**Digital Technologies & Hangarau Matihiko 3.7A\_3.8A**

**Level Three, Credits 12, Assessment Internal**

# 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.

# 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:

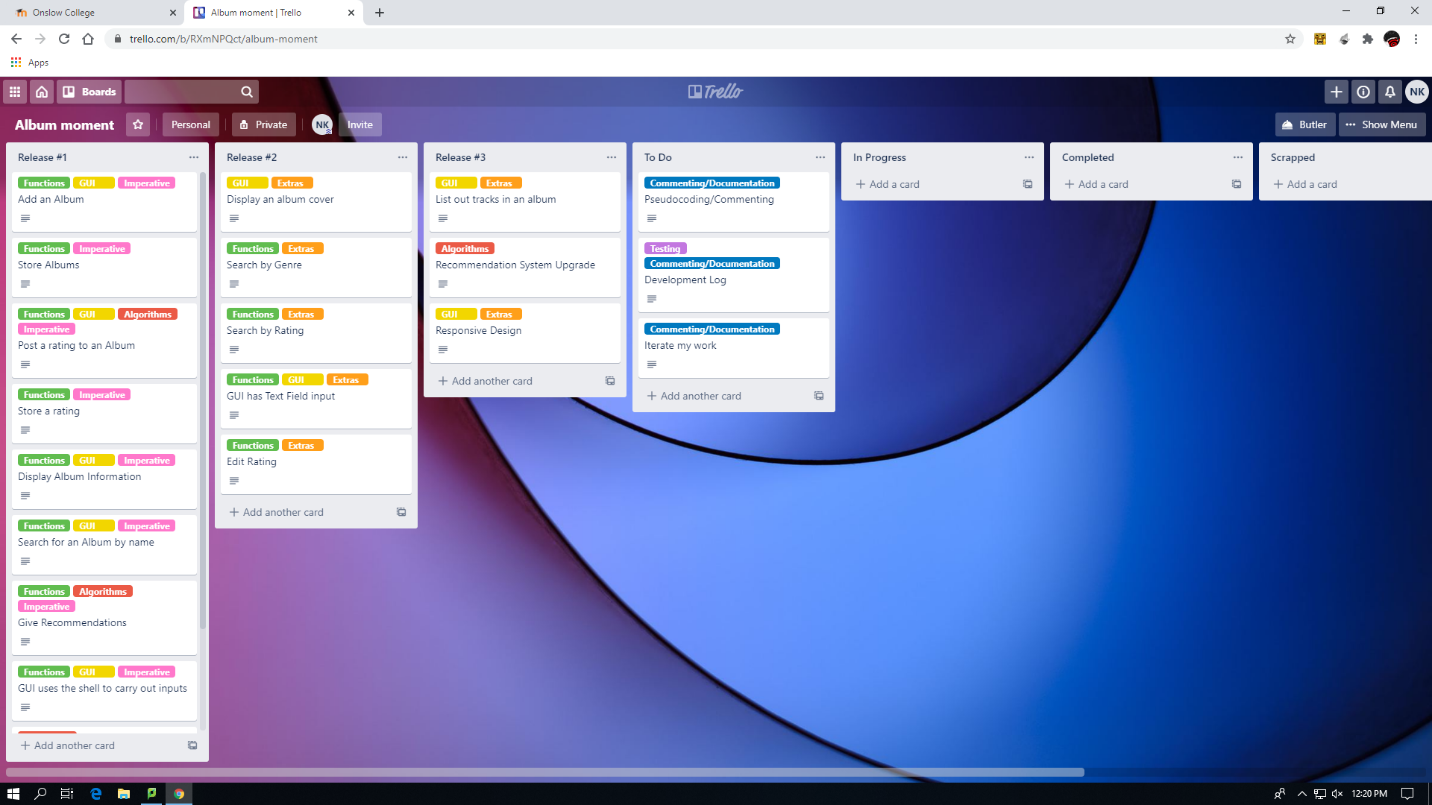
* Be able to add a movie or musical album (name, director/artist, genre)
* Search for a movie or musical album
* Rate a movie or musical album
* Recommend a movie or a musical album specific to the user based on their rating
* Have a GUI

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

# Decomposing the outcome

* ***decomposing the digital technologies outcome into smaller components***

*Decompose your digital technologies outcome into smaller components. User stories is one method that is commonly used in an AGILE methodology*



When decomposing my project, I thought about all the different things I should aim to accomplish and then all the things that aren’t as imperative but are important on an accessibility or aesthetics standpoint.

