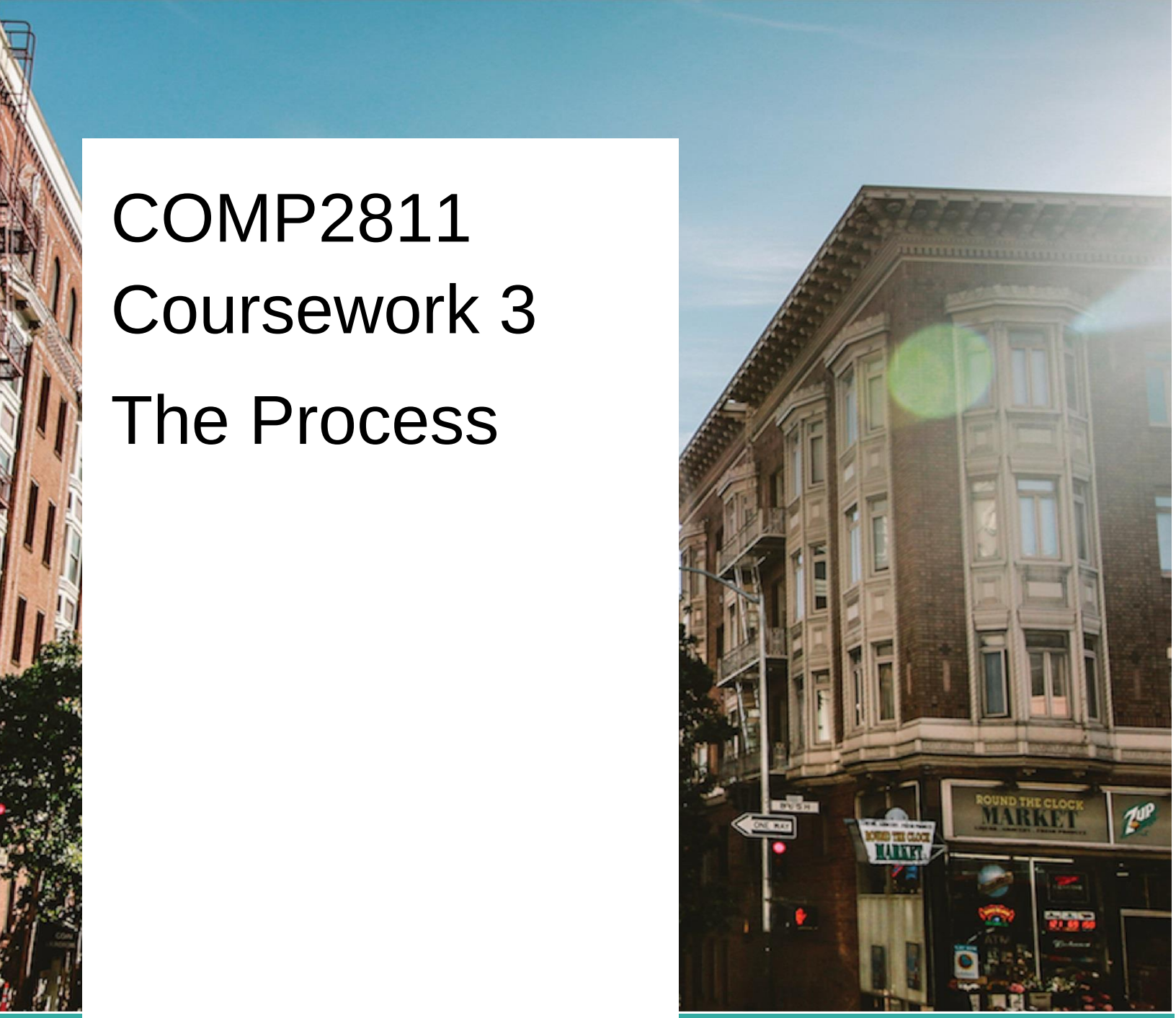


COMP2811 Coursework 3

The Process

[18 NOVEMBER 2021]

[Coursework Group 9]

UNIVERSITY OF LEEDS

Group Members

Shrawan Sreekumar – el20ss
Harith Al-Safi – el20hzaa
Tom Milner – el20tm
Dylan Charnock - el20dlgc
Jorge Bishop - el20jlmb

Requirement Analysis

From the information provided in the project brief, as well as the consensus between the group members, we have identified the following areas for improvement in the *Tomeo* application:

- Implement the native system explorer functionality to the application, to allow the user to specify the source of the videos from the local system.
- Add a “Stop” and “Play” button to the interface.
- Add a “Preview” feature that allows users to scroll through all detected videos at a particular destination folder
- Be able to filter videos based on different tags (Location, Priority, Length etc.).
- Add a “Map” function that shows the location where the videos were taken.
- Add a “Volume Control” to the interface
- Implement a “Navigation bar”
- Add a “Speech to Text” functionality, as well as adding subtitles to videos.
- Add additional language support
- Add a share feature that allows users to upload content to hosting sites.

PACT Analysis

People

The *Tomeo* application is primarily aimed at outdoor enthusiasts and travellers, who are looking for a simple, effective, and robust means to organize their video collection. We expect the audience to be cantered around hikers, cyclists, and extreme sports lovers, who collect videos from action cameras, drones, and video cameras. As a secondary group, we also expect v-loggers and travellers to utilize the application for categorizing their video library. Beyond this scope, we also consider single-time users, as well as small businesses looking to, for example, view and categorize videos for a project. The simple, straightforward interface will cater to users with diverse levels of specific knowledge, and the language support will allow them to navigate around the application with relative ease.

The intended audience are of a modern era, with age ranges assumed to be between teens to late middle-aged users who can assess the use of technology with a certain degree of maturity, as well as engage in outdoor activities in a safe manner. The application will be tailored to suit a gender-neutral user group, where a lot of the in-built functionality will take control of how the library is managed. This will allow the user to prioritize what they want to get done, rather than get occupied with the app's complexity.

Activities

Due to the nature of the activity of the primary audience, the application will be used infrequently. It is likely that the users would, at most, use the app on a weekly basis. This links back to why the interface must be simple, as it also aids the users by being easy to remember. That being said, it would still take a certain amount of exploration to get around the interface. The navigation bar will allow users to intuitively move around the app. Tomeo, like its competitors, is designed for continuous use. Users will be exploring the library for extended periods of time during their quiet hours. The stop and play buttons will equip users to pause their viewing and get back to it when time allows it. Tomeo will also be designed to be instantaneously responsive, *ideally*. This quality is due to the unpredictability of the videos themselves. A three-hour long video would take

substantially longer time to parse compared to a five second moving clip. However, users would expect to experience consistent and fast-paced processing times with the application. The presence of a preview function will let users access important files without having to wait for the videos to play through till that point.

Being a personal video library management app, *Tomeo* lacks collaborative features that would otherwise allow other users to work in tandem to organize a certain collection of media. The application is plotted to work offline, but certain features and updates would require an active network connection. It only handles user sensitive data like personal data and videos, and therefore has no safety critical functionality. The user would mostly interact with pre-defined functions in the application, meaning there's little chance of making errors.

Context

Tomeo is being developed and will be deployed in a desktop/laptop environment. This is partially due to the storage and processing overheads of the media. Attempting to host the application as a web-app would require integration with third-party storage and security tools, which can add to the overall expense of the application, as well as introduce security and privacy risks. Other devices like smartphones lack the inherent ability to interface with recording devices like Go-Pros and action cameras and will need a pc to interface with it in the first place. With regards to a physical context, the only restricting factor would be the portability of the hosting device (desktop/laptop) and the nature of the media being viewed on the app. The volume control being implemented will ensure that social sensitivities are met when using the application outside (such as in a library/cafe).

By maintaining its offline profile, *Tomeo* does not introduce any privacy risks to the user. While a help and tutorial function can be hosted on the web, it would not create any legal issues. The application is not designed to be run in a professional environment but can be implemented without concerns of deskilling the workforce.

Technologies

Tomeo is a standalone application that requires only a basic desktop/laptop machine with a working I/O interface and storage to house its content media. While additional processing power is always an advantage in running video viewing/processing software like itself, it is not hindered by a lack of it either. The input data is directly read off the source files upon being directed to the relevant location by the user, and the output is shown on screen which can then be viewed. Categorizing the content is subjective to the user's requirement, although some recommendations will be provided by the app after analysing the data.

Use Cases

Scenario 1:

Tom (35) is an avid skiing fan. He works at a Snowboard and Skiing Rental in Zermatt, Switzerland. He enjoys traversing the high peaks and gorges on skis, while recording his antics for his YouTube channel. Most of his video clips are recorded using a Go-Pro attached to his ski helmet. Tom decides to use *Tomeo* to organize his clips and remove some unwanted footage. He plugs the Go-Pro into his laptop and opens the application. Upon applying the "Recent" filter, his last recorded videos are displayed on the app. He then proceeds to click through the videos, removing any unwanted content using the in-built "Delete" feature. Tom then decides that he wants to check some of his older content on the Go-Pro. However, the "Recent" filter does not display the videos from that time-period. He chooses to use the "Search" feature to locate the video by date. He is then shown all the videos created on that date. Tom manages to successfully organize his video library.

Scenario 2:

Jerry (23) loves to travel. It has been his dream to go on a world cruise and he has recently been given the opportunity. He wants to record his journey as he travels around the globe. He decides to organize the clips he captures based on the location where he took them. He transfers his video content to a laptop the evening he leaves the country. He opts to use *Tomeo* to help him organize these

videos. Jerry opens the application on his laptop. He navigates to the "Open Folder" option and selects his source folder of videos. The videos are displayed on screen in the order of date created. The app suggests grouping based on a cluster of videos that were created in/around the same time. Jerry selects this option. He then proceeds to modify the clusters to add a location tag, which reflects across all the relevant videos. Jerry does not think one of the videos has been clustered correctly. He wants to change this specific video, but the auto-cluster option does not allow this. Jerry is still satisfied with the application. The feature has saved him more than an hour of his time. He decides to manually change the metadata for the specific video from the app separately.

