Video approval app for iPad

Andreas Jakobsson

May 10, 2011
Master’s Thesis in Computing Science, 30 credits
Supervisor at CS-UmU: Håkan Gulliksson
Examiner: Jerry Eriksson

UMEÅ UNIVERSITY
DEPARTMENT OF COMPUTING SCIENCE
SE-901 87 UMEÅ
SWEDEN
Abstract

The goal of this master thesis project was to design a concept for an iPad app that solves the problem of collecting feedback around a video clip in a video production. The concept targeted an end user which used the mobility of the iPad to give feedback on video clips from wherever, collaborating with other people involved in the video production. A background study was performed covering mobile app development, guidelines for designing mobile systems and guidelines for designing collaborative systems. With data gathered from interviews with people working with video, a design process including ideation, sketching, interface mockups and interactive mockups iterated the concept into a proposal for a video commenting tool for iPad. The app lets the user add comments onto the timeline with text, drawings and voice recordings as well as keeping him or her up to date with the discussions around the video clip.

The result of the project can be used as a reference document for further development.
# Contents

1 Introduction 1

2 Problem Description 3
   2.1 Problem Statement .......................... 3
   2.2 Goals ................................... 4
   2.3 Limitations ................................. 4
   2.4 Outline .................................. 5

3 Method 7
   3.1 Design Process .............................. 7
   3.2 Background and Data gathering .............. 8
   3.3 Sketching and Prototyping ..................... 8
   3.4 Evaluation ................................ 9

4 Background 11
   4.1 Smartphones and Tablets ..................... 11
   4.2 Native Applications vs. Web Applications .... 12
      4.2.1 Web Programming ......................... 14
      4.2.2 Discussion and chosen framework .......... 16
   4.3 Collaboration and Mobile Work ............... 18
      4.3.1 Guidelines for collaboration and mobile work 19
   4.4 Designing for the iPad - Design Guidelines .... 20
      4.4.1 General mobile interface guidelines .......... 23
   4.5 Conclusions from Background Studie .......... 24
      4.5.1 Smartphones and Tablets .................. 25
      4.5.2 Collaboration and Mobile Work ............... 25
      4.5.3 Design Guidelines ........................ 26
5 Data Gathering

5.1 Digital Video Production and Metadata . . . . . . . . . . . . . 27
  5.1.1 MAM and Vidispine . . . . . . . . . . . . . . . . . . . . . 28
5.2 Interviews . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29
  5.2.1 Interview with Erik Lövgren, Dimension Magic . . . . . . 29
  5.2.2 Interview with Swedish TV4 . . . . . . . . . . . . . . . . 30
  5.2.3 Meetings with Parham Azimi, Cantemo . . . . . . . . . . 32
  5.2.4 Interview with Dan Wallis . . . . . . . . . . . . . . . . . 33
5.3 Conclusions from Data Gathering . . . . . . . . . . . . . . . . . 36
  5.3.1 What iPad app can help the reporter in the field? . . . . 36
  5.3.2 What iPad app can help in the making of a documentary? 36
  5.3.3 What iPad app can help in the making of commercials . 36
  5.3.4 The chosen scenario . . . . . . . . . . . . . . . . . . . . . . 37
  5.3.5 The chosen end user . . . . . . . . . . . . . . . . . . . . . 37

6 Design Process

6.1 Ideation and Sketching Phase . . . . . . . . . . . . . . . . . . . 39
  6.1.1 Precision scrubbing . . . . . . . . . . . . . . . . . . . . . 40
  6.1.2 Dragging in and out points . . . . . . . . . . . . . . . . 42
  6.1.3 Representation of current comments . . . . . . . . . . . . 42
  6.1.4 Orientation dependant layout . . . . . . . . . . . . . . . . 43
  6.1.5 Drawings . . . . . . . . . . . . . . . . . . . . . . . . . . . 45
  6.1.6 Voice recording . . . . . . . . . . . . . . . . . . . . . . . 45
  6.1.7 Thumbnails as support in browsing . . . . . . . . . . . . 45
  6.1.8 Ways of minimizing text input . . . . . . . . . . . . . . . 46
  6.1.9 Collaborative work . . . . . . . . . . . . . . . . . . . . . 46
  6.1.10 Evaluation of Ideas and discussion . . . . . . . . . . . . 47
  6.1.11 Conclusions from Ideation and Sketching Phase . . . . . 47
6.2 Mockup Phase . . . . . . . . . . . . . . . . . . . . . . . . . . . . 50
  6.2.1 Evaluation of Mockups . . . . . . . . . . . . . . . . . . . 57
  6.2.2 Conclusions from Mockup Phase . . . . . . . . . . . . . 59
6.3 Implementation of Prototype . . . . . . . . . . . . . . . . . . . 60
  6.3.1 State of the implementon . . . . . . . . . . . . . . . . . . 64
  6.3.2 Conclusions from Implementation phase . . . . . . . . . 67

7 Conclusions

7.1 Problem in the Workflow . . . . . . . . . . . . . . . . . . . . . 69
7.2 Core functions . . . . . . . . . . . . . . . . . . . . . . . . . . . 70
CONTENTS

7.3 User interface for the iPad ........................................ 70
7.4 Lessons learned and future work .................................. 70

8 Acknowledgements ....................................................... 73

References ........................................................................ 75
Chapter 1

Introduction

"To have a good idea, you must first have lots of ideas."
- Linus Pauling

Technological development is an ongoing sprint towards progress. We have seen an digital revolution in the last years with faster and more capable computers, the evolution of the Internet and in recent years also more capable mobile devices continue to change how we live and act.

At the same time, the working climate of today tends to strive for optimization and efficiency. Often software systems are developed to get rid of extra steps and speed up the process if possible. With the demand for quick and efficient task completion, people sometimes need to work on the go - with laptops, modern smartphones and now with tablet computers.

When looking at video production as an example, there is a growing need to produce video faster and reach target devices as fast as possible. Technology is embraced by the video industry and cameras and other equipment are upgraded in a fast pace. Coming from a business where tapes have been switched for flash drives and hard drives, new systems for handling the massive amount of video data and leveraging the benefits of digital files and Internet connectivity have been developed.

With this in mind, this master thesis project will examine the realm of digital video production and how a software system designed to handle large amounts of media material can work together with a tablet computer. The problem is stated as followes - How can the Apple iPad help in the video production workflow making users more mobile and efficient?

The project is made in cooperation with CodeMill AB, a software development company based in Umeå, Sweden. CodeMill has been a key partner in the development of a powerful Media Asset Management system called Vidispine owned by Vidispine AB. With Vidispine, all stored digital media is made available and manageable from wherever in whatever format you need.
The stakeholder’s interest in this thesis is as an investigation of the possibilities and challenges when leveraging the “always available” capabilities into an application specially designed for a mobile device such as the iPad.

The author of this thesis is a student from the Master of Science in Interaction Technology and Design program and the work is an interaction design project that resulted in a concept for an iPad application for giving feedback on segments of short video clips.
Chapter 2

Problem Description

With the introduction of the iPad, the type of mobile device known as a tablet had a commercial breakthrough. More like a netbook computer then a phone, but with a form factor more suitable for quick interaction and consumption of content.

But, just putting an iPad in the hands of people will not automatically make them more efficient as compared to when using a laptop or smartphone. With the application area in mind, video production, what are the uses where the iPad can come in handy? We have yet to see all the ways people adopt the use of tablets as tools for work and entertainment and one part of the problem is figuring out how the iPad fits into the chosen application area.

2.1 Problem Statement

For this project the stakeholders initially only had a loose idea around the problem the application could solve, namely using the iPad to enter metadata for a video, both for the whole and also for parts of the video. The general guideline though was to research where the iPad would do most good in the video production workflow with the Vidispine system as a backbone.

Since the initial research question for the thesis was somewhat loosely stated much work with the project was done iteratively with increasingly specified goals and limitations. This influenced the work process quite a lot and led to that goals changed over time. But, the overall goal of the project kept, it was to develop a concept for a video application using gathered information and user studies and in the end suggest a possible design for this thought up application.
2.2 Goals

In the beginning of the project the goal was to identify a good application area and a problem that could be solved with an iPad app. Some time into the project a interesting problem area was identified after an interview with a video business professional. Namely, collecting feedback from colleagues, experts or stakeholders on video clips during a video production project. The goal was then set to design a concept for collecting this feedback in an iPad application.

The following sub goals were stated:

Collect information about the video production workflow and state reasonable assumptions about the specific challenges in the field.

Specify core functions for an video annotation tool used to collect feedback about a video clip.

Create a user interface optimized for the iPad - What are the guidelines and best practices for tablet design in general and for iPad in particular? How can these be used in this particular application?

2.3 Limitations

The scope of the project included an overview of video production, user studies, and design of the user interface of an application. Some limitations were stated for the project to constrain it:

- The design of the concept was limited to a number of prototypes in different development stages, further discussed later under the Method section, with the highest level of detail being a digital prototype with interaction possibilities for the key concepts of the design. This means there was never any intention for implementing a complete application. The prototypes are just meant to demonstrate the basic ideas.

Some technical demands and limitations were stated also for the project:

- The concept should be based on the capabilities of the Vidispine system. This means that even if the Vidispine system is not connected to any of the prototypes developed, the constraints passed by Vidispine should be considered.

- The target device should be the iPad. In a real life scenario, the system would probably have to function on a number of devices reaching from desktop browsers to smartphones and to other tablets. This is not taken into consideration in the project.

- Although the target device is an iPad, since the software should be accessible from multiple devices, web technologies should be used and the implications of this should be considered.
2.4 Outline

Chapter 3 - Method
In this chapter the different methods used during the project are presented.

Chapter 4 - Background
Here the background study is presented. Development for mobile devices, mobile and collaborative work and basics about video production are covered.

Chapter 5 - Data Gathering
During the project both existing software relevant to the application were examined and interviews were conducted to collect data. This data was later on used to make design decisions. The collected data are presented in this section.

Chapter 4.4 - Designing for iPad - Design Guidelines
The official human interface design guidelines for iOS and iPad [43] were studied as well as existing iPad user interfaces. The main findings are presented in this section.

Chapter 6 - Design Process
This chapter presents the design process for the project with its different phases.

Ideation and Sketching Phase - This section contains the work process of the early design stage of the project.

Mockup Phase - This section contains the work process from the mid level stage of the design process. Also, much of the comments and feedback gathered came from this stage and are described.

Implementation of Prototype - This is the last stage of the design process and also the last stage of the project. The JavaScript framework Sencha Touch was tried in the last stage of prototype development. The framework is explained in short and evaluated in relation to a possible productification of the concept.

Chapter 7 - Conclusions
Finally, the goals are revisited and discussed based on the results of the project.
Chapter 3

Method

As mentioned earlier this project was conducted with the goal of creating a concept for an application. A design process with different stages of ideas and designs coupled with evaluations was planned. In this section the different parts are introduced.

A plan was developed in the beginning of the project based on the author’s experience in the design process. The plan was organized as a 20 week plan containing the following basic steps:

– Formulate goals together with stakeholders
– Pre-study of the field and interviews with practitioners
– Ideation and Sketching
– Mockup design
– Evaluation and redesign
– Implementation
– Writing of thesis

The basic structure of the plan was followed but the pre-study and interviews ran in parallel throughout most of the project. In the end of the project this thesis was written alongside the last stage of the prototype work. The planned evaluation methodology and implementaion goal changed due to time limitations. This will be discussed later.

3.1 Design Process

The core of interaction design is the focus on the interaction between a user and a system. To design something useful it is essential to understand the situation
and the user’s mindset, motivation and mental model. Therefore the design process focused on iterative research and ideation - trying ideas and concepts through different stages.

### 3.2 Background and Data gathering

The goal of the data gathering was to get a general idea about what video production was all about as well as understanding a potential user. In the beginning, popular video editing software were reviewed and articles about video production were read. Later on, interviews were performed with different professionals.

A literature study around mobile app development, tablet design, collaboration and mobile work was also done with the common goal of getting inspiration and ideas for the design.

