

## Process document for Video Player

### Authors

Michael Wiciak

William Neild

Brooklyn Mcswiney

Lai Ting Yeung

### Introduction

This group has created a responsive and rich-in-functionalities video player, using the desktop Qt framework (version 5.15.2). We have chosen to call this video player HTPV (Hot Tomato Video Player) in iteration 3 and henceforth this project will be addressed as such. Throughout our three iterations, we followed agile methodologies, as shown by our Kanban board screenshots provided. Each iteration involved a short sprint, first, we prototyped our goals, then attempted to implement them for a set number of days and evaluate what we did at the end of the sprint, often using direct user feedback. The evaluation from previous iterations influences our next prototypes. HTPV was designed initially as a tablet application but due to its responsiveness, it could easily be used for both Desktop and Mobile devices, evidenced in the Further Material section.

### First iteration

#### Prototyping

This cycle aims to implement basic functionalities into Tomeo that any video player needs, such as basic media player controls (Pause, and Play buttons). The second goal is to provide a more user-friendly interface, getting rid of Tomeo's unwanted functionalities like random video displays. The reasoning behind those goals is to allow users to start using the basic functionalities and experience our initial user interface designs, allowing us to collect feedback early in production. This follows the agile methodology we have chosen for this project and helps us shape

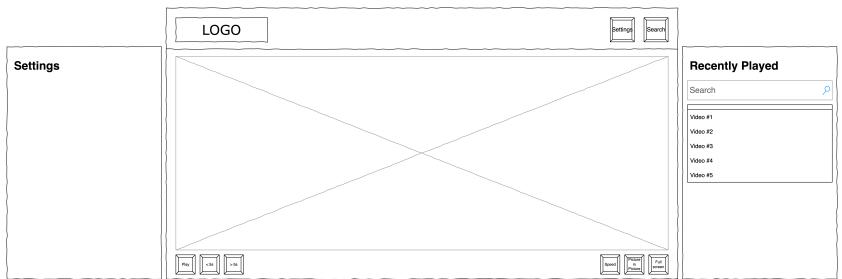


Figure 1: Wireframe model of the video player made using [draw.io](#)

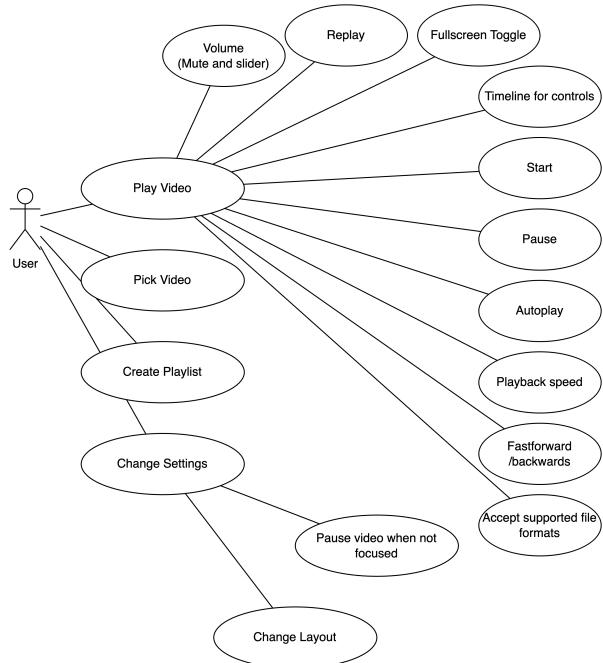


Figure 2: Use Case diagram for HTVP using [draw.io](#)