Scenario 3:

Lequisha (27) is a cycling enthusiast. She enjoys exploring new roads and off-road paths on her mountain bike. He keeps a record of her exploration using a tracking tool and uses her native file manager to organize her videos. She hears of *Tomeo* from her friends. Interested by the "Map" feature in the app, she decides to give it a go. She opens the application on her desktop. She navigates to the open folder option and selects her source folder of videos. She selects the filter by "Location" option. The application reads the metadata from the videos and offers to group the videos by location. The location of the group of videos is displayed on the Map. Lequisha tries to export the map to another file but is unable to do so. She proceeds to then open the map on a browser page and downloads it to her tracker.

Platform

We have chosen to design the *Tomeo* prototype for production on Desktop/Laptop machines. This is primarily done to prioritize user experience when using the application. Some of the features of *Tomeo* might not be as pleasant to use on a smaller screen device as it is on a Desktop/Laptop. Also, due to the nature of the application, it maintains consistency with other applications of its kind like VLC Media Player and Windows Media Player. By having a desktop/laptop-oriented interface, it will ensure that users have a more intuitive experience when using the app for the first time. It also ensures that the application itself has enough resources to function properly (in terms of processing power and speed). It also gives the user flexibility in terms of how the videos can be accessed (Go-Pros and action cameras can be directly connected to

desktops/laptops instead of having to bring in additional adaptations to connect them to smaller devices).

Design Cycles

Cycle 1:

Prototype:

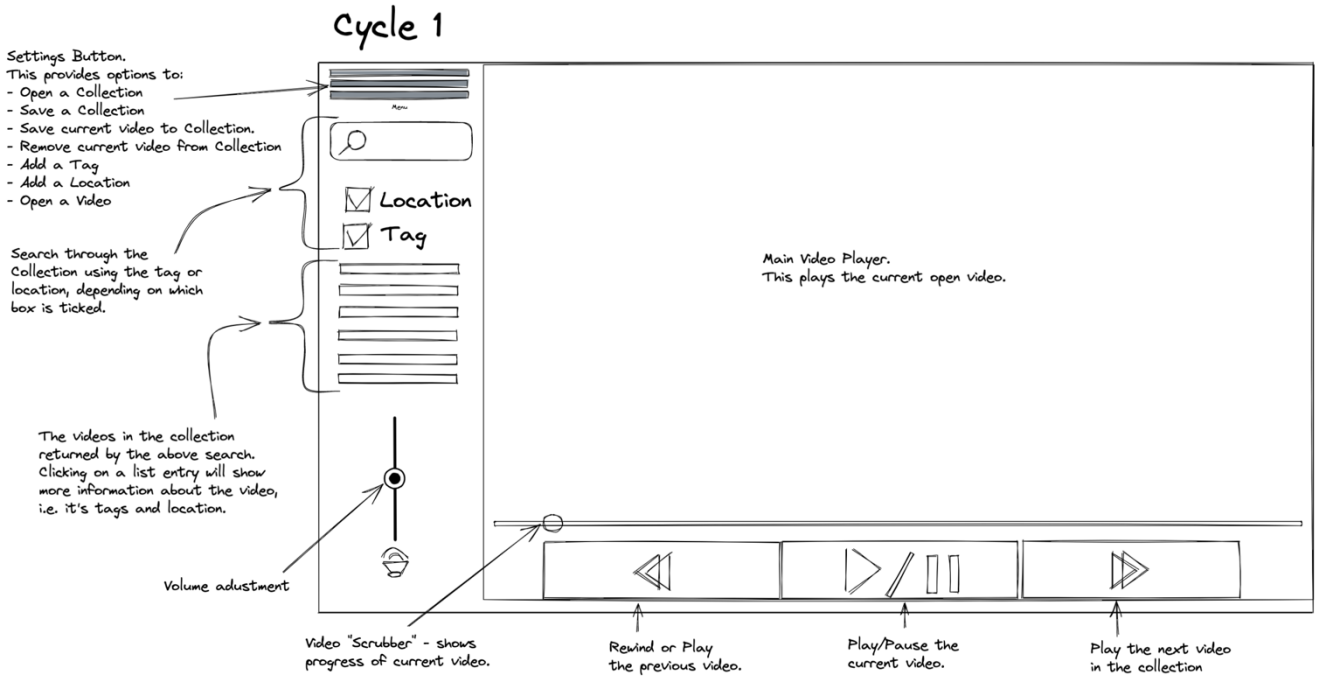


Fig 1.1 Excalidraw prototype

The goal of this first cycle was to implement the basic features of the application – tagging, location, collections (more info later), and video player functionality. We decided this was the highest priority because once we had the basic functionality in place, we would be able to experiment more with the design and usability aspects of the program.

To prototype we used a sketching software called Excalidraw (https://excalidraw.com/), as it allowed us to collaboratively work on the prototype together. Excalidraw also makes it extremely easy to create reproducible components which can be reused throughout the design. An example of this feature being used is the icon used for the skip and rewind buttons; the rewind icon is the skip icon, just mirrored horizontally.

As mentioned earlier, we produced the idea of a "Collection", which is a set of videos that the user wants to group together. For example, in "Use Cases – Scenario 1", Tom could group all of his skiing videos together into a "Ski Trip" collection. This is a conceptual model that users can easily understand, as a collection is essentially a sub-group of a larger collection of videos. Different collections can contain the same videos, as different collections can serve different purposes. For example, Tom could form a collection called "Snowsports", and this would likely contain a lot of the same videos as used in the "Ski Trip" collection, but not all of them, as some of the "Ski Trip" videos could be video diary entries etc.

In this design, the videos of the current collection can be seen on the left-hand side. They are displayed in a list view, so the user can easily navigate between them, and are searchable using the search bar above. The user can add to the collection, remove from the collection, and open or save the collection using the menu at the top-left of the program. This leads on to the next features we decided to implement: "Location" and "Tags". We decided to add the ability to "tag" a video with an arbitrary string – Tom, for example, could give a video of him skiing a black run the tag "Black Run". He can then easily find all the videos of him skiing black runs by just searching the collection for "Black Run", and all the correspondingly tagged videos in the collection would be listed. The same idea goes for the location – a user can give a video a location, and then will be able to search for all the videos in that collection from the same location.
The rest of the buttons and controls are for manipulating the video itself. Users can play, pause, rewind, and skip the video, as well as adjust the volume and the current video progress. These controls are all very intuitive and are detailed in the prototype sketch.

Technically speaking, the design is an instruction-based interface - the user has access to buttons, menus, and controls that allow them to interact with the program. We decided this would be the best approach as it is a popular technique for interacting with desktop applications and would be familiar to most users. All the controls "afford" being clicked on and indicate their function by either signifier in the form of an icon on the control itself (e.g., play/pause, rewind), or by using consistent mappings that are recognisable to most users (e.g., the video progress and volume sliders).

The volume and video progress sliders provide feedback by moving the circle to the desired location when the user interacts with them. This means that immediately, the user can see how far through the video they are and the volume the video is playing at, as well as how much louder/quieter the video could be.

Code:

Fig 1.2 Tomeo – Base program implementation

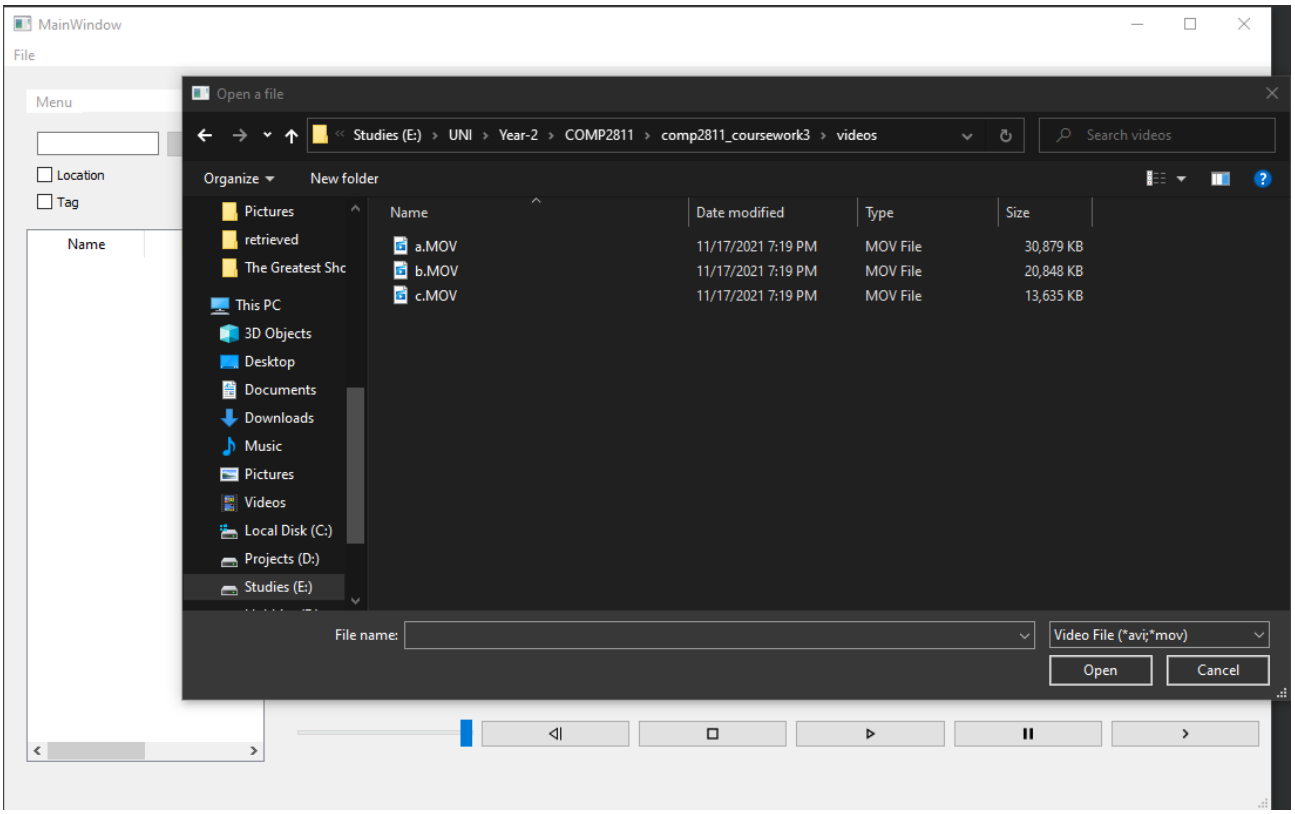


Fig 1.3 Tomeo – Open file function implementation

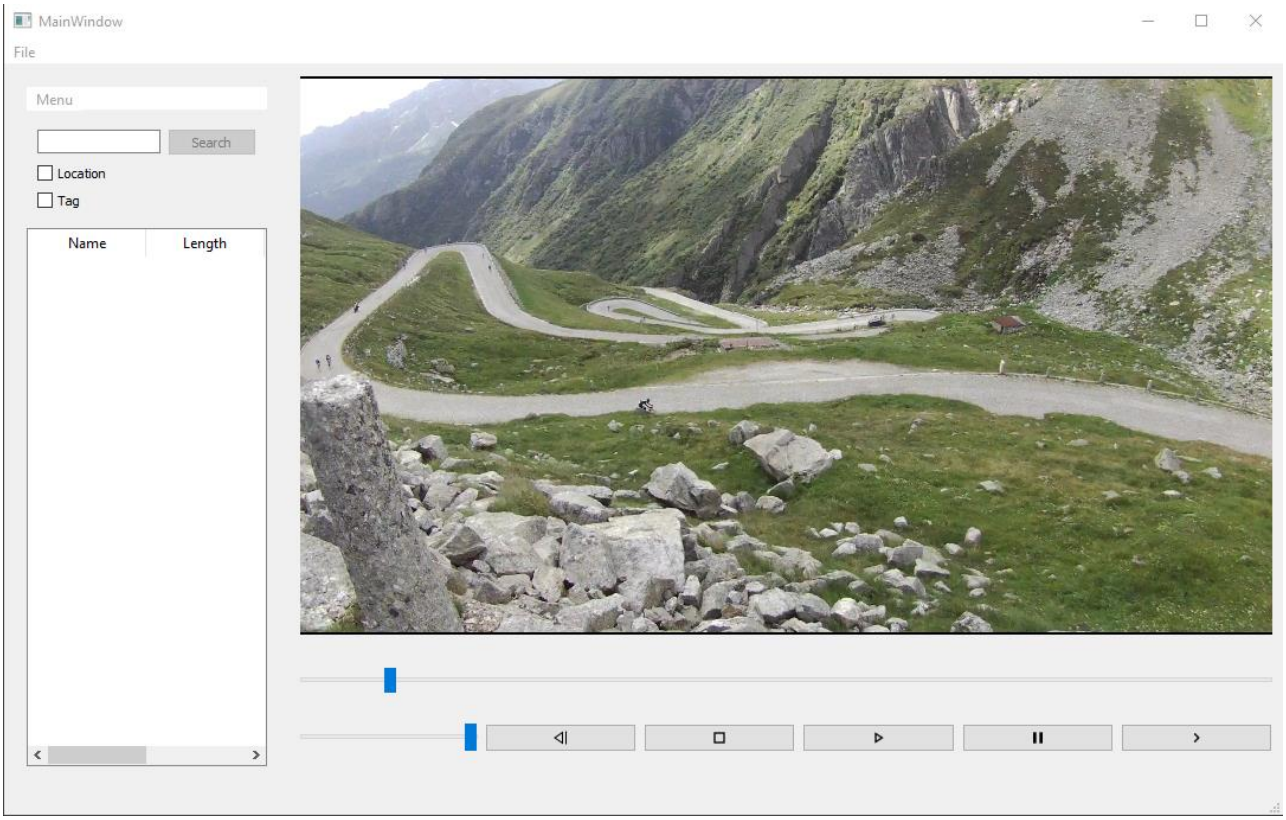


Fig 1.4 Tomeo – Media playing

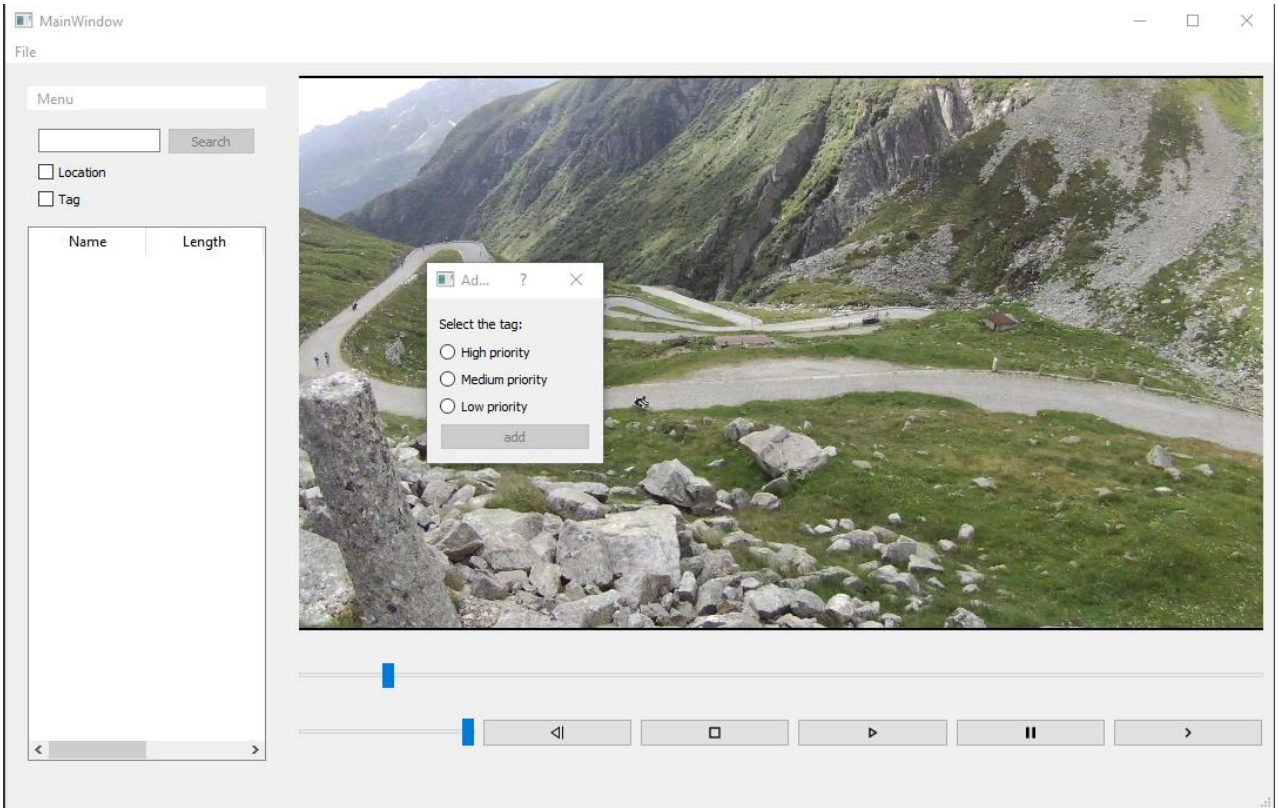


Fig 1.5 Tomeo – Add tag functionality implementation

Differences between the prototype and implementation:

- We decided to move the volume slider to make more space for the collection side bar. Additionally, it keeps it more in line with the layout of the other video player controls.

Evaluation:

For this cycle we used heuristic evaluation, comparing our interface to Nielsen's 10 general principles for interaction design; we chose to use heuristic evaluation because it gave us a good overall impression of the usability of the product. As this was the first cycle, there were many prominent issues with the software before and after implementing the prototype. There would have been too many problems to conduct a participant study, the results would have been

much more easily collated with the evaluation type we chose. Heuristic evaluation is good for revealing many distinct types of usability problem, which makes it more of a sensible technique over something such as cognitive walkthrough – which would have only unearthed problems relating to a specific task. We plan on building on this prototype and this evaluation gave us a key set of problems for which we need to work on within the next cycles. We feel that complicated and time-consuming evaluation techniques should be saved for when issues relating to the program are not as obvious.

The evaluation uncovered a lot of profound issues with the interface and functionality of the program, we realized that we had made it very hard to reorder videos in the collection which contradicts one of our main usability goals. Our interface also had a lot of features which did not properly state their use - we lacked appropriate signifiers and affordances. For example, the search bar had check boxes which filtered based on either location or tag, but there was no way of knowing this was the case and both could be selected at once. There were problems with the video progress bar and volume slider, it was not clear which was which. Moreover, the progress bar did not state the time that we were through the current video, this could cause errors if the videos were exceedingly long, and the user could not get back exactly where they were after a mis click. We realized that in upcoming iterations we need to make it clear what all our buttons do – to avoid the need for documentation. Overall, the evaluation gave us a clear set of guidelines for the next cycle, the changes were accepted, and we will improve on this design in the upcoming prototype.

Heuristic evaluation:

Problem Number	Issue	Heuristics Violated	Severity (/5)
1	Can't see when the video is paused on the pause button	1, 5, 6, 9	2
2	Volume slider and video progress bar aren't tagged	1, 4, 5	2
3	Time not shown on the video progress bar	1, 4, 5, 6	3
4	No tooltips on the navigation buttons	2, 3, 4, 5, 6	3
5	It is not clear what the 'Location' and 'Tag' check boxes do	4, 6	4
6	Sideways scroll for the video library	6	1
7	Unclear when a search has been executed	1, 3, 5, 6, 9	4
8	Changing the order of the videos is constrained	3, 6, 7	3

Heuristic evaluation based on Nielsen's heuristic. Evaluated twice.