### 3.3 Sketching and Prototyping

When generating ideas and trying them out, sketching with pen and paper offers fast iterations and it can help to look at concepts instead of details. Therefore the early ideas were discussed with other people through simple sketches, keeping the focus away from graphical details.

As the project progressed the ideas were transferred to more detailed ideas using lofi- and hifi prototypes. With lofi prototypes the goal was to demonstrate the flow and basic ideas behind its concept and structure with the hope that many of usability issues are addressed early on in the process while it is still cheap to fix the problems (takes less time to make the changes). Vizri et al. have showed the power of lofi prototyping by comparing the number of usability issues found with a lofi paper prototype and a hifi prototype product with the result being that the two prototypes exposed almost the same amount of issues. When considering the development cost between the two, lofi prototyping is an effective method. [51]

Lumsden and MacLean go further and examine lofi prototyping as a method of usability evaluation for mobile development in particular. The initial study compared paper prototypes with so called pseudo paper prototypes which are paper based prototypes scanned into a computer and made clickable with a PowerPoint software and then showed on a tablet during testing in a more natural use context. The pseudo paper prototypes exposed more unique usability issues and the authors argue that the issues would be more valid because of the context affecting the use, but they could not show this with statistical significance.

A final point is that paper lofi prototypes have the quality of making the user more comfortable with criticizing the system since it does not look like a final design. There is a risk that a person testing a prototype with a lot of detail and polish holds back, consciously or unconsciously, because of the work
3.4 Evaluation

Since this was a project with interaction design focus, evaluation was important. Planned methods were end user interviews with the think-out-loud method and also heuristic evaluations in the final design or designs. The heuristic evaluation was planned to be performed by the author himself because of his understanding gained through his education. In the end, time forced the heuristic evaluation to be an ongoing parallel process during the design phase which is not optimal. The designer of a system can get blind for issues that first time users can experience and testing should be done with pre stated heuristics to force the designer/tester to really look beyond the initial experience.
Chapter 4

Background

The project included a background study in development approaches and design guidelines for smartphones and tablets as well as guidelines for collaboration systems. This chapter presents the findings that are the basis for this project.

4.1 Smartphones and Tablets

When Martin Cooper from Motorola first demonstrated a mobile telephone in 1973 its weight was two kilos and it was extremely expensive. Since then the number of mobile phones have grown to the point were half of the worlds population own one and you can get phones for free with some deals [49]. According to Gartner [48], [50] we have seen a steady increase of sold mobile terminals with sales figures for the third quarter 2007 being 289 million new mobile units worldwide rising to 417 million new units in the third quarter of 2010. Also, the growing number of the so called smartphones have reached a share of 19.3 percent of the overall mobile phone sales in the third quarter of 2010.

There are many definitions of what a smartphone is but in the Gartner report it is defined as a communication device based on an open operating system, allowing the user to easily install applications [50]. Others also add the criteria of having a touchscreen.

Smartphones have been around for some years but with the release of the iPhone in 2007 [42] there was a boom for ordinary consumers. With its App Store users could easily browse and install new apps directly on the phone or through iTunes. With a new release every year since then, the current model iPhone 4 has a 3.5 inch multitouch display with a resolution of 326 dpi which is high with today’s standards [10]. While Symbian is the highest selling smartphone operating system with 36.6% of the smartphone market with all the different variations included, iOS have taken a 16.7% part during its 3 years and in only 2 years the Android operating system has taken a 25.5% cut [50].
Chapter 4. Background

The point here is that smartphones are evolving and we are starting to see new functions, user interface elements and interface metaphores that are widely recognized by the users.

Along with the smartphone there is another interesting mobile device, namely the tablet. With a form factor in between the pocket sized smartphone and a small portable netbook computer it has been around for several years but has not yet had a commercial breakthrough among general consumers. With the Apple iPad released in 2010 [41] and other tablets running the Android operating system there was somewhat of a breakthrough. Gartner, a company working with mobile market analysis, defines a media tablet as a slate device with a touchscreen that runs a lightweight OS such as iOS, Android, WebOS or Meego. Gartner also forecasted the total worldwide tablet sales to reach 19.5 million units in 2010 and projects the numbers to increase to 54.8 million in 2011 and be as high as 208 million units in 2014 [26].

4.2 Native Applications vs. Web Applications

With the iPhone and the App Store model, smartphones became more common among private consumers and application (app) development have had a boom in the last three years. Apple announced that the App Store’s 350 000 apps had together been downloaded 10 billion times by the 22 january of 2011 [4]. Android market which is the biggest contendant is growing rapidly. No official statistics are given but the website AppBrain.com keeps its own statistics and puts the number of apps in Android Market at around 150 000 in March 2011 [2].

To develop an app for the iPhone, iPad and iPod Touch for publication in the App Store you need to use the language Objective C. With the iOS SDK developers can make use of the device’s hardware such as the camera, accelerometer and gyroscope as well as data such as contacts, pictures, music etc. The iOS has different versions of the SDK for the iPhone/iPod Touch and the iPad.

But, there are other ways to reach mobile users apart from using native applications. Todays smartphones are shipped with a powerful browser that can render webpages with results similar to a desktop experience but also with the ability to zoom in and out to fit the content on the screen, instead of as the earlier versions of mobile browsers that rendered scaled down websites. The browser in iOS is based on the WebKit browser engine which is also used in some applications running on Mac OS X, the Safari browser, the Google Chrome browser and the Android browser [?]. The WebKit engine is capable of handling much of the modern web standards that are emerging with the HTML5 draft [5].

With the powerful browser it is possible to run so called ”web apps”. They are seen as the equivalent to the native app but are developed using web technologies such as HTML, CSS and Javascript and are delivered over the Internet,
meaning that they are not installed on the device but simply visited with the browser. Apple is open to the use of web apps and has documentation available to help developers to develop iOS adapted websites or iOS adapted web apps [9].

**Pros and Cons**

Jonathan Stark, author of the book *Building iPhone Apps with HTML, CSS, and JavaScript*, writes about his bad experience with iPhone development and the App Store in the preface of his book. The book is dedicated to teach how to use web technologies and he lists some pros and cons for going native or with a web app. Stark lists the following [47]:

**Positive aspects of native** - When going native, you get the benefits of having millions of users with registered credit cards that can easily buy your application. Also, you can use the development tools provided by Apple in the form of Xcode (code editor), Interface Builder (graphical interface builder with drag and drop). Finally you get access to much of the hardware features of the device such as the camera, microphone, gyroscope etcetera.

**Negative aspects of native** - The down side of native applications are that you have to pay to become an Apple developer, Apple has to approve your application which can take between a couple of days to several weeks and whenever you fix bugs or add features the approval process must be run again. Also, you have to learn Objective-C and do the development on a Mac computer.

**Positive aspects of web** - With web application development, you have the benefits of doing your development in any text editor on any operating system and the result can run on any device with a web browser. Since web applications do not need to be submitted to the App Store you also cut out the approval process which leads to no registration cost and updates being available instantly.

**Negative aspects of web** - The down side of web applications is that you have less access to hardware features so depending on what your application should do, sometimes a web application can not manage it. Furthermore, marketing and selling an application demands your own payment system.

Critical voices can be heard over the Internet about the app hysteria. One example is from blogger Mark Suster, a developer and venture capitalist, that has experience with working with mobile carriers and their mobile portals back when Wap was first introduced. He compares that situation of being forced to go through a process of being approved by carriers to show up in there portals with todays situation with Apples App Store approval process and censure.
Also, when programming for native platforms you may need to make one version for iOS, one for Android, one for Blackbeery and one for Windows Mobile to cover the market. He argues that it is not sustainable to have developers doing apps for the different platforms when you have an opportunity to resolve the platform differences with one web app [3].

4.2.1 Web Programming

But how do you develop a web app? You can try to mimic the look of a native app by coding your own HTML/CSS from the ground up, but it is time consuming and it is not trivial to recreate the expected behaviour and animations for the different platforms. Therefore different frameworks have been developed to prevent the recreating of the wheel for every project. In this project a study was made of some different possible alternatives.

When programming for the web the usual language is Javascript. It is a scripting language supported by all modern web browsers and with Javascript frameworks like the popular open source jQuery there are plenty of helper functions that decrease the need to program common actions from the ground up [13]. When using jQuery the developer can use methods for traversing the HTML-dom, selecting and manipulating elements and also animate changes on the site with very little code. With jQuery UI there are also complete user interface components for common tasks such as displaying time pickers, dialog windows, buttons, lists as well as support for interactions such as drag, sort, resize etcetera [16]. Similar frameworks targeted especially to mobile application development also exist. Some have connections to jQuery but there are also different approaches.

**jQTouch (beta)** - jQTouch is a plugin for jQuery that focuses on programming animations, themes and event handling for WebKit based phones such as iPhone, Nexus One and Palm Pre. The project is said to work on bigger devices such as iPad but it is not the main focus and there is no effort to automatically adjust the interface to bigger screens. The plugin was developed by David Kaneda and is now maintained by Jonathan Stark, the author of ”Building iPhone Apps with HTML, CSS and JavaScript”, and is in beta version 2. The plugin is 12 kb, jQuery is 57 kb in minified versions and they are available for free [12] [11].

**jQuery Mobile (alpha)** - The jQuery project provide a version optimized for touchscreen devices. It is based on the jQuery core and uses the same type of syntax. As of today the framework is in alpha version 4 and supports iOS, Android, Blackberry, Palm WebOS, Nokia/Symbian, Windows Mobile, bada, MeeGo. Their goal is to make it possible to design and develop a single web app that works on all major mobile browsers with the project mantra ”write less, do more”. There are components like lists, dialogs, buttons and forms and the whole framework is made to be easily themed either by choosing between shipped themes or providing your own custom
4.2. Native Applications vs. Web Applications

theme in CSS. The framework size is 12 kb in its minimized version and is available for free [14]. While the previous mentioned alternative jQtouch is a plugin that supports some modern WebKit browsers jQuery Mobile aims to provide a consistent experience also over less capable browsers. They use a progressive enhancement technique[19] meaning that the web page first is made with semantic and standard compliant HTML so that as many devices as possible can read the content. Then more modern techniques such as CSS for styling and animations as well as advanced JavaScript are added for devices that can handle them. [15]

Sencha Touch - On June 14th 2010 Ext JS announced that they would change their company name to Sencha and at the same time begin collaborating with the previously mentioned jQtouch [20]. A few days later on the 17th of June Sencha Touch was released in a beta version [23]. While jQtouch still remains the same in a foundation called Sencha Labs, Sencha Touch is a separate framework for web apps using modern HTML, CSS and JavaScript techniques. Both jQtouch and jQuery Mobile use an approach were you develop ordinary web pages with extra classes and JavaScript functions to add look and feel but Sencha Touch has another approach. In a blog post by David Kaneda, the creator of jQtouch and now lead designer for Sencha Touch, he describes how there is a place for both projects with jQtouch being easier for web developers (used to web technologies) and Sencha Touch being more suitable for software developers (used to more classical programming languages). Sencha Touch uses JavaScript to work with the Sencha Touch API which in turn build the HTML DOM upon execution, therefore you do not use HTML to do the actual development. The framework is about 120 kb minified and is available for free. Support is available through a forum but Sencha also sells premium support packages [22].

The framework has rich support for streaming, resolution independent user interface components, animations and event handling along with support for a Model-View-Controller (MVC) design pattern[33]. With a MVC pattern there is a separation between the data model which describes and holds the data, the views that present the data to the user, and the controller that reacts to the users interaction and updates the views and models accordingly.

Phonegap - With Phonegap there is an additional approach for using web technologies in apps. First you develop a web app using HTML, CSS and JavaScript in whatever environment you want. Then Phonegap is used as a wrapper to make code run on a device as a native app. This means that Sencha Touch, jQuery Mobile or jQtouch can be used to develop a web app that is later wrapped and compiled with Phonegap as an iPhone app and by that opening up the opportunity to upload it to the App Store. Phonegap also has an API providing access to the device’s native
capabilities such as camera, compass, contacts, files, geo location and notifications so that the web app can bridge some of the limitations of not being a native app. The framework is available for free together with a cloud based service for building a project for multiple platforms all at once. The cloud based building service will not be free for commercial projects [18].