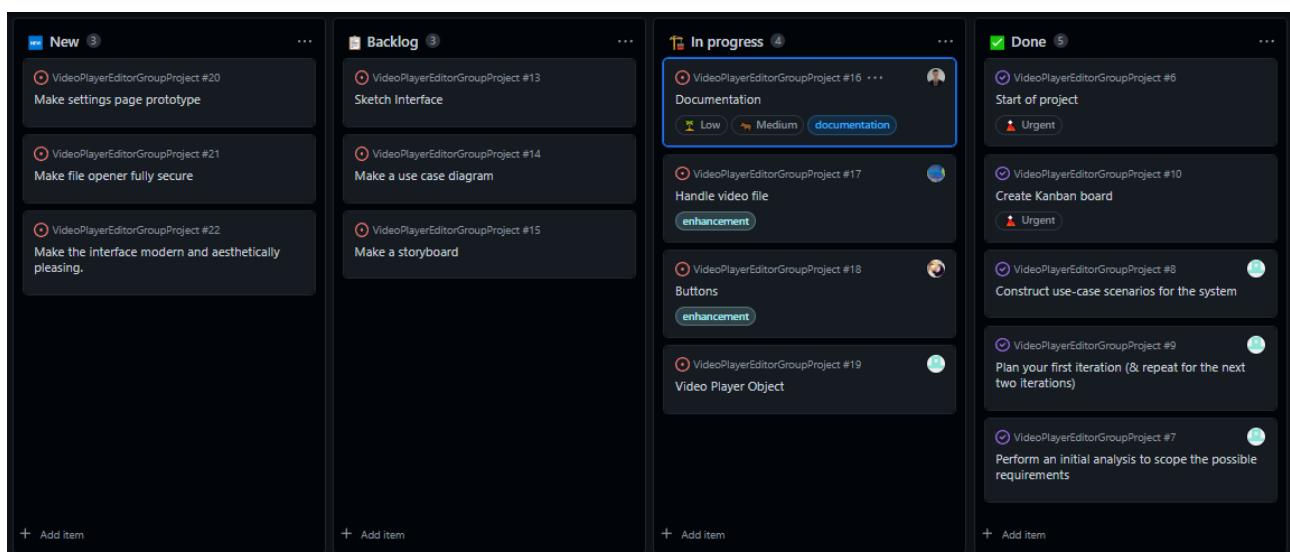


Figure 3: Kanban board after prototyping meeting during the first iteration

future iterations to the exact requests/needs of users, rather than our speculation of what the users might want.

For this iteration we created a wireframe model of HTVP, illustrating our initial UI design, as shown in Figure 1, with the motivation of it being user-friendly. In Figure 2 we created a use case diagram, showcasing all of the functionalities we feel the system needs to have at the end of all three iterations, allowing us to create UI designs that fit all those methods and make them accessible to users. This grants us to pick a few desired functionalities and attempt to implement them in this first iteration, and see early if there are any technical difficulties so they can be addressed now so there is no issue with this in later iterations, with motivation to make HTPV a useful and working software.

VLC's (1) and YouTube's (2) user interface and functionalities have inspired our initial prototypes. For the first iteration we have chosen to work on: a pause and a play button, making the video selection no longer random, importing videos, a volume slider, a mute button, a skeleton for the settings page and creating a clearer UI layout.

## Implementation

By the end of the first iteration, we manage to implement all of the functionalities we wanted to achieve in the first iteration. Due to our agile methodology, during implementation, we created out-of-the-scope functionalities in this iteration as we felt they were needed for a better user experience. These include fast-forward, rewind buttons, volume slider, playback speed, current video timeline, and running the program without requiring a command line.

Some of the initial requirements we prototyped have changed during implementation: the setting is now a side panel instead of a pop-out which makes more sense as Tomeo is meant to be a mobile-first application.

Although those requirements were implemented, they posed technical difficulties: Adding a video was harder than intended as the video player had to be separated from Tomeo.cpp, bug with the current Qt version made it so only WMV works on Windows so no playback speed support on Windows.

