department of Design Sciences, Faculty of Engineering LTH at Lund University.

First off we would like to thank Jon Lindeheim and Marcus Wiedner at Axis for making this thesis possible. We would also like to thank Joakim Eriksson, our supervisor at LTH, for his guidance and ideas that helped us carry out our work. Lastly, we would like to dedicate a big thank you to the Small Solutions team for their support and honest feedback, and for making us feel at home during our time at Axis.

Contents

1	Introduction	11
1.1	Purpose	11
1.2	Scope	12
1.3	Axis Communications	12
1.3.1	History	12
1.3.2	Axis Companion	13
1.3.3	Cameras and Recorders	13
1.4	Terminology	13
2	Theoretical Background	15
2.1	Design and Usability	15
2.1.1	User-Centered Design	15
2.1.2	Donald Norman's Design Principles	15
2.1.3	Usability Evaluation	16
2.2	Video Sharing	18
2.2.1	Streaming Media	18
2.2.2	Security and Video Integrity	18
2.3	Software Development	19
2.3.1	Agile Development	19
2.3.2	Analytics	19
3	Technical Background	21
3.1	Cloud Services	21
3.1.1	Amazon Web Services	21
3.1.2	Netlify	22
3.2	Development and Tools	22
3.2.1	Microsoft Visual Studio	22
3.2.2	.Net and C#	22
3.2.3	React	22
3.2.4	Git and Version Control	22

3.2.5	Code Review	23
4	Methodology	25
4.1	Investigation and Ideation Phase	25
4.2	Design and Development Phase	26
4.3	Verification and Evaluation Phase	26
4.3.1	Cognitive Walkthroughs	26
4.3.2	Usability Test	26
5	Investigation and Ideation	27
5.1	Saving Recordings in the Axis Companion	27
5.1.1	Functionality	27
5.1.2	Analysis	29
5.2	Export Feature Usability Test	30
5.2.1	Overview	30
5.2.2	Results and Analysis	30
5.2.3	Discussion	31
5.3	Competitor Analysis	32
5.4	Video Players and User Behaviour	33
6	Design and Development	35
6.1	Lo-fi Prototyping and Conceptual Model	35
6.1.1	Lo-fi Dialog Design	35
6.1.2	Lo-fi Website Design	35
6.1.3	Conceptual Software Ecosystem	37
6.2	Hi-fi Prototyping	38
6.2.1	Spike	38
6.2.2	Further Development	40
6.2.3	Design Improvements	41
7	Verification and Evaluation	45
7.1	Cognitive Walkthroughs	45
7.1.1	Participants	45
7.1.2	Test Execution	45
7.1.3	Test Result	47
7.2	Usability Testing	48
7.2.1	Participants	48
7.2.2	Test Execution	48
7.2.3	Questionnaires	49
7.2.4	Test Result	49
8	Discussion	53
8.1	Legal Aspects	53
8.2	Security Related Aspects	54
8.3	Design Suggestions	54
8.3.1	Export Flow	54

8.3.2	Export Dialog	55
8.3.3	Website Design	57
8.4	Limitations	57
8.4.1	Prototype Limitations	57
8.4.2	Other Limitations	58
9	Conclusion	59
	Bibliography	61
	Appendix A Usability test of the export flow (local export only)	67
A.1	Test Scenario Manuscript	68
A.2	SUS-questionnaire	69
A.3	Interview Questions	71
	Appendix B Verification test: export and share	73
B.1	Test Tasks	74
B.2	Interview Questions	75

Abbreviations

AWS Amazon Web Services.

CCTV Closed-circuit television.

GUID Globally Unique Identifier.

SUS System Usability Scale.

UCD User-Centered Design.

UI User Interface.

VMS Video Management System.

Chapter 1

Introduction

Axis Companion is a video management system (VMS) developed and maintained by Axis Communications. Software such as Companion is used to setup, manage and monitor surveillance sites [1]. A site consist of one or multiple cameras and a recorder. This kind of software application normally features a certain set of core functionalities, one of them being a method for exporting recorded material to a storage medium other than the recorder itself.

Whenever an event has taken place that needs to be viewed by someone without access to Companion, such as the police department, involved legal personnel or even the courthouse, the recorded video evidence needs to be shared somehow. This is usually accomplished by performing a local export of the recorded video material in Companion, meaning that the video files are transferred from the recorder and saved locally on the computer connected to the site. The resulting exported video files are most often too large to share online in any practical way, meaning they would then have to be transferred to a flash drive and physically handed over to the intended recipient.

The current process of sharing video material is very time consuming, not only for the user looking to share the material but also the intended recipient, such as the police. They often have to transport themselves to wherever the recordings are located and retrieve the files in person. The team at Axis that developed the Companion application have acknowledged the issues that exist in the current solution. They are interested in finding an alternate solution to this process that is simple to use, while also being secure and less time consuming.

1.1 Purpose

The purpose of this thesis was to develop a functional prototype of a cloud-based video sharing solution that allows the recipient to view the exported material in an ordinary web browser instead of a specific media player.

The prototype will be used to evaluate the potential of this kind of solution and present the advantages over the current solution. Legal and security aspects will also be researched and discussed in order to facilitate whether this solution is durable enough to be an alternative to the current solution in all possible use cases.

In order for the cloud-based solution to be an even further improvement, the solution needs to feel intuitive and easy to use. To achieve this, the usability of the current export feature will be examined and evaluated so that it can be improved, in addition to the implementation of the new features required for cloud sharing. The prototype will be developed using well known design principles and tested in order to make it as user-friendly as possible.

Would the cloud-based video sharing solution be an improvement over the current video sharing process? Could it even serve as a substitute down the line?

1.2 Scope

Development of the prototype proposed by this thesis project was composed of two distinct parts:

- Implementation of a cloud-based online sharing feature in the Companion application.
- Development of a media player website where the recipient can view the shared video material.

After completion, the prototype was tested in order to evaluate usability.

To further improve the user experience of the export and share functionality we examined and evaluated the pre-existing functionality for exporting recordings from the Companion.

Legal and security related aspects of the solution are all highly important for a fully fledged product. These were also taken into consideration, although mostly researched and discussed for the purposes of further development rather than fully taken into account in the development of our prototype.

1.3 Axis Communications

A brief description of Axis Communications and the Axis products used in the project.

1.3.1 History

Axis started in 1984 as a company that developed servers for network based printers. In 1996 the very first network camera was developed at Axis. With this invention, Axis introduced themselves to the CCTV-surveillance business which at that time had not seen this type of camera. After about four years, the company's focus had completely shifted towards network cameras and the development of partner programs, which were programs that could be used to install and control the cameras. From the year 2000 until today, Axis

has been a major player in the global CCTV-market with over 3000 employees globally and a total revenue that in 2019 was calculated to over 10 billion SEK [2].

1.3.2 Axis Companion

As mentioned in the introduction, Axis Companion is a video management system. A VMS provides the user with all the necessary functionality to create and control the surveillance of their facilities. They are developed for both mobile devices and desktop computers, Companion being one of the computer versions of Axis' software. Companion is targeted towards small to medium sized businesses with 1-16 cameras per site. The application is written with the .Net framework and can be installed on computers running Windows.

1.3.3 Cameras and Recorders

Axis also develops the hardware necessary for surveillance. The camera models used in this thesis project are the M3045-v mini dome camera, a Bullet LE camera and an Eye mini L camera. The recorder is a 4 channel network video recorder, connected to the cameras through Ethernet.

1.4 Terminology

A brief explanation of some recurring terms used throughout the thesis.

Export flow: Refers to the user interaction required from start to finish when saving recordings in the Companion.

Export mode: Refers to the added interactions that are available when the user has pressed the "save recordings" button in the recordings view of Companion.

Export job: Refers to an export made in the Companion with a specific timeline and chosen cameras.

Chapter 2

Theoretical Background

Literature and theories relevant to the thesis project will be presented in this chapter.

2.1 Design and Usability

This section introduces some theory regarding design and usability.

2.1.1 User-Centered Design

User-Centered Design (UCD) is a design process comprised of different phases, where the intended users' needs are taken into account in each step of the design process [3]. The phases of the UCD process may vary slightly on a case to case basis, but the process is characterized by an iterative multi-stage process that normally has four different phases during each cycle: research, ideation, design and evaluation.

The research phase consists of identifying the user, collecting user data or directly observing users while using the system to get a better understanding of their behaviour and needs. The ideation phase is where the observations are put to use to in order to come up with new concepts for the design. The design phase is when the system is implemented and a prototype is created, which is then tested and verified in the evaluation phase before repeating the iterative cycle until the product is finished.

2.1.2 Donald Norman's Design Principles

This section will briefly introduce Donald Norman's design principles [4] which have been taken into consideration while designing the prototype.

Discoverability: Refers to how well the current state of a system informs the user of what

actions are possible within the system.

Affordances: Attributes that let the user know *how* to use an object.

Signifiers: Easily confused with affordances, signifiers gives the user information about what an action *does* (rather than *how* to perform an action).

Mappings: Derived from the mathematical term, mapping explains the relationship between items, or in particular the relationship between controls and the performed action of an object in this context.

Constraints: In order to narrow down the amount of possible interactions with an object, constraints are used to show what interactions are *not* possible.

Feedback: After performing an action, feedback in this context is the information sent back to the user to tell them what action was performed and what it accomplished.

Conceptual Model: A highly simplified explanation of how something works. An example of this is how the Windows operating systems use a desktop with files and folders. This creates a conceptual model of a virtual desktop containing documents (although these are just conceptualizations - there are no actual files or folders inside a computer) that is similar to a traditional, physical desktop.

2.1.3 Usability Evaluation

According to Nielsen, five users is the optimal number of users for each iteration of formative testing. Adding more users after the third user will teach you less and less about usability based problems, and five is considered the sweet spot before returns of cost and time contra problems detected begin to diminish drastically [5]. A graph of these diminishing returns is shown in figure 2.1. For this reason, it is better to perform usability testing over multiple smaller test sessions with just a few users each time, then making improvements based on feedback from the testing before the next iteration, rather than spending time on larger more elaborate tests with more users involved in each step.

Cognitive Walkthrough

The cognitive walkthrough is a test method that can be used to examine the usability of a product. This is accomplished by defining a user profile or persona that the testers, usually referred to as an evaluator, takes on the role of. The evaluator is then asked to perform certain tasks from this user profiles perspective.

A cognitive walkthrough usually has four questions that are asked after each action sequence of a task has been walked through [6]:

- Will the user try to achieve the right effect?
- Will the user notice that the correct action is available?

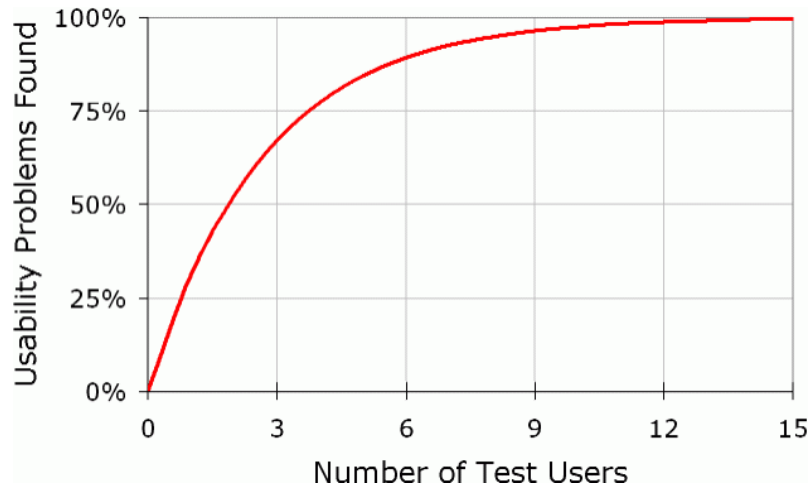


Figure 2.1: Diminishing returns curve for number of users during usability testing [5].

- Will the user associate the correct action with the effect they are trying to achieve?
- If the correct action is performed, will the user see that progress is being made towards solution of their task?

These questions help determining if a user interface (UI) makes inaccurate assumptions regarding the user's knowledge and experience level, and if the system is lacking feedback. They also help in assessing if performed actions align with user expectations.

Cognitive walkthroughs are quick and inexpensive compared to traditional usability testing and can be performed during any stage of a products development. Another advantage is that they can be performed without first hand access to end-users of the product in question.

System Usability Scale

System usability is a highly subjective matter which can not truly be measured in any absolute way. There are however multiple assessment templates focused on usability evaluation, such as the System Usability Scale (SUS). SUS is used to gauge the usability of software systems in terms of user efficacy, efficiency and satisfaction. In this thesis project, SUS was utilized to assess the export workflow of the Axis Companion client in particular.

SUS consists of a Likert scale with ten items where each item is a positive or negative statement regarding the user experience of the system in question [7]. The Likert scale usually ranges between 1 and 5 indicating agreement or disagreement with the statement, where 1 corresponds to "strongly disagree" and 5 to "strongly agree". The scores of each one of the user tests are then compiled and a formula is used to calculate a SUS rating of 0-100 for the system. Based on over 5000 observations of systems where SUS was applied, it has been concluded that a score of 82 ± 5 is a very good score, and scores in this range tends to only be given by promoters [8]. 68 is considered to be the average score that a system should strive for. Anything below 60 is in need of improvement while a score below 50 is considered unacceptable [9]. A more complete grading system of SUS scores can be seen in figure 2.2.