Titanium The company Appcelerator provides a development platform called Titanium. It comes either as an IDE or as a separate API. Just as Sencha they offer tools to develop both for desktop and mobile. Titanium Mobile is the API for mobile apps. As with Phonegap, Titanium Mobile can be used to wrap web technologies in native iOS code or native Android code and can then be published in the App Store or on the Android Market. In versions prior to 1.0 Titanium Mobile was quite similar to Phonegap in the sense that you used HTML, CSS and JavaScript to build the app inside a so called WebView - a native component for rendering HTML. Then the provided API was used to make use of native capabilities such as camera, compass, accelerometer. From version 1.0 it differs from Phonegap through skipping the WebView as the main component and instead having the developer program pure JavaScript that through the API render native components. This means that Titanium can use the JavaScript code to instantiate either iOS or Android components, each one using their own native components [28] [27] [29].

4.2.2 Discussion and chosen framework

The frameworks examined span from being jQuery plugins to applications using JavaScript to instantiate native components. I have personally used HTML and CSS together with basic JavaScript through jQuery to make web pages,
and native Android code for Android apps before the project, but have no experience with coding Objective C. Since CodeMill wanted the project to investigate possibilities for using web technologies to create an iPad app it was decided that a framework would be used as a basis for the implementation of concepts.

The following are my impressions of the frameworks after reading the company web sites and user impressions about the platforms:

**jQTouch** - The jQuery plugin is in a beta and seems to give a good basis to work on. Since the focus is on iPhone and not on iPad there are no special support for tablet size displays. Although it would be possible to use it anyway it did not seem like the right way to go in this project.

**jQuery Mobile** - Supported by the jQuery community and with project sponsors like Mozilla, Palm, Blackberry, Nokia, Adobe and dotMobi this system is likely to become a successful project even if the current version is in alpha. The project has the ambitious goal to provide a common platform across all the major mobile browsers instead of jQTouch that focuses on just iPhone and Android. With the announcement in August 2010 the development has already surpassed jQTouch in the number of UI elements and styles. If standard HTML should be the basis for the app so that web developers are more at home or if the web app should work on less capable devices this would be the way to go in my eyes.

**Phonegap** - While this system is not offering user interface elements it is supposed to be used together with ordinary web technologies. Therefore it should be looked at as a tool to bridge the gap between web apps and native apps. Both jQuery Mobile and Sencha Touch writes about how to use Phonegap to do this using their framework.

**Titanium Mobile** - Titanium Mobile is also more of a bridge to native capabilities. But with Titanium you can actually instantiate native elements which mean that if the goal is to have as high similarity to the real thing this could be a good choice. However, only Android and iOS are supported as of now. It would be interesting to look into how well the look and feel translates between the different platforms but this framework/IDE was not selected as the target framework since it seemed to have less examples to learn from and less of a community.

**Sencha Touch** - Compared to jQuery Mobile and jQTouch, Sencha Touch is more of a software development approach then a web development approach since you use JavaScript towards the Sencha Touch API to generate the HTML, CSS and behaviour. This framework seemed to be the best fit for this project since it offered components that can be found on the iPad such as the Popdown and modal layovers with look and feel much like the native counterpart with minimal code. The examples and developer community also seemed to be more helpful then that of
Titanium. More details about Sencha Touch will be discussed later in the paper.

4.3 Collaboration and Mobile Work

Supporting collaboration with technology goes under the term Computer Supported Cooperative Work (CSCW) [39] and is the basis of many papers about collaboration. There are several aspects described in the research that are important for making collaboration smooth and efficient.

Dourish and Belloti write about Awareness in their paper about a collaborative writing software. Awareness is "an understanding of the activities of others, which provides a context for your own activity. This context is used to ensure that individual contributions are relevant to the group’s activity as a whole, and to evaluate individual actions with respect to group goals and progress" [36]. In other words, awareness is knowing about what other people are doing, what am I supposed to do and how will that benefit the common goal.

Another aspect is described by Dourish. "Appropriation is the way in which technologies are adopted, adapted and incorporated into working practice." [35].

Dourish provides three design principles for supporting the Appropriation of a software. They come from experiences with the “Placeless” document management system which uses a document properties model where documents are assigned private, public, active or passive properties. These properties are metadata associated to documents and the discussions about them are interesting for this thesis project. The design principles are [35]:

**Supporting multiple perspectives on information** - With a properties model it is possible to present documents in different ways in different applications or modes with the original content staying the same. Different people can have different perspectives and different goals when working with the documents and this should be supported by providing different views/applications for different people.

**Preserve visibility** - The system should be designed so that the user not only understands what they can do but also what the consequences will be to the system as a whole. This is similar to the Awareness described earlier.

**Make information sharing an application matter** - Rather than information being an infrastructure matter, the user should be in control of what is shared and how it is shared and by that making the sharing a part of the application. If information is part of the application instead of part of the infrastructure it is more likely that the functionality is in line with the users activity at the time.

The notion of *Articulation work* is another aspect of collaboration.
"Articulation consists of all the tasks needed to coordinate a particular task, including scheduling subtasks, recovering from errors, and assembling resources" [38]

A complex task is hard to specify into the last little detail so during a collaboration there is a need to split tasks into subtasks, distribute tasks between resources and then merge results into the overall project result.

### 4.3.1 Guidelines for collaboration and mobile work

Herskovic et al present a framework for general requirements to support mobile collaboration [40]. The authors take so called Mobile Shared Workspaces (MSW) as examples and focus on what the system architecture should support. The guidelines also tell something about what the user’s needs are. Here are a summary of the guidelines but with an interpretation placing the end user as the point of interest when possible:

**Autonomy** - The user should be able to work autonomously throughout the system when being mobile. In other words, users should not rely on collaboration since connectivity and rapid change of context can easily interrupt the tasks. They should be able to perform work by themselves in first hand.

**Flexibility** - The user should be able to choose the level of involvement in a collaborative group, moving between online and offline.

**Consistency and Availability** - This is more of a challenge for the system, keeping the data produced by the participants consistent and available between the users when they at any time can switch from online to offline. If we look on this from the users perspective, a typical question from a user would be - when was this information last updated?

**Connectivity** - Another challenge for the system and the infrastructure is connectivity. The guideline suggest forming Ad-hoc networks when possible to increase the interaction capabilities among the participants.

**Heterogeneity and Interoperability** - The user should be able to participate in the work regardless of the device used. Different versions of the system should be available to leverage hardware capabilities for those who have them but still let lesser capable devices be a part of the process.

**Communication** - The user should be able to communicate with other participants using messages, notes, documents and alarms. It should be assumed that users are not available at all times so mechanisms for asynchronous communication should be present as well as mechanisms for tracking whether users received messages or not.
**Awareness** - The user should understand what is being done by the participants. The authors give examples of what the mechanics could be. When offline, a user sees a document’s last known modification date and authorship. When online, a user sees a list of connected users and their location.

**Protection** - A user should not be able to destroy another user’s work by accident and a user should be able to protect information by setting different authorizations for other users or user groups. This is similar to when Dourish talks about making sharing something the user is in control of.

### 4.4 Designing for the iPad - Design Guidelines

Apple provides design guidelines for mobile app development in the form of the document *iOS Human Interface Guidelines* [43]. What was originally a split document for iPhone and iPad became a combined document for the iOS. The purpose of the document is to guide developers towards designing a good app by explaining how mobile design differs from desktop design, specifics of the iOS system, human interface principles, user experience guidelines and also design strategies and use cases.

The specific sections for iPad in the user experience guidelines can be summed up as:

**"Enhance Interactivity (Don’t Just Add Features)"** - With the bigger screen compared to the iPhone it is tempting to add more features. More features are not automatically improving the user experience. "Feature creep" is a term used for this and means that more features are added as development progresses. This can lead to a worse user experience as well as causing delays in the development.

**"Reduce Full-Screen Transitions"** - With the iPhone it is customary to replace the whole window while navigating, using the deck of cards metaphor. With the iPad you should keep this behaviour to a minimum and try to animate only the specific part of information you want to update since a lot of movement can draw the focus away.

**"Restrain Your Information Hierarchy"** - Do not give the user the feeling of having to visit many screens to find what they want. They mention using a "Split View" or a "Popover" to present information relevant to the main content, see figure 4.2 and figure 4.3

Other ways of creating hierarchy is to use Tab Panels or segmented buttons to jump between views.

**"Consider Using Popovers for SomeModal Tasks"** - The documentation covers different situations and suggests when to use a Popover instead
4.4. Designing for the iPad - Design Guidelines

Figure 4.2: Example of Popover from Apple’s guidelines

Figure 4.3: Example of Split View from Apple’s guidelines
of a modal windows. A modal window typically dims the rest of the interface and pulls more of the user’s focus away from the content behind it, compared to a Popover.

"Migrate Toolbar Content to the Top" - With the increased screen real estate toolbars and control buttons can fit in a top toolbar for better organization and more space for the main content.

The guidelines above were aimed towards the iPad but there are a lot of other interesting guidelines in the document. Another reviewed in the project was:

"Consider Adding Physicality and Realism" - It is not a bold statement that Apple produces software that look well polished. In the document it is suggested that you can add physicality and realism where it is appropriate because it can tell the user much about how the app can be used. In their own address book app they use graphics mimicking a real book with pages that the user can flip and tabs as quick navigation between initial letters for the contacts.
4.4.1 General mobile interface guidelines

The iPad is considered a mobile device. It is mobile in the sense that it can be handled while standing up and is lightweight enough to be carried around. It cannot however be operated with one hand with ease while walking around as a smartphone. But, with this considered it is still useful to look at implications that mobility has on usability for the iPad.

A big challenge when designing for mobile use is the shifting context. In 2002 Kim, Kim, Lee [44] performed an empirical study of usability problems when using mobile Internet in South Korea, with the focus on how different contexts influenced the results. The main findings were, firstly, that people do not use mobile Internet evenly across contexts but wait for good situations. Secondly the goal, user movement, availability of hands, sound level and distractions all influence the use patterns of mobile Internet. Lastly, different contexts trigger different usability problems. Overall it is clear that the context has a big impact on the usability and user experience.

But, this only points out that there are big impacts when going from a desktop environment to a mobile environment. Gong and Tarasewich used Schneiderman’s *Golden rules of Interface Design* [46] to formulate more detailed design guidelines specific for mobile use. The following is a summary of the 15 guidelines along with short explanations:

**Enable Frequent Users to Use Shortcuts** - The user should be able to perform common tasks with as little interaction as possible. Time is often critical when interacting with a mobile device.

**Offer Informative Feedback** - When the user interacts with the system it should give feedback to acknowledge what effect the action had.

**Design Dialogs to Yield Closure** - The user should complete a task in steps so that the user feels that in the end, the task is completed.

**Support Internal Locus of Control** - The user should initiate the actions in the system rather than the system initiating actions leaving the user forced to respond.

**Consistency** - The user should be able to move information between the mobile device and desktop environment without losing consistency. An example for this would be a pdf document moved between desktop and mobile without loosing the possibility to open and read it.

**Reversal of Actions** - The user should be able to reverse actions.

**Error Prevention and Simple Error Handling** - As with a desktop application, there is a need to limit errors. With a mobile device the input methods can be less fluid than with a full set keyboard and this needs to be considered.
Reduce short-term memory load - This is perhaps the most important guideline since there are often more distraction from the surroundings and a user is most likely interrupted from the task at some point and will struggle with reminding things.

Design for multiple and dynamic contexts - Be prepared for changing conditions in term of brightness, noise level, time of day, location and so on. The user can switch between interacting with one hand, two hands or maybe no hands with voice input. But, this scenario should consider that voice input can be inconvenient depending on social context.