Cycle 2:

Prototype:

The main goal of cycle 2 was to add functionality for opening a set of videos that could be played in the media player and added to the video collection. This cycle also focused on fixing heuristic violations in cycle 1. The reason this was the highest priority was because it directly aligned with our main use cases, being able to organize a set of videos into a collection. In the previous cycle it was cumbersome to reorder and add to a collection, this became amplified when

dealing with a large video library. After our evaluation and an internal debate, we produced the idea of having a separate side bar to display opened videos, ready to be viewed and organized into a new collection.

We did two prototyping techniques for this cycle, the first was a sketch done in Excalidraw to determine the new layout. After this was done, we did a workflow layout on a whiteboard so we could see how adding the videos to the sidebar would work.

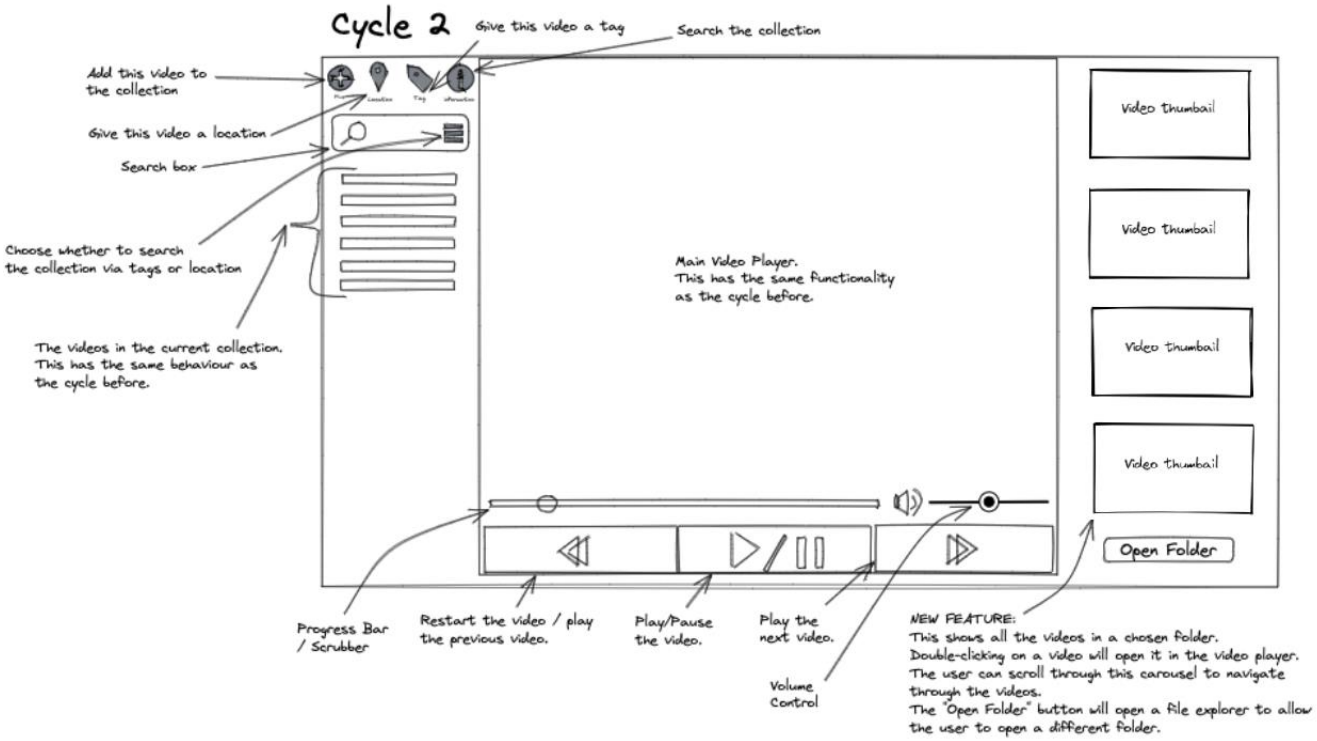


Fig 2.1 Excalidraw sketched layout

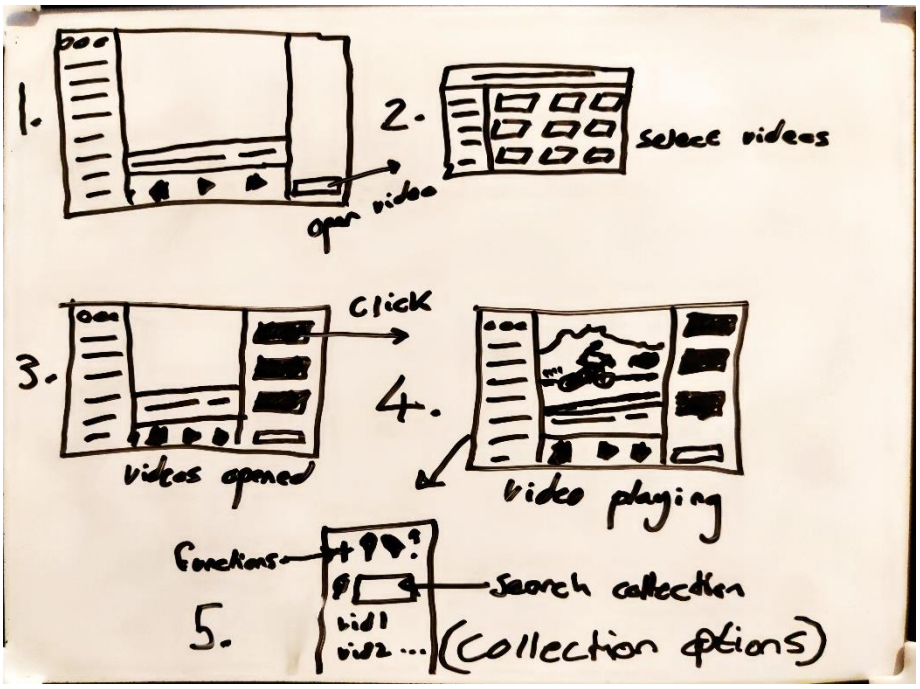


Fig 2.2 Video sidebar workflow whiteboard sketch

The reason we chose this design was we wanted something that was familiar and had perceived affordances that a casual user (as with our use cases) would be able to recognize. As the whiteboard shows, if the user clicks open video, it displays the thumbnails in the file explorer. When the user opens these videos, the thumbnails are now displayed in the sidebar, these now act as inherited signifiers showing the user that these videos can be opened in the media player. In our use cases, we have presumed that our user base knows how to use a computer to a reasonable degree; having the thumbnails represent the videos in the sidebar represents a consistent mapping between how the file manager works and how our program operates. This reduces our need for extensive help documentation or overwhelming text signifiers, producing a clean user interface. We believe this style is in line with design patterns in similar software, which was a large motivator to adopting this design. If we chose to use the video name instead of thumbnail, then the user might believe this affords opening in the media player. Displaying the thumbnail dramatically closes the gap between perceived and actual affordance, explicitly stating what kind of operations the sidebar supports.

For this cycle, we chose two different prototyping techniques, we engaged in a lengthy discussion about the design, attributes, and workflow so it made sense to

articulate them in more than one form. Initially we drew a few different prototypes in Excalidraw and eventually concluded that Fig 2.1 was the best design, we felt like this was extremely important as it was such a prevalent feature in our program. We were researching various possibilities related to the functionality we wanted to implement, sketching on Excalidraw allowed us to quickly change the design and evaluate how it would work in our program. Next, we decided to draw a workflow on a whiteboard, we did this because it allowed us to cement various paths of reasoning which led us to the sketched design. It enabled members of the group to understand how an end user might interpret the thumbnail layout and how they would be inclined to react to the presented affordances. There were a lot of ideas between the group for this feature so low fidelity prototyping made the most sense.

Code:

Here is a quick demo of the improvements -
Cycle 2 Quick Demo

Differences between the prototype and implementation:

- The layout of the bottom navigation buttons is different to the prototype, although the play and pause were still merged into a single button.
- The menu buttons in the top left and the collection searching functionality were edited, we realized we had more functions than could be presented in four buttons, so the design had to change.
- We changed the backend of the collection functionality considerably; this was not apparent in the prototype but allows us to easily deal with complex functionality in the later cycles.

Evaluation:

For this cycle we have decided to use a cognitive walkthrough style of evaluation, we will use this type of evaluation as during this cycle we have implemented a feature which is supposed to afford itself to ease of use and familiarity. Additionally, we have created a massive amount of backend functionality which would not lend itself well to a heuristic evaluation. We also believe it is too early in development for other types of evaluation techniques. We believe cognitive walkthrough is perfect as the new sidebar is a complicated feature, to evaluate it sufficiently we need to put ourselves in the mindset of our userbase (users relating to our use case scenarios and demographic). In doing so, we will be able to more clearly see issues that users might have when trying to accomplish tasks which we might become ignorant of as developers.

The evaluation showed that we had mostly designed an easy-to-follow workflow, there were only a couple of trivially fixable issues. It was immediately obvious how to perform the actions in the walkthrough, until we got to the point where you want to view a video in the media player. After a brief time of clicking, you would be able to work out how these videos are displayed but we want to make it immediately obvious how this would be performed. This warrants a change that will be fixed in the next cycle. Furthermore, performing actions on a video that is currently open potentially requires traversing four different menus. Although these menus have icons it is not an ideal design; we want actions relating to an open video to jump out to our users and have an obvious, easily repeatable workflow. Overall, we are happy with this cycle and have decided that the changes will be accepted. We will improve upon this design and fix issues in the next cycle.