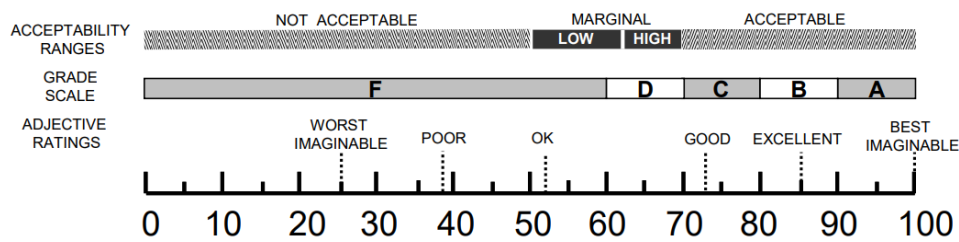


Figure 2.2: Grades based on SUS-scores [9].

2.2 Video Sharing

Some additional theoretical considerations relevant to the thesis in regards to streaming and sharing video are presented in this section.

2.2.1 Streaming Media

Streaming media can be described as "the act of sending sound or video to a computer, mobile phone, etc. directly from the internet so that it does not need to be downloaded and saved first" according to the Cambridge dictionary [10]. The purpose of streaming video in this context is to allow the recipient of the shared video to see the video playback directly in a common web browser of choice, rather than having to download the video clip and play it on their own computer. Local video playback would possibly require video codecs and additional software not bundled into their operating system in order for playback to work. These requirements are circumvented by allowing the media stream directly from the web.

2.2.2 Security and Video Integrity

The main security concern of the project was the integrity of the saved video recordings. Videos cannot be allowed to be altered after being exported from the Companion client.

The first step to ensuring this is encrypting data in transit, in this case meaning encrypting the video recordings while uploading them to Amazon Web Services S3 (more on AWS in section 3.1.1). Thankfully, all Amazon web services use the Transport Layer Security (TLS) protocol encryption for all of their data traffic, as well as for function calls between different services, by default [11].

S3 also provides server-side encryption for the data while at rest using the Advanced Encryption Standard specification with a 256 bit key length (AES256), which is a block cipher using a symmetric key algorithm. AES256 has been reviewed by the National Security Agency (NSA) and declared sufficient to protect top secret information [12] (the highest level of classified information in the United States [13]), meaning it certainly meets the requirements of this project as well.

Watermarking is a very common technique to uphold the integrity of a video. This is done by placing a pattern of some sort on top of the video that can only be seen using a

specific algorithm that is generated together with the pattern. If the pattern has disappeared from just one frame in the video, it has most likely been edited.

2.3 Software Development

This section gives some theoretical background regarding software development relevant to the thesis work.

2.3.1 Agile Development

Agile software development is an iterative approach to software development based around making continuous smaller deliveries rather than having a huge launch at the end of the development cycle, like a more traditional waterfall development model would. In agile software development, the results are constantly being tested and evaluated while project requirements and plans are revised. This allows for continuous improvement and frequent releases in short development cycles, with quick response to change and new customer expectations [14].

There are many agile development frameworks where some, such as Scrum and Kanban, focus on managing the workflow of a project, while some, like Extreme Programming, rather focus on the practices involved in the work [15]. Practically, a combination of different frameworks and agile practices are usually applied to software development projects, and even teams within the same company usually applies slightly different agile methods.

Spike is a term originating in Extreme Programming that refers to gathering information or researching a specific technical area during a short time-boxed interval [15]. A spike is often performed when it is difficult to estimate the cost of implementation due to unpredictable technical solutions. The main goal is to reduce the risk of technical problems occurring further along the development process and being able to perform a more accurate estimate of costs involved in the project. The spike result is normally a solution in its simplest form, just to ensure that there is a viable solution.

2.3.2 Analytics

There are many services today that offer analytics to companies as well as private users. Examples of these are Google Analytics, Amazon Analytics and Azure Analytics. The term analytics includes both the collection of data from users as well as the presentation and analysis of the data. The collection is done by sending events to the cloud that contain some sort of information about the user or what the user is doing. In most cases, this data is completely anonymous and can not be traced back to the user.

There are many benefits of implementing analytics in your website or application. There are few restrictions to what you can analyze about your product using analytics. By sending logs every time the product crashes for users, this can allow you to better understand what causes the crashes and in turn assist in fixing the issue to prevent them. It can also be used to analyze user behaviour in your product that can answer questions such as: which functionalities are frequently used? In which part of the product does the users spend most time? Are there any functionalities that are not used at all? This information

can then be used as a basis for decision making when it comes to what areas of the product should be prioritized for development.

Chapter 3

Technical Background

This chapter will introduce key technical components of the project, including both the hardware required as well as the software used for development and deployment.

3.1 Cloud Services

Notable cloud services used in the project are introduced here.

3.1.1 Amazon Web Services

Amazon Web Services (AWS) is a collection of web services providing a broad spectrum of data infrastructure services, such as storage options, analytics, databases, networking and computing power [16].

The relevant services utilized in this project are S3, which is a cloud storage service, and Lambda which is a serverless computing platform.

S3

S3 cloud storage consists of data containers called buckets [17]. When you create a bucket you also have to specify a region. The name of the bucket together with the chosen region create a unique identifier so that the user can access their bucket from outside of the S3 management console. Access permissions for buckets can be modified to for example allow anyone to read or download files from the bucket.

Lambda

Lambda is a serverless computing platform where runnable code can be uploaded and triggered to execute by certain events in other AWS's [18]. Lambdas can be written in

multiple programming languages such as Python and Java. They are used in conjunction with other Amazon services like S3. For instance, a Lambda can be configured to trigger whenever a specific event has taken place in an S3 bucket. Examples of this could be to modify or transform files as they are requested from the bucket or to send alerts via email whenever a file has been created.

3.1.2 Netlify

Netlify is a cloud computing company offering web hosting through continuous integration with a Git repository [19]. This means that the code in the connected Git repository will be used to create a website. When code is pushed to the Git repository, the website will be automatically built and deployed with the new changes.

3.2 Development and Tools

Different tools and components used for development is presented here.

3.2.1 Microsoft Visual Studio

Visual Studio is an integrated development environment (IDE) developed by Microsoft [20]. This is the code editor used for developing the Axis Companion and consequently used to implement the client side aspects of the new features introduced to the application in this thesis project.

3.2.2 .Net and C#

.Net is a developer platform that can be used to develop programs for Windows [21], such as the Companion application.

C# is one of the programming languages that can be used with .Net. It is an object oriented language that is used in a wide variety of software today [22].

3.2.3 React

React is a JavaScript library that is used to create graphical user interfaces for websites and mobile applications [23]. In this project, React was used to create the website where the recipient of the shared video material can view the video recordings.

3.2.4 Git and Version Control

Developing software simultaneously in a team would be a very difficult task without some type of version control system. One of the main purposes of version control is to make it possible for developers to simultaneously make changes in different parts of the program and eventually merge these changes without creating conflicting code. Version control also

keeps track of every change that has been made to the source-code and makes it possible to revert specific changes [24].

Git is one of the most widely used version control system available today [25]. It was developed in 2005 by Linus Thorvalds, the creator of Linux, and is now being further developed as an open source project. In figure 3.1, the most common workflow for a developer using Git is presented. The master branch is what most projects use as their main branch where all finished and reviewed changes eventually end up.

3.2.5 Code Review

Code review is an activity that is practised in most open-source projects and industrial systems today [26]. Whenever a developer has made changes to the source-code of a project, he or she submits the changes for code review. This allows the rest of the development team to inspect and review the changes before they are merged with the master branch.

There are multiple reasons for performing code reviews. A study [26] asked a number of developers that worked for Google what they wanted to achieve by performing code reviews. Some of the key themes identified were: education, maintaining norms, gate-keeping and accident prevention. These are all factors that can be achieved through code reviews.

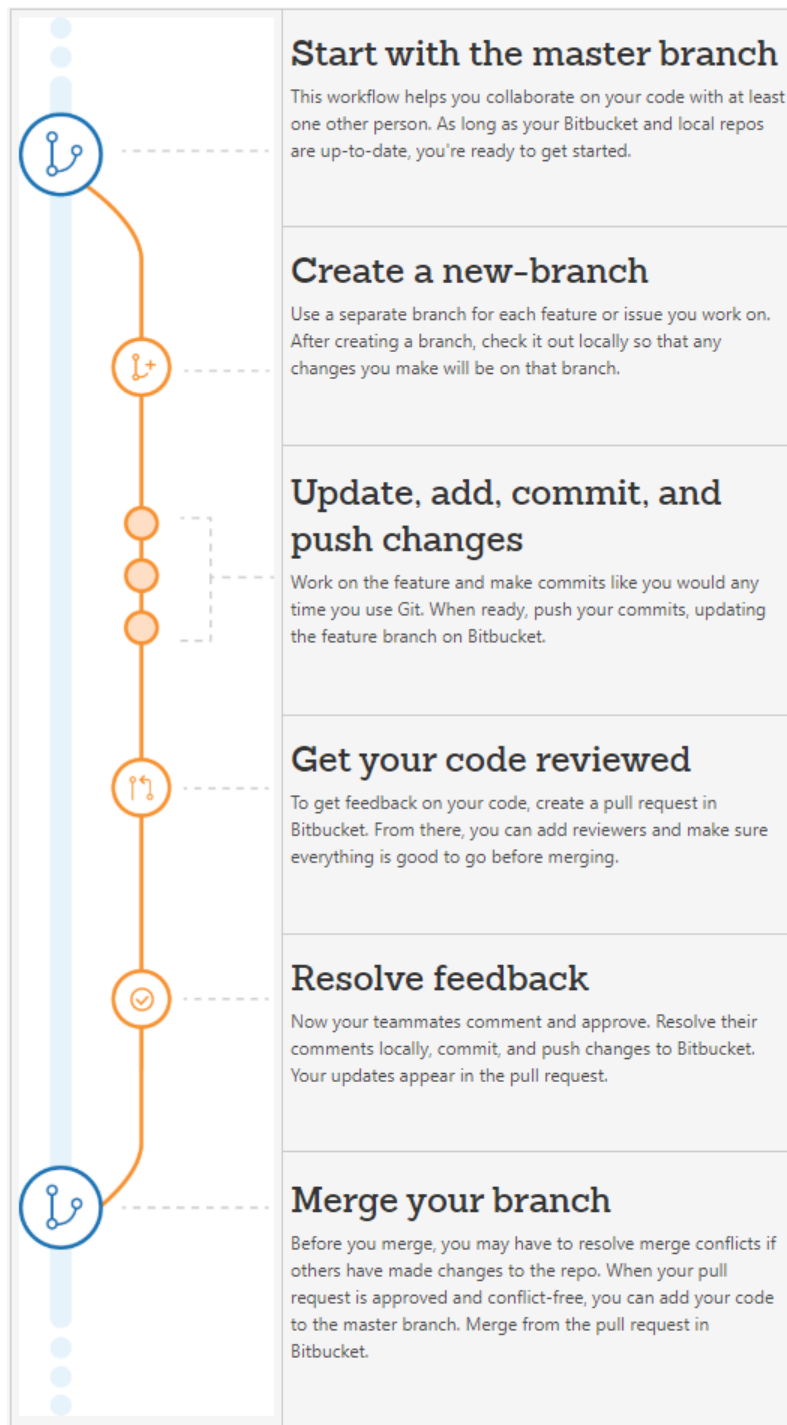


Figure 3.1: A visualization of a Git branch structure made by Atlassian [24].

Chapter 4

Methodology

This chapter explains the methodology used in the development process, as well as how user data was gathered and utilized in order to improve the prototype. Since a combination of agile development and a user-centered design approach was taken, the development process has been a cyclic iterative process done in phases.

4.1 Investigation and Ideation Phase

The investigation phase initially consisted of figuring out what we wanted to achieve, for what purpose, and how to reach the thesis goals. The required features of the product and the user needs were determined, and the workflow of the Companion export function was analyzed. In addition to this, a usability test of the Companion export flow was conducted with users lacking prior experience of the Companion application. This was done in order to gain a better understanding of what pain points exist in the way that the export feature currently functions, as well as what users would expect from the video sharing features introduced by this thesis project.

Apart from examining the Companion application itself, Axis competitors VMS's, and in particular their respective export features workflow, were also analyzed.

In UCD, understanding the users is the core principle. Contact with actual end-users were limited in this thesis work however, due to most of Axis' clientele being confidential. Instead, analytics were implemented in the Companion to get a better understanding of user habits and behaviour.

In preparation of designing the media player website, a study in the most popular video players and general user behaviours was conducted.

4.2 Design and Development Phase

The development process was loosely based on the agile software development framework Extreme Programming, while focusing on the UCD principles. While UCD was not fully enforced in the work process of this thesis, it was still highly important to get continuous feedback during development in order to keep improving the prototype. This was accomplished by having regular meetings with our supervisor at Axis and by holding demonstrations to present our current progress and showcase the prototype during different stages of development. This way we could get valuable feedback from other Axis employees.

Apart from work on the design aspects, it was difficult to estimate what areas would require the most work due to the different technical components involved in the back-end of the prototype. In the case of this thesis, the technical considerations involved in the project were difficult to assess since a product like this had not been made before. Hence, figuring out potential problems in the technical infrastructure as early as possible was crucial. For this reason we, in conjunction with our Axis supervisor, decided to perform a spike (defined in section 2.3.1) as the first step of our development process.

Once the spike was complete and a proof of concept had been created, certain improvements were made to the prototype before proceeding to user testing.

4.3 Verification and Evaluation Phase

During the last stage of the project the prototype was tested and evaluated. This phase consisted of two stages: cognitive walkthroughs and usability testing.

4.3.1 Cognitive Walkthroughs

Cognitive walkthroughs, as described in section 2.1.3, were performed with members of the small solutions team at Axis. The method employed was based on a more streamlined cognitive walkthrough approach [27] where the evaluators were asked two questions, rather than the four questions of the original cognitive walkthrough method, at each action sequence of the task.