Design for Small Devices - In this guideline the authors argue that devices will get smaller and smaller in form of bracelets, rings etc. This is not relevant to this project.

Design for Limited and Split Attention - Similar to reducing short-term memory load. The user will probably be interrupted and the system should therefore not only rely on visial feedback but tactile and sound feedback can also be considered for notifying the user.

Design for speed and recovery - The user will need to perform tasks at great speed and will just as fast abort the process. Therefore it is important to start the application quickly and save the state of the application for later.

Design for "Top-Down” Interaction - The Top-Down approach means presenting the user with a short summary of content giving the option of further exploring the content on the user. One example would be a message being received on the device, the first couple of words are shown, the user can then decide to open it or continue with whatever he was doing.

Allow for personalization - The authors argue that the mobile device is more likely to be used by one person, therefore it is important to give the option to personalize the interface of an application to the individual user’s habits.

Design for Enjoyment - Emotion influences the use of an application. The user should feel that the application is well polished and beautiful.

4.5 Conclusions from Background Studie

The background study was used in different ways in the design process of the project. This section presents how the background study becomes relevant to the project.
4.5. Conclusions from Background Study

4.5.1 Smartphones and Tablets

When looking at the sales figures of Smartphones it is comforting to see that there is some truth behind the Smartphone hype. Sales are increasing and the trend points upwards. Phones are becoming more and more capable of handling rich media and the Internet browsing experience is improved. This means that there are new possibilities for applications and systems that a lot of people can interact with only by reaching into their pocket. If you then believe the prognoses for the number of tablet devices that could be sold in the future it gets really exciting. The Smartphone has already forced designers to look into a new design paradigm with touch screen interaction and with the tablets form factor and I am sure that we will see even more interaction behaviours unique to touch screen devices being investigated and design patterns conformed to these new use cases.

Web apps

But, with the smartphones comes the "app hysteria”. It seems that every company wants an iPhone app today although the market is still not so big that it is profitable to develop the app in all cases. Also, there is not only the iPhone on the smartphone market We have Android, Windows Mobile, RIM, Symbian that each have their own programming language. It is not hard to imagine the added complexity that comes with developing an app to reach as many of the Smartphone users as possible. Here I see a big shift coming very fast and that is with approaches towards developing a shared code base that covers multiple mobile operating systems. With the capable mobile browsers you can create a web app with a user experience that is quite similar to that of a native app. Open or closed source frameworks that let the developer achieve this in a cost efficient way are being developed, refined and are attracting a big developer community. It is also quite fascinating to see the business models for these companies and organizations. The frameworks are often free of charge and the income comes from support packages and other premium services which makes it cheap to try them out, thereby encouraging the developer community to provide help in the Internet forums.

For this project the information about the different frameworks was used to pick a good alternative that would be tried out. More about this will be covered in the Implementation section of the Design Process.

4.5.2 Collaboration and Mobile Work

The concept developed in this project was not isolated to the iPad app itself. The concept fits as a piece into a bigger puzzle, a workflow that needs to function efficiently with multiple people involved. Because of this, notions and guidelines for CSCW systems were studied to learn about the fundamental challenges that come with collaboration and working away from the somewhat
static office desktop environment. The guidelines worked as a structure to highlight problem areas in collaboration and to foresee specific challenges associated with working on the go. More about specific reasoning and design proposals are presented in the Design Process section.

4.5.3 Design Guidelines

The guideline document from Apple is very impressive and highlights many of the unique challenges when designing for mobile devices. It is of course focused on their own products and is therefore very concrete in the examples, although it seems useful even if the target platform is not iOS because of the important lessons that are given both in user interface design and in interaction design. The iOS guidelines worked as a reference document during the project to look up what was considered standard interface elements and if there were standard design patterns to solve common tasks. During the design process I aimed to create an experience that Apple users would recognise.

Moving on to the mobile design guidelines that were collected from research literature, they are much more general in their character and do not give suggestions for how to solve the challenges presented on a detailed level. But in the same way as the collaboration and mobile work guideline gave structure to the design and evaluation process, these guidelines were helpful in highlighting possible pitfalls.
Chapter 5

Data Gathering

The goal of the project was to develop a concept for how an iPad app could help in the digital video production and during the project this goal was refined to aiming at an application for collecting feedback on video clips. In the beginning of the project time was spent looking into what tools were used in video production with focus on software support and support for digital workflows. Later on interviews were conducted to get a better feel for how the work situation looked in different areas with the hope to find incitement for solving a specific problem. In this section a very brief overview of video production is presented as well as summaries from the interviews.

5.1 Digital Video Production and Metadata

With the introduction of digital video a new type of editing came about, Non-Linear Editing. With digital video you no longer need to play the tape to the time you wanted to access or to duplicate sequences onto a new tape to form a clip. Instead you can use software that lets you jump to whatever frame you want and freely rearrange sequences on a timeline without distorting the source [37]. Their are a lot of software for digital editing and some of the most popular are Avid Media Composer, Apple Final Cut Pro, Adobe Premier and Sony Vegas.

As expected the size of the media files created in a video project are quite big. The variables for the file size per hour of footage are bitrate and compression. The file sizes can vary between a few gigabytes per hour up to 1000 gigabytes. This makes transferring files and encoding and working with files a bottleneck. During the interviews this came up and will be discussed later on.

For a production company where files are continuously added into a project over time or in a archive of old material the amount of material quickly gets to vast to be mass stored on a hard drive without some sort of organization. It is here Media Asset Management Systems (MAM system) come into the picture. A
MAM is a sub-category of DAM (Digital Asset Management) which is explained by a tech analytic company as having these core functions [31]:

**Repository** - Stores the content with basic functions like version control, categorization, uploading, downloading.

**Metadata index** - A set of metadata associated with the content that describes the content itself but also its relationship with other content.

**Search engine** - An integral part is being able to search through the vast information using the metadata.

**Access and rights subsystem** - Managing rights on content both on a file basis but also on a metadata field basis.

**Workflow or collaboration engine** - Support for working collaboratively with content.

So what is metadata? Metadata is simply put just information about information. If you take a picture with a compact camera there is probably attached metadata about the time of day, the date, camera settings or even location depending on the camera’s capabilities. The metadata is vital when searching for material since it can tell something about the content without having the user open up the file and inspecting it or trying to figure out its content by looking at the file name. But, to leverage the metadata in a search system you need to have an idea about how the metadata is structured. Metadata schemas are ways to organize metadata and there are a lot of different standards for this. One standard is the Dublin Core which contains a basic set of 15 text fields that should be able to make a good base for describing and catalogue most resources [7]. Different application areas have their needs for describing the information handled so different schemas focus on different data.

The activity of entering metadata associated with video is called ”logging” and the Wikipedia entry mentions three typical goals when logging metadata:

**Identify** the video by names, duration or time code.

**Describe** the video in terms of quality, rating, events in the video, objects or people present.

**Classify** the video for easily retrieving video of a curtain type (ie. news team need a certain background clip for a news segment)

### 5.1.1 MAM and Vidispine

The Vidispine system is a MAM developed by Vidispine AB and CodeMill AB and it is the backbone of this thesis project. This section does not aim to describe the system architecture or the features in detail but to introduce the functions that were considered when the concept was developed. Beyond
version control, replication across servers, metadata storage, rights system these are the most important functions [30]:

Transcoding video - Vidispine transcodes video between many different formats making it possible to request a video file in a format suitable for the task at hand.

Time based metadata - Metadata can be associated with a time code or an interval of time in a video which increases the searchability within a file.

Vidispine is an API - Vidispine does not develop its own graphical user interface but functions as a server back end that software can communicate with through an API. Callbacks makes it possible to among other things react if a file changes status, metadata is changed or some task is finished.

Collaboration - Vidispine is designed to work across multiple locations with multiple users.

Easy to access - Because of the focus on back end logic Vidispine works well for access over the Internet. It can stream a video or even intervals of a video on demand.

5.2 Interviews

During the project some minor interviews were performed as well as a bigger one, all with people involved in the video production business. In this section the lessons learned from the interviews are presented, used for the reasoning made further down the road of the project. The interview format was informal without a script. The informal approach was chosen because not much was known about their workflow and the aim was to spark discussions about problems and bottlenecks. The basic premise of the project was explained as an exploration of what an iPad could do in their workflow and they where asked to explain their basic workflow today.

5.2.1 Interview with Erik Lövgren, Dimension Magic

Erik Lövgren has his own business called Dimension Magic and it is a small video production company based in Umeå. Dimension Magic works with digital, tapeless cameras - meaning that the material is stored on flash memory cards and not on tape. A typical project starts with a brief from an idea and a short plot. This turns into the foundation for a budget. Later on a schedule is done that covers the video shoot down to the last detail. Everyone involved knows what to do on the set and time efficiency is one of the most important aspects. On bigger productions there can even be a test shoot to test out the scenes before all props, actors and events are added to the shoot. He mentions that a video shoot for a commercial can have a cost of around 1000 SEK per
minute in this business. There is no time to do anything other than planned. Lövgren talks about the hierarchical roles on a video shoot with the director, producer and scenograf in the top. The rest of the people normally has no influence over the production. In his case he often work alone or with a few others, but still the time at the shoot is minimized. When editing the material he uses Final Cut Pro and when archiving the material he uses external hard-drives and organizes the projects in different folders. He does not work with such a large amount of material that he feels it the search function in Mac OS X is not sufficient.

**Where can you see the iPad being used in video production?** During the interview the basic premise for the concept of using an iPad during video production was explained and Lövgren was then asked to give his thoughts about it. The premise was that the iPad could be used on the set to review the recorded material and input written or sketched comments, enter metadata about what the material contained and communicate with people outside of the set in a form of collaboration.

He instantly questioned how the material could be accessed on the iPad directly on the set, talking about the time it takes to read the large files created by the cameras. The file sizes can be extremely big. If using low resolution versions, some information about color depth and depth of field is lost. On the question if it would be useful to quickly send clips to other persons off set he responded that it could be useful for spotting errors but the producer or director that are in charge are often present on the set so they can see these things in other ways. Entering metadata on the spot does not interest him since he does not work actively with metadata in his process.

Some other ideas were suggested by Lövgren during the interview. With the growing number of social functions on the Internet methods like crowdfunding has become increasingly popular, meaning that individuals such as consumers or fans are asked for money to finance content [6]. Lövgren wondered if the iPad could be used to interact with fans during a shoot to give exclusive sneak peeks, asking the audience for help in smaller decisions and making them more engaged during the process.

**Conclusions from the interview**

- Time is money. Time is an important aspect in video production and a tool like an iPad app will only be embraced if it speeds up the process of the production.

- Metadata is not used by everyone.

**5.2.2 Interview with Swedish TV4**

Two interviews were performed with people from the Swedish TV company TV4. The first interview was with a reporter working for TV4 Umeå and
the second interview was with two people working in the news room at TV4 Stockholm as a system developer and a reporter.

**Interview 1 - Arvid Sundqvist, TV4 Umeå**

Sundqvist works as a reporter and editor. They use tapeless cameras and are using a software called Liquid (recently bought by Avid) but they plan to switch to Final Cut Pro in the beginning of 2011. A normal workflow for him is to go out on an assignment alone with a camera and microphone. He can shot some video, makes a short interview or make a voice overlay. The goal is to record as little material as possible to cover what you need. As an example he estimates 7 minutes of raw material being cut down to two minutes of video. Afterwards he returns to the TV house and edits the segment himself. When done, the segment is posted to a central web system that stores the material making it available for other people working at TV4. When posting the material some metadata is entered with some fields being mandatory. Example of metadata:

- Who filmed the material.
- Who edited the segment.
- What is in it (e.g. on roof, in classroom etc).
- Where it is filmed, the location.

**Could the metadata be entered earlier, with an iPad for example?**

- If it would be super easy to do it and it would follow into the editing software it could be interesting. But then we are talking about doing it directly on the camera. If he has been to four different locations it would be convenient if it could be labelled so that he does not need to search among all clips when editing later on. Other then that scenario he does not see how it would be time efficient in everyday assignments for a field reporter.