Cognitive walkthrough:

- Task: open a set of videos, find a specific video, watch the video, tag it, and then add to the collection

Step	Correct Action?	Perform?	Associate and interpret?
Select open videos	✓ - appropriate signifier	✓ - open space with a relevant button displayed	✓ - file explorer is a basic computer function
Select videos in the file explorer	✓ - sufficient file explorer tooltip shown, only video files shown, and it is made clear	✓ - user base should know how to use the file explorer	✓ - thumbnails now displayed in the side bar gives adequate feedback
Scroll the list to find a specific video	✓ - scroll bar appears and thumbnails identify videos	✓ - scroll bar is a basic functionality of many programs	✓ - videos scroll when the scroll bar is used
Click the video to open in the video player	× - it is not clear how to view these videos (medium severity)	✓ - user expected to be able to click	✓ - video is instantly shown in the central media player

Click the play button	✓ - the play button is a familiar action	✓ - clicking play works as expected	✓ - the video plays and the play button changes to a pause button
Click the tag button and tag the video	✓ - appropriate text signifier on the tag option	× - traversing menus is required, menu symbols are not obvious (low severity)	✓ - a text box pops up with appropriate instructions on how to tag
Click the add to collection button	✓ - appropriate text signifier on the add to collection option	× - traversing menus is required, menu symbols are not obvious (low severity)	✓ - the video name and data is added to the collection sidebar

Cycle 3:

Prototype:

The main goal of this cycle was to provide the application with aesthetic rework. It was also designed to include a vertical-rolling carousel that displayed the videos from a collection to the user. The cycle also focused on improving the previous heuristic discrepancies that we came across during our evaluation. The rework was prioritized during this cycle to improve usability for the end-users. We realized that some of the functionality introduced in Cycle 2 was not as intuitively understood due to the way it was presented in the application.
The lack of perceived affordance became clear when a team member pointed out that he could not understand how to interact with menu icons implemented in Cycle 2. Upon further discussion within the team, we concluded that the icons conveyed the impression of a single functionality, verses what we tried to implement (a drop-down menu system). Alongside this, it pushed us to take a fresh perspective on the design of the application, which led to Cycle 3 being set for reworking the UI of *Tomeo.*

Fig 3.1 Modularly drawn layout

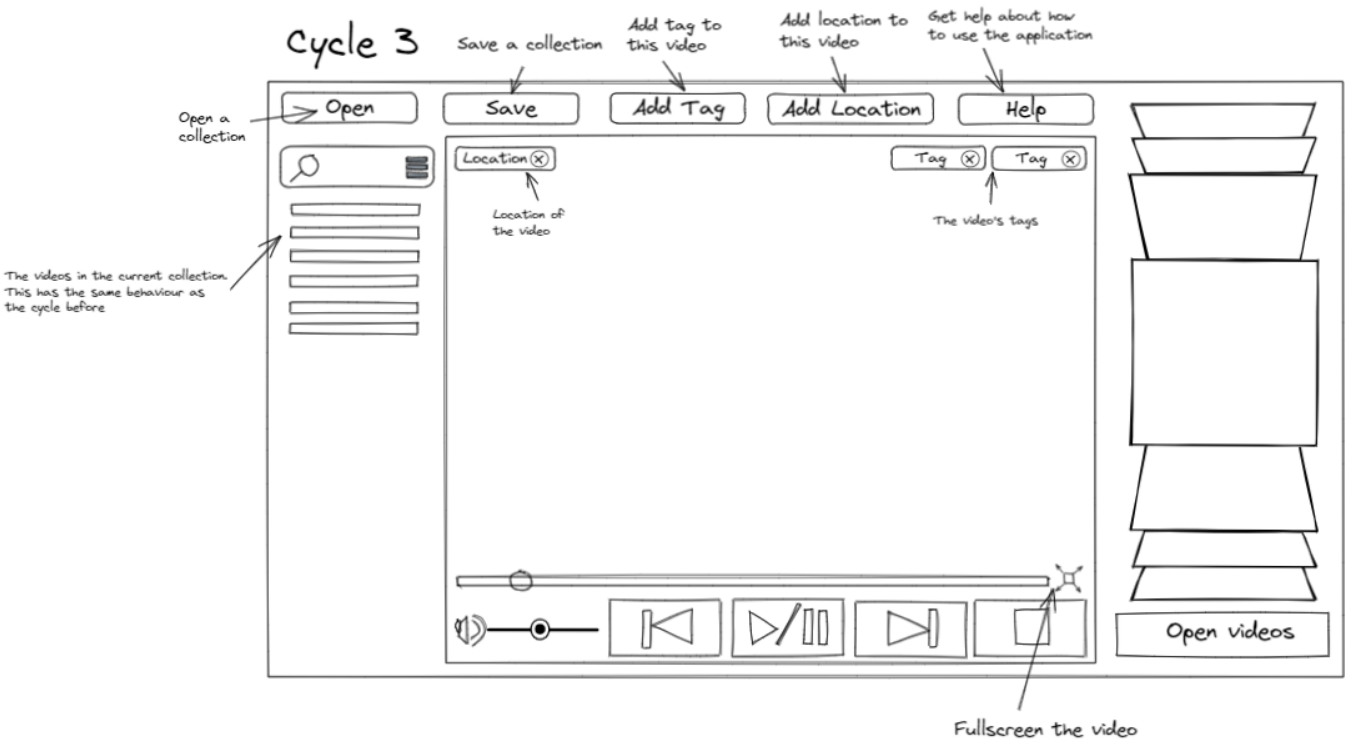


Figure 3.2 Excalidraw sketched layout

We did two prototyping techniques for this cycle, the first was a real-life model of the new layout to support the design brainstorming session. After this was done, we did a sketch on Excalidraw to finalize the design and make it presentable, as well as easily shareable.

We chose this design rework to improve on the affordance failure of the previous Cycle. We implemented better signifiers in this iteration that users can recognize instantly from previous experience (we assume in the use case that the users have experience utilizing basic applications on a PC). This involved scrapping the icon-based menu for a more standard menu-bar with relevant drop-down menus. The style follows more mainstream applications like Adobe Premier and VLC Media Player in its presentation of these affordances. The menu items have been grouped based on how close they are functionality wise. For example, the 'Open' drop down menu will contain functions to open videos and collections. We have also introduced some user feedback when playing videos, by creating pop-ups over the video that show the user the location where the video was taken and its associated tags. These pop-ups can be closed manually by the user or disabled temporarily by the user. We will also try to implement a full-screen option for users to view media across their entire screen. Another major implementation will be the introduction of a vertical carousel-view that will display a collection of videos opened by the user. This will build on from the previous static collection-display view to give the user a smoother viewing experience.

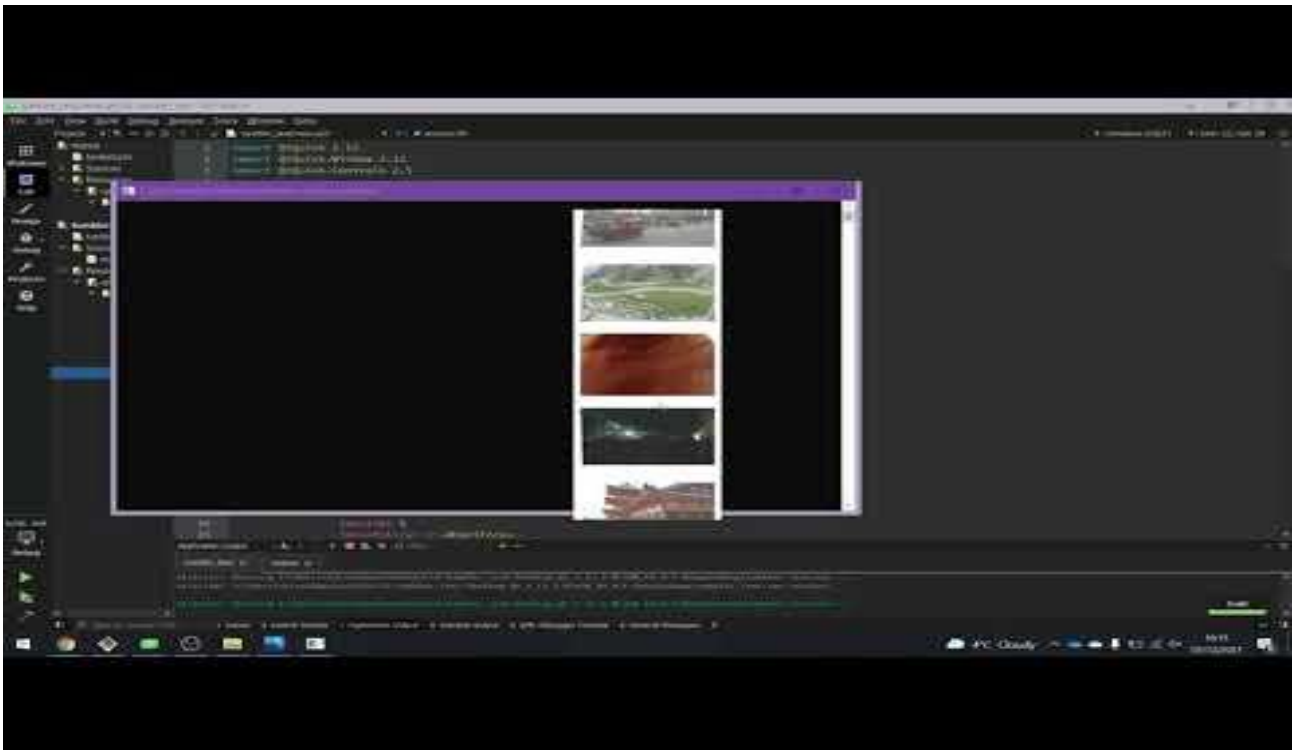
The prototyping technique used in this Cycle involved creating a paper-based model of the application GUI. We found this technique to be highly effective as it allowed the team to sit down and re-position the different components (individual paper strips) to describe our reworked design. It also served as a session where the whole team could sit down and truly convey their ideas across the board, explaining why they chose to set the components where they did. Since we had an innovative idea of 'what' components we wanted in the application from the previous Cycle, this session was focused more on 'where' we were going to have the components placed. Once we finalized the design, we drew the design up on Excalidraw for easy sharing across the team and to have a presentable looking final design. This exercise also had a positive effect on the team, allowing us to narrow down our focus on what we really wanted the application to look like, design-wise.

