Digital Technologies & Hangarau Matihiko 3.8A

Level 3, 6 Credits, Internal Assessment

Student Documentation

This document serves as evidence of your work for **AS 91901: Apply user experience methodologies to develop a design for a digital technologies outcome**

# Introduction/Kupu Arataki

This assessment activity requires you to plan, develop and create a complex computer program.

You will be assessed on

* How effectively you use project management tools and techniques to plan and manage the development of a digital outcome
* How effectively you decompose the problem into smaller components, and test and refine your media outcome so that it is a high-quality response to the task
* How well you have addressed relevant implications
* How well you synthesise information from the planning, testing and trialling of components to develop a high-quality response to the task (e.g. well-structured, logical, flexible, robust and comprehensively tested program)
* Discuss how this information assisted in the development of a high-quality outcome.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Colours used:** |  | #302c34 | #609cd4 | #282c34 |

# Problem Statement

Recommender systems are commonly recognised as playlist generators for video and music services like Netflix, YouTube and Spotify, product recommenders for services such as Amazon, or content recommenders for social media platforms such as Facebook and Twitter.

“In October 2006, Netflix released a dataset containing 100 million anonymous movie ratings and challenged the data mining, machine learning and computer science communities to develop systems that could beat the accuracy of its recommendation system, Cinematch” (Bennett & Lanning, 2007).

Given a dataset of movies or music albums, users and their ratings, you are to create a recommender system.

## You must:

* Be able to add a movie or musical album (name, director/artist, genre)
* Search for a movie or musical album
* Rate a movie or musical album
* Recommend a movie or a musical album specific to the user based on their rating
* Have a GUI

You may possibly want to use persistent storage (i.e. store the data in a file)

Table of Contents

[Introduction/Kupu Arataki 1](#_Toc80802525)

[Problem Statement 2](#_Toc80802526)

[S.W.O.T Analysis 3](#_Toc80802527)

[Decomposing the Outcome 4](#_Toc80802528)

[Explain relevant implications 5](#_Toc80802529)

[Sprint Tracking 1 7](#_Toc80802530)

[Sprint Tracking 2 9](#_Toc80802531)

[Sprint Tracking 3 11](#_Toc80802532)

[Project Summary 13](#_Toc80802533)

# S.W.O.T Analysis

Conduct a SWOT analysis for the project management tools you are considering using for your project.

## [Trello](https://trello.com/) for Project Management

#### Strengths

* Easy to visualise tasks that: need to be done, currently being worked on or are completed
* Mobile app means it is convenient to change and add tasks on the go
* Checklists within a card are very useful

#### Weaknesses

* Trello requires payment in order to assign parent/child cards. This doesn’t work for our project
* Doesn’t handle big projects very well
* No Gantt charts
* Only simple descriptions can be given for tasks

#### Opportunities

* To provide a fresh and clean alternative to current models available.

#### Threats (an aspect that prevents the tool from being useful)

* Being unable to create parent/child tasks is a pain in that you must create a full task for every decomposed task

## [ClickUp](https://clickup.com/) for Project Management

#### Strengths

* Allows nested subtasks and checklists
* Organise projects using folders for better management
* Calendar view shows you how long you have to finish tasks in order to stay finish on time
* Allows you to upload media
* Software organises projects in an intuitive manner

#### Weaknesses

* The amount of features can feel overwhelming
* Complicated interface that is not always easy to pick up and use

#### Threats (prevent the tool from being useful)

* Complicated interface could lead to wasting time figuring out how to use the software rather than managing the project

## [Monday](https://monday.com/lp/projectmanagement/bundle/?marketing_source=adwordssearch&marketing_campaign=au-s-project_management-b-desk-monday&aw_keyword=%2Bproject%20%2Bmanagement%20%2Bplatforms&aw_match_type=b&cluster=project_management&subcluster=&gclid=Cj0KCQjwsZKJBhC0ARIsAJ96n3VWqhG2mPzFpywfo87hxyw3RfYef8KTnuYyFcr5z08O4gp8nRTAgU8aAv53EALw_wcB) for Project Management

#### Strengths

* Works well as a visual management aid
* Shows the status of projects
* Customisable project management styles such as Kaban, Gantt chart

#### Weaknesses

* There is a steep learning curve when coming to using Monday to aid in managing your project
* Relies heavily on having good organisational skills
* Page response time is lacking when boards have many tasks

#### Threats (prevent the tool from being useful)

* Complicated interface could lead to wasting time figuring out how to use the software rather than managing the project

# Decomposing the Outcome

Decompose your digital technologies outcome into smaller components. User stories is one method that is commonly used in an A.G.I.L.E methodology

**Graphical user interface, text, application, website

Description automatically generated**

# Explain relevant implications

What relevant implications do you need to consider in the design of your digital technologies outcome? Explain at least three relevant implications and how they relate to your project.