4.3.2 Usability Test

The final stage of the verification and evaluation phase consisted of a usability test where each tester took on two separate roles. First, they acted as a user of Companion who wants to share a recording with an email address, where the export flow including the added video sharing functionality was reviewed. Secondly, the testers acted as the recipient of the shared video and the user experience of the media player website was evaluated.

Chapter 5

Investigation and Ideation

In this chapter we will present the results from our investigations and conceptual design processes.

5.1 Saving Recordings in the Axis Companion

In this section we will examine the already existing functionality for exporting video from Companion, from an interaction design perspective. The numbers written in parenthesis in this section refers to the highlighted items in figure 5.1 and figure 5.2.

5.1.1 Functionality

The main window of the companion has three different views that can be navigated to using the buttons in the top left (1). One of these views is the recording view which is shown in the figures. The user can play recorded video from a specific time by moving the green pointer (2) along the timeline as long as video actually was recorded during that time. It is possible to tell the camera to only record if the motion sensor detects movement which will result in longer periods of time without recorded material.

The user has three methods of accessing the export view which is shown in figure 5.2: clicking the save icon (3), pressing Ctrl+S with the keyboard or right-clicking the lower section of the window and selecting save. In the bottom left corner, the user can select which cameras to export video from by checking the boxes (4) of each respective camera. By moving the grey pointers (5) the user can select an interval on the timeline that they want to export from their chosen cameras. Timestamps (6) show which hours of the day that the user has selected using the pointers. When the user then presses the save button (7), the chosen material is downloaded as videos onto the users computer.

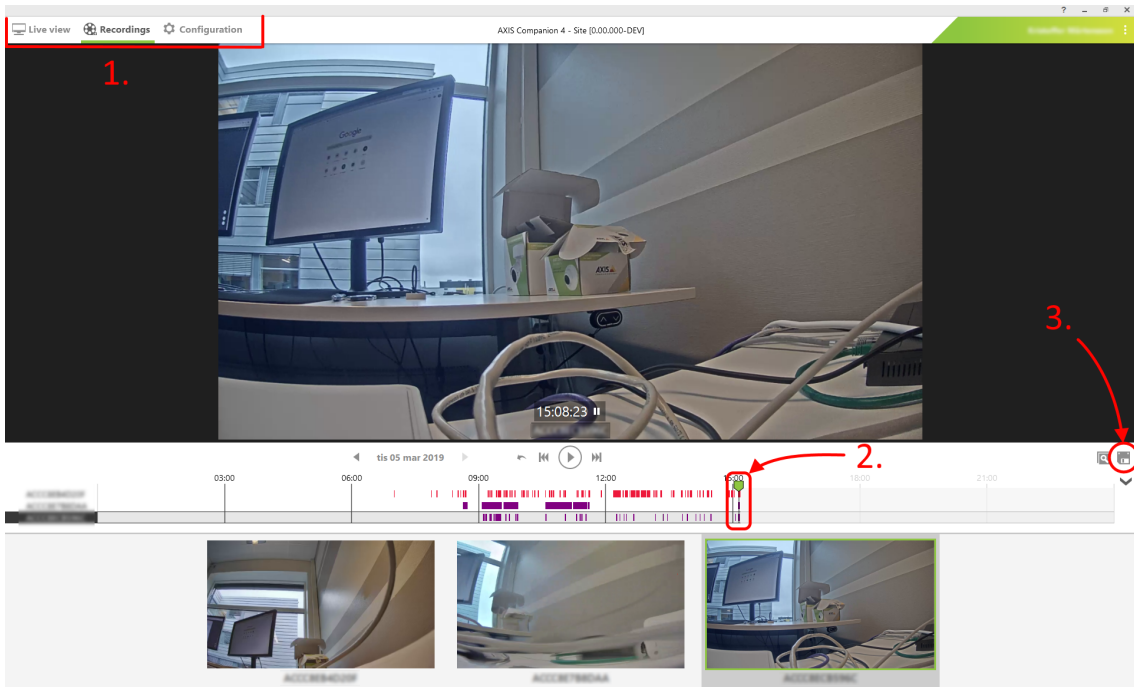


Figure 5.1: Axis Companion's recordings view.

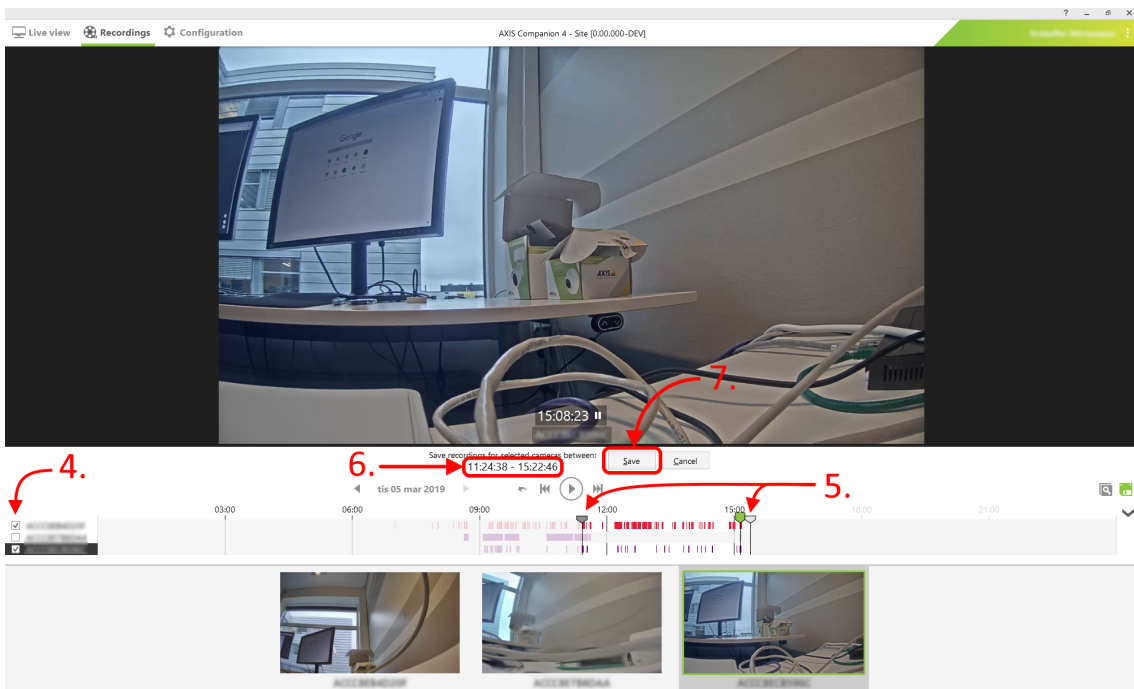


Figure 5.2: Axis Companion's save recordings view.

5.1.2 Analysis

Navigating the different views (1) is done using a top navigation design with tabs. The advantages of this design are that the different views can be navigated to at any time and it also leaves more horizontal space for the main window [28]. This is advantageous in regards to the export flow since it allows for a wider timeline and a wider video preview window.

Interactive timelines are not a very common concept in software applications, apart from appearing in other types of video editing softwares. It is therefore hard to properly compare the chosen solution with the pointer (2) with other solutions to interactive timelines. One aspect of the pointer that works very well is the instant feedback that the user is given. As you are moving the pointer along the timeline, the exact time the pointer is marking is shown at the bottom of the video. This makes it easier for the user to understand what moving the pointer along the timeline actually does.

The floppy disk (3) is a common signifier for save functionality across many software applications. The expected behaviour when using this signifier is that work you have just done will be saved so that nothing gets lost. Since the common theme presented to the user is saving recordings and not exporting recordings from their recorder which arguably is what the user is doing, it is understandable that this signifier was chosen. But what actually happens when the user clicks the button is not the same as the expected behaviour of this signifier. Nothing is immediately saved when the button is clicked, the user is instead shown what we refer to as the export view where they are asked to provide more information regarding which clips they want to export. It is possible that this could confuse some users and that another icon such as the common signifier for downloading material, see figure 5.3, would be better suited in this purpose.



Figure 5.3: A typical signifier for downloading.

Once the user is in the export view, they have to select which cameras to export from and the time interval. The cameras are chosen with the boxes at the bottom left of the view (4). The user is given clear feedback to which cameras are selected and which are not with checked or unchecked boxes. There is however no feedback or tooltip telling the user what they are actually doing by checking the boxes. A user that is unaware that multiple cameras can be selected for export might not choose to use this functionality due to not understanding it.

The time interval is chosen with the grey pointers (5). Just like with the green pointer, the user is given instant feedback to what time the pointers are currently marking (6), which is very helpful for the user.

5.2 Export Feature Usability Test

In this section we will present the format, results and conclusions from the first usability test. In this test, the Companion's export feature, such as it was prior to our video sharing implementation (where video recordings were just saved locally on the computer), was evaluated.

5.2.1 Overview

The purpose of this test was to let users that were completely new to Companion export video from the program in order for us to understand how intuitively this functionality is implemented.

This test was performed in the usability lab at IKDC, where we setup a camera site using three cameras which were used in the test. The participants of this test were four engineering students with a high degree of technical expertise.

The flow of a test session is summarized in figure 5.4. The testers first had to agree with our consent form stating that the results from this test will be used in our thesis and that they were being recorded. Then the user was given a task where they were asked to save recordings of an event that had taken place at a specific time. The whole task can be found in appendix A. For further analysis of the interaction, we recorded the screen while the tester was performing the task. Once the task was completed, the tester was asked to answer a SUS-questionnaire which can be found in appendix A.

The last part of the test was an interview which was more focused on how the user experienced using this functionality in order to complete the task. The questions asked during this part can also be found in appendix A.

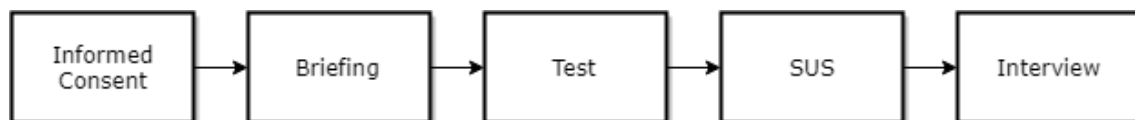


Figure 5.4: Test flow for the export feature usability test.

5.2.2 Results and Analysis

The calculated average SUS-score from the questionnaire was 53.5 which is lower than average, but acceptable (see 2.1.3 for more information about SUS). Due to the low number of testers, the SUS-score has a low degree of statistical significance, but it was still clear that the functionality was not perfectly implemented. Many interesting thoughts about the design came up during the interviews and through our observation of their sessions in the Companion.

All of the testers immediately navigated to the recordings view in order to complete the task. The next step in completing the task was figuring out how to get into export view. While trying to accomplish this, the testers started second guessing themselves whether they actually were in the correct view for exporting video, since they were unable to find a way to do it. This led some of them to navigate into other views, such as configuration view,

in search of a way to export. What all four of the testers eventually found, was that you could right click the bottom part of the recording view and select save recordings. When asked if they ever thought about using the button with the save icon, they answered that they either did not see it or that they did not think that the save icon held the functionality that they were searching for. None of the testers tried any combination of keyboard buttons to try and export video. The usage of this signifier was discussed in the 5.1 section and the fact that none of the testers chose to click that button further facilitates the point that this signifier is misplaced.

Once the testers had found how to save recordings, the next step was to figure out exactly at what time the event took place and which cameras that captured it. All the testers found it straightforward to move the marker along the timeline to preview the recorded material at different timestamps. One tester did however point out that it would have been helpful to be able to view all the cameras at the same time in the main window, in order to see immediately which cameras captured the event they had found. All of them tried moving the new markers along the timeline and in turn understood that they could use those to select the interval for their clips. One thing that became an issue while doing so was that the timeline was fully zoomed out initially, which made the markers hard to place on exactly the timestamp that they wanted. This led some of them to try and click on the interval so that they could manually type in the start and end time of the interval but this was not supported. What some of the testers eventually found was that they could use the scroll wheel on the mouse to zoom in on the timeline which made choosing the interval a bit easier.

The final step in the task is selecting the correct cameras to export from. This could be done by checking the boxes as explained in 5.1 but none of the testers even noticed this functionality. So what they did was that they manually exported from one camera at a time.

5.2.3 Discussion

In the final question of the interview, we asked the testers if they could come up with their own conceptual idea of how the sharing function could be added into the current workflow of saving recordings. All of the testers answered unanimously that they would prefer to have a dialogue pop up after they clicked the save button, where they could choose to export locally or share and also put in the email of the receiver. One of the testers expressed that they also in this dialogue would like feedback on the clips that they are going to export. Information such as what the time interval was, how many cameras they chose, how many clips that resulted in and the size of the clips.

Other features that were mentioned during the interviews was for example the arrow that moved the green pointer 5 seconds back. The testers felt that it was first unclear exactly what it did and once they figured it out, they thought it was very strange that there was no arrow to move the video 5 seconds forward as well. This leads us to believe that the arrow should either be removed or there should be another arrow that allows the user to move 5 seconds forward. There is also something called investigation mode that can be accessed by right clicking the top section of the main window or by clicking the icon that looks like a magnifying glass. Some of the testers went into investigation mode while searching for the export mode via right clicking the screen and they were all confused to what the

purpose of this mode was. Investigation mode gives the user the possibility to look frame by frame at events that span across a couple of seconds. The main purpose of this is to take an as accurate as possible snapshot of an event that has taken place.

5.3 Competitor Analysis

After examining the Axis Companion itself, the nearest competing surveillance VMS’s export solutions were also analyzed in order to gain a better understanding of how our system should work.