Code:

Differences between the prototype and implementation:

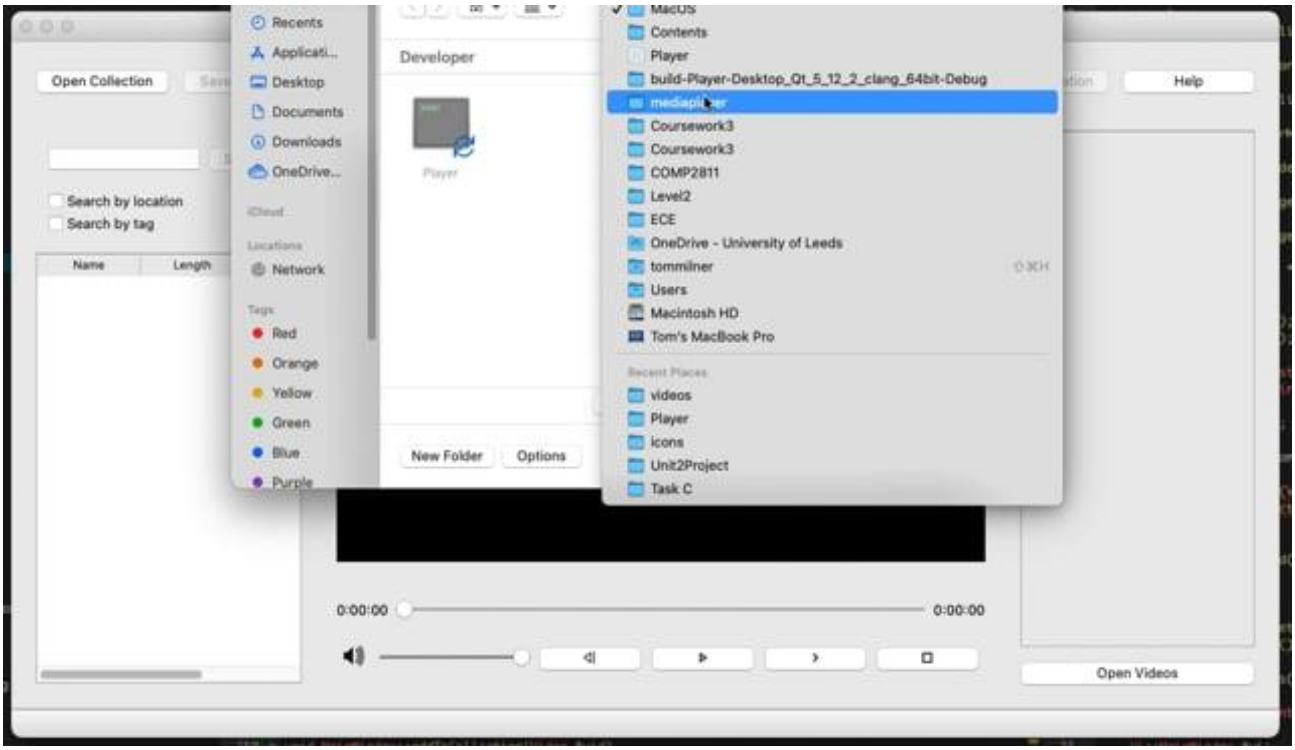
- We coded and implemented a QML carousel for the right-side video collection, unfortunately this did not work on the lab computers, so we had to revert to the old implementation
- Tumbler evidence video:

Tumbler video



- We added 3 more collection control buttons for removing videos from the collection, removing tags from videos and for adding a video to the collection.
- Implementation video:

Cycle 3 Demo.mov



Evaluation:

For cycle 3 we have used heuristic evaluation again, comparing our program to Nielsen's 10 general principles for interactive design. We have chosen this evaluation type as we have added a lot of new features to our program that we will need to examine and check the issues that can arise from these features. Going through the evaluation and seeing what the issues are with specific tasks helps with finding a solution to these problems so that the next cycle can improve by fixing the mistakes found. At this point in the project, we almost have a fully functional program with a lot of useful tools and features. Therefore, this evaluation will be useful for finding small problems that are noticeable but do not cause the program to be unusable.
These issues could be small cosmetic issues with the interface or some functionality that is yet to be added properly. Additionally, as we did not do a heuristic evaluation in the last cycle, there is likely to be a build-up of violations inherited from that cycle.

The program's usability at this cycle is almost perfect, allowing the user to open, edit and save collections of videos. They can also add tags and locations to each video, which will show on the video player window as an overlay at the top, allowing the user to quickly remove a tag or location by just clicking the X on it. The user can also put the video into Fullscreen mode by pressing the button on the bottom right of the video player.
The buttons for adding locations, tags and opening and saving collections have also moved from the top right. They will now be larger, more obvious buttons at the top of the screen which gives the user more information about what each button does. On top of this, there is also a new 'Help' button which the user can click to find out how to use the application as a sort of tutorial.

The errors we encountered when going through cycle 3 of the project were to do with the video collection and some aesthetic issues. One error we had was that, when we added a new video to the current collection, the length of the video would not appear/update on the list on the left side. Currently, the help button we have implemented does not open any tutorial document. An aesthetic problem we came across was that the forward and backwards buttons for the video do not match, this is not a massive problem but for consistency, it would be better if they were the same style. We do not have any keyboard shortcuts for opening or saving a collection or opening videos from a folder. The video folder on the right does not revolve around like a carousel like in the design, this is due to an error with QT and UML files making it incompatible, however this is not too severe as we can still use a scroll area instead. We have also identified that a user can add multiple videos to the video folder on the right but if they want to remove multiple videos, they must go through and remove each one individually. The final issue we found is that a user cannot create a new collection by adding videos to collection and then saving, instead they must open an already made collection and add the videos from the right to that collection.

Heuristic evaluation:

Problem Number	Issue	Heuristics Violated	Severity (/5)
1	When adding a new video to the current collection, the length of the video is not shown	1, 2, 4, 8	3

2	Currently the help button does not lead to any instructions on how to use the program	6, 10	1
3	The symbols for going forward or backwards are not the same	2, 4, 8	1
4	No keyboard shortcuts for saving or opening videos/collections	3, 7	1
5	Video folder on right side does not scroll through like a carousel	3, 8	1
6	There is a way to open multiple videos but no way to delete multiple videos	3, 5	3
7	There is currently no way to create a new collection, only able to open a previous one and save it.	3, 4, 7	4

Heuristic evaluation based on Nielsen's heuristic. Evaluated twice.

Cycle 4:

Prototype:

Following Cycle 3, there were two major goals tackled in this cycle. The first was to implement the "Map" functionality, and secondly to rework the way the collections list was displayed on the UI. While the team was quite satisfied with the level of technical detail that was implemented, we wanted to bring forth an idea that was not mainstream in the media player market. This need birthed itself in the form of a 'Map" functionality. Aside from this, we also chose to rework the collections list that was displayed on the left-side of the app. This was prioritized in this cycle because the previous implementation of it did not meet our standard of what we considered 'user presentable.' Also, the existing design of the

collection list required the user to scroll to the right to view further details of each file, which while clearly presented an affordance to do so, was also unprofessional.

Fig 4.1 QT Designer prototype

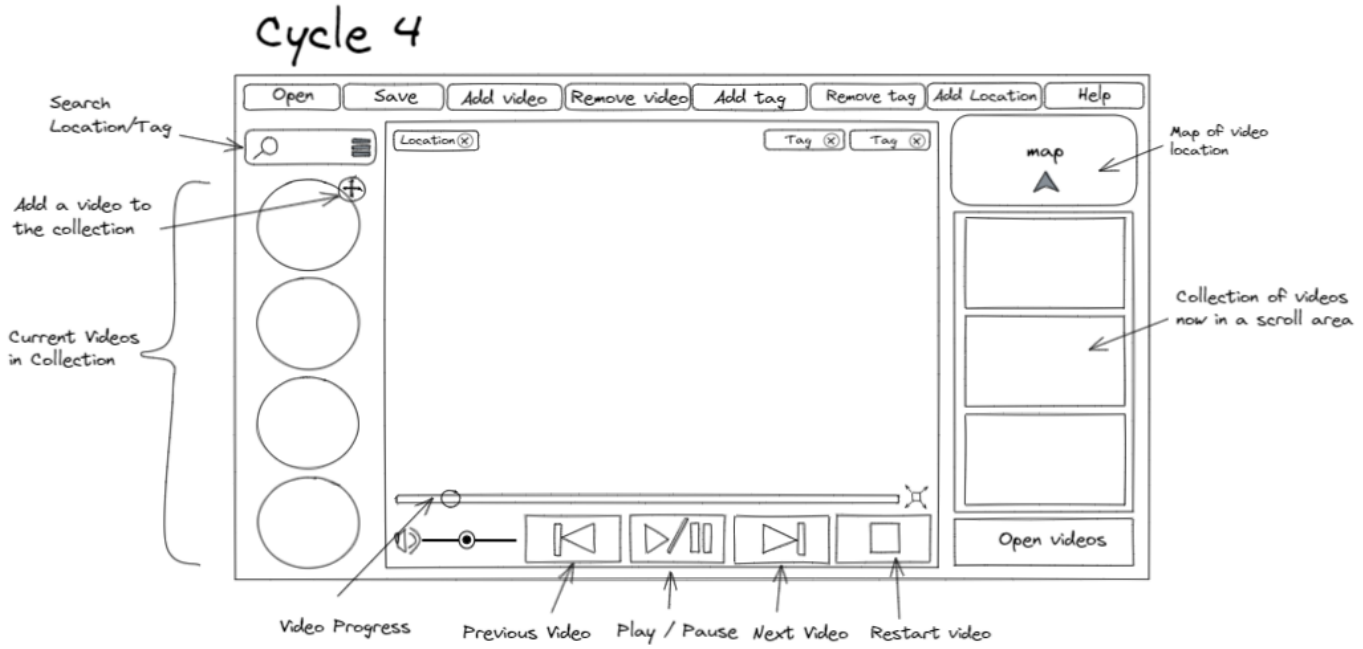


Fig 4.2 Excalidraw prototype

Our prototyping technique for this cycle was to create a layout for the Cycle in Qt Designer. Aside from letting us reuse code from the layout itself, it also gave us an incredibly detailed perspective at what the end GUI would look like.