Sundqvist also has worked as a documentary film maker before his time with TV4. The biggest difference from news reporting is the amount of material collected. About 65 hours of video was edited down to 1 hour for a project he mentions. During the filming the camera could roll for a whole day and in the evening the digital tapes were ingested into a computer. Everything had to be watched and what the sequences contained and what the tapes contained was managed with a handwritten list. This worked but could have been done easier. The biggest problem, according to Sundqvist, was to keep track of the material and it would have been reasonable to work with metadata to log it.

**Interview 2 - Stefan and Ola, TV4 Stockholm**

The basic workflow for Ola when working as a reporter is similar to that of Sundqvist. He goes out with his equipment in a car, takes shots of the surroundings, makes interviews, records a speaker voice and then returns home.
A difference between the Umeå office and the Stockholm office is that at the Stockholm office there is a central place called Media Desk where the material is ingested. You leave the camera’s memory card together with a hand written form with metadata similar to that previously mentioned. Also, they add a category (News, morning show ”Nyhetsmorgon”, Sports, Economy and so on), raw material ID, memory card number, special rights for the material. All of the ingested material is stored in Ardome which is a MAM system so it is accessible for everyone to edit with Final Cut Pro.

**Could the workflow be optimized with an iPad?** - During the interview the participants were asked to give their thoughts on how the workflow could be optimized with an iPad. Overall they were skeptical to bringing more equipment to the field. Since they often return back to the office and hand over the memory card there is nothing to gain by entering the metadata on beforehand. If you carry something with you it is a laptop. It happens that the memory card is emptied onto the laptop and sent back to the office over 3G/4G if there is a rush but it is not common. One minute of video usually gives a filesize of 200 Mb. They went on to discuss the possibility of docking the camera in the car while charging it and then accessing the material on an iPad or iPhone when driving back to the office. They mention entering in- and out points as an interesting function to shorten the time spent in editing. Another possibility is doing rough editing outside the office, for example preparing material on an iPad when relaxing in a sofa. Although it is not that common to prepare segments over two days.

**Conclusions from the interview**

In the news business there is also a need to do things as fast as possible. With technical development we have seen a change in how the news team are composed. Instead of a whole news team with several people with specified responsibilities, now one or two people go out and do the segment by themselves. That means it is critical to carry as little equipment as possible.

The MAM is central to the news room workflow.

Metadata is used more in news to increase the searchability.

**5.2.3 Meetings with Parham Azimi, Cantemo**

A company collaborating with Vidispine is the Stockholm based Cantemo which develops web user interfaces to the Vidispine system. During the project a couple of meetings have been had with Parham Azimi at Cantemo, talking about metadata systems and search interfaces. He has worked as a sounding board for ideas. The foremost lessons from the meetings are that metadata is getting more important as the number of media assets in a organisation grows.
The ability to find material you need or browse the material you have efficient is essentially to leverage the value of the material.

5.2.4 Interview with Dan Wallis

A longer interview was performed with Dan Wallis who works for the British production company Hogarth. His job at the company is to optimize the workflow with new tools. Hogarth is a part of the multinational company WPP and during the interview he talks in general terms about different aspects of video production workflows.

For Hogarth the most common task is to adapt TV commercials or magazine commercials to different markets. As an example, if a person wears a cross or a women wears a skirt they can remove the cross and change the skirt to a longer one, or maybe just change the language of the ad. But as this is not the main focus of the project, most questions and reasoning was done around production of primary video content, that is the production of new video content.

After a request to produce a commercial with a certain budget, Hogarth plans the entire production. People involved are a TV producer, a ”data wrangler” (person who calculates disc space needed and handles other technical aspects), a light/cameraman and a person that handles all the casting and bookings. The goal is to keep the team as small and cost effective as possible. It is not uncommon that the director grabs the camera or do logging of the material afterwards. Wallis points out that it has become more important to know how to handle many different tasks nowadays. You need to be flexible to be more efficient.

Digital tapeless cameras are used during the shoot and often the cameras are hooked into a server on the site so that the material is streamed into a computer to skip the need to ingest the material later on from the memory cards, the memory cards are bottle necks and if they can be skipped it saves time. From the server some or all the material is uploaded to a Final Cut Server (FCS) which is the MAM system used by Hogarth at the time. The parts that are good are selected on the site or later on and are simply called ”selects”. Selects are often done in Final Cut Pro (FCP) and are always logged with tags such as ”girl red dress”, ”green tree”, ”wind blows” etc.

Offline and online editing - When working with editing the clips a lower resolution is used. These clips are called proxies and act as placeholders in the editing software since they are less CPU-demanding then the master files. When the editing is finished with the proxies, the editor can render the complete sequence with the master files to get the full resolution version. If the proxy is good enough they never need to look at the master files before this final rendering. Traditionally online editing is performed on very powerful computers and the cost of this can be very high. With computer power getting cheaper, Hogarth can run most of the online editing on well-equipped Mac Pro computers and keep the cost of the production down.

Collaboration and approvals - Hogarth has offices around the world and
are always trying to send the work to those who can do it as good and as fast as possible. An example can be a video sequence that is done being edited and then checked into FCS over in Britain. An Asian office can then be asked to fetch it from the server and continue with a Chinese voice over. These kinds of workflows are getting increasingly important.

When selects are being made during the workflow smaller proxies can be made automatically for the customer to watch. A central bottleneck in the current workflow is that the customer is given these proxies over e-mail or ftp and communication can sometimes use telephone lines. Confusion can easily happen when talking about file names and frame numbers. This is something that Hogarth is working to improve.

Their are different processes for approving material. Hogarth works with three different processes - simple, dependent or concurrent approvals.

**Simple approval** - In a simple approval the material is sent directly to the customer for feedback.

**Dependent approval** - In a dependent approval the project needs to be approved in several steps. For a person to receive the material the person beforehand needs to approve it or else the material is returned for more work. One example is that a pharmaceutical commercial is sent to a person working for a pharmaceutical company to check if all the medical facts are correct. After that the material is sent to a lawyer so he can check the content against current regulations in the area. Next step is a person working with the creative aspect to check if the feedback is in some way changing the message of the commercial. Finally there is a lawyer doing a final check to sign of the commercial as ready. A scenario could be that the commercial is approved for every country but France because they have a different set of regulations in the area.

**Concurrent approval** - In a concurrent approval scenario the material is sent to multiple people at the same time, without the chain of approvals.

**MAM, metadata and archiving** - Almost all metadata about the camera settings are saved after a shot. Sometimes also temperature and wind direction. All this in case of a sequence needing to be reshot. But it is very uncommon do that since they can go back to the old material and use that. After a project is finished they keep the selects, the FCP production files and the finished masters. If the client pays extra they store the raw material also. A range of metadata is saved for the project. All the titles that the involved people have of the project (the commercial agency, the post production company, Hogarth can have many different titles for the material), the customers name, the brand name, the producer, music rights and where the rights are valid. Then there are tags for the content of the material.

"The problem is, trying to actually find that thing out of hundred of thousands of things initially, that is the biggest challenge. You
have to have metadata to make that work. [...]

The moral of the story on metadata is, as long it is accurate, you never can have too much.

**Does every media company use MAM systems?** - Everybody is not using such a system but they should. Without them there can be issues recovering material and sending material around. It is not uncommon to send harddrives with material around the world with the postal service. There is a number of problems that can happen, including discs getting corrupted, discs getting stuck in customs, files getting leaked onto the Internet or in the worst case scenario - discs getting lost.

**Approval tool** - What came to be the problem to solve in this thesis is how you collect feedback from people involved in the video production. As Dan Wallis expressed it:

"What we don’t do at the moment but what we want to build is this annotations tool that would allow someone to review the video in a frame accurate style, so using probably HTML5, and to stop the video, draw circles around things and use copy, stickie notes, also to stop the video and do a voice recording. "Ok, I like to change this, this is the reasons why...” and so on”

When talking about the features for an approval tool he also talks about the users. They will not be expert users that are used to working with editing software and familiar to keyboard shortcuts for maximum speed. They are lawyers, financial people, marketing people and so on.

**What is the scenario for the future?** -

"I think as the cost of CPU and computers goes down... I think everyone will do it on shot. We have even done approvals on shot. [...] We make the selects on shot, keep the talent there and we send the files to our client for approval on those shots. [...] So we don’t end up with a lot of shots and the client goes - you know what, I don’t like any of that. [...] That is going to happen more and more.”

Hogarth are already doing work where cameras are streaming material onto an on site server and there are possibilities for optimizations in the digital workflow. The overall goal is to have everything served easily to people needing to do work on the material. Wallis sees online tools for light editing and for logging being accessible on the production site and distributed around the world. Hopefully image recognition will take steps forward towards automatically extracting information about what is in the video without users needing to do it manually.
5.3 Conclusions from Data Gathering

In this section the information gathered is summarized into conclusions that can be used in the project.

5.3.1 What iPad app can help the reporter in the field?

The reporter is in a big rush and has to carry a big camera. Since the memory card is brought back to the station and ingested, sometimes even edited by the same person, it seems like a bad idea to force the use of a tool to enter metadata in the field. A future scenario with better connectivity through 4G network or WiMAX it would be possible to stream material back to the office in reasonable high resolution to let others edit it in more or less real time. In this case the people on the location could use their phone to add on metadata to the material if needed.

For this thesis the scenario and problem area was discarded because of the lack of a real need.

5.3.2 What iPad app can help in the making of a documentary?

With the large amount of material collected for a documentary there is a need to sort through the material and find what you are looking for. Entering metadata could be done with an iPad or a smartphone in the field with the benefits of having a more on the go form factor. Or it could be done later on with a more laid back interaction style.

In comparison with the news scenario it seems more suitable to investigate what an iPad application could do in words of metadata entering since the time out in the field can be longer with possible windows of time to do quick annotations on the recorded material. However, this scenario was not picked for further investigation.

5.3.3 What iPad app can help in the making of commercials

When talking to Dan Wallis from Hogarth it became obvious that they were looking at computer technology to optimize their workflow. They rely on metadata to find and work with material and they are looking for better logging tools to do logging on set. What seemed to be the most interesting problem area was the approval process where material was sent out to many people with approvals being done by multiple people in the same time or in a chain of people.

The scenario chosen for the thesis was based on the Hogarth use case.
5.3.4 The chosen scenario

When working with a video production there are multiple people involved. When making selects or later on versions of the final clip you need to collect feedback from stakeholders in the project. These people can be outside of the office so a mobile tool should have the possibility to reach them on the move. Therefore an iPad application can be suitable for having people receive video clips, reviewing them and entering comments that is sent back to the central MAM for easy access in the company workflow. In a collaborative approach, other peoples comments should also be available in the app for discussions.

5.3.5 The chosen end user

It is hard to pinpoint the target group in the scenario since the people that are giving feedback on a clip in the Hogarth scenario can come from different backgrounds. The assumptions about the use case and the end user is summed up in the following list:

Not expert users. Overall it can be assumed that they are not expert users of video editing software. Expert users are used to navigate in a clip with editing software using keyboard shortcuts with great precision. To translate this knowledge into the concept is therefore not a must.

The user has limited time. Since the target platform is the iPad it is assumed that they pick it up more or less on the go and wants to be in and out as quick as possible.

The user does not want to enter text. Since they are assumed to be on the go and lacks physical keyboard it is best to keep away from text input.

The task is not always the same. The concept has to be flexible in the sense that the feedback entered and the workflow around the concept will most certainly differ depending on the company and the type of video clip.

From the guidelines collected during the background study, a lot of valuable notes can be taken but the following can be viewed as the core from the users perspective:

The user will start and stop tasks abruptly. The work is done on the go, multiple guidelines talk about the ability to move between offline and online, to support split attention and a shifting context. The user have to be able to drop what he is doing and continue later on.