## Evaluation

Evaluation for this iteration will be done using qualitative surveys based on

## Tomeo Feedback

3 Responses 09:50 Average time to complete Active Status

1. How would you rate your experience from 1-10? (1=Very bad, 10=Very good)

3 Responses 8 Average Number

2. How easy was it to open a video file in Tomeo? (1=Very hard, 5=Very easy)

3 Responses 3.67 Average Number

3. How easy was it to watch a video in Tomeo? (1=Very hard, 5=Very easy)

3 Responses 4.67 Average Number

4. How easy was it to control the video in Tomeo? (1=Very hard, 5=Very easy)

3 Responses 4.67 Average Number

5. How easy was it to swap between loaded videos in Tomeo? (1=Very hard, 5=Very easy)

3 Responses 5 Average Number

6. How easy was it to find the settings page in Tomeo? (1=Very hard, 5=Very easy)

3 Responses 5 Average Number

7. How easy was it to find the recent page in Tomeo? (1=Very hard, 5=Very easy)

3 Responses 5 Average Number

8. If you gave any question a 3 or below please explain why.

1 Responses Latest Responses  
"I do not often use video players so took a while to see how to loa..."

9. Any suggestions for new features you'd like to see?

3 Responses Latest Responses  
"Autoplay between videos"  
"Not that I can think of "  
"Been able to delete from play list "

Figure 4: User responses to our first iteration

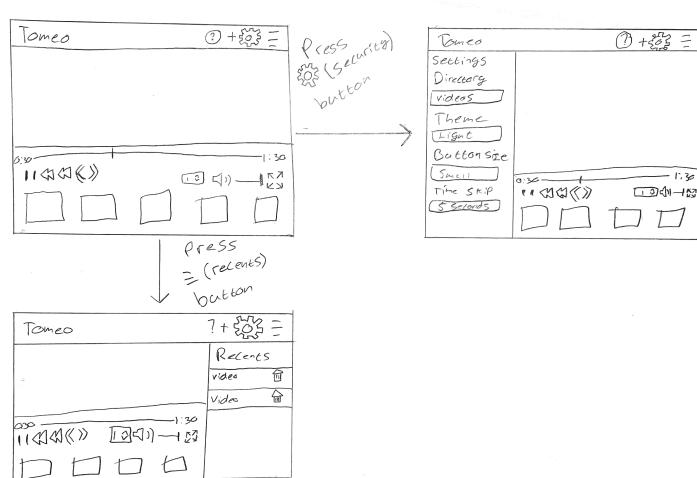


Figure 5: Paper prototypes for iteration two

actual user experience. The users were first asked to fill out a consent form to allow us to collect their data. The first version of the software was shown to various users and asked to fill out a questionnaire in which they rated their user journey for how easy it was to access and use certain functionalities, such as video control using buttons. The questions had a normalised scale so we could easily interpret the

results and compare each question, allowing us to pinpoint straight away what section needed work in the next iteration. Providing a comments section in the questionnaire, allowed the users to leave more detailed feedback such as functionalities they required to be added to HTVP. The questionnaire was implemented using Microsoft Forms which stores collected data securely, adhering to UK's GDPR laws.

The results of the questionnaire are shown in Figure

4. We concluded that many of our basic functionalities were easy for users to use and the layout helped with that. However, it was difficult and tedious for users to add new videos to the player and a new system is needed to ease this in future iterations. We accept the results of this iteration.

## Second Iteration

### Prototyping

The goal of this cycle is to add functionalities requested by the users through the questionnaire used. The second goal is to implement our functions further to meet the needs of Figure 2 and fill out some of the skeleton sections with actual working functions. The reasoning behind these is to make the software satisfy what a typical user would expect typically expect from a video player and UI improvement to make our existing functionalities clear and accessible to the majority of non-technical users.

To make quick prototypes, we used paper sketches to add and improve UI designs in Figure 1, as shown in Figure 5. The motivation behind those was the idea of making the software, even more, user-friendly and creating space for the functionalities the users requested.

For the second iteration we have chosen to work: add a recent tab, double click on the video in recent to open it, add drag and drop valid videos to the player, add guidance for new users as a base image, add a function to delete videos from recent, add a next and previous video buttons, file path select in the settings tab, change themes of the window, change default time skip amount, change the default button size.