IPVM, one of the leading providers of expert reports and reviews of physical security technologies, has conducted a test comparing six of the most popular VMS’s export features [29], reviewing Milestone, Exacq, Avigilon, Genetec, Dahua and Hikvision. Their test criteria, outlined on the left axis of the table in figure 5.5, gave a clear idea of what features are to be expected from a good export solution and what pitfalls to avoid. However, the test also made it clear that no VMS currently support a feature similar to the online video sharing feature that we have implemented for Companion.

IPVM EXPORT SHOOTOUT SUMMARY						
	AVIGILON	DAHUA	EXACQ	Genetec	HIKVISION	milestone
Ease of Selecting Video	●●	●	●●	●●	●●	●●●
Single / Multicamera Export	●●●	●●	●●●	●●●	●●●	●●●
Timeline Adjustment	●●●	●	●●	●●●	●●●	●●●
Maximum Export Length	●●●	●	●●●	●●	●●	●●●
Number of Formats	●●●	●●●	●●●	●●●	●	●●●
Standalone Video Player	●●	●●●	●●●	●	●	●●
Export File Management	●	●●	●●	●●●	●	●
Ease of Exported Video Playback	●●●	●●	●●●	●●	●	●●●
Watermarking / Encryption	●●	●	●●	●●	●	●●●
File Size Estimation	●	●●	●	●	●●	●●●

Figure 5.5: A summary of the IPVM VMS comparison [29].

There were four key differentiators of the test that showed the most variance between different manufacturers:

- **Standalone player sizing:** Each of the VMS’s in the test (as well as Companion, which was not in the test) provides its own video player that comes bundled with the videos while exporting and saving videos locally. A large player size means that more disk space is required and would also makes the export itself slower.
- **Management of the exported video:** Refers to what happens after the video has been exported and what the user has to do in order to view the video file(s) that has been created.

- **Watermarking and encryption:** Watermarking essentially refers to a digital checksum which is utilized in order to ensure video integrity, and encryption is used to prevent unauthorized playback.
- **File size estimation:** The ability to provide an estimate of the total exported video file size prior to initiating the export procedure.

These differentiators were ultimately the deciding factors in the IPVM performance rankings of each software, making them important considerations in our work as well.

All of the VMS's in the test, as well as Companion, has a bundled video player in order to ensure that the users computer is able to show video playback (if, for example, the operating system lacks the necessary video codec required to render the video). Some manufacturers provide large, relatively advanced video players which are slower to transfer onto the users computer, making them less portable which is considered a major drawback in most use cases. A strength of our video sharing system is that it does not require any additional software; it just plays the exported video directly in the recipients web browser.

After video has been exported and stored locally, most VMS's typically forces the user to deal with the Windows Explorer file manager in order to view and open the video files. IPVM concludes that the user is better served by having this functionality more automated by the VMS, and a strong performer would ideally automatically open a video player with a playlist already populated by the exported video files [29]. Our use case is slightly different, since it is not the same user that both exports and watches the video clips, but we tried to automate this as much as possible by creating a web based video player with a playlist that is automatically populated with all of the exported videos as soon as the recipient clicks the link leading to the web portal.

The importance of watermarking and encryption are largely dependant on the use case. For legal reasons, watermarking could be very important to ensure data integrity if the video is supposed to be used as evidence in a legal matter. In cases where the video material is sensitive, encryption is also essential to ensure that no one unauthorized gets access to the video.

File size estimation is significant in order to give the exporting user an approximation of how time consuming the video export will be and how much disk space will be required. And, while not a focus of this thesis, it is worth mentioning that file size is also relevant from a financial standpoint in our use case, since we are using cloud services. This means that the file size of exported videos has to be calculated prior to the cloud upload taking place, in order to ensure that the service is not being abused by superfluously large uploads (or by taking a more naive approach and simply limiting the allowed video duration of the export).

5.4 Video Players and User Behaviour

YouTube is currently the most used video system on the web [30]. Its popularity made YouTube a suitable choice to base the design of a web based media player around in order to create an easily accessible media player that most users feel familiar with even when they use it for the first time.

The vast majority of users only use basic video player functions, such as play/pause and volume control [30]. This shows that, for most users, there is little interest in advanced functionality beyond the most important functions. More advanced functions are better omitted unless crucial, since they could potentially impair the user experience and serve to confuse less experienced users.

Chapter 6

Design and Development

This chapter presents the design and development process of the prototype.

6.1 Lo-fi Prototyping and Conceptual Model

In this section we present the results from our initial lo-fi designs and conceptual modeling.

6.1.1 Lo-fi Dialog Design

Our initial plan to include the sharing functionality in the current export flow was to include a dialog after the user had pressed save. We wanted this dialog to hold more functionality than just an input box for an email address. Another thought behind it was that it could be used in the future to add further functionality to exporting as well as sharing. An initial drawing of how we pictured this dialog can be seen in figure 6.1. In this design, the user could see which cameras they had chosen and the time interval. They could also select whether they wanted to save the recordings locally or share the selected recordings or do both at the same time.

6.1.2 Lo-fi Website Design

The website where the recipient views the shared recordings has to be easily accessible to any type of user, regardless of their level of technical expertise. Our aim was to first and foremost create a site that has a high degree of familiarity to a majority of users.

As mentioned in the investigation chapter, section 5.4, YouTube is the most commonly used web based video player. In order to give our media player a familiar feeling, our initial lo-fi design, seen in figure 6.2, was based on a common HTML5 web player such as YouTube.

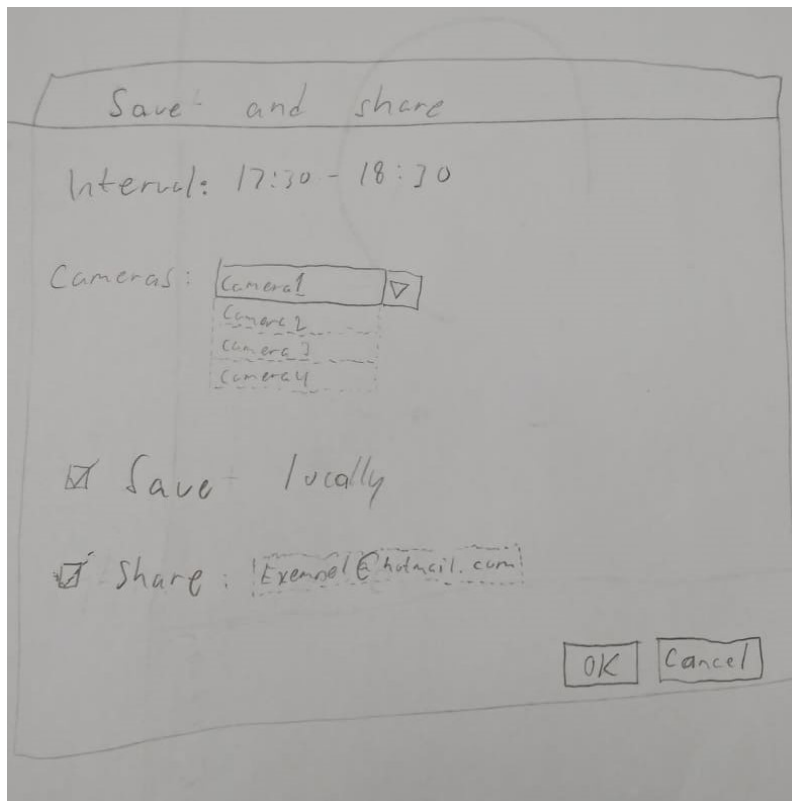


Figure 6.1: Picture of lo-fi design of export dialog.

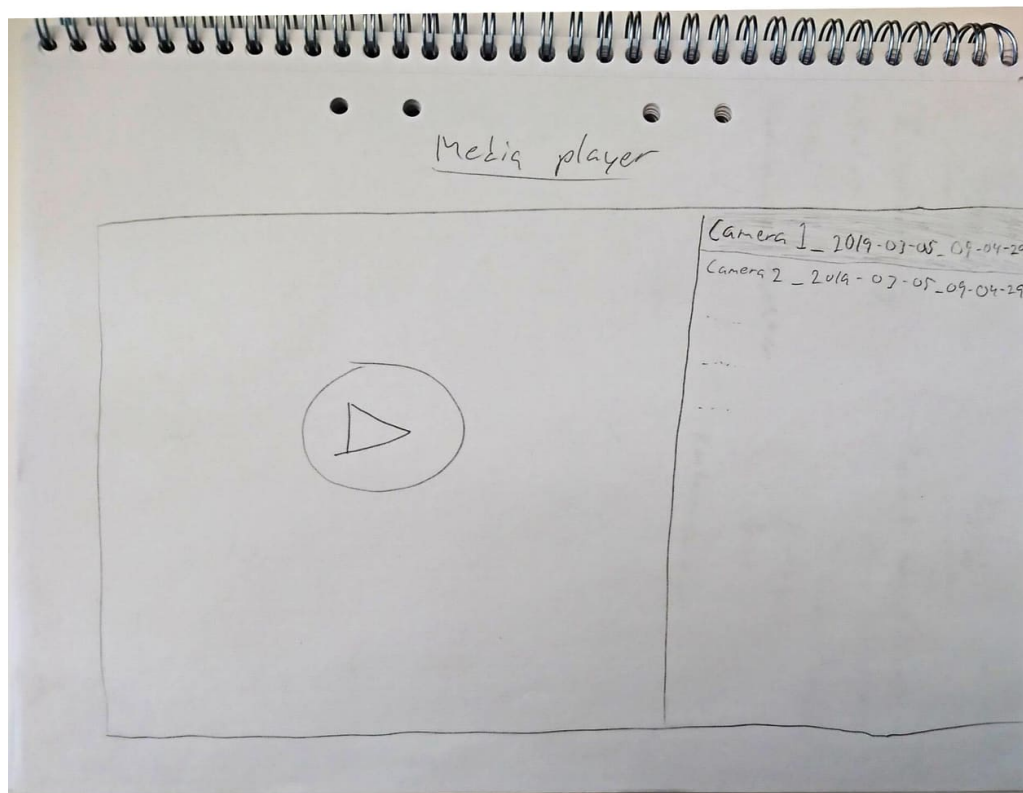


Figure 6.2: Lo-fi design of the web based media player.

In the competitor analysis, section 5.3, it is mentioned that Axis as well as most of their competitors bundles their own media players with the video files when a user performs a local export. This is done in order to ensure that the user is able to view the exported material. Axis' media player that is bundled when performing an export in Companion is called Axis File Player (seen in figure 6.3) and was also used as a reference point for the website. More so as a reference for what features to include in the media player, rather than for the design itself however. The goal was to keep all necessary functionality of the original player while maintaining a simplistic website design.

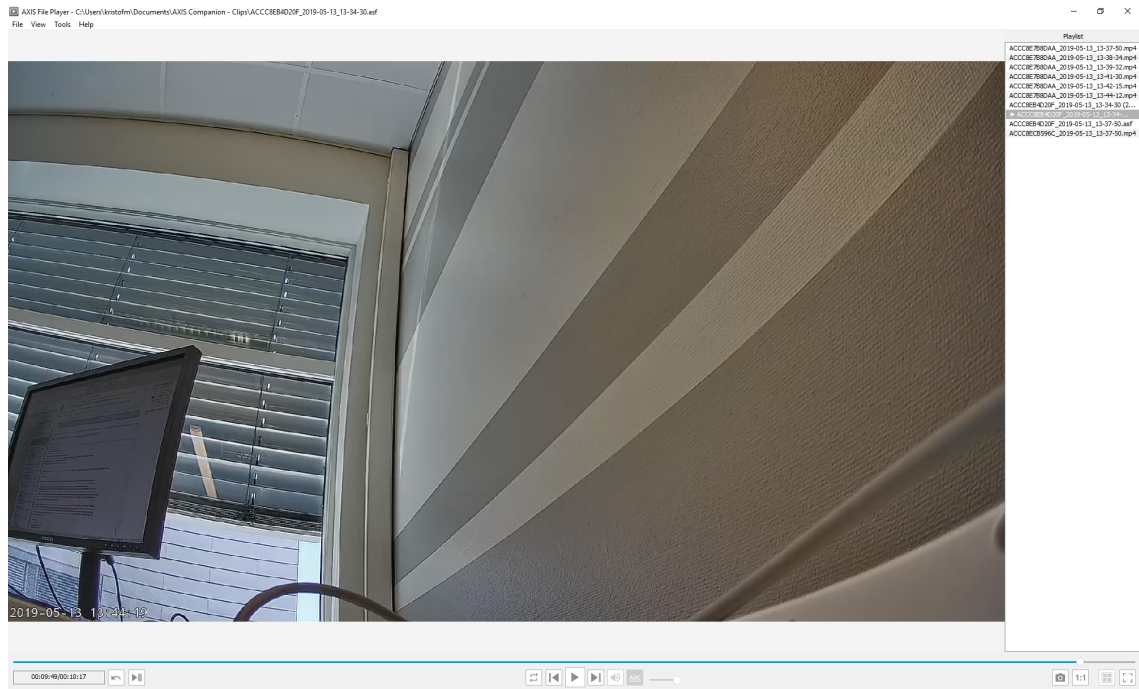


Figure 6.3: Axis File Player.

6.1.3 Conceptual Software Ecosystem

At a very early stage, we began with conceptual modeling of how our solution would work end-to-end. We came to the conclusion that we needed four main components in place in order for this solution to work.

- A service that could upload the saved recordings to some sort of storage.
- An appropriate storage solution.
- A service that could be responsible for sending the email to the intended receiver.
- A website where the receiver could view the shared recordings.