The user should be able to work autonomously. Just as the guideline says, the collaborative aspect should not make the user dependent on others to complete his tasks. In other words - you can not force multiple people to use the app at the same time.
The user wants to be in control of what is shared. It should be clear what the consequence of an action will be for other users to make the user feel in control.
Chapter 6

Design Process

6.1 Ideation and Sketching Phase

After deciding on the use case, as described in Chapter 5, the sketching phase investigated possible features and user interface concepts. The Wallis interview and discussions with Vidispine and Cantemo ended up in a list of features for an video approval tool.

- Basic features
  - Make an interval selection of a clip - with precision being important.
  - Make a comment with the use of:
    - Text
    - Drawings
    - Voice recording
  - See what others have commented on.

- Other features discussed
  - Export comments to a text based version and send with email.
  - Stills grabber to save images or to send them with mail.
  - Predefined buttons for common annotations. Used to mark things with one tap.
  - Shot boundary detection for better overview of the content.

With the basic functions as a base the sketching phase continued to explore how they could be realized. The following sections are summaries of the ideas and exploration.
6.1.1 Precision scrubbing

It is important to have precision when selecting a frame or an interval, even when collecting feedback, since the cost of a commercial can come down to how many frames an actor is seen in or how long the clip is. Also, there can be specific rules for instance on how long a warning text must be in frame for it to be valid. In traditional editing software you have buttons to step frame by frame and to navigate between points in a clip. Overall, editing can be a very short cut intensive task.

![A professional editing bay](image)

Figure 6.1: A professional editing bay

In figure 6.2 the video player in FCP is shown. If you click and draw the mouse over the time line the video is scrubbed. In the middle-segment there are buttons to control the playback. From the left: jump to in point, play interval, play, play around current frame, jump to end point. The in and out points are set with the rightmost buttons down to the left.

The thoughts around how to control the video on the iPad circled around either adopting a similar approach as with this kind of popular editing software or if the touch screen interface could be leveraged to get rid of these controls. In the Apple Video app for iPad the user has a time line for dragging the timeline thumb back and forth. But, when grabbing the thumb a text informs the user that dragging the thumb up or down instead of left or right adjusts the scrubbing speed.

The concept, illustrated in figure 6.3, builds on our fine motorics abilities and that it is more natural to drag your finger then to drag a mouse cursor with high precision. The feature is making good use of the touch screen but it is quite hidden to the user. When scrubbing, the only hint about how to do it is the help text. The affordance for the function is in other words really low.
6.1. Ideation and Sketching Phase

Figure 6.2: Image of the viewer from the Final Cut Pro 7 manual [8]

Figure 6.3: Conceptualization of fluent scrubbing in different speeds
This feature is used in other video apps aswell, e.g. Netflix, so a factor that can relieve the low affordance is that it could be a wide spread design pattern with the influence of Apple.

### 6.1.2 Dragging in and out points

- In editing software it is normal to have the possibility to drag the markers for an interval with the mouse cursor or by tapping the keyboard during playback (The "I" and "O" key for in and out point for FCP). Dragging the mouse would be even more straight forward when using a touchscreen as it takes the direct manipulation paradigm even further. Figure 6.4 shows some early experimenting with handles for in and out point.

![Figure 6.4: Early sketches of interval handles](image)

### 6.1.3 Representation of current comments

A central piece of the concept other then the interval selecting would be to see previous comments made by the user himself or by other people. This is central to the idea of having a collaborative workflow were distance should not impose a limitation on discussions. Previously mentioned notion of Awareness described by Dourish and Belloti stated that it is essential for collaboration to understand what others are doing in the system as well as how you fit into the actions.

The presentation of peoples comments could be done in many different ways. Common ways of commenting on video or on blogs on websites are having the comments come as a list below the content such as in figure 6.5.
6.1. Ideation and Sketching Phase

Figure 6.5: Comments below a video on youtube.com

As mentioned, a central part is the ability to comment over a specific time interval. How could this be visualized? Early attempts simply added a small representation of the timeline in connection to each comment indicating which interval the comment was referred to.

When sending a clip for review spontaneous and uninfluenced reactions are preferred. Therefore some of the concepts initially hide the comments to give the user a chance to first watch the video before reacting to what others have said about it. This will be demonstrated in the mockup section.

6.1.4 Orientation dependant layout

With a tablet or smartphone it is almost standard to have access to an accelerometer sensor to read the physical orientation of the device and then adjust the user interface to the orientation. With the iPad you get a portrait mode or a landscape mode, demonstrated in figure 6.7. As mentioned in the background study Apples guideline document talks about using a split view, meaning that e.g. a list is shown to the left and a bigger content area covers the rest of the right side when in landscape mode. When rotating to portrait mode the list is hidden and the content area now covers the whole display. To access the list you press a button in the title bar, showing the list in a popdown over the content. When looking at apps in the App Store it is quite common to use this pattern since it adapts to the screen without extensive change of the proportions of the list or the content area.

Some of the ideation was around the concept of changing the interface when the device orientation was changed. There are plenty of examples from existing iPad applications where the orientation gives different interaction possibilities and functionality. One example is the ABC Video Player for iPad which lets you watch programs from their network. When holding the iPad in portrait mode while watching a video, relevant videos and other information is shown
Figure 6.6: Visualizing comments on an interval. The majority of the sketches visualized comments with blocks parallel to or on top of the timeline.

Figure 6.7: Demonstration of the proportions between landscape and portrait mode
6.1. Ideation and Sketching Phase

below the video screen. But, when rotating to landscape mode the video is shown in fullscreen, hiding the rest of the interface [1].

6.1.5 Drawings

If you have tried to scribble something with a mouse pointer you know it is quite hard but with a touchscreen it becomes much easier and more natural. Approaches tested for drawing on the touch screen were primarily simple free-hand drawing with an eraser or clear button but the possibility of inserting objects such as circles, arrows, boxes and text fields was also considered.

6.1.6 Voice recording

The possibility to do voice recording seemed to be an obvious functionality considering the use case. The user is presumed to be on the move and a quick voice memo can be more convenient then to use the on screen keyboard input.

6.1.7 Thumbnails as support in browsing

Thumbnails are used in different ways to convey the content of a video through still images. There are different aspects to consider when using thumbnails. Lee et. al. developed 6 different user interfaces on top of the Fischlar system, an operational recording, indexing, browsing and playback system for broadcast TV programmes. The goal was to investigate how thumbnails can be used in video browsing interfaces. The authors describe the results as a three dimensional design space with the three axes being [45]:

Layeredness - The number of thumbnails used can range from a single image to in theory every frame in the sequence. Depending of the user’s goal you have to do a suitable selection of thumbnails.

Provision of temporal orientation - The ways of communicating the temporal position of the thumbnail to the user.

Spatial vs. Temporal Visualisation - The alternative to putting out a row of thumbnails after each other (Spatial Visualisation) is to put them on top of each other and flip through them, replicating the style of the actual video clip (Temporal Visualisation).

The authors concluded that the best approach is very dependent on the user’s goal and nothing from the reference can be directly transferred to the use case. However, the idea of different dimensions can act as an inspiration when trying out different ways of using thumbnails.

In the sketches done in the exploration phase, the most common way of using thumbnails was a poster image - a singel frame that represents the video when displaying many video items side by side. When relating this to the
Layeredness from Lee et. al, it is a low level of detail and says very little about
the content, depending on the poster image. An idea that seems commonly
used is to change the poster image when the user moves the cursor over the
thumbnail left to right. This would correspond to a temporal visualisation in
the design space, looking to present the story of the clip.

Another concept that was considered was displaying thumbnails spread out
along the timeline, either spread out evenly or at detected scene changes. The
goal would be to give the user a hint of the content and make it easier to
browse. This would correspond to the spatial visualisation end of the design
space.

Taking inspiration from iOS and the Gallery app multiple images behind
a poster picture could be made visible by pinching over the image, spreading
the pictures below and increase their sizes. This corresponds to spreading and
looking closely into a pile of images.

6.1.8 Ways of minimizing text input

During discussions the idea of having pre made notations available came up.
This is seen in live logging tools where predefined events such as ”Goal” for
”Team 1” by ”Player A” can be entered simple by pressing buttons instead of
having the person entering the information by writing. Wallis suggested that
these predetermined notations could be entered when the project in hand was
first created back in the MAM system. The notations would then be available
as a tag field where options can be toggled on and off.

Other ways of minimizing text input are of course to draw comments or to
do voice recordings, features previously discussed.

6.1.9 Collaborative work

Inspiration regarding collaborative work was found in the online collaborative
tools of Google Documents. The service is an online editor for text documents,
spread sheets, presentations and more. Some user interface features supports
the Awareness level among the participants.

In figure 6.8 some hints of collaboration support are numbered and high-
lighted with red arrows.

1. An icon of a lock and text tells us that the document is not publicly
available and that it is shared with 2 other people.

2. Text tells us that the document was updated a couple of seconds ago by
a curtain user.

3. One other user is editing the document at this moment. The area is
clicked to show a chat.

4. Text is updated in real time and each user has his own color for the text
cursor.
6.1. Ideation and Sketching Phase

5. You can highlight text and write a comment about it. This is indicated as textboxes next to the document, where other people can reply and discuss.

The inspiration was transferred into a sketch shown in Figure 6.9. (1) Users that view the video together. Each have their own color. (2) Similar to the Google Documents example each user’s cursor is shown on the timeline. This would be valuable if the use case demanded real-time reviews in group, preferably coupled with a voice chat.

6.1.10 Evaluation of Ideas and Discussion

After the basic functionality was specified and ways of realizing this in the user interface were thought out the plan was to create paper prototypes to test possible concepts early on. But, in reality the ideas were bounced around a whiteboard at the office of Vidispine and then translated to the next level of mockups to speed up the process. Figure 6.10 demonstrates one session. With more time, the ideas could have been put together into a flow of a couple of fundamental screens of the interface. This would both test the specific concepts in more detail and reveal challenges in the user interaction flow. As mentioned earlier, it is cheaper to reiterate problematic designs that have low detail.

6.1.11 Conclusions from Ideation and Sketching Phase

**Precision scrubbing** - The design pattern used is not common and user testing with interactive prototypes is necessary. Does this way of controlling the scrubbing feel precise enough to replace buttons for stepping one frame at the time?
Figure 6.9: Sketch of real time collaboration mechanisms

Figure 6.10: Communicating ideas around a whiteboard
Dragging in and out points - A lot of discussions circled around what the behaviour would be when a user decides to make a selection. One alternative would be that when the user taps the "Add Comment" button while the video is playing a start point marker is laid out without interrupting the playback. The "Add Comment" button switches to something like "End interval" and the user would end the interval by tapping it. This corresponds to how video editing tools work. But what if the user wishes to comment on a single frame instead of an interval? It would be necessary to either stop the playback and put the interval markers back together, or using a dedicated button for single frame comments.

Another alternative would be to stop the video playback when "Add Comment" is tapped and allow the user to select an interval if needed. The later alternative was chosen since it seemed to add less complexity to the interface even at the price of extra taps for making an interval. When looking at realizing this project, the "final decision" would depend in what the expected default action would be.

Representation of current comments - It was decided that comments would be hidden when first opening up a video so that the user would not be influenced by them at first glance.

Orientation dependant layout - Two alternatives for handling rotation change were considered and mockups were made of both of them. They will be further discussed in the Mockup Phase section.

Drawings - Using objects like circles, arrows, boxes were perceived unnecessary. Only freehand drawing will be available.

Voice recording - Should voice recording playback be synced with the video? This was discussed and it would certainly be good to make sure that the recorded message is coupled with the content being discussed. But, questions like "What if the chosen interval is shorter then the voice recording?" and "Is it intrusive to have the clip played back when you start the recording? Will it collide with the goal of having a distraction free user interface for mobile use?" tipped the decision to making voice recording something that is coupled with the interval selection but not synced to the video in the interval.