## Intellectual property

One of the relevant implications I will need to take into consideration while developing my digital outcome is Intellectual Property. According to [Wikipedia](https://en.wikipedia.org/wiki/Intellectual_property), “Intellectual property is a category of property that includes intangible creations of the human intellect.” Every kind of content has some sort of intellectual property associated with it. Take a musical album for example, the album will have a cover image will most likely be owned by the label.

While developing this outcome I must make sure to not infringe on anyone’s intellectual property. This does not necessarily mean I am unable to use other people's creations however, if I am to do so I will make sure that I accredit them in an appropriate manner.  If I did not take this into consideration, I could be facing legal concerns later down the track to do with copyright and or other legal infringements.

## Functionality

Another thing I need to take into consideration while developing my program is the functionality of it. According to the Cambridge Dictionary functionality is the “quality of being useful, practical, and right for the purpose for which something was made”. The functionality of my outcome is especially important, if the outcome were no to function properly, nobody would be able to use it, thus making it useless.

As for the outcome I will be developing, I must make sure that the user is able to navigate through all the menus and different screens with ease to not ruin the experience. One of the ways I can make my outcome more functional is by drawing the outputs to the GUI for the user to visually see as opposed to printing them in the console.

## Sustainability and Future Proofing

Finally, I must also take sustainability and future proofing into consideration while developing my program. According to Cambridge Dictionary future-proofing means to “design software … so that it can still be used in the future, even when technology changes”. Future proofing a program is especially important now days with the speed at which technology is developing and changing, if you were to ignore this fact, the program you are developing may only be able to be used for a short period of time.

In context of the outcome I am producing, I must make sure that I am developing the program in one of the latest versions of java as to insure the longevity of the program. I will also me making sure I program using constants so that it make it easy to alter the outcome in the future if need be.

# Sprint Tracking 1

|  |  |  |
| --- | --- | --- |
| Sprint Number: | Start Date: | End Date: |
| Number 1 | 26th August | 7th September |

## Planning:

During the first sprint I am going to focus on developing the Content and Collection classes.

The Content class will be an object that contains 5 key fields: an ID, Name, Creator, Genre and Rating. The class will also include Getter and Setter methods as required. As for the Collection class, this will create an object that stores all the Content objects to that can be referenced and used later on in the program.

## 

## Development

The component I will be trailing in my first sprint will be the method of which the Content objs are stored:

* Either using a HashMap – (Requiring a key and value)
* Or using an ArrayList – (Requiring just a value)

Text

Description automatically generated

## Feedback

|  |  |  |  |
| --- | --- | --- | --- |
| **Component:** | Using a HashMap to store Content objs (& ArrayList) | | |
| **Name:** | Jarrod Roberts | **Date:** | 31st Aug |
| **Feedback:** | The HashMap is a good option as it uses the Integer as basically an index value similar to an ArrayList however it looks like accessing items with the HashMap is a longer process than that of using an ArrayList  The ArrayList on the other hand seems like it would be a better fit for the purpose and is a much straighter forward option to use | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Component:** | Using an ArrayList to store Content objs (& HashMap) | | |
| **Name:** | Gavin Moon | **Date:** | 31st Aug |
| **Feedback:** | The Array List seems like the obvious option to use. There is no need to use a dictionary (which is basically what a HashMap is) to store all the Content objects.  The HashMap is a more complex way of storing the objects an doesn’t fit the purpose. | | |

### Outcome of the feedback

The outcome of my feedback is that in order to make my code a little more readable I should use an ArrayList for the type used to store the collection of Content objects.