Since Axis was already using many of the services provided by AWS, our Axis supervisor recommended that we looked into their services for our solution. After doing some research on the different services and possibilities, we drew an initial model of the software ecosystem that can be seen in figure 6.4. The main components consisted of Axis Companion, AWS S3, AWS Lambda and a React website hosted by Netlify.

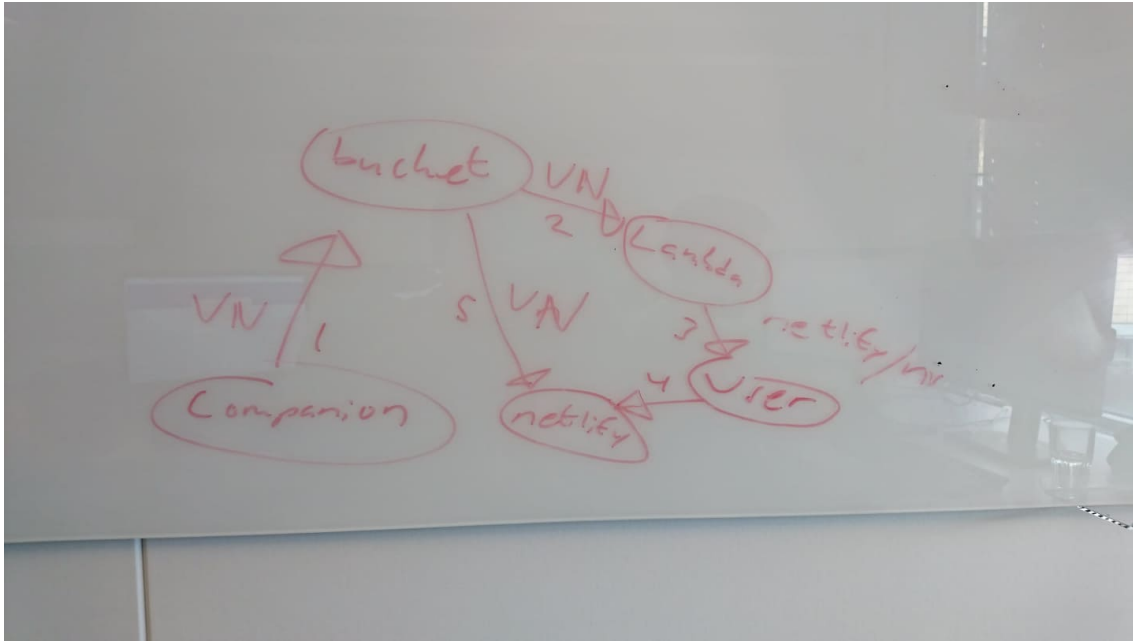


Figure 6.4: Picture of initial conceptual model of software ecosystem.

6.2 Hi-fi Prototyping

In this section we present the implementation process of our hi-fi prototype.

6.2.1 Spike

In addition to the conceptual design and lo-fi prototyping of graphical user interfaces, we decided to perform a spike on an end-to-end solution for our problem. The main reason for this was to identify which parts of our solution that required the most work. It was also a great exercise for us to get a better understanding of the software ecosystem that we had to create in order for this to work. A simplified representation of the software ecosystem can be seen in figure 6.5. The numbers in the figure represent the order in which things happen before the video is shown to the receiver. These will be referred to as different steps.

The first step refers to the sender using Companion with intent to export and share some recordings with a specific recipient. At this point of our development we made no changes to the user interaction when exporting video. The only thing we added was a placeholder input box where we could manually enter an email address belonging to the intended receiver. We did however have to make changes on the back-end of the application. We had to implement a service that would upload the clip that the user was exporting to the bucket that we had created in AWS. This service was also responsible for taking the email address that we input in the text box and attaching it to the uploaded clip. This was done by adding the address as metadata to the clip. The second step in the figure represents the uploading of the clip to the bucket.

We now needed a solution for sending the email to the receiver to let them know that

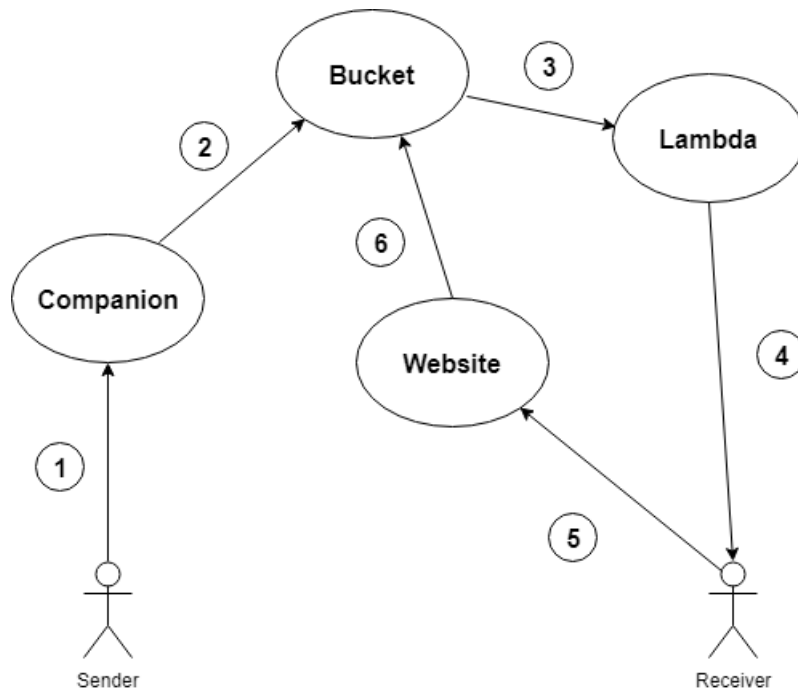


Figure 6.5: Picture of simply explained software ecosystem.

someone had shared a clip with them. This brings us to step three and four. We created a Lambda script written in Python that was configured to trigger on an S3 event called "All object create events". This meant that the Lambda's code is executed every time something is uploaded to the targeted S3 bucket. The Lambda could then retrieve information about the file that was uploaded. This meant that we could both get information about the name of the clip, as well as the metadata of the file, which in turn meant that the email address of the receiver was retrieved. We then used this information to send an email to the receiver using a Python simple mail transfer protocol (SMTP) library. The email contained a link to our website with the name of the uploaded file as a URL parameter such as: "www.ourwebsite.org/?clip=NameOfTheClip". The email was sent from a Gmail account that we created for this purpose.

The fifth step refers to when the receiver clicks the URL attached in the email in order to navigate to the website and watch the clip that was shared. The website at this point contained nothing more than a simple media player that could take a URL as video source and play it in the browser. The media player had basic controls such as play/pause, volume control, playback rate, fullscreen toggle and a progress bar, but did not contain any functionality to handle multiple video clips. The name of the clip that was passed as a URL parameter when the receiver navigated to the website, was used to create a new URL that pointed to the clip that was placed in the bucket such as: "http://aws-region.MyBucket/NameOfTheClip". This URL was passed to the media player which in turn began streaming the clip from the bucket and the receiver could now view the shared clip in the browser, see figure 6.6.

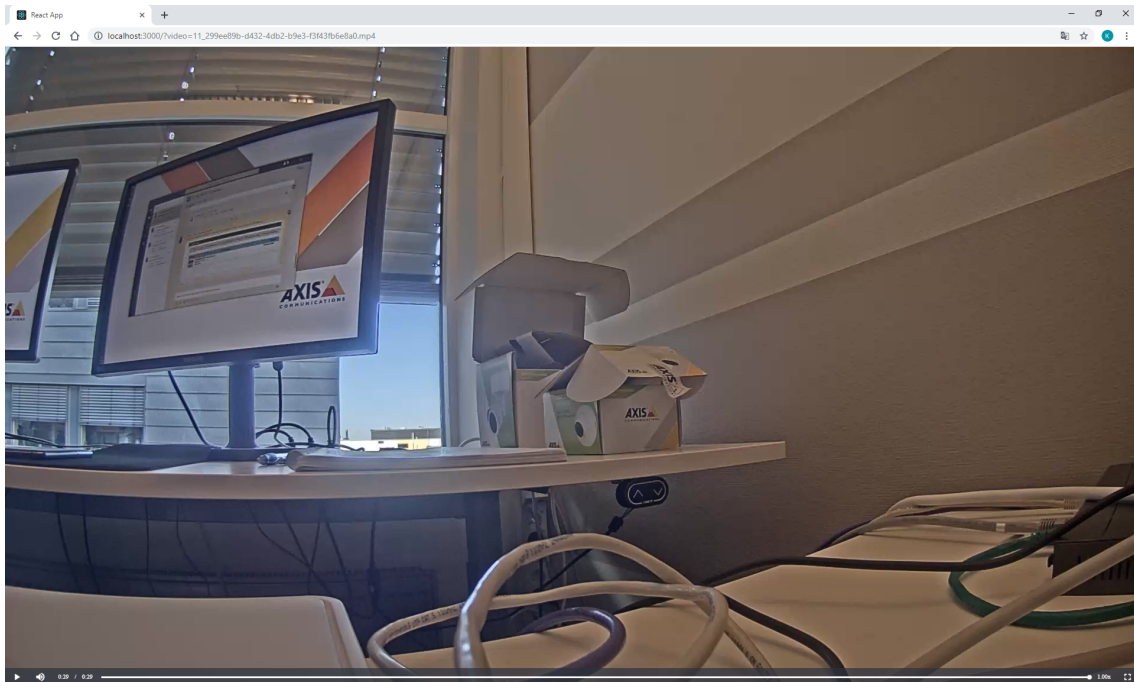


Figure 6.6: The resulting website of the spike, streaming a shared video clip.

6.2.2 Further Development

The result of our spike could be considered as a proof of concept of what we set out to accomplish in this thesis. However, this was still just a basic prototype with significant limitations and some immediate improvements were required.

The Export and Sharing Process

When the user exports recorded material from the client, it very rarely results in all the material being exported in one file. In some cases it could even result in very many files depending on how many cameras are chosen and how large the chosen interval is. Our prototype only worked if the exported material came in just one file and since this was rarely the case, we needed to further improve our product.

There were not many changes needed to step one and two in figure 6.5, we only needed to make sure that all files that were being locally exported were also exported to the cloud. Step three however needed to be fixed. The Lambda we had created triggered whenever a new file was uploaded to the bucket, and if one export job resulted in multiple files, then multiple emails would be sent to the receiver. Our solution for this was to create an additional bucket. We implemented so that all of the files except the last one from each export job was placed in one bucket while the last file was placed in another. The Lambda then only triggered when the last file was uploaded since that was to a different bucket, resulting in only one mail being sent.

We did not have to make any changes to the Lambda or the email that was being sent to the receiver. The website however needed an upgrade. Each file contained one clip so the website had to be able to handle a playlist of clips instead of just one clip. For

this to be possible, all of the clips being exported in a single export job needed a unique identifier so that they all could be retrieved together and put into a playlist on the website. We solved this by generating a globally unique identifier (GUID) [31] in the client for each new export job and named all the clips with an index followed by the GUID such as: "5_d2aac976-4kf4-40cf-8d5b-c9deae397f6e.mp4". By looking at the index of the last clip that was uploaded, we knew how many clips belonged to this export job. We could then populate the playlist on the website without actually looking through the entire bucket and filtering out clips that belonged to a specific export job.

6.2.3 Design Improvements

Before the final testing of our prototype we improved the design of the sharing functionality in Companion as well as the design of the media player website.

Companion

Our first implementation of the sharing feature, UI wise in Companion, was very barebones just to get a functioning prototype up and running, and needed to be improved. The way we integrated it into the export flow was that instead of having the input box in the export mode, we implemented a checkbox that the user could select if they wanted to use the sharing functionality, which can be seen in figure 6.7. If this box was checked when the user clicked save, a dialog popped up where the user could enter the email address of the intended receiver. The dialog can be seen in 6.8.

It is worth mentioning that most of the users during the first usability test advocated that the sharing functionality would be best added in a dialog that appeared after the user had pressed the Save button. The reason why we chose to place the checkbox before the dialog that appeared was to give the user a chance to find it before clicking the Save button, which is a finalizing interaction in the export flow. A user that wants to share saved recordings would probably not click the Save button without knowing they are also sharing them at the same time. We predicted that the risk of a user getting stuck while searching for the sharing functionality was to big and that it was better to let the user feel sure about having found it, before finalizing the export.

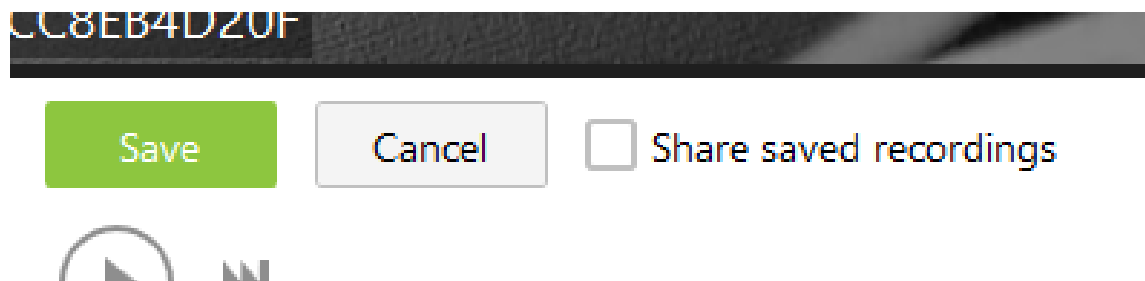


Figure 6.7: The checkbox where the user can select to also share the saved recordings.