Thumbnails as support in browsing - The mockups later discussed both use the poster style visualization when browsing videos and the temporal visualisation style of having the poster image change when the user scrolls over it. Thumbnails have the potential to give the user a hint of the content within a video at a glance and also marking out scene changes seems interesting and could be further developed for quicker navigation in a clip.
**Collaborative work** - Real time collaboration mechanisms as demonstrated by Google Documents were discussed during the Wallis interview. The use case of having multiple people in the same clip doing annotations simultaneously was not seen as the normal use for Hogarth but it is an interesting function for supporting discussions. Showing where other users are looking within a clip could be a bit overkill but indicating that other people have commented is reasonable.

### 6.2 Mockup Phase

The next step in the process was to translate the ideas to user interface sketches with focus on interface flow and interface layout. The tools used was the software OmniGraffle Professional for Mac together with stencils for common user interface components for the iPad and iPhone. The goal of the mockups was to have users interact with them on the iPad and test the structure and flow of the interface, not the graphical design. To achieve this colors were used as little as possible and instead the interface relied on gray scale and layout to communicate the functionality and affordance.

The process was to first merge ideas into one concept to see if it was going in the right way of what CodeMill and Vidispine had in mind as well as doing hallway usability testing on unexperienced users. After gathering feedback on the first mockup an additional concept was tested based on another fundamental idea combined with the feedback already collected.

![OmniGraffle Professional with Stencils](image)

Figure 6.11: OmniGraffle Professional with Stencils

Figure 6.11 shows the working area and the downloaded stencils for Omn-
iGraffe Professional. The stencils provide perfect proportions of the standard interface elements and for people that are new to the tablet format and iOS elements this is very useful.

The mockup design process was iterative and many versions were made while tuning details and trying new ideas but they all focused on one of two concepts - adapting to rotation and fullscreen video.

**Mockup A - Adapting to rotation**

The iPad guidelines states that the interface should adapt to the screen rotation of the device. This mockup was based on adapting the layout depending on the orientation. The layout turned out to be very similar to the built in app for YouTube.com seen in figure 6.12.

![mockup](image)

**Figure 6.12: Screenshot of the built in YouTube app.**

The basic flow of the app can be seen in the flow chart in figure 6.13. Since multiple users should be able use the app, the first thing that shows when using the app is a login screen. Depending on security policies this could be a one time event and the latest user stays logged on. The screen that follows is called the inbox and lists the videos the user is expected to approve of. With the profile button, the user can control user account information and log out. Furthermore, there is a "History" button with every video in the inbox that shows a popup with all events that has happened to the clip in the system. In the flowchart this is shown as a dotted line to "All Comments" since it can be used as a shortcut to read comments on the video without loading the video.

Figure 6.14 and figure 6.15 shows the mockup in portrait mode and landscape mode while on the review screen. In the portrait mode you have a video
Figure 6.13: Flow chart for Mockup A

panel on the upper part and a tab panel on the lower part for information, making comments and reading other peoples comments.

The upside of adapting the interface to the rotation is that you can optimize the layout to the screen ratio. In portrait mode the screen is split in half but in landscape mode the screen ratio allows the screen to be divided in two areas leaving a sidebar free for a panel. The main function, to enter comments, is lifted to the sidebar that is viewable all the time.

Mockup B - Fullscreen video

With Mockup B the basic concept was to start with fullscreen video and put the interface on top of it, to put the focus on the video experience and if needed complemented with other components.

The strategy is to hide controls to reduce complexity. As Colborne says:

"Designing simple user experiences often turns out not to be about "How can I make this simpler" but rather "Where should I move the complexity?". The secret to creating a simple user experience is to shift complexity into the right place, so that each moment feels simple." - Giles Colborne [34]

In comparison with Mockup A this version starts with fullscreen video playback and the user chooses to show the playback control when tapping on the screen, chooses to list the comments or look at metadata by tapping buttons in the toolbar. All this to move the complexity away and create simplicity. The
Figure 6.14: Mockup A - portrait mode
Figure 6.15: Mockup A - landscape mode
Figure 6.16: Mockup B - portrait mode
Figure 6.17: Mockup B - landscape mode
approach to move secondary information and tools away from the main content is also mentioned in the older iPad Human Interface Guidelines document.

The flow of the app is basically the same as in Mockup A but with a couple of adjustments, as seen in figure 6.18

![Flow chart for Mockup B](image)

**Figure 6.18: Flow chart for Mockup B**

### 6.2.1 Evaluation of Mockups

The mockups were evaluated in two ways. First during a demonstration for Cantemo which focused on if the functions felt reasonable. Later, hands on tests with think out loud protocol was performed with different people. All the tests were performed while sitting around a table in a quiet setting or in a café setting.

**Feedback from Cantemo**

**Mockup A** - The concept of presenting the comments directly under the timeline with the interval represented with a colored bar was understood and appreciated. The comments were sorted in a list grouped by contributor but this seemed not to be the most reasonable way of doing it. Instead a more standard time based ordering was suggested, more in line with a conversation metaphor. When discussing the workflow between the production company and the people giving the comments it was clear that the comments would be handled as "tasks" for the production company to react upon. Therefore it was suggested that every comment could have a button for checking the comment off, like a to do list, for people with
that sort of user rights. When looking at the fields in the details view Cantemo commented that the sort of information that is good to present here is very dependent of individual companies workflows. This was earlier brought up in the interview with Dan Wallis from Hogarth. Lastly, when looking at the difference between the layout in portrait mode and landscape mode Cantemo preferred the portrait mode since it seemed less cluttered. There is more information on the screen at a given time when in landscape mode.

Mockup B - The discussion around Mockup B was shorter but some feedback was given. The fullscreen style was appreciated because it felt less cluttered. Again, what metadata the ”Details” dropdown should contain is dependent on the company workflow so it needs to be flexible. When looking at the list of users that are involved in a clip it was pointed out that the list could be very long since groups of people could get permission, quickly adding to a long list. In this version the ”Approve” functionality, that was meant as a sign off from the user, was dropped from the concept since it presumably added complexity and decreased the flexibility for how different people want to use the system. Cantemo preferred this.

Feedback from other people

There where no specified user group available for the project so friends were asked to perform think out loud tests of the mockups. There were 7 people looking at the mockups and out of that 6 people were students at ”Interation Technology and Design” so it should be noted that the feedback from them had obvious influences from their education as interaction designers. The following is a summary of the findings

Mockup A

- The mockup does not cover how to remove or edit comments.
- The majority of the people misinterpreted the ”Review comment” icon that was shaped as a typical play button to an arrow icon that indicated that if the row would be tapped, it would swipe in more information for the comment, much like a nested list.
- One person asked if it was really necessary to rotate the screen at all since he felt that the portrait mode was the best.
- Comments are more connected with the time line in comparison to Mockup B

Mockup B

- The mockup does not cover how to remove or edit comments.
– The icon for making replies to comment was confusing and was interpreted as a "go back" icon. Even the three people that owns an iPhone misread the iOS standard reply button. They often do this on their phones too.

– Feels simpler since there is less on the screen.

– More taps to reply to a comment, but not curtain if it is worse since it is less cluttered then Mockup A.

– Comments are not as easy to use as in Mockup A.

6.2.2 Conclusions from Mockup Phase

The two mockups contained the same basic functionality but with different main concepts for the layout. The first using a split layout with a tab panel, the second using a fullscreen video mode with controls and information laid on top of it. When looking at the basic functionality it became clear that the functionality was what Vidispine and Cantemo had in mind when thinking of the problem at hand. The most important keywords that were repeated in the feedback was "simple". Even though there were no attempts to measure the difference in error or efficiency between the mockups or in comparison to similar systems the overall feeling was that the approach taken was not bad or confusing.

After the hands on tests the users were asked which mockup they preferred with mixed results. The majority picked a mix between the concepts. The fullscreen concept was perceived as clean and clutter free and the comment style on the split view was appreciated for the mapping between time line and comment.

– Mockup A - Good Things
  • Comments are well mapped to the time line.

– Mockup A - Possible Improvements
  • Sort comments by time, not by user.
  • Add functionality for removing or editing comments.
  • "Play" button for reviewing a comment is misleading, change the icon.
  • Demonstrate how a comment is played back.

– Mockup B - Good Things
  • Clutter free layout.
  • Less confusing when rotating.
Mockup B - Possible Improvements

- Change the "Reply" button to something other than the standard icon or add a text label.
- Can replies be less taps away?
- Demonstrate how a comment is played back.

Next step - since there was no clear winner of the two concepts and most people liked elements from both versions a mix would be a good next step in the mockup phase with additional testing. Because of the limited time left in the project, the lessons were instead written down and the next phase was started immediately. Mockup B was chosen as the foundation for the prototype phase because it was perceived as less cluttered and more in line with the idea of having the important content taking the center of the focus.

6.3 Implementation of Prototype

The last step of the process was moving the ideas and knowledge from the mockup phase into a prototype phase that initially had two goals. One goal was to make a more robust demo in the sense of more interaction capabilities and more dynamic content. The other was to test a framework for web application development. Because of the project running late the demo goal was put second and the evaluation of the framework put first hand. This section will go over the lessons learned from using the JavaScript framework Sencha Touch.

Initially the documentation for Sencha Touch was skimmed through and the demo app was set up. It does not take much code to have a demo application with a basic list component up and running. On a normal HTML web page you use HTML-tags around content such as `<h1>heading goes here</h1>` or `<p>this is a paragraph</p>`. Tags like these are used to build up the entire page inside of the `<body>`-tag. For a Sencha Touch App you also create a HTML-file but you do not fill it with content tags. See the code in figure 6.19 for an example from the Getting Started section of Sencha Touch [21].

The HTML-file only includes the stylesheet for Sencha Touch, the stylesheet created by the developer as well as the Sencha Touch JavaScript source file together with the developers JavaScript file. The body-tag is left empty and will be filled with tags generated programmatically through Sencha Touch using JavaScript. The next part is the custom made JavaScript file where the developer states some initial parameters like app icon, startup image and then which components should be created in the app. See figure 6.20 for example code from the Sencha Touch documentation [21]. The code for updating the twitter data is removed to shorten the example.

The next step was trying to implement the concept developed in the mockup phase. A video component was used to play a video clip which was quite easy as seen in figure 6.21. A panel component with xtype set to video followed by
size, position and the source file parameter was put in the example code. This worked perfectly on the iPad since the video component creates a HTML-video element and therefore can be handled by the iPad despite not having support for Flash [24].

The concept sketches used custom controls for the video playback and by default the native controller is used. A custom controller was developed using Sencha Touch components such as buttons and a slider and a working controller was successfully implemented during the project with the ability to skip back and forth using the time line thumb, seeing the current time code and to start and stop the playback. More about the performance will be discussed in the Discussion section further down.

Another central part of the concept was the ability to draw on top of the video for commenting. This was successfully implemented using a community submitted plugin called signaturePlugin [25]. With this plugin it was possible to add a component that handled touch events and traced the user input. The component was made transparent and added on top of the video component, making it possible to draw directly onto the clip. The plugin supports saving the drawing in a base64 format which is convenient for storing for example in a backend system such as Vidispine.

After experimenting with different components the MVC support was investigated. The basic idea is that a model defines the structure of the data used by the app. The guide on the Sencha webpage was followed and figure ?? shows how a simple model is implemented [17].