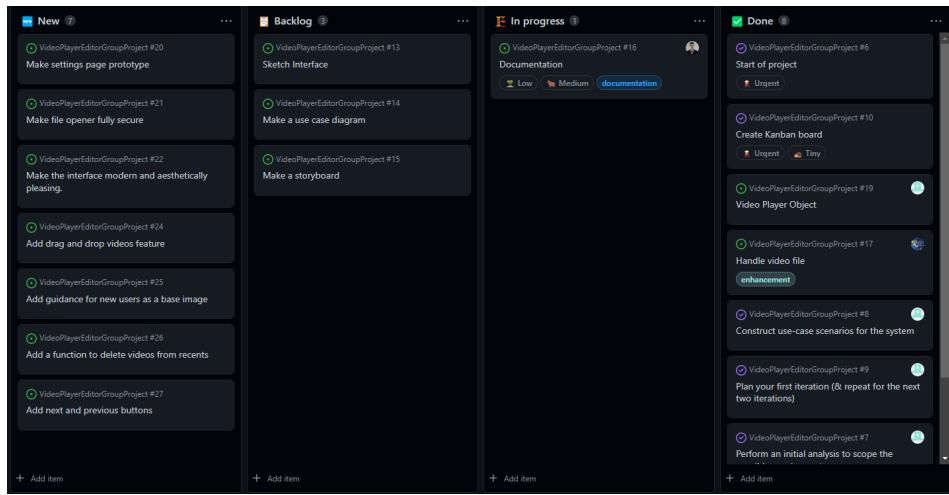


Figure 6: Kanban board after prototyping meeting during the second iteration

## Tomeo Feedback Iteration 2

3 Responses 05:22 Average time to complete Active Status

1. How would you rate your experience from 1-10? (1=Very bad, 10=Very good)	<b>9</b>
Responses	Average Number
2. How easy was it to open a video file in Tomeo? (1=Very hard, 5=Very easy)	<b>4.67</b>
Responses	Average Number
3. How easy was it to watch a video in Tomeo? (1=Very hard, 5=Very easy)	<b>4.67</b>
Responses	Average Number
4. How easy was it to control the video in Tomeo? (1=Very hard, 5=Very easy)	<b>4.67</b>
Responses	Average Number
5. How easy was it to swap between loaded videos in Tomeo? (1=Very hard, 5=Very easy)	<b>4</b>
Responses	Average Number
6. How easy was it to find the settings page in Tomeo? (1=Very hard, 5=Very easy)	<b>5</b>
Responses	Average Number
7. How intuitive did you find the settings page? (1=Not Very intuitive, 5=Very intuitive)	<b>4.67</b>
Responses	Average Number
8. How easy was it to find the recent page in Tomeo? (1=Very hard, 5=Very easy)	<b>4</b>
Responses	Average Number
9. How intuitive did you find the recent page? (1=Not Very intuitive, 5=Very intuitive)	<b>3</b>
Responses	Average Number
10. How useful did you find the help? (1=Not useful, 5=Very useful)	<b>4.33</b>
Responses	Average Number
11. If you gave any question a 3 or below please explain why.	Latest Responses "Recent page didn't show any videos" "Didn't realise the next video was at the bottom at first"
Responses	
12. Any suggestions for new features you'd like to see?	Latest Responses "-Maybe change the arrows for next video/skip because its ..."
Responses	

Figure 7: User responses to our second iteration

## Implementation

There were differences between the prototype and the implementation. A help button was added to review so the user could toggle whether they wanted to see the guidance on how to use the software, button size/theme button is now a dropdown, not an option select, as drop-down buttons are more intuitive to users, sidebars now push everything to the side instead of over the main screen, as it makes everything more viewable.