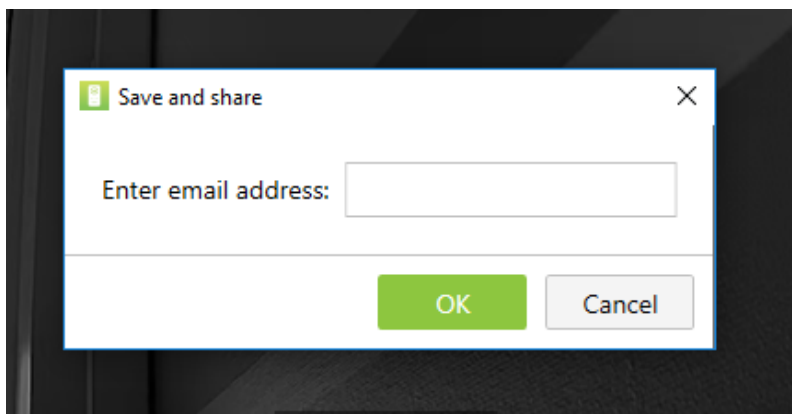


Figure 6.8: The dialog window where the user can enter the email address of the intended receiver.

Media Player Website

During this stage of development, the design of the media player website was also improved. The new website design can be seen in figure 6.9. The website was re-worked to properly handle multiple clips and designed to display all of them in a scrollable playlist. To the right of the media player the user can maneuver between the clips in the playlist, making it easy to select which clip to view. All of the clips in the playlist are sorted in descending order, primarily based on which camera the material was recorded on and secondarily based on the starting time of the clip.

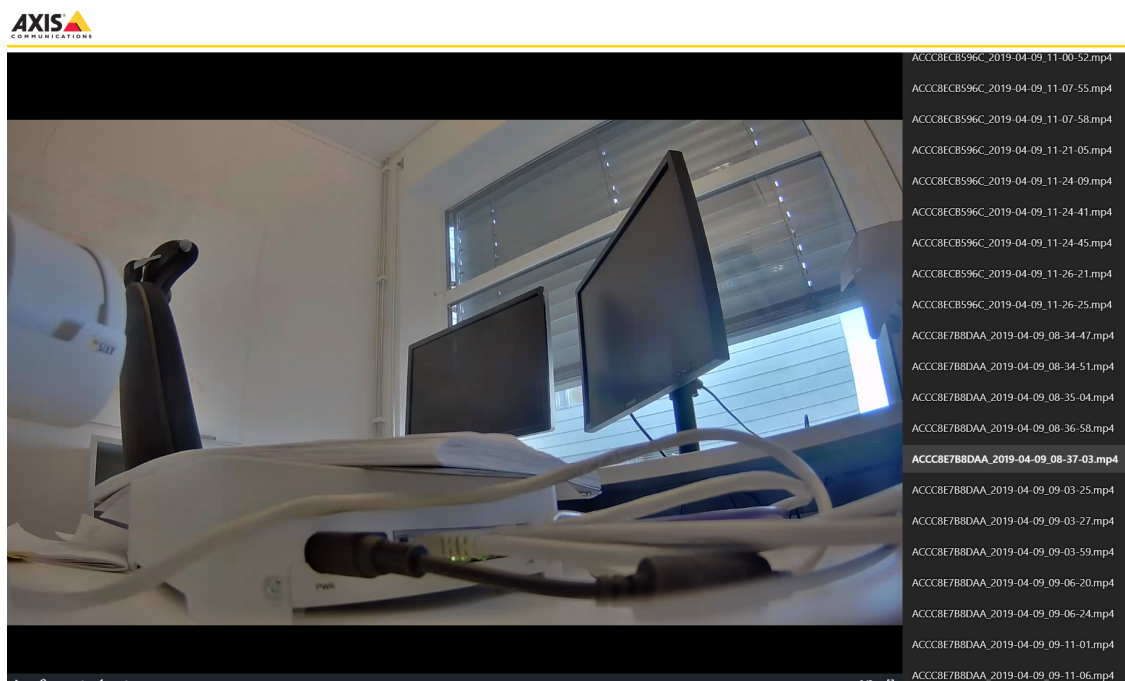


Figure 6.9: The website with multiple video clips in a playlist.

The addition of playlist functionality meant that the website would take slightly longer to load (depending on the recipient's network bandwidth) in case the export job contains

many clips since meta data for each video clip is downloaded, in order for the playlist to be generated. Because of this slight loading time in certain use cases, we felt that it was important to give the user some kind of feedback to ensure them that the page is actually loading properly. To accomplish this we added a spinning loading overlay which can be seen in figure 6.10.

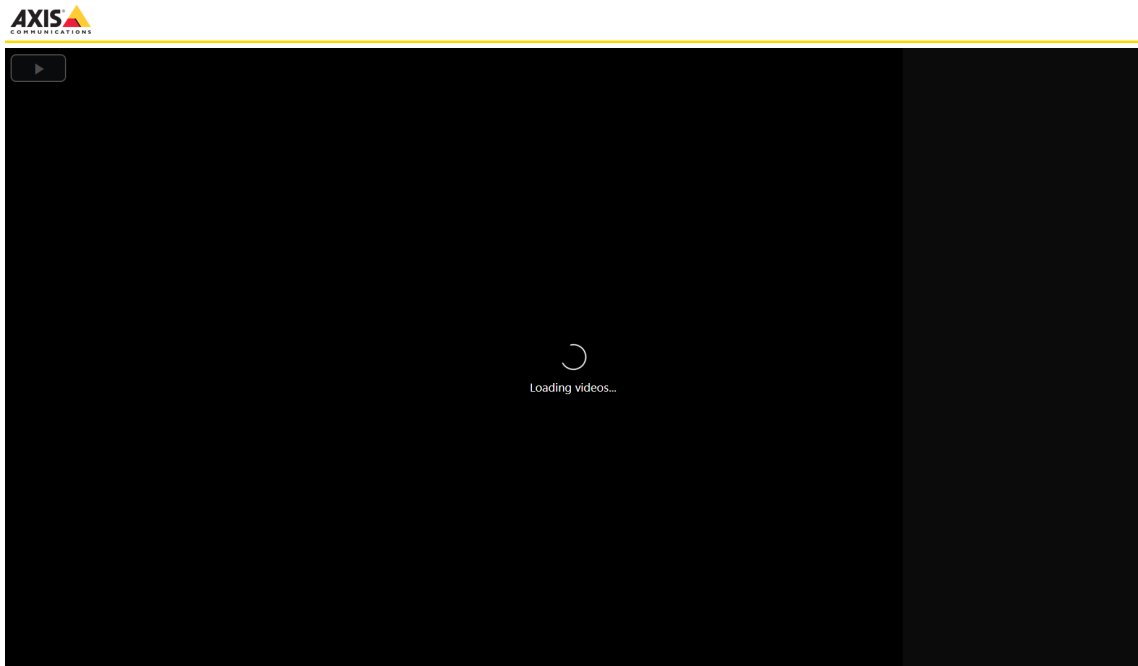


Figure 6.10: The loading overlay which is shown while the playlist is being populated.

Some additional quality of life improvements were also made to the website, while the media player itself retained a familiar design. A navigation bar at the top of the page with the Axis logo was added as well as a new web browser tab icon, also featuring the Axis logo, and the browser tab title was changed to "Axis Media Player". These additions were made in order to instill a more formal impression, making the website appear more trustworthy to the recipient. The navigation bar can be seen in figures 6.9 and 6.10, and the new browser tab design is featured in figure 6.11 (the previous design can be seen in the top left corner of figure 6.6, which is basically just the standard React appearance).

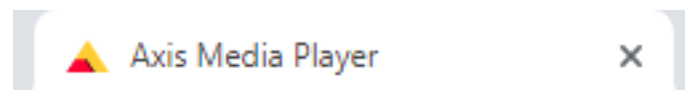


Figure 6.11: The new web browser tab design, as it appears in a Google Chrome tab.

Chapter 7

Verification and Evaluation

This chapter describes the verification and evaluation phase of the thesis project. This phase consisted of two parts: cognitive walkthroughs and usability testing. All tests were performed with the latest version of the prototype, outlined in section 6.2.2.

Both test types were performed at Axis, using the same setup that we used during development of the prototype. The camera site used for testing consisted of three network cameras and a recorder. The only software required for the tests were Axis Companion and a web browser (Google Chrome).

The following sections will detail how the tests were conducted and the conclusions drawn from the test sessions.

7.1 Cognitive Walkthroughs

Cognitive walkthroughs, described in section 2.1.3, were the first stage of the verification process.

7.1.1 Participants

The cognitive walkthroughs were conducted with three experts; Axis employees who were Companion developers. They acted as evaluators and performed the walkthroughs individually.

7.1.2 Test Execution

The sessions began with a short briefing about the cognitive walkthrough test method. However, since the evaluators were Axis employees, as well as experts on the Companion application and relatively up-to-date on our thesis work, not much further briefing was required.

The test was comprised of two tasks. The first task was to export and share a video recording of a specific event. In other words, this task was meant to evaluate the whole workflow of exporting video recordings, and not just the features that we have implemented in particular. The second task was to view the shared video recordings by opening the link provided in the email that was sent in task one.

Rather than a more exploratory usability test, the cognitive walkthroughs were focused around having each evaluator perform specific steps. Task one consisted of an action sequence of 13 steps that each evaluator performed. The action sequence of task two consisted of 6 steps. The tasks and their respective action sequences can be seen in figure 7.1.

Cognitive Walkthrough

Task 1:

Export and share a video recording of an event that took place at around 12:27 o'clock with an email. The user is signed in to ACC and the walkthrough begins in the "Recordings" view.

Action sequence:

1. Click "Save recordings" (the floppy disc to the right).
2. Additional information as well as additional possible interactions appear.
3. Hold the pointer on the timeline and zoom in by scrolling the mouse wheel.
4. Move one of the time-picker pointers to 12:25 and the other pointer to 12:30.
5. A recording interval to be exported has been chosen.
6. Check the checkboxes on the bottom left that are next to the cameras ACCC8E7B8DAA and ACCC8ECB596C, make sure that camera ACCC8EB4D20F is not checked.
7. Check the checkbox called "Share saved recordings".
8. Click "Save".
9. The "Export and Share" dialog window appears.
10. Enter the email address _____ in the input field.
11. Click "OK".
12. A progress bar for the export appears.
13. When the export is done, the progress bar disappears.

Task 2:

View the shared video recordings by clicking the link contained in the received email. A web browser has already been started and the email sent in task 1 is open.

Action sequence:

1. Click the link in the email.
2. A browser window containing the Axis Media Player appears.
3. Wait for the loading overlay to disappear.
4. The first video will start playing and a video playlist appears on the right.
5. Click on the second item in the playlist to the right.
6. Another video will start playing.

Figure 7.1: The cognitive walkthrough's tasks and action sequences. The email address used in the test has been omitted.

After each step of these action sequences, the evaluator was asked two questions, based on the streamlined cognitive walkthrough approach suggested by Spencer [27]:

- Would the user have known what to do at this step?

- If the user did the right thing, will the user know that he or she did the right thing and is making progress towards the goal?

7.1.3 Test Result

This section will present the most striking feedback given from the walkthroughs. Steps where all the evaluators agreed that a user would have felt confident in their actions will not be presented as thoroughly as the steps that contained more issues. If all evaluators agreed that a user would feel confident at a certain step of the action sequence, that step is omitted from the list below.

Task 1

1. **Click “Save recordings” (the floppy disc to the right).**

"The floppy disc is an outdated metaphor and its function may not be obvious to all users."

The very first step proved to be one of the most problematic. The evaluators found the floppy disc icon, supposed to signify "save", to be outdated. Its placement far to the right side of the screen also made it unnecessarily difficult to find.

3. **Hold the pointer on the timeline at 12 o'clock and zoom in by scrolling the mouse wheel.**

The zoom-function on the timeline, while not crucial in order to export flow, can come in very handy while picking a specific time interval. However, it did not feel apparent that scrolling the mouse wheel would have this effect and a new user may not locate this functionality unless they happen to right-click on the timeline.

6. **Check the checkboxes on the bottom left that are next to the cameras ACCC8E7B8DAA and ACCC8ECB596C, make sure that camera ACCC8EB4D20F is not checked.**

"If the user doesn't already know that it's possible to export and share multiple video clips at the same time, it's likely that they would struggle to find this functionality."

Similarly to the floppy disc button mentioned above which is used to get into the Save recordings view, the list of cameras is also located near the edge of the screen. After pressing Save recordings the list of camera checkboxes appear to the far left and can easily be missed, potentially causing the user to export from one camera at a time without noticing that it is actually possible to chose multiple camera feeds in the same export job.

7. **Check the checkbox called “Share saved recordings”.**

"Some clarification about how the sharing is actually going to happen could be given to the user here, by adding a tooltip on mouse hover for example."

The evaluators felt confident that the user would have known what to do at this step, but feedback was slightly lacking.

10. **Enter the email address xyz@mail.com in the input field.**

In this step, the evaluators discovered two issues that could cause confusion:

- *"Am I supposed to enter my own email address, or the recipients email address?"*

- *"What is the email being used for? How does sharing happen? Will the files be attached to an email, or will the email just contain a link that somehow displays the video clips?"*

12. A progress bar for the export appears.

"The progress bar provides enough feedback to let the user know that the export is in progress, but the wording should be improved to make it clear that the sharing is also happening simultaneously."

When the export was happening, the evaluators trusted that the sharing also took place. However, some more explicit feedback letting the user know that the sharing is also in progress was desired.

13. When the export is done, the progress bar disappears.

"The file explorer window that shows up after exporting indicates success of the local export, but you don't get any reassurance that the sharing actually worked. However, since there is no indication that sharing didn't work, the user would assume that it did work properly."

Feedback indicating that the sharing process succeeded is missing.