An ArrayList seems to be better as it takes up less lines of code in order to accomplish the same tasks as a HashMap. Reducing lines means that it the program should run better as well as making it more readable. Making an ArrayList the more suitable option for storing a collection of Content objects

## Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Testing:** | **Values Entered:** | **Expected Result:** | **Result:** | **Comment:** |
| **General:**  showCollection()  Iterating String[] | NULL | Both string arrays will get printed displaying the genres of the content objs in each collection |  | Works as expected |
| **General:**  getCollection()  ArrayList<> | NULL | Should return the collection as a ArrayList |  | Works as Expected |
| **General:**  getCollection()  HashMap<> | NULL | Should return the collection as a HashMap |  | Works as Expected |
| **General:**  addContent() | Genres = {"Space", "Drama"} | Should add the genres to the content obj |  | Did not work as expected  Changing String[] to ArrayList<String> |
| **General:**  addContent() | addContent("","", new ArrayList<String>(  Arrays.asList("Crime",  "Thriller")),5); | Should add the genres array list to the content obj |  | Works as expected |

## Evaluation

Sprint reflection and summary

Spring 1 has been conducted over 2 weeks, starting on the 26th of August and finishing on the 7th September.

Starting off the sprint consisted of developing the Content class which allows a Content object to be created. The Content object stores some necessary information such relating to the piece of content such as, the id number, name, creators name, genre and the users rating.

After that the Collection class was next. The collection class is quite simple. By creating a collection object, you will be able to use it to store a range of Content objects with in it.

What major changes and achievements did you complete in this sprint?

The major change and achievements completed during this initial spring method the way in which the Content objects were going to be stored using the Collection object. HashMaps and ArrayLists were the two options available. Thanks to testing and feedback given by classmates it was made clear that using an ArrayList was the most sensible option of the two.

Provide evidence of your project management tools being used

A picture containing graphical user interface

Description automatically generated

# Sprint Tracking 2

|  |  |  |
| --- | --- | --- |
| Sprint Number: | Start Date: | End Date: |
| Number 2 | 7th September | 4th October |

## Planning:

During the second sprint I am going to focus on developing the first half of the GUI class and full Card class.

I will primarily be focusing on making the GUI functional. This includes creating buttons and text fields, where necessary, for the user to interact with the program. Methods such as addContent(), viewAllContent(), findContent() as well as all the drawing content to the canvas methods to make the program usable.

I will also be developing a Card class that contains fields storing the positional data including width and height of a card and button. The object will also be able to link to a Content object in order to create content cards that will be viewable in the viewAllContent() method and wherever necessary.

Provide evidence of your project management tools being used to plan the development of your outcome at the beginning of your spring here

Graphical user interface, text, application, email

Description automatically generated

## Development

The component I will be trailing in my second sprint will be the location of the buttons

* Console Menu (text menu)
* Side Bar Buttons
* GUI Buttons (using the mouseListener)

Graphical user interface, text

Description automatically generated

## Feedback

|  |  |  |  |
| --- | --- | --- | --- |
| **Component:** | Visual On-screen GUI Buttons, (Sidebar GUI Buttons & Console Menu) | | |
| **Name:** | Jarrod Roberts | **Date:** | 24th Sep |
| **Feedback:** | The onscreen buttons are really nice visually, making for a more intuitive design beating the other two variations. Selecting using the text and side bar menus are tedious and requires two decisions having to be made in the brain (e.g. selecting the letter and choosing the outcome). | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Component:** | Sidebar GUI Buttons, (On-screen Buttons & Console Menu) | | |
| **Name:** | Gavin Moon | **Date:** | 24th Sep |
| **Feedback:** | The sidebar buttons are a good option as they are clear and functional for the user. They also minimise on code as ECS100 handles everything. Looking at the console (text-based) menu, the design seems bulky and un-intuitive when compared with the onscreen buttons. The console also means that the user must veer from the GUI to provide the input. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Component:** | Console Menu, (Sidebar GUI Buttons & On-screen Buttons) | | |
| **Name:** | Ryan Joe | **Date:** | 24th Sep |
| **Feedback:** | The console (text-based) menu is the worst option of them all. It is unappealing and a pain to have to click off the GUI to make to input your decision. Looking at the Sidebar buttons and the onscreen buttons, these options are both better and more intuitive than the text-based menu. | | |

### Outcome of the feedback

The outcome of my feedback seems to be that the preferred button style is that of using the onscreen style. This is due to the fact that selecting buttons onscreen is more intuitive rather than having to click off the screen to enter your option into a menu.

Although the onscreen buttons will inevitably be a lot more work, it will be beneficial in the long run as it will improve the user experience making my outcome more desirable to use.

## Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Testing:** | **Values Entered:** | **Expected Result:** | **Result:** | **Comment:** |
| **General:**  addContentGUI() | **Name:** YSIV  **Creator:** Logic  **Genres:** Rap, Hip-hop  **Rating:** 4.6 | The content object will be added to the collection  The genre text will be split via the “,” and then added to an ArrayList. |  | Works as expected |
| **General:** | **Choice:** A  **Choice:** V  **Choice:** F  **Choice:** R  **Choice:** Q | The correct function is run when the user provides an answer to the menu |  | Works as expected |
| **General:** | **Clicked:** Add Content  **Clicked:** View Content  **Clicked:** Find Content  **Clicked:** Change Rating | The correct function is run when the user provides an answer to the menu |  | Works as expected |
| **General:**  setupGUI() | NULL | Should print buttons in a grid format:  2 columns & x rows |  | Unintended output  Need to multiply the x value by x%2 |
| **General:**  setupGUI()  addButtonsToList() | **Buttons in HashMap:**  Add Content  View All Content  Find Content  Etc… | Should print a total of 5 cards |  | Unintended output  Not all cards display |
| **General:**  setupGUI()  addButtonsToList() | buttons.put(0, "Add Content");  buttons.put(1, "View All Content");  buttons.put(2, "Find Content"); | Should print out all the cards in a grid format |  | Works as expected  Moved row = 0 outside the for loop |
| **General:**  setupGUI() |  | All buttons added to the addButtons func should be printed to the GUI |  | Works as expected |
| **General:**  viewAllContent() | NULL | Currently just the content that comes standard is being shown |  | Works as expected |

## Evaluation

Sprint reflection and summary

Spring 2 has been conducted over a total of 3 weeks, starting on the 7th of September and finishing on the 4th of October.

This second spring consisted of beginning development on the GUI part of the outcome. This began with deciding on a component I was going to test. After some thought and planning I settled on developing what could be considered the user experience of the outcome. I trailed a range of different kinds of buttons including a text-based menu in the console, side bar buttons and an onscreen button drawn on the canvas of the GUI.

The trailing provided some feedback that helped me in making a refined decision as to which option would be the most appropriate. I ended up settling on the third option, the onscreen buttons.

I also

What major changes and achievements did you complete in this sprint?

Some of the major achievements I have accomplished during this sprint include:

* Developing the setupGUI() and doMouse() methods which sets up and handle the users interaction with the GUI’s menu
* Developing the Card class, allowing an object to be created that hold all the information regarding the individual buttons or content when you pass a Content object into it.

Without these I wouldn’t be able to create onscreen buttons and content cards as I did as I did.

**Graphical user interface, application

Description automatically generated**

# Sprint Tracking 3

|  |  |  |
| --- | --- | --- |
| Sprint Number: | Start Date: | End Date: |
| Number 3 | 5th October | 10th October |

## Planning:

During the third and final sprint I am going to focus on developing the last portion of the GUI class.

I will primarily be focusing on the user experience of inputting data, altering data and viewing data while using the program. In order to do this I will be trailing several methods of inputting the data such as a basic input through the console, text fields along the side of the GUI and a popup window using an external library called UiBooster.

I will also be running tests to improve and finetune the program

I will primarily be focusing on making the GUI functional. This includes creating buttons and text fields, where necessary, for the user to interact with the program. Methods such as addContent(), viewAllContent(), findContent() as well as all the drawing content to the canvas methods to make the program usable.

I will also be developing a Card class that contains fields storing the positional data including width and height of a card and button. The object will also be able to link to a Content object in order to create content cards that will be viewable in the viewAllContent() method and wherever necessary.

Graphical user interface, text

Description automatically generated

## Development

The component I will be trailing in my second sprint will be user experience of inputting information into the program

* Using the console as a text-based menu input
* Using the sidebar with text fields
* Using an external library to create a popup window