## Evaluation

We decided to reuse the questionnaire from the first iteration, with some changes to the questions to cover the added functionalities, as it allows us to compare our current results with past results and see if some of our improved designs are the user experience worse. The results are shown in Figure 7. We also decided to perform heuristics analysis, as shown in Figure 8, to see if our current UI designs have made it more user-

Heuristic Criteria based on Jaboc Nielsen	Does HTVP satisfy the criteria and why?
<b>Visibility of System Status</b>	The systems are all visible on the main screen and the recents/settings tabs are available upon simple/clear button click. All of the system work with no major delay.
<b>Match between system and real world</b>	The system uses standard icons and placement of systems seen in other common video players so users, who used other videos in the past, should straight away know how to use it.
<b>User control and freedom</b>	If the users press a video control button by mistake, they can click that button again to undo that action easily.
<b>Consistency and standards</b>	The system uses standard icons, language and placement of systems seen in other common video players, such as video control buttons below the video.
<b>Error Prevention</b>	When the users tries to play a file that isn't a valid video file accepted by the system, an error message will show up, explaining the error to the users
<b>Recognition rather than recall</b>	All of the button represent a singular change in the software so users doesn't have to remember a complex process to solve a problem.
<b>Flexibility and efficiency of use</b>	The most frequent action is selecting a new video to play and in the settings menu, there are shortcuts such as default file path when adding a new video so the users don't have to each their file directory every time to find a video in a often used folder. Also the user doesn't have to add every video they want to play from the same folder by finding it in a directory each time as the panel at bottom of the software does that for the users.
<b>Aesthetic and Minimalist Design</b>	The software has no long texts and uses icons/buttons for the most common actions. The basic systems are available in the main screen and the advanced functionalities can be found in panel which only opens if the user presses a specific buttons.
<b>Help users recognise, diagnose and recover from errors</b>	The only error message given to the user lets them know they tried to add a file that wasn't of a valid/accepted video file format. The message states this clearly and lets the users know what file extensions are accepted are accepted by the software.
<b>Help and documentation</b>	A help button is provided which guides the users to add a new video and a HOWTO text file explains how to use the software. However, there is nothing directly in the software that explains how to use all of the features of the software.

Figure 8: Heuristics analysis of second iteration

friendly and what aspects of heuristics analysis we haven't thoughts enough about and should improve on in the next iteration.

From this, we learned that our current program has bugs in the recent tab and video control in edge cases and the access for users to learn how to use the software is inadequate. However, the software has still improved compared to the first iteration.

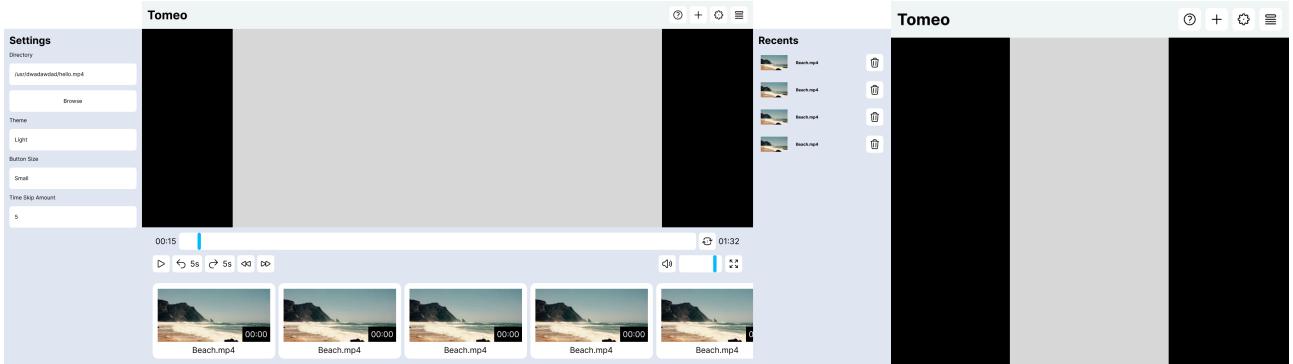


Figure 9: Desktop layout diagram for HTVP, done in Figma

### Third Iteration Prototyping

The goals for the last iteration were focusing on bugs and polishing out our current functionalities. The second goal is to create layouts, making the software easy to use on a variety of resolutions, focusing on adding standard desktop and mobile layouts. The reasoning behind it was that based on the user feedback, our new features from integration 2 contained edge case bugs, which heavily impacted the user experience.