Task 2

The evaluators found this task to be very straightforward with little risk of confusion for a typical end-user. Since this task was so well received, the feedback from this action sequence will not be broken down step-by-step. There were however a couple of things that the evaluators felt could be improved upon. The main points of suggested improvements are summarized below.

A possibility to add a description of the clip (while sharing in the Companion rather than something that could be done on the website, which was the intended subject of evaluation in this task) would have been appreciated. Also, being able to add your own comments to the clips in order to make it easier to remember certain events could also be useful for future viewing.

7.2 Usability Testing

This section presents the second part of our verification tests.

7.2.1 Participants

There were a total of nine participants for this test, which they all performed individually. All of the participants were fifth year engineering students with high technical expertise that had no previous experience with the Companion.

7.2.2 Test Execution

The setup of the verification usability tests was similar to that of the initial usability tests that were performed, as seen in section 5.2. First, we briefed the user about the Companion

before the user conducted the test. Following the test they answered a SUS-questionnaire and lastly we performed a short interview that was followed by an open discussion about the solution.

The major difference between this and the first usability test, seen in section 5.2, was that this time, the sharing functionality was included and the test divided into two separate tasks. The first task was to export and share with a given email, an event that we specified. The second task was to view the email that was sent in the first task, and click the link provided in order to view the shared recordings. The tasks can be found in appendix B.

It is notable that the SUS-questionnaire was meant to evaluate the export flow as a whole, not just focusing on the parts we implemented, in order to see if the addition of our sharing functionality had somehow interrupted the workflow in comparison to the first usability test. The interview on the other hand was focused primarily on the specific features that we have implemented.

7.2.3 Questionnaires

The SUS-questionnaire that we used was the same as in the first usability tests, which can be seen in appendix A.2, with the small addition of replacing "export" with "export and share" in all the statements. The idea behind the interview questions was for us to get some answers to questions that we had about our design. It was also a chance for us to pick their brain a bit and maybe get some new thoughts and suggestions about the entire solution itself.

7.2.4 Test Result

The user feedback given from the usability test as well as the result of the SUS-questionnaire are presented below.

SUS-questionnaire

The average SUS-score from the questionnaire was 76.1, which is well above average and considerably higher than the score from the first user test which had a mere 53.5 average score (see 5.2 for details of the first user test, and 2.1.3 for more on SUS). Although the first test's SUS-score lacked statistical significance, due to the limited number of testers, the higher score from this user test at least gave us no reason to believe that the addition of sharing functionality made the usability worse.

Task 1

1. Was the sharing functionality easy to find?

"Yes, it was not difficult at all. It had a very natural placement."

The overall consensus to this question was that the functionality was easy to find. Many of the testers indicated that the reason they found it easy was because the checkbox was located in the center where the majority of the interaction took place. One of the testers stated that it might have been even easier to find if it was placed to the left of the Save

button since we read from left to right. This might have prevented a user from clicking the Save button to finalize the export before even seeing the checkbox with the text.

2. Was it clear what you were doing when you checked the checkbox next to “share saved recordings?”

"Yes, I expected that I was supposed to enter the email address in the next step."

"I was missing feedback in order to understand how the clip was going to be shared."

These were the main opinions lifted in this question. Since the testers already knew that there was an email with which they were going to share the saved recordings, they knew that clicking the checkbox would enable this functionality. But when asked to consider if they still would have understood this without previous knowledge, most of them answered that they probably would have expected either feedback to what they were doing or a prompt asking them how they wanted to share the clips. Many services today have functionality to share images, links, videos et cetera and when the user accesses this functionality, they are always prompted how they wish to share them. This was what some users meant would be the expected behaviour when clicking the sharing checkbox.

3. Was there anything confusing about the dialog window that appeared?

"No, but consider the wording of the text. The way it's written now makes you unsure whether it's your own email or the recipients that you're supposed to input."

This was the main concern with the dialog. Simply changing the text to "Enter email address of recipient" would have been a sufficient solution to this was what most of our testers argued.

4. What did you think of the feedback you received whilst the recordings were downloading and after they were done?

"There could have been a text saying 'sending' after the saving is complete, to make it clearer."

Most testers felt that they trusted that the sharing had been done despite not receiving feedback about it, but the overall opinion was that it would probably be best to have some more direct feedback stating the fact that sharing was going on. The suggestions to how this could be done was for example having another progress bar simultaneously that kept track of the progress of the sharing or once the saving was done informing the user that the saved recordings also had been shared with the given email address.

5. Do you think the general user would feel comfortable using the sharing functionality, even if it involves sharing sensitive video material?

"Email is generally not an encrypted channel, the user would need to trust a third-party email service."

"Regular users would probably feel comfortable with using it."

A couple of our testers were concerned about how safe it would be to use email for this purpose. This was based on their previous knowledge of how the email protocols work and the level of security they generally have. Making sure that the email service is secure and links will not be intercepted is definitely of interest.

However, most testers thought that the end-users of Companion would have faith that the built in functions in the program are secure. This includes the sharing functionality.

6. Do you think the sharing functionality is well integrated with saving recordings in the Companion? Would you have preferred a different approach?

"Yes, the process had a nice flow. The sharing felt like a natural part of the program."

Most testers felt that the way the functionality was integrated right now felt natural. From our own observations we did not notice that anyone hesitated to click the checkbox or in any way halted their work progress due to not finding or understanding the checkbox itself. There were however some alternatives presented when they were asked about a different approach.

A couple of testers felt that it was a bit odd to have the sharing and saving functionality all in one place. What they suggested was to lift the sharing functionality from the export mode and have this separately, in a mode of its own. This could also open up the possibility to share clips that have been saved previously which was a functionality some users claimed would be highly desirable. Another alternative was similar to that which most testers suggested during our initial usability tests, to have a dialog appear where this functionality could be used. This dialog could either appear when you click the save button or when the clips have all been saved. If the input box for the email address was located next to the checkbox where you could choose to share your clips, all in the same dialog, there would be less confusion to how the clips were going to be shared. When we asked the testers if they thought that a user trying to share clips would click the Save button while searching for this functionality, they agreed with our initial reasoning that this might have presented an issue.

Task 2

1. What was your first impression when you got to the media player website?

"Felt familiar, nothing strange at all. Simple and good."

None of the testers felt that their first impression was in any way negative towards the website that they had navigated to. They immediately found the media player to be a familiar web-based media player.

2. Was the media player and the playlist easy to navigate?

"Yes, very intuitive and simple. Hard to make an error."

The testers were all unanimous that the navigation was intuitive since it was the same as on most other media players they had used before.

3. Is there anything else you would like to point out about the website?

"Perhaps add the possibility to split view the shared clips simultaneously."

"Group all clips from the same camera to make it clearer that they come from the same camera."

Since it was possible to view the different cameras' recordings at the same time in the Companion application, some users mentioned that it would be useful to be able to do this on the website as well.

Grouping the clips was a suggestion on how to make the playlist part of the media player more readable. The readability of the clips was something more users felt some concern about. The way that the clips are named right now made them rather cumbersome to read so a solution to that was something they felt would improve the experience overall.

Chapter 8

Discussion

In this chapter we discuss all the topics that are not directly related to our design and development process, as well as project limitations and some design improvement suggestions that were not implemented.

8.1 Legal Aspects

It is very hard to say if it is even possible for this type of solution to be usable in the form of video evidence, at least in this day and age. Different countries and different institutions all have different policies of what can be considered evidence in terms of video and what simply is hearsay. Video integrity plays a major part towards making video recordings admissible as video evidence or not. When the video is presented, there can be absolutely no doubt about the fact that the video is an exact copy of the video that was recorded on the day of the event.

This can be compromised by technical issues such as compressing videos in certain formats which can change just a few pixels in just a few frames. It can also be compromised by people who get their hands on the material and manually edit parts of it. This is prevented using watermarking.

Because of how hard it would be to take all these things into account, we have not spent a lot of time to make sure our solution can replace flash drives in terms of video evidence. There would have to be a lot of investigations into what has to be done in order for cloud services and browser based media streaming, to be considered safe enough to be used as video evidence. We simply decided that this was not in the scope of our thesis.

8.2 Security Related Aspects

To make sure that the sharing service is not abused by anyone, there has to be no doubt that the person sending a clip is a Companion user and that the receiver requesting the shared clip is the intended receiver. The way this is done is by implementing authentication on both parts of the communication. Every time someone tries to upload something to the S3 bucket, we need to authenticate that user as a Companion user. When it comes to the receiver, we need to make sure that any requests to look in the buckets is coming from someone who is an intended receiver. It is not enough to block them from accessing the website since requests to the bucket can be made using other services. This is something that would have to be implemented in order for our solution to be used in a production version of the Companion.

Another thing that have been discussed is having links to shared clips expire after a certain amount of views or a certain amount of time. This could be done by regulating the authentication of the receiver by limiting the amount of requests that can be done to the bucket or letting their authentication expire after a certain time. Since neither authentication nor sessions were necessities for our prototype, we have chosen not to spend time implementing neither, during our thesis.

8.3 Design Suggestions

In this section we will present our suggestions for improvements on the different user interactions that are involved in our thesis.

8.3.1 Export Flow

In the analysis and discussion sub-sections of sections 5.1 and 5.2 we presented some issues that we or the testers found with the export flow. The first issue encountered was accessing the export mode when in the recordings view. This was an issue for two main reasons: the user did not know that there even was an export mode that had to be accessed when in the recordings view, or the user simply did not find the available interactions to access this mode.

The simplest solution for this issue would be to simply include all the added functionality in the export mode into recordings view from the beginning. If the user could see the save button together with the timestamp indicators directly when they navigated to the recordings view there would be no confusion to whether they were in the correct view to export recorded material. What this potentially could lead to is that users would instantly press the save button when they see it which would trigger the export function directly on the automatically selected camera and interval but since the function can be canceled, we feel it would still be an improvement. Another parameter to consider is if additional buttons and pointers to interact with would make the entire view feel harder to understand and more frightening to interact with at first glance by a new user. The discussion about minimalistic design versus added functionality is something that every developer should consider and it is definitely appropriate in this case. However, since the user tests indicated

that the added interactions in the export mode are well integrated into the already existing UI components, it would not cause a great deal of confusion for new users.

Choosing multiple cameras to export from at the same time was a functionality that many of our users failed to discover. One reason for this was that they were unaware that it was even a possibility but mainly it was because of the location of the checkboxes, see figure 5.2 number (4). Since most of the interactions are done in the center of the screen, the users did not find the checkboxes. The users that did find the checkboxes all said that it made sense to have them there since they were part of the timeline where a lot of the interaction is done. It is therefore possible that the best solution would be to keep the checkboxes and add either a tooltip telling the users that they can select multiple cameras with the checkboxes or add another method of interaction to select multiple cameras. It could for instance be an alternative on the menu that pops up when you right click the timeline or you could allow the users to Ctrl-click the cameras in the bottom to highlight multiple cameras for export.

Investigation mode can be accessed by clicking the button next to the save recordings button, figure 5.1 number (3). In this mode, the user can investigate a short period of recorded material by looking at the video frame by frame. The main functionality of this is to take snapshots of specific frames. It is therefore arguable that it is not actually a part of saving recordings and might be better placed somewhere else. A lot of our users clicked on investigation mode when trying to figure out how to save recordings which only led to confusion.

Another thing that a lot of our users did was to click on the time interval showed on 5.2 number (6) in an attempt to change the timeline by editing these timestamps manually. This would probably be the preferred interaction for some users over the pointers since it allows for exact timestamp specifications. Placing the pointer on an exact time can be difficult, especially since many users did not find the timeline's zoom function, so this would be a good addition.

8.3.2 Export Dialog

In figure 6.1, you can see the very first ideas we had towards an export dialog that could be implemented as the primary method for sharing saved recordings. Since we made that lo-fi design, the dialog have always been in our mind the best alternative for our solution. We have throughout our thesis thought of how this dialog could be evolved to hold more functionality which was relevant to saving recordings in the Companion. The reason we were not able to work more with the dialog in our prototype was simply that the extra functionalities we were looking to add with that dialog, was too hard or too time consuming for us to do on our own. We do however want to present what we had in mind for the dialog.

The added functionality that we had in our lo-fi design was that the user would get feedback about which cameras they had chosen and what time interval. The user could also choose whether to save locally, share or do both. Both of these are very reasonable things to add to the dialog. Providing additional feedback about the save job that the user is doing, will provide a sense of security for the user that they have chosen the right cameras and timestamps. This could lead to less users checking the exported material one extra time after the export. Allowing the user to select what they want to do with the chosen recordings will help users to not have to do more actions after exporting like removing

locally saved recordings because all they wanted to do was to share. Both these things would make the export feature much quicker to use overall.

Another functionality that has been discussed, both during our tests and during meetings with other employees of Axis, is the possibility of being able to leave a comment or note with your shared recordings. There have been a couple of different suggestions to how this could be done. The easiest way would be if the export dialog contained an empty text field where the user could simply leave a comment with whatever information they want about the saved recordings. This comment would then be shown in either the mail sent to the receiver, or posted somewhere on the website for the receiver to see. Another version of this would be to let the user see all of the clips that have been created and preview these in the export dialog. The user could then leave a separate comment per clip, specifying what significant events to look for in that specific clip. This implementation would be better in order to minimize the effort required of the recipient, but it would require more from the sender.

It could also be done using manually set timestamps in the saved recordings. If the sender could pick a specific timestamp in the recording, and leave a comment with that timestamp such as "This is when the culprit enters the store", a lot less work would be required from the receiver to search through the material. This would however also require more work from the sender, but it could be added as an optional feature. An example of how this could look on the website for the receiver can be seen on figure 8.1. Another solution to this that is slightly different but still fills the need for a method of sharing information between sender and receiver about the recordings, is opening a channel of communication between sender and receiver. The sender could be given the option to fill in their own email address as well as the address of the receiver which would lead to the sender receiving the email as well. This could then be a platform for the sender and receiver to discuss the shared recordings.