Graphical user interface, text

Description automatically generated

## Feedback

|  |  |  |  |
| --- | --- | --- | --- |
| **Component:** | Console, (Sidebar Text Fields & Popup Window) | | |
| **Name:** | Jarrod Roberts | **Date:** | 7th Oct |
| **Feedback:** | The console doesn’t seem to the most effective option of variation for this component. The size of the GUI is too small to present both the text input and buttons making the user must extend the window. For the other options, the sidebar text fiends are similar as they are a reduce the area of the GUI. The popup window seems the most effective variation. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Component:** | Sidebar Text Fields, (Console & Popup Window) | | |
| **Name:** | Daniel Stevens | **Date:** | 7th Oct |
| **Feedback:** | The sidebar text field is a clear way to receive user input, however, aesthetic & design of it doesn’t fit in with the design of the GUI. The usability is also very poor as the colour of the labels are light making it hard to read. In comparison with the other options, I think it’s more effective than the console however the popup window beats it in usability. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Component:** | Popup Window with UiBooster, (Console & Sidebar Text Fields) | | |
| **Name:** | Samuel Martin | **Date:** | 7th Oct |
| **Feedback:** | The popup window is the most effective of the three variations. The window is intuitive in that its simple, clear, and easy to understand what is being asked. The slider bar further fits this intuitive nature of this variation. The popup also fits the aesthetic of the GUI. Compared to the other options, the popup seems to tick all the boxes the others don’t, making it the most suitable variation of the component. | | |

### Outcome of the feedback

The outcome of my feedback seems to be that either the side bar and the UiBooster window are both good ways for inputting data into the program, however the UiBooster window is the better option as its more visually pleasing, intuitive, and functional.

## Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Testing:** | **Values Entered:** | **Expected Result:** | **Result:** | **Comment:** |
| **Expected:**  addContentGUI()  Console Input | **Name:** YSIV  **Creator:** Logic  **Genres:** Rap, Hip-hop  **Rating:** 4.8 | Program asks the user for the content’s information using the GUI’s console |  | Works as expected  Also adds content to the collection |
| **Unexpected:**  addContentGUI() | **Name:**  **Creator:**  **Genres:**  **Rating:** | If everything is blank the content will not be added to the collection |  | Works as expected |
| **Expected:**  addContentGUI()  Sidebar Input | **Name:** YSIV  **Creator:** Logic  **Genres:** Rap, Hip-hop  **Rating:** 4.8 | Program presents the sidebar text fields when the user clicks on the “Add Content” button |  | Works as expected |
| **Expected:**  addContentGUI()  UiBooster Window | **Name:** YSIV  **Artist:** Logic  **Genres:** Rap, Hip-hop  **Rating:** 4.8 | Using the UiBooster lib a popup window appears with a form that the user can use to enter the content data into |  | Works as expected |
| **Unexpected:**  addContentGUI() | **Name:**  **Creator:**  **Genres:**  **Rating:** 1 | Content won’t be added to the collection and a message will be presented describing system status | UI displays a message saying “Content Failed to Add” | Works as expected |
| **Expected:**  function() |  |  |  |  |

## Usability Testing

|  |  |  |  |
| --- | --- | --- | --- |
| **Name:** | Nerissa Linwood | **Date:** | 5th Oct |
| **Feedback:** | Testing the addContentGUI() method I found that if the user adds a space at the end of the content, they add they also need to add the space back to the input when searching for the content | | |
| **Result:** | - Must .strip() the input from the user before it is added to the collection  - When searching for content must see if input equals the content name ignoring case | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name:** | Mary Jane Robiony-Rogers | **Date:** | 6th Oct |
| **Feedback:** | Testing the rateContentGUI() method I suggested that you could use a dropdown box full of all the titles rather than having the user input the text | | |
| **Result:** | Will need to iterate through the collection creating an away with all the titles so that the UiBooster form can handle them. I could also do this with the find findContent() method | | |

## Evaluation

Sprint reflection and summary

Spring 3 has been conducted over a total of about a week, starting on the 5th of October, finishing on the 10th of October.

The third and final sprint consisted of finishing off the GUI class. This means I have had to do some usability testing to which I made improvements on my outcome based on the feedback given. I have also trailed different ways the user is able to input data into the program as to find the methods that best suits the outcome. As a result of the trials, I was able to make an informed decision as to which option would be the most appropriate. I ended up settling on the UiBooster window.

It also helps that using the UiBooster window is easier to store the values of the inputs as opposed to using the sidebar. The only drawback to using the UiBooster window and library is that I have needed to convert my project from a standard Java project to a Mavan Java Project. This meant that I’ve had to learn how to use and set up a Mavan project as well as the new UiBooster library provided which provided a large learning curve.

What major changes and achievements did you complete in this sprint?