Based on user feedback, some of our buttons were unclear about what they do and needed to change. Adding a variety of layouts will allow the software to be able to run on a variety of devices, making the software as accessible to as many users as possible. Based on the heuristics analysis in the evaluation section of the second iteration has let us know we need to add a way for users to learn how to use the software while using it. This can be done using a help button which will display a short text, explaining what each button does. To prototype our final iteration, we decided to do Hi-Fi Wireframe using Sigma, showing the two different layouts. This can be shown in Figure 9 for desktop and Figure 10 for mobile.

### Implementation

In this iteration we intend to implement: vertical and horizontal layouts, a help page with guidance, a saving directory of the last opened video, change the time skip buttons to be clearer, implement an icon for HTVP, playlists that show the video thumbnail, duration and name of the video, dark light contrast themes implemented, change default button

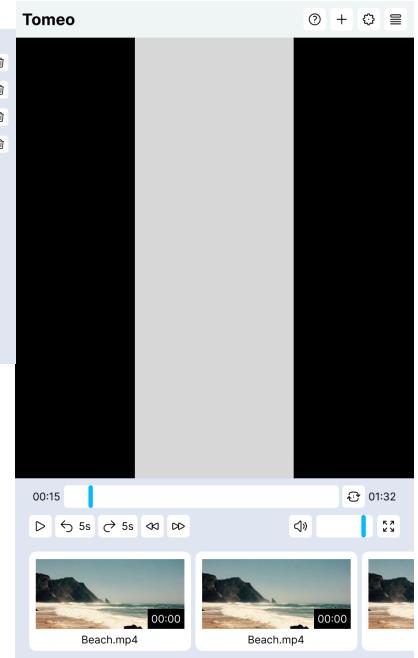


Figure 10: Mobile layout diagram for HTVP, done in Figma

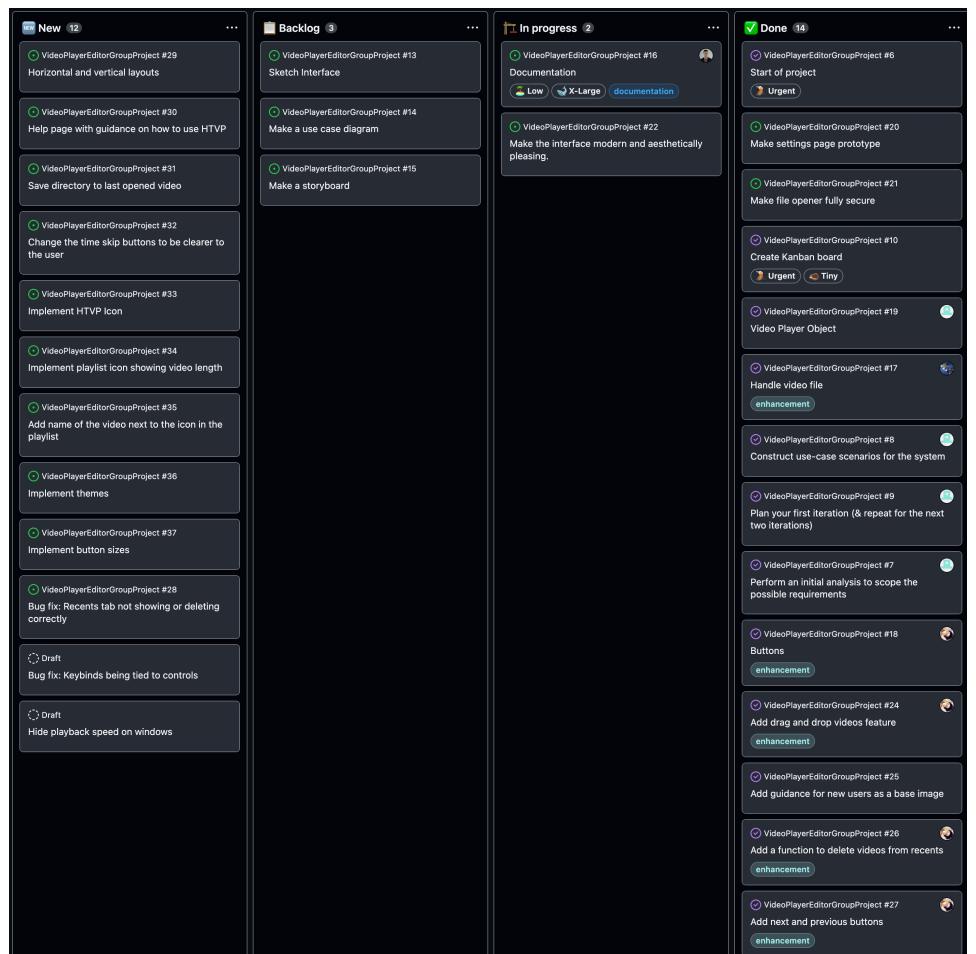


Figure 11: Kanban board after prototyping meeting during the third iteration

sizes. Bugs we need to fix: the recent page not showing any videos randomly, the pause button needs to work when the video is fullscreen by pressing the space bar, and hide playback speed on windows as the Qt version used doesn't support it.

## Evaluation

We decided to reuse the questionnaire from the second iteration so we can evaluate our software against every other iteration and can pick the iteration that the users liked the most as the final software, results are shown in Figure 12. We also performed automated tests to show our software in various resolutions. Since the only large change between iterations 2 and 3 was the help menu, the heuristics analysis is still valid and now fully satisfied. According to this, we decided to use third iteration software as it performed the best in the questionnaire.