Our design in this Cycle included some of the improvements we wanted to make based off the previous evaluation. This included fixing some of the errors involved in adding/removing videos from the collection. The major design implementation here was the introduction of the "Map" functionality. The Map will be designed to take co-ordinates off the file metadata for a video and display a marker on the Map centred on the location where the video was captured. A collection of videos captured in the same location can be grouped and displayed as a collection marker. As stated in Scenario 3, we expect the map widget to present itself like any other virtual mapping application, and frequent users of Google Maps (such as our userbase who travel frequently) will find it quite familiar to navigate. It will also contain the ability to scroll into and out of areas, as expected by its perceived affordance. The mapping of the widget also works adjacently to the user's intuition, as most applications (especially video game UIs) have their map widgets placed in the top-right corner of the screen.

The rework of the collections list will remove any unnecessary information from the immediate screen. It will also streamline the process of adding videos from the open folder on the right side of the application to the relevant user defined collection using a drag-and-drop functionality (or similar method).

Code:

This a demo for cycle 4 implementation
Cycle-4 Video

Differences between the prototype and implementation:

- Compared to the prototype we were not able to embed the search box into the collection viewer, however the functionality was still intact.
- The prototype had a full screen button, whereas the actual implementation did not. The reason was that other features seemed more superior at the time being
- The add button above the collection viewer is not implemented, since it did not display any real functionality as we already have an add video button
- The prototype has the play/pause buttons with the progress bar as overlays, however this did not look appealing when implemented in the actual software, thus we did not add it as part of the implementation

Evaluation:

With the map added and the collections now displayed correctly, the program is extremely near completion. All the functionality has been added and

so for the final evaluation we decided to go with a Cognitive Walkthrough, so that we could analyse many parts of the program in one swoop.

We decided on a task that would make the user have to use many parts of the program without it being too similar to the last Cognitive Walkthrough:

"Open a pre-existing collection and remove a video from it. Then, change the location to 'China' on a different video and watch it in Fullscreen."

User: Familiar user; traveller.
Context: Laptop, user is on a bus inspecting their latest videos.

Cognitive Walkthrough:

Task:

Step	Correct Action?	Perform?	Associate and interpret?
1. Select "Open Collection""	✓ - appropriate signifier	✓ - open space with a relevant button displayed	✓ - file explorer is a basic computer function
2. Select the collection from the file explorer	X – It is not fully clear which files are the collection files, as they have a standard .txt file extension. (High severity)	✓ - User base should know how to use the file explorer.	✓ - Videos of the collection are displayed on the left – adequate feedback.
3. Scroll through videos in collection.	✓ - thumbnails can be seen just out of reach; numbered thumbnails hint that there is a continuing list.	✓ - scroll bar affords scrolling.	✓ - contents of the side bar moves as the user scrolls.
4. Select video to delete	✓ - one video is selected at the start, implying the user can select others.	✓ - user expected to be able to click.	✓ - video is highlighted.

5. Right-click video to delete	✓ - tooltip implies that the user can access a menu	✓ - user expected to know that right click usually brings up a menu – external consistency with all mouse-compatible computers.	✓ - context menu appears.
6.Click "Remove"	✓ - familiar language.	✓ - familiar menu – uses native externally consistent with Operating System.	✓ - video disappears from left side. X – not immediately obvious that the video has disappeared (low severity)
7. Select a different video (repeat of steps 3-4).			
8. Double click video to open in player	✓ - familiar action; video is highlighted X – still not immediately obvious (low-severity)	✓ - Double-clicking is a familiar gesture.	✓ - video appears in player, relevant location and tags appear. Video immediately starts playing.
9. Click location	✓ - we can see the location we want to change.	✓ - icon button is a familiar signifier.	✓ - change location prompt displayed.
10. Start typing	✓ - blinking cursor suggests typing.	✓ - typing works as expected	✓ - letters appear, "Add" button changes colour.
11. Click "Add" button	✓ - appropriate signifier	✓ - familiar design - consistent with OS.	✓ - dialogue appears that must be accepted for user to move on.
12. Click "Ok"	✓ - appropriate signifier	✓ - dialogue design is	✓ - dialogue disappears.

		consistent with OS.	✓ - location overlay is instantly updated.
13. Click Fullscreen button	✓ - appropriate signifier; consistent across video sites.	✓ - clicking is a familiar gesture.	✓ - Video enlarges to fit screen; rest of the application is hidden.

Cycle 5:

Prototype:

Being the final cycle, we decided to use this iteration of development to refactor our code, touch up on the details of the UI and implement a consistent stylesheet across the application. Having laid out all the widgets, an exceptionally good majority of which we managed to fully implement functionality-wise, the last step was to ensure that the UI was visually appealing to our target audience. We utilized our knowledge of 'colour theory' and 'grouping' to effectively place our widgets in their final positions across the application UI. In the previous evaluation, we realised that some affordances (such as opening videos) were not too obvious. This led us to also implement short-cuts (which have been designed with familiarity in mind). The list of short-cuts has been added to the supplied 'Help' document in the project directory. We also focused on refactoring some of redundant code and improving the efficiency of some of the other parts of the source to ensure that the final application would be robust and ready for the next level of updates post launch.