The models are put into stores to hold the data and so called proxys are created to handle the reading, writing, updating and so on. During the development in this project, the ”localstorage” proxy that comes with the framework
Ext.setup({
  tabletStartupScreen: 'tablet_startup.png',
  phoneStartupScreen: 'phone_startup.png',
  icon: 'icon.png',
  glossOnIcon: false,
  onReady: function() {
    var timeline = new Ext.Component({
      title: 'Timeline',
      cls: 'timeline',
      scroll: 'vertical',
      tpl: [
        '<tpl for=""»,
          '<div class="tweet">
            <div class="avatar"><img src="{profile_image_url}" /></div>',
            '<div class="tweet-content">
              <h2>{from_user}</h2>',
              '<p>{text}</p>',
            '</div>',
          '</div>',
        '</tpl>'
      ]
    });

    var map = new Ext.Map({
      title: 'Map',
      getLocation: true,
      mapOptions: {
        zoom: 12
      }
    });

    var panel = new Ext.TabPanel({
      fullscreen: true,
      cardSwitchAnimation: 'slide',
      items: [map, timeline]
    });

    var refresh = function() {
      var coords = map.geo.coords;
      Ext.util.JSONP.request({
        // Stripped from example.
        // An external call is done to twitter using JSONP and the data is updated in the app.
      });
    };

    // These are all Google Maps APIs
    var addMarker = function(tweet, position) {
      var marker = new google.maps.Marker({
        map: map.map,
        position: position
      });
    };

    map.geo.on('update', refresh);

    var tabBar = panel.getTabBar();
    tabBar.addDocked({
      xtype: 'button',
      ui: 'mask',
      iconCls: 'refresh',
      dock: 'right',
      stretch: false,
      align: 'center',
      handler: refresh
    });
  }
});

Figure 6.20: App with a tab panel that either lists tweets or shows them on a map
var pnl = new Ext.Panel({
  fullscreen: true,
  items: [
    {
      xtype: 'video',
      x: 600,
      y: 300,
      width: 175,
      height: 98,
      url: "porsche911.mov",
      posterUrl: 'porsche.png'
    }
  ]
});

Figure 6.21: Example code for a video component

app.models.Contact = Ext.regModel("app.models.Contact", {
  fields: [
    {name: "id", type: "int"},
    {name: "givenName", type: "string"},
    {name: "familyName", type: "string"},
    {name: "emails", type: "auto"},
    {name: "phoneNumbers", type: "auto"},
  ]
});

Figure 6.22: Example code for a model
was used. It uses the HTML5 local storage functionality to store the data within the browser much like cookies are stored. When thinking ahead this could easily be changed to a proxy that worked towards a server backend like Vidispine. More about this will be discussed later on.

Further on, the logic of the application were separated into controllers and the presentation parts were separated into views to complete the MVC pattern.

### 6.3.1 State of the implementation

As mentioned, the time limitations for the thesis project combined with inexperience with JavaScript and Sencha Touch cut the implementation phase short. This section will briefly present what the state of the implementation was when suspended with focus on how the application is constructed.

![Figure 6.23: Start screen for app.](image)

Figure 6.23 shows the starting screen for the application. This corresponds to the inbox screen in the mockups but it is implemented with a list component. The list is set up with a reference to a data store and is given a template for how each item should be presented. In this case the list should write out the title of each item in the video store followed by the source file path. The list will automatically render a list item for every entry in the given store. This screen is highly simplified but provided a good example for how to render data from a store. When the user taps a list item a message is dispatched to the controller with information about what item was tapped and what function should be run. Furthermore there is the "Log in" button in the top toolbar.
6.3. Implementation of Prototype

which shows a popdown that should contain user information but in this demo it is empty.

![Figure 6.24: Reviewing a video.](image)

When an item has been tapped the screen is changed to the review screen as seen in figure 6.24. When pressing play the video starts to play and the user can skip through the video with the time line. The precision scrubbing is yet to be implemented. When the user taps the “Add comment” button in the form of a plus icon the video is paused and the controller is extended to house to commenting controls. At the time of implementation Sencha Touch had not yet implemented a multi handled slider for the framework which made the interval selection function necessary to custom code which was not completed.

In the toolbar of the video review screen there are buttons for Comments and Details as seen in figure 6.25 and figure 6.26. The popdowns are built with floating panels that are set to render by respective button. In this way Sencha Touch delivers an experience similar to the native popdown component. The comment popdown consists of a list which is populated with the Comments store. The behaviour of expanding list items was tested and it is possible to achieve with some workarounds. The Details popdown is built using the fieldset component which gave a native-like look and feel.
Figure 6.25: The comments in a popdown.

Figure 6.26: Metadata for the clip.
6.3.2 Conclusions from Implementation phase

A goal for the implementation phase of the project was to create a more robust and interactive prototype. There are still a lot of pieces missing for accomplishing this since the interval selection and precision scrubbing are still not implemented. These are important parts as they are central to the concept and not yet fully evaluated by users, only imagined by them when trying the mockups. Despite of this the implementation phase proved to be very interesting and a valuable experience since the framework Sencha Touch was tried out. When conducting the implementation the aim was never to hold it up to production quality but still something can be said about the experienced performance:

**Video playback** - The first video file that managed to play on the iPad was then used throughout the testing. The playback was smooth for the test file but nothing can be said about the performance for other file formats and bitrates since there was no testing of this.

**Drawing** - The drawing function was implemented as lines drawn between registered input points on top of a HTML canvas. On the desktop the high update frequency made the curve very smooth. When testing the same code on the iPad the update frequency seemed a bit lower, leaving the curve a bit less smooth. It still worked reasonable well but the problem was more obvious when drawing on top of a playing video. The code could presumably be better optimized but it is clear that the iPad has less processing capacity than a regular desktop.

**Animations** - When scrolling lists and opening popups there were no lag experienced but sometimes transitions between screens could slow down. There are guides on the Sencha web page describing how to optimize the rendering of views to keep the memory load down and this is something that should be further examined.

**Theaming** - The framework has support for switching between different stylesheets that control the overall look. This is called theaming and gives the opportunity to make big changes to the look throughout the app by switching the theme. The theaming functionality was never tried and although it seemed pretty strait forward in guides and the documentation, no comment can be made about how easy or powerful it really is. When talking to Cantemo, their use case included styling interfaces for different companies and this is a good asset for that.

Furthermore, something could be said about the learning curve of Sencha Touch. Before this project I had some experience in HTML and CSS and had limited experience in JavaScript. Even if the sample apps were very easy to get going, it took quite some time to really understand how to connect the different views together and set up a good app architecture. Over the whole course of
the project the documentation for Sencha Touch improved and more guides, videos and blog posts were posted. It could be so that if all these resources had been there from the start, the learning curve would be less steep, but the overall impression and my view of the Sencha Touch framework is that it is easy to use when doing exactly what the examples are demonstrating - but quite demanding when trying to make custom components that are not offered out of the box. The steepness could stem from the fact that everything is done in JavaScript, leaving people without JavaScript skills a bit behind from the start.

Lastly, it should be made clear that even if the implementation was not completed and the learning curve was seen as steep - I absolutely see potential in the framework and its capability to do production grade web applications.
Chapter 7

Conclusions

The problem statement for the thesis project was to examine how feedback on a video clip could be collected from people during a video production project. The overall goal was to design a concept for collecting this feedback with an iPad application.

Revisiting the sub goals, they were stated as followed:

Collect information about the video production workflow and state reasonable assumptions about the specific challenges in the field.

Specify core functions for an video annotation tool used to collect feedback about a video clip.

Create a user interface optimized for the iPad - What are the guidelines and best practices for tablet design in general and for iPad in particular? How can these be used in this particular application?

This section will look back on the goals and sum the conclusions, give final suggestions for the concept and to suggest how to move on onto future work.

7.1 Problem in the Workflow

From the interviews performed it was made clear that collecting feedback efficient was important and an opportunity to solve the problem with a mobile app was seen. Video production is done with punctual planning when on set and time spent in editing and other post production is kept to a minimum. In this climate the need for fast feedback is obvious and there is no time for confusion around what frames of a video clip a discussion is referring to.
7.2 Core functions

The project specified a number of core functions for the annotation tool that could solve the challenge in the workflow. In the Ideation and Sketching Phase section, the functions were listed and ideas on how to realize the functions were examined. Later on, in the Mockup Phase, the functions were put together into an application flow and evaluated by peers and users. Since the iteration did not come as far as a complete design suggestion, the resulting feedback given and suggested interface solutions should be used as an inspiration when designing and implementing an annotation tool for a more specific workflow.

7.3 User interface for the iPad

The sketches and mockups all focused on the iPad as a target device and specific guidelines for iPad were collected and summarised. If the concept should be realized it would be wise to take a fresh look at the whole eco system of software in the work process before more interface designs were made. As mentioned before in this report, a likely scenario would put a web interface higher in priority when supporting this tool because of people usually being more used to a desktop environment and having access to laptops. The questions to be answered here are - what parts of this concept are specific to the mobile touch screen interaction paradigm and how should it be translated to a web based desktop environment?

7.4 Lessons learned and future work

It was hard to narrow down the problem statement even though Vidispine at an early stage presented the idea of making a tool for entering metadata over time intervals for video. The reason behind this was the quest to find end users that could be interviewed and observed. This desire came from of the need to better understand the context and the specific challenges. When information later was collected through the data gathering phase, the plan was to both produce sketches, mockups and an interactive prototype. When looking back at this process there was a point in time where the project could have taken another path that could have been fruitful. Namely, to cut out the next step of interactive prototyping and to continue to iterate and refine the mockups that actually provided both the possibility to navigate through the application flow on the iPad. If this path would been taken, the final result would have been a more refined interface suggestion that could be demonstrated more like a demo. But, as mentioned in the conclusion part for the Implementation Phase section, the fact is that the path that was chosen proved to be fruitful in the form of insights into the Sencha Touch framework and the results from the Mockup Phase can still be used as a first exploration of the purposed concept.
7.4. Lessons learned and future work

Many of the guidelines stated in the background study were used during the design process both as basis of design decisions as well as inspiration for new functionality. They have the potential of being useful in future work as a check list when evolving the presented concept or creating new applications that are meant to be used on the go or in collaboration with other people.

From a personal view the project turned out to be very educational in many ways. It is my biggest project during my time at Umeå University and I achieved my goal of doing it alone instead of in a pair. This was my goal since I wished to test my abilities to work independently in a big project and I am happy that I did it. Although there are negative aspects working alone in a project like this. I have on countless occasions felt the need to talk a design over with someone to get new perspectives and to discuss the tiny details of interface sketches. CodeMill and Vidispine were helpful but they had limited time and they of course did not have the same insight into the project as I did. The result of the project would probably be more detailed and more in depth if done with a partner but then again, the experience would not have been the same.

The project plan was changed during the process and with that came new priorities in the individual phases. One thing that have become clear is the fact that the longer you keep the concept open with many use cases, the harder it gets to tie together the final concept. I feel that many good ideas are mentioned in the report but are hard to communicate when there are so few mockup sketches. I will give the same advice as many master thesis student have given before me, even if there is a exploration phase the project scope should be as precise as possible from the start to lower the time spent on narrowing the project down. Another reflection is that the time needed to find users should not be underestimated. It was not obvious what the end user looked like for the use case and it ended with much of the user testing being done on friends and random people in my surroundings. In the ideation phase, many sketches were made but it were mostly the mockups that was given to other people for think out loud sessions. If the project would be done again, less time would be done exploring different use cases and more time would have been given to iterating mockups.

Finally, I want to go back to a very interesting area of research that is mentioned briefly in this thesis. When evaluating mobile applications current usability evaluation techniques and prototyping tools are put to a test. Tests done in lab environments or over a coffee table will most definitely miss the effects that a highly variable use context will put onto a user experience. It will be interesting to see how lofi prototyping techniques will be developed to capture the variable context that mobility brings and to see how remote usability tools can be used to monitor users while they use devices in their everyday life.
Chapter 8

Acknowledgements

First, I want to thank the people at CodeMill and Vidispine for their generosity and for making me feel at home at both companies. Johanna Högberg, supervisor from CodeMill, was truly a big help in guiding me forward in the thesis project work and kept me calm until things started to fall into place. Eskil Bylund from CodeMill was an invaluable asset when tackling the JavaScript programming and always provided great input when discussing interface sketches around the whiteboard at the Vidispine office.

Furthermore, I want to thank the people that agreed to be interviewed and the people that helped me evaluate the concept. Especially Cantemo showed great interest and arranged the big interview with Dan Wallis that turned out to be a real turning point in the project.

Håkan Gulliksson, the supervisor from Umeå University, was as always genuinely curious about my work and was helpful in the formulation of the thesis project. You made me feel confident that the project would be fruitful.

Finally, I of course want to thank my wonderful girlfriend Elin, my loving family and all my friends for giving me great support and always being there for me.
References