## Conclusion and Reflection

In this report, we have looked at implementing an interactive video player that is responsive and works on a wide variety of devices. We added features that would be expected from the most popular video players already available in the market. This was achieved by creating overall three versions of the video player and each process iteration was directly influenced by the feedback given to us by users. The process is now complete and we are happy with our implementation. However, some sections could have been improved. We noticed that Qt creator and the Qt Framework work inconsistently for cross-platform / cross-OS applications. Throughout development, there were many bugs when we attempted to use the software on Linux, macOS and Windows 11. We attempted to compensate for these by creating specific guards and they should be minimised in our final iteration. However, we had to disable some functionalities on certain OS, such as the playback speed button on Windows and the Full-Screen button on Linux because of this. In the future, we would prefer to update the Qt to a more modern version where most of these bugs were fixed. That could have saved us a lot of time during implementation. We have implemented many features and they interact with each other, similar to industry standards. We were happy with this as we created software that we actually could actively use in our day-to-day lives. For the future we would also like to implement it over a longer period, allowing us to do more iterations and make it as accessible and useful to users as a video player could be. This would reduce the number of unintended interactions (bugs) currently present in our iteration 3 release.

## Further Material

The testing shown in the process video for iteration 3, shows the software working in resolutions for a typical tablet, smartphone and desktop. Figure 13 shows how the software looks on a standard mobile resolution.