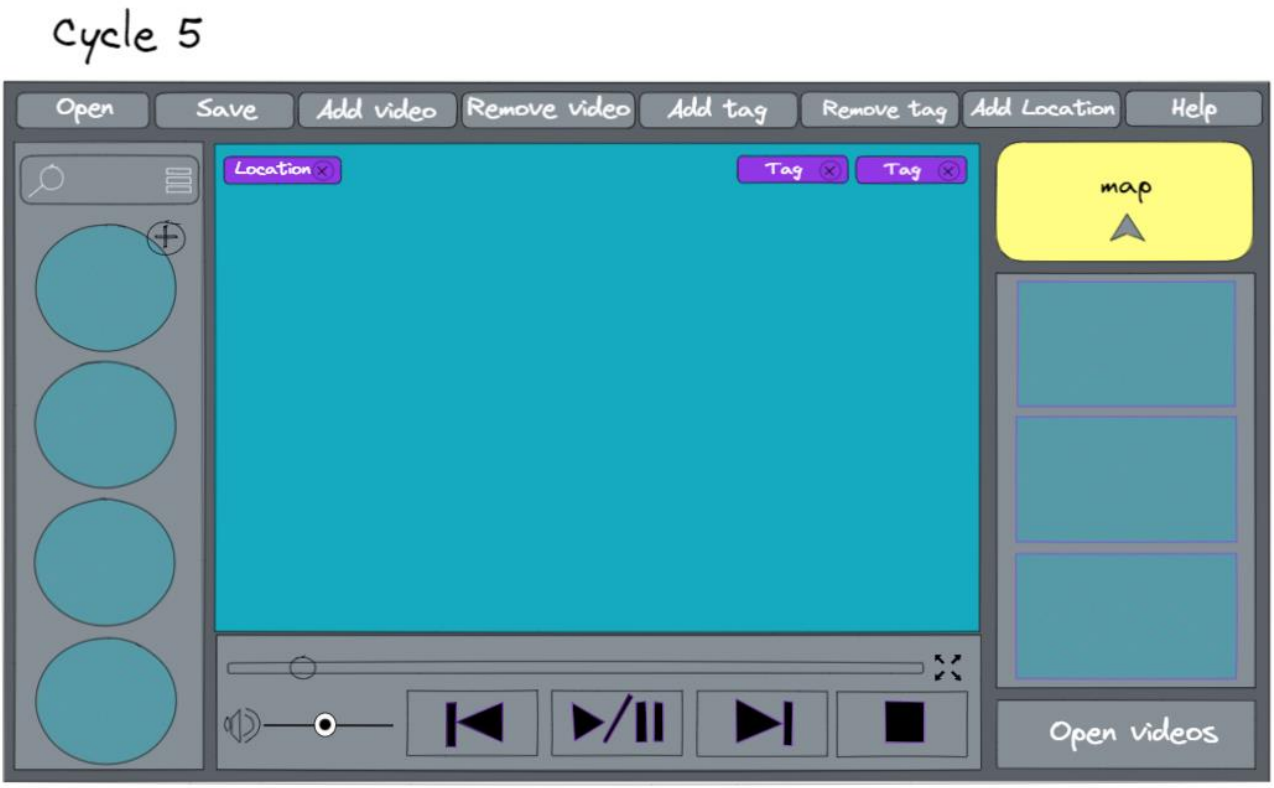


Fig 5.1 Excalidraw dark mode prototype

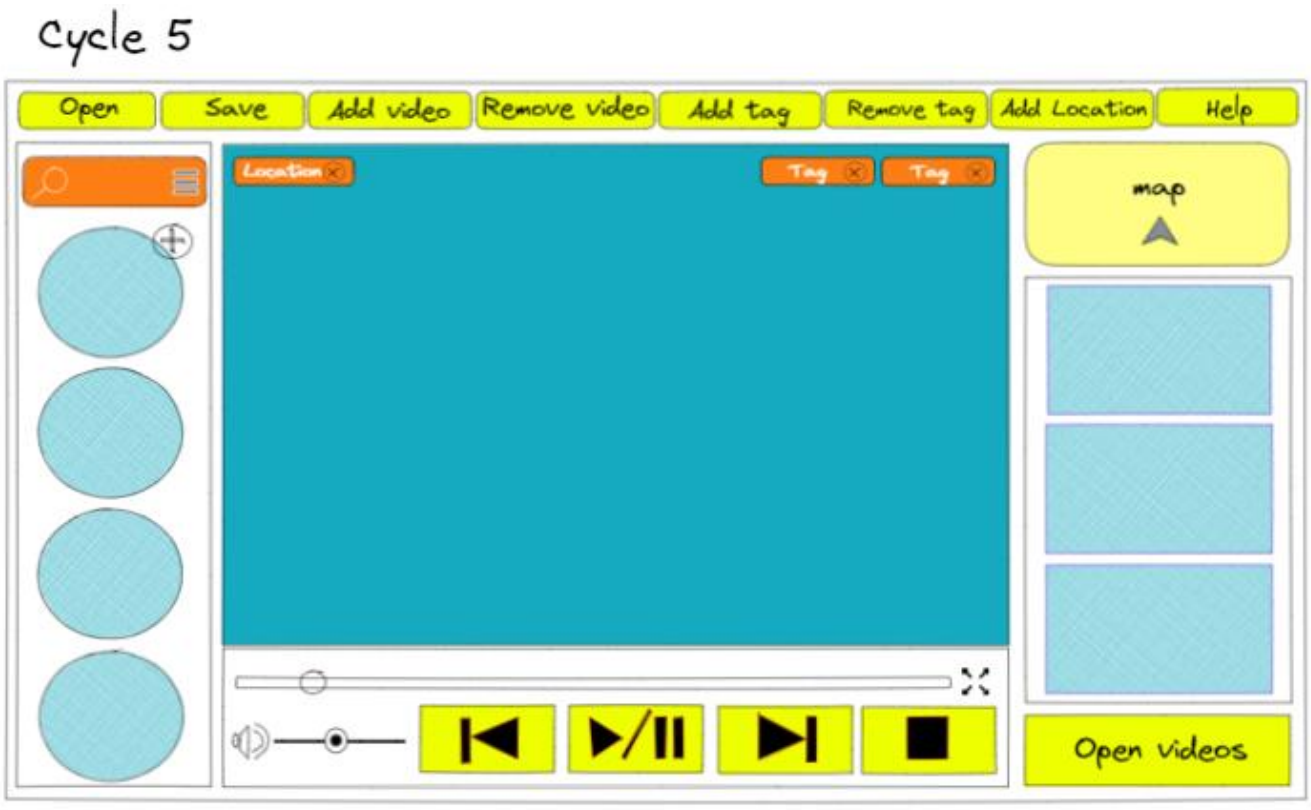


Fig 5.2 Excalidraw light mode prototype

A fresh product into the media player market, it is essential that *Tomeo* makes a good impression on our intended user base, as well as a wider audience. Having achieved the best possible implementation of our designed application within the given timeframe, the last essential step was to ensure effective presentation at first glance. This involved resizing and realigning the widgets, standardising the spacing between the component widgets and establishing a uniform stylesheet. The rework will be done alongside the code refactoring, which will also allow us to have a final check through the program internals. The above design created on Excalidraw is not completely true to detail. The blue represents areas where video thumbnails and the actual media is to be displayed. We have decided to go with a combination of dark colours, mainly shades of grey, black and white for writing and highlights. The colour grey gives users the impression of neutrality, allowing it to appeal to a wider range of audience. The presence of black in the background will bring the lighter grey widgets to the forefront, and the white will allow us to highlight icon writing and borders due to the contrast it provides. As our application is primarily aimed at outdoor and extreme-sports enthusiasts. Hence, the neutral colour scheme will also help to further emphasis the media that is being viewed and managed on *Tomeo*.

We chose to complete the final design on Excalidraw to save time and allow for quick changes to be implemented. At this stage in the cycle, we were discussing the different options for stylesheet colour combinations and widget sizing. These discussions meant frequent changes, which had to be reflected in the design in real time. Excalidraw proved to be the ideal tool for the purpose at hand and therefore justified its reuse in this cycle as a prototyping method. During the last cycle, the team had to work remotely due to other time constrains, which meant the design had to be edited by multiple people that were connecting online. Again, Excalidraw was the most sensible choice when it came to designing the final cycle. It also allowed us to quickly share the design in a portable format to the rest of the team, saving time and improving the overall efficiency of the team.

Code:

This a demo for cycle 5 implementation
Cycle-5

Differences between the prototype and implementation:

- The main difference was the spacing, such as the right side and left side had the same spacing whereas the middle layouts did not. We did try to design the spacing correctly however we were not able to implement it through those layouts.

Evaluation:

For this cycle, we chose a controlled participant study to evaluate our prototype. We chose this technique because this cycle focused on the style guide and spacing, which is imperative to the connotations that our program produces. As colour theory and preference to various levels of contrast are entirely down to individuals, culture, past experiences and intrinsic taste – it made sense to conduct a participant study. In many sets of heuristics, consistency, visibility and aesthetic design are key headings; we want the user to be able to efficiently navigate and visually process the features of our program. As with our use cases, we are targeting inquisitive individuals who are casual users of our program. Evaluating the design internally within our group would not truly support our use cases, which is ultimately how we chose this evaluation technique. As this is the final cycle, we felt that it was appropriate to evaluate the prototype so that we can use the data within the same cycle. Additionally, the chosen technique gives us an idea of how unbiased individuals feel about the design.

For this evaluation, we asked participants a series of questions relating to the dark mode design, light more design and a comparison of both. For the dark mode 4/5 participants found the design visually appealing, only 1/5 participants said the same about the light mode. This was the same for each colour design being in line with modern computer software. In contrast, the participants found the light mode buttons easier to read and there was no difference in what participants thought about the clarity of the functionality.

An important statistic we received is that the participants felt that the dark mode design was overwhelmingly more professional than the light mode, a 4.2/5 average rating compared to a 2.5/5 average rating for the light mode design.

This data reinforced itself during the comparison section; all participants would choose the dark mode design over the light mode. After collating the data, we have chosen to move forward and implement the dark mode prototype.

Design 1:

1. *Do you find this design visually appealing?*
   Yes, somewhat, yes, yes, yes

2. *Is this colour design in line with modern computer software?*
   Yes, somewhat – could be darker, yes, yes, yes

3. *Out of a maximum of 5, how do you rate the professionalism of this design?*
   5/5, 3/5, 5/5, 4/5, 4/5 = 4.2/5 average

4. *Out of a maximum of 5, how easy is it to see the button text on the top of the design?*
   4/5, 3/5, 4/5, 5/5, 5/5 = 4.2/5 average

5. *Out of a maximum of 5, How clear is the functionality of this design?*

   4/5, 4/5, 4/5, 4/5, 5/5 = 4.2/5 average

Design 2:

1. *Do you find this design visually appealing?*
   No, Way less than design 1, No, No, Yes

2. *Is this colour design in line with modern computer software?*
   No, Not quite, No, No, No

3. *Out of a maximum of 5, how do you rate the professionalism of this design?*
   2/5, 2/5, 3/5, 2.5/5, 3/5 = 2.5/5 average

4. *Out of a maximum of 5, how easy is it to see the button text on the top of the design?*
   4/5, 3.75/5, 5/5, 5/5, 5/5 = 4.55/5 average

5. *Out of a maximum of 5, How clear is the functionality of this design?*

   3/5, 4/5, 4/5, 5/5, 4/5 = 4.2/5 average

Fig 5.3 A sample of anonymised collated participant study data (Document available in the submission)

Demo Video

Final Video

Ethics Statement

We have committed to complying with the University of Leeds Ethics requirements during the evaluation of Cycle 5, where we conducted a controlled study. We have created appropriate information sheets and consent forms to comply with the University Ethics standards and accepted practices in storing data, receiving appropriate consent and selecting participants. We also made sure to provide the research participant privacy notice.

When asking participants whether they would like to take part, we made sure that they read all the required documents. We also reiterated important details, for example, being able to back out at any time. We made it clear that they were not required to take part, and there would be absolutely no penalty in refusing to do so. Even though the documents were exhaustive, we had a moral and legal obligation to look after the best interests of all our potential participants.

No part of the study was conducted until each participant had completed the consent form, we took a secure copy and stored it appropriately. The participant was asked to keep all of the documents, including the consent form.

As part of collecting research data, we have made sure to let participants know exactly where and how their research data will be used. We have made them aware that their data, which for the sake of this project does not involve sensitive personal information, will be shared with the concerned parties at the University of Leeds. During the tenure of this project, the data collected will be stored securely on an online cloud storage medium offered by the University of Leeds - as stated in the University regulations webpage. However, due to the nature of project, we have no control over the lifetime of the data once the project has been submitted. We told participants that the school would require the consent form copies, but evidently this is not the case, therefore we will destroy the consent form copies.

Here is a link to the forms we created for our controlled study:
Information sheet and consent form
The documents are also available in the submitted zip, we are required to remove the research data from OneDrive after submission, but the data will be available in the submission.

Description

Tomeo is designed and developed using the Qt Creator application and its associated libraries. Initially, Tomeo will have to be compiled and its libraries linked using the Qt.pro file.

Upon running the program, *Tomeo* will display its GUI, allowing users to open local videos and watch them on screen. Having been designed with user familiarity in mind (based on patterns of media player layouts), it should be relatively simple to navigate through the application. The users can arrange videos in their own collections, edit the video attributes by adding tags, remove videos from existing collections and in essence, operate the UI like they would a workspace. The interface also allows for sorting and searching through files, with a full list of features being provided in the 'Help' document.