As mentioned above of the major achievements I have accomplished during this sprint include:

* Converting & learning how to use a Maven Java Project as to manage using Java packages
* Learning how to use and incorporating the UiBooster library into my outcome in the form of popup window forms. These have been used in the methods: addContentGUI(), viewAllContentGUI(), findContentGUI() & changeRatingGUI()
* Drawing all the content cards to the screen in a grid format along with drawing the searched for card to the screen

Without these I wouldn’t be able to develop and create the user experience my program currently has.

Graphical user interface, text, application, email

Description automatically generated

# Project Summary

## Addressing relevant implications

Throughout the project I have needed to follow and comply with a set of relevant implications. Those have been Intellectual Property, Functionality as well as Sustainability & Future proofing of the outcome.

I have made sure to follow the **intellectual property** implication by accrediting all functions and code snippets that have been inspired by someone else’s code. An example of this is in the GUI class where I have used created the wait() method I used for debugging my code where I have credited the stack overflow post I received the help from. Another example of an instance I have needed to use someone else’s help was using the UiBooster library in order to create the popup forms used to receive user input. I have credited them under the credits button on the menu and at the bottom of the GUI.java file.

I have also followed the **functionality** implication, making the outcome easily navigable. I have done so by creating onscreen buttons, providing a more intuitive design and experience for the user. I have furthered this by also out putting all the results to the GUI in a thought-out design, as opposed to the small console in small text. I have also incorporated a third part library known as UiBooster that allows the user in input data into a popup window form using a range of drop down boxed, text fields and sliders.

In addition to the other implications, I have also developed the outcome with **sustainability and future proofing** in mind. This means I have used one of the latest versions of the Java Development Kit (16) as to ensure longevity of the program. With sustainability in mind, I have also developed the program following convention, most notability using constraints. This was to ensure that later development can be done with ease. In addition there is also scope for the program to use the Movie DB API in order to provide better recommendations to the user.

## Synthesising information gained from: Planning, Testing, Trailing of components & Usability testing

Throughout this project I have used techniques such as, planning, testing, and trailing of components in order to gain a better understanding of the how my program works in its intended environment.

The initial planning stage was incredibly helpful as it allowed me to visualise just how I was going to approach the tasks and in what order I was going to approach them in.

The planning stage, however, would not have been as useful without the testing and trailing of components I also carried out. Testing was an essential part of developing the program as I was able to find surface level bugs and fix them in order to make the program useable. The trailing of components and usability testing was also very useful as it allowed me a wider and deeper insight into how new users received and used my program. As a result of all of this, I was able to develop a more intuitive and user-friendly outcome.

As for the trailing of components throughout each sprint. During the initial first sprint I settled on trailing the functionality of an ArrayList vs HashMap. Due to the fact this component is a programming feature, I asked a couple of 13DTC students for their opinion. As a result, I ended up settling on using an ArrayList.

During the second sprint I wanted to start developing a better user experience thus choosing to trial the look and feel of the buttons used in the program. As this component is centred around the user experience, I was able to ask some potential stakeholders their opinions on the separate designs, thus resulting in settling on the onscreen button you see in the current design.

Finally, during the third sprint I settled on trailing separate ways of inputting data into the program. As this is also a user experience-based component, I collected a few potential stakeholders to give me feedback as to which I would make an informed decision to which option would be the best. This resulted in using the UiBooster window that you can see in the outcome.

## Discussing how this information led to the development of a high-quality digital outcome

Using this project managing process in order to guide the development of my outcome has been very helpful tool in providing me with means of iteratively improving it as to better the user experience.

During the beginning stage, where I planned out the development of the outcome via decomposing all of the tasks into my project management tool of choice, ClickUp, I had somewhat of an idea for how the outcome was going to look, feel and interact with the user.

Further down the line however, after I started consulting with potential stakeholders and end users and receiving their feedback, I found that my initial idea was not going to be the most desirable and/or functional component. Just like that, the project management process showed me how valuable it is to develop multiple options of a component, then go through testing to find which one best suit the intended social and physical environment.

For example, during the second sprint I had intended and planned to use sidebar buttons for the user to control the program. However, after trailing a range of options with individuals it turned out that creating onscreen buttons were the most desirable option as it felt more intuitive to use. This meant I had to change up my initial plan a wee bit but resulted in a higher quality outcome being produced. This happened during all sprints which you can see in the documentation recorded in the pages above.

## Stakeholders / End Users

* Teenagers
* Adults (all ages)
* Movie enthusiasts

## Who is doing trailing

* Classmates (covering the teenagers)
* Adults (my parents)