## Appendices

Iteration 1 process video

## Tomeo Feedback Iteration 3

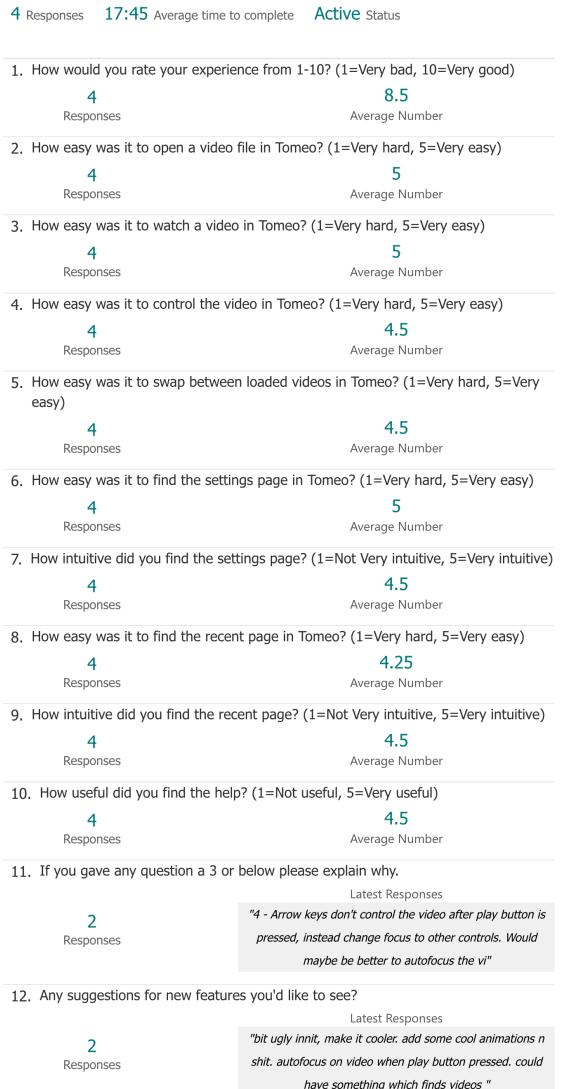


Figure 12: User responses to our third iteration

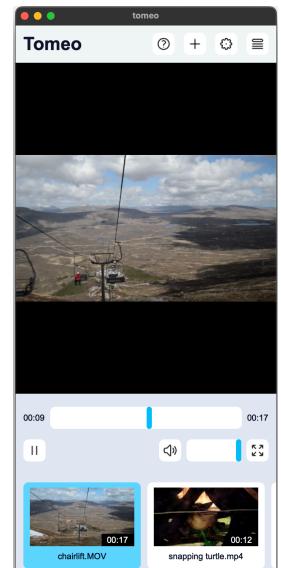


Figure 13: HTVP on mobile.

Iteration 2 process video

Iteration 3 process video  
Showcase video of the final product  
Filled questionnaires document

Signed consent forms  
Code of the final product

## References

Colours were chosen from Tailwind CSS (3), a CSS framework. Icons in the application are taken from Heroicons (4), which provide free SVG icons. Simple sketches were drawn using draw.io Desktop (5). UI design sketches were drawn using Figma (6). We used GitHub Projects (7) for our Kanban board.

---

<sup>1</sup> VideoLAN. *Official download of VLC media player, the best Open Source player*. [Online]. 2022. [Accessed 28 November 2022]. Available from <https://www.videolan.org/vlc/>

<sup>2</sup> YouTube. *YouTube*. [Online]. 2022. [Accessed 28 November 2022]. Available from <https://youtube.com/>

<sup>3</sup> Tailwind CSS. *Rapidly build modern websites without ever leaving your HTML*. [Online]. 2022. [Accessed 12 December 2022]. Available from <https://tailwindcss.com/>

<sup>4</sup> Heroicons. *Beautiful hand-crafted SVG icons, by the makers of Tailwind CSS*. [Online]. 2022. [Accessed 9 December 2022]. Available from <https://heroicons.com/>

<sup>5</sup> JGraph. *Release 20.6.2· jgraph/drawio-desktop*. [Online]. 2022. [Accessed 28 November 2022]. Available from <https://github.com/jgraph/drawio-desktop/releases/tag/v20.6.2>

<sup>6</sup> Figma. *Figma: the collaborative interface design tool*. [Online]. 2022. [Accessed 14 December 2022]. Available from <https://www.figma.com/>

<sup>7</sup> GitHub. *Planning and tracking with Projects*. [Online]. 2022. [Accessed 18 November 2022]. Available from <https://docs.github.com/en/issues/planning-and-tracking-with-projects>