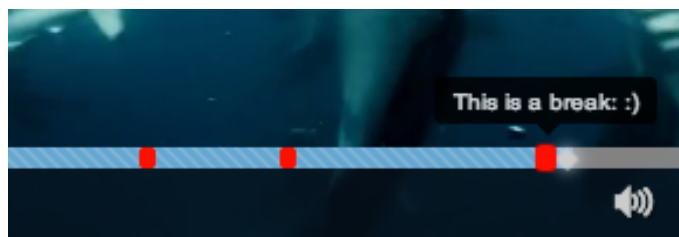


Figure 8.1: An example of how bookmarks on specific timestamps could look.

The dialog could also let the user choose whether they want to watermark their saved recordings or not. Since this is not necessary if the shared material is not meant to be video evidence, it would be a nice addition to let the user choose this if they want to. Another useful feature would be to let the user see an estimation of how large the downloaded material will be, if they choose to save the recordings locally. Also allowing the user to enter multiple email addresses into the email-input box, as well as a way to save some commonly used addresses for future use would be helpful features.

8.3.3 Website Design

During the usability verification tests, some users mentioned that having a split view in the website would be a great feature. This would allow the receiver to more quickly look through the recorded material and find the relevant events.

Also included in the user experience of the receiver is the email sent when the Companion user has shared recorded material. To make the receiver feel safe with clicking the link leading them to the website, the email has to inspire confidence that it is legit. This was something that many of our users mentioned during verification tests as well. The email has to be sent from an address that looks professional and the email itself needs to look professional.

8.4 Limitations

In this section we will first present the known limitations of our prototype and then other limitations we have had during this project.

8.4.1 Prototype Limitations

In order for the solution that we have developed to become fully functional with the Companion, there are a few limitations that needs to be looked over. Some of these are connected to the different aspects previously mentioned in this chapter and will not be discussed further. There are however further technical limitations.

When exporting from the Companion, the user can choose to export in two different formats, one is the Microsoft native format .asf that is not a very common video format, but since Companion is developed with .Net, it makes sense to support the format. The other format is .mp4 which is a video format that has far more widespread use. The video player that is currently being used on our prototype website does not support .asf format. We have therefore implemented it so that when a user that has chosen to export their recordings with .asf format is trying to share their clips, they are informed that only recordings exported with .mp4 format can be shared.

There is however another issue with the formats. One of the cameras that have been used during our development, the Eye mini L camera, can only have its recordings exported in the .asf format. What this means is that even though a user has chosen .mp4 format, if one of the cameras selected to export from, is the mini camera, some of the exported clips will be in the .asf format. Because of this we have implemented it so that whenever a clip with this format is downloaded locally, it is ignored by our cloud export service and subsequently not uploaded to a bucket. It is possible that the user should be informed after exporting that some of the clips could not be shared due to this limitation, but that is currently not implemented.

Another known limitation is when the user enters anything in the email input box that is not a valid address. The input box does not currently check with any sort of regex patterns if the entered value looks like a valid email address. This could rather easily be implemented to prevent some mistakes when entering email addresses. If this was implemented there would however still be a limitation with input addresses that look like a

valid email address, but do not actually exist. All the exported clips would be uploaded as usual and the Lambda would attempt to send the email to the given address. The way this is handled by most email services today is that the address that sends the email receives an automatically generated email that states that the email that you tried to send, could not be sent to the intended receiver. This means that the temporary email that we setup to send the emails to the receiver, is the only place we could notice these types of errors. We currently do not have any solution for that.

8.4.2 Other Limitations

Conducting our thesis at a company that works with sensitive security related products has presented certain challenges. When focusing on the interaction design aspects of a project, you always have to keep the end-user in mind while implementing. Arguably the best way to do this is to interview the end-users to get to know them and also let them test your prototype to ensure that it fits their needs. This was simply not a possibility for us. There is no record of who exactly the end-users of Companion are since information about a company's security system is sensitive and should not be available to just anyone. This meant that we could not come in direct contact with the end-users which is a limitation to our design process.

In an attempt to still retrieve some information about how the end-users uses the export functionality in the Companion, we added it to the analytic service that the client uses. We implemented so that whenever an export job is completed, we track information about how many cameras they exported from, how long the time interval they selected was and how many clips that resulted in. Our plan was to use this information when making certain decisions about how to design our prototype. However, the Companion version to which this code was added did not get delivered to the end-users in time for us gather this information. We did get some information from Companion beta-testers but it was not enough to draw any real conclusions from.

Chapter 9

Conclusion

This master's thesis set out to investigate if a cloud-based video sharing solution could be a viable alternative to the dated local export feature, that not only Companion but also its competing video management systems currently incorporate.

In order to examine this solutions potential, we developed a prototype that was evaluated by conducting cognitive walkthroughs and usability tests. The video sharing functionality itself was very well received. The testers had little negative feedback in regards to this feature, and were especially happy with how smooth and responsive the media player website was where the recipient views the shared video recordings.

We believe that the prototype developed during this thesis not only proves that cloud-based video sharing can be implemented, but that it can also be extremely easy to use and would serve as a useful feature for the CCTV-industry. There are many use cases ranging from sharing recorded material within the company, to sharing material as video evidence for the police to be played in courthouses as video evidence. In almost all possible cases, a cloud-based solution would be more convenient and save both time and money for all parties involved.

The main issues that arose are connected to using the service in terms of video evidence. If the legal protocols that concern video evidence evolved to also include video evidence that is shared via completely secure cloud-based solutions, it is very possible that the type of solution we have developed during our thesis, will completely take over from the solutions available today.

Bibliography

- [1] Axis Communications. *AXIS Companion video management software: Video surveillance made simple.*
- [2] Axis Communications. Axis passerar 10 miljarder i omsättning. <https://www.axis.com/sv-se/newsroom/press-release/axis-passerar-10-miljarder-i-omsattning>.
- [3] Stephen W. Draper Donald A. Norman. *User Centered System Design : New Perspectives on Human-Computer Interaction.* CRC Press, University of California, San Diego, 1986.
- [4] Donald A. Norman. *The Design of Everyday Things.* The Perseus Books Group, New York, United States, 2013.
- [5] Jakob Nielsen, Thomas K. Landauer. A mathematical model of the finding of usability problems. *CHI '93 Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, pages 206–213, 1993.
- [6] Cathleen Wharton, John Rieman, Clayton Lewis, Peter Polson. The cognitive walk-through: A practitioner's guide. In *Usability Inspection Methods*, pages 105–140. John Wiley & Sons, Inc., New York, United States, 1994.
- [7] John Brooke. Sus - a quick and dirty usability scale. *Usability evaluation in industry*, pages 189–194, June 1996.
- [8] Jeff Sauro. *A Practical Guide to the System Usability Scale: Background, Benchmarks, & Best Practices.* CreateSpace Independent Publishing Platform, Denver, Colorado, 2011.
- [9] Aaron Bangor, Philip T Kortum, James T Miller. Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of Usability Studies*, 75(2):114–123, May 2009.

- [10] Cambridge University Press. *Cambridge Business English Dictionary*. Cambridge University Press, New York, United States, 2011.
- [11] Jonathan Kozolchyk. How to prepare for aws's move to its own certificate authority. *AWS Security Blog*, 2018.
- [12] National Security Agency. Commercial national security algorithm suite. <https://apps.nsa.gov/iaarchive/programs/iad-initiatives/cnsa-suite.cfm>.
- [13] U.S. National Archives and Records Administration. Executive order 13526 of december 29, 2009, classified national security information. *CHI '93 Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, 75(2):207, January 2010.
- [14] Robert C. Martin. *Agile Software Development : Principles, Patterns, and Practices*. Harlow : Pearson Education Limited, Upper Saddle River, New Jersey, 2014.
- [15] James Shore, Shane Warden. *The Art of Agile Development*. O'Reilly Media, Sebastopol, California, 2008.
- [16] Amazon. Amazon web services website. <https://aws.amazon.com>.
- [17] Amazon. Amazon s3. https://aws.amazon.com/s3/?nc2=type_a.
- [18] Amazon. Aws lambda. https://aws.amazon.com/lambda/?nc2=type_a.
- [19] Netlify. Netlify documentation. <https://www.netlify.com/docs/>.
- [20] Microsoft. Visual studio 2019. <https://visualstudio.microsoft.com/vs/>.
- [21] Microsoft. What is .net? <https://dotnet.microsoft.com/learn/dotnet/what-is-dotnet>.
- [22] Microsoft. A tour of the c# language. <https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>.
- [23] ReactJS. Tutorial: Intro to react. <https://reactjs.org/tutorial/tutorial.html#what-is-react>.
- [24] Atlassian. What is version control. <https://www.atlassian.com/git/tutorials/what-is-version-control>.
- [25] Atlassian. What is git. <https://www.atlassian.com/git/tutorials/what-is-git>.
- [26] Caitlin Sadowski, Emma Söderberg, Luke Church, Michal Sipko, Alberto Bacchelli. Modern code review: A case study at google. <https://ieeexplore-ieee-org.ludwig.lub.lu.se/document/8449249?arnumber=8449249&SID=EBSCO:edsee>.

- [27] Rick Spencer. *The streamlined cognitive walkthrough method, working around social constraints encountered in a software development company*. ACM, New York, United States, 2000.
- [28] Microsoft. Navigation design basics for uwp apps. <https://docs.microsoft.com/en-us/windows/uwp/design/basics/navigation-basics>.
- [29] IPVM. Vms export shootout - avigilon, dahua, exacq, genetec, hikvision, milestone. <https://ipvm.com/reports/export-shootout>.
- [30] Johana María Rosas Villena, Bruno Costa Ramos, Renata Pontin M. Fortes, Rudinei Goularte. Web videos – concerns about accessibility based on user centered design. *Procedia Computer Science*, 27:481–490, 2014.
- [31] Paul J. Leach, Michael Mealling. *A Universally Unique Identifier (UUID) URN Namespace*. The Internet Society, 2005.

Appendices

Appendix A

Usability test of the export flow (local export only)

A.1 Test Scenario Manuscript

Test scenario - Axis Companion Client: export recording feature (local)

Axis Companion is a video management solution that can be used to monitor your surveillance system at home or at your workplace. Your cameras and their recorders make what we call sites and we have already prepared a site for you. Your task today is to enter the site we have created called "TestSite", password is "abc" and save some recordings onto your computer. Somewhere between 12:50 and 13:00 today an event has taken place (the event is that Petter is holding up a paper) and we want you to save a recording of that event as well as what happened 30 seconds before and 30 seconds after. Not all cameras on the site have captured the event, and we only want you to save recordings from the cameras that actually captured what happened. Good luck!

6. I found the various functions in the export recording workflow were well integrated. *

	1	2	3	4	5	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

7. I thought there was too much inconsistency in the export recording feature. *

	1	2	3	4	5	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

8. I would imagine that most people would learn to use the export recording feature very quickly. *

	1	2	3	4	5	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

9. I found the export recording feature very cumbersome to use. *

	1	2	3	4	5	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

10. I felt very confident using the export recording feature. *

	1	2	3	4	5	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

11. I needed to learn a lot of things before I could get going with the export recording feature. *

	1	2	3	4	5	
Strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly agree

A.3 Interview Questions

Interview Questions:
ACC export recording feature (local)

1. Did navigating the menus feel intuitive?
2. What was your first impression when you got to the recordings view?
3. Did any of the tools or alternatives of this view make you confused?
4. Was it difficult to figure out how to get into "export mode"?
5. Did you find that using the timeline for previewing and deciding time interval was intuitive?
6. Is there anything else about the exercise you just performed that you want to share?
7. If there was added functionality to also share the saved recordings with someone via email, how would you have wanted that functionality to be added into this workflow? (Via a dialog, via extra alternatives next to the save button, an entirely different view etc.)

Appendix

B

Verification test: export and share

B.1 Test

Tasks

Final Usability Test

Task 1: Axis Companion is a video management solution that can be used to monitor your surveillance system at home or at your workplace. Your cameras and their recorders make what we call sites and we have already prepared a site for you. Your first task today is to save some recordings onto your computer and share these recordings with this email address: "abc@hotmail.com", using the built in functionality in the Axis Companion. To get you started we have navigated to the recordings view where the saving is done. Somewhere between 13:44 and 13:47, an event has taken place (the event is that Petter is holding up a paper) and we want you to save a recording of that event as well as what happened roughly 30 seconds before and roughly 30 seconds after. Not all cameras on the site have captured the event, and we only want you to save recordings from the cameras that actually captured what happened. Good luck!

Task 2: The email account "abc@hotmail.com" is logged in on Google Chrome on this computer. Open the email that just now have been sent from an address called "pythonpk@gmail.com". Your task now is to view the recordings that have been shared. It will suffice to simply click on each of the clips and let them start, you don't have to look through the entire clips. Good luck!

B.2 Interview

Questions

Interview Questions: Companion export and share recording feature

Task 1:

1. Was the sharing functionality easy to find?
2. Was it clear what you were doing when you checked the checkbox next to "share saved recordings"?
3. Was there anything confusing about the dialog window that appeared?
4. What did you think of the feedback you received whilst the recordings were downloading and after they were done?
5. Do you think the general user would feel comfortable using the sharing functionality, even if it involves sharing sensitive video material?
6. Do you think the sharing functionality is well integrated with saving recordings in the Companion? Would you have preferred a different approach?

Task 2:

1. What was your first impression when you got to the media player website?
2. Was the media player and the playlist easy to navigate?
3. Is there anything else you would like to point out about the website?