My project consists of parts that are involving:

* Functions (classes and different methods to do different things)
* GUI (parts involving the buttons, display and the link to the functions)
* Algorithms (processes that calculate what to do or when to do specific functions)
* Commenting/Documentation (Pseudocoding before beginning development and documenting my progress iteratively)
* Testing (Testing out processes and documenting my findings after every test)

Making sure that I stay on top of all of these different sections of my project, I have divvyed them up into their own respective releases. This can also be interpreted as my most imperative features versus my extras that I have come up with. Some are labelled as imperative as they are the ones slated to be a part of the 1st release whereas ones marked with Extras are parts that are intended to be released alongside fix-ups and re-releases.

# Considering Relevant Implications

* ***addressing relevant implications.***

*What relevant implications do you need to consider in the development of your outcome? Describe which you will address in its development.*

*Examples of relevant implications include:*

* *social*
* *cultural*
* *legal*
* *ethical*
* *intellectual property*
* *privacy*
* *accessibility*
* *usability*
* *functionality*
* *aesthetics*
* *sustainability and future proofing*
* *end-user requirements*
* *health and safety.*

**Intellectual Property:**

My end users are people that are looking for music in their line of genre. With that in mind, following copyright laws is a good way to avoid any problems with the project.  
  
Albums are created by people who own the property that I will be documenting, I will need to address this in my project by possibly adding disclaimers and telling the user that I don’t own the information and photos that I display.

This is important to my project because Intellectual Property is important when dealing with other peoples works. If I didn’t address this implication then this could come with different troubles that would mean that the services my project provides would be stopped and that would do harm to my code and the validity of my code.

**Privacy:**My end users are the users of my project looking for new music, possibly teenagers and other people of similar age. Privacy is very important and sharing other people’s information isn’t safe  
  
When people give me feedback, I need to be sure that their privacy is kept in mind when documenting. I also need to know which albums I should include as the ones that are already public are an obvious choice as they are part of media. Stuff involving creative commons or worldwide names is a better choice than smaller creators.

This is important to my project because people’s privacy is important and users will expect me to respect it with my code and my commenting, if I don’t take into account my users and my stakeholders’ privacy then users could not trust my code and therefore not want to use it. I can address this by not asking for any information from my users to do any processes in the system.

If I don’t ensure to keep my stakeholders’ privacy in mind, it could jeopardise the validity of my documentation and possibly create problems with my stakeholders, the people who mean a lot to my development.

**Accessibility:**

My end users are people looking for music in their line of genre. With that in mind, a good use of accessibility would do wonders to making my users time with this project

When developing my project, I need to think about how some people may not have an easy time using my project. To avoid this problem, I should use recognised ideas like arrows for changing albums, responsive design. Text fields in the shell that have a good layout and do not make the user look for the answer.

This is important to my project because being able to use the project successfully is important to making it easier to use and a choice above other applications, if my code wasn’t accessible then users wouldn’t want to use it.

If I don’t create processes that people can recognise and use effectively, my code could be too confusing to use, or could be not as intuitive as it can be. Making my project accessible makes it easy to use and more appealing to users as otherwise it will be too clunky and won’t be worth a user’s time.

**Aesthetics:**

My end users are people looking for music in their line of genre. With that in mind, a good aesthetic to my project would do well to make it appealing to users.

When creating my project. I need to make sure that my GUI has a clean layout, all of the information that the user needs is clear to them but isn’t cluttered. It would also do me well to use colours and photos to help the user to use my project more efficiently.

This is important to my project because a nice design is an incentive to a use to continue using the application. If the code didn’t look good or had a poor layout then they wouldn’t be inclined to use it again, one way I can address this is by using recognisable images to indicate different parts of the graphics pane or being clear and concise in the shell

If I didn’t make my project aesthetically pleasing, it could be harder to use and thus not be a success. Making my project nice to look at and well set out makes using it easier and development can progress consistently. Messy code = messy development.

# Sprint Tracking | Sprint 1

|  |  |  |
| --- | --- | --- |
| **Sprint Number** | **Start Date** | **End Date** |
| 1 | 29/06/20 | 10/07/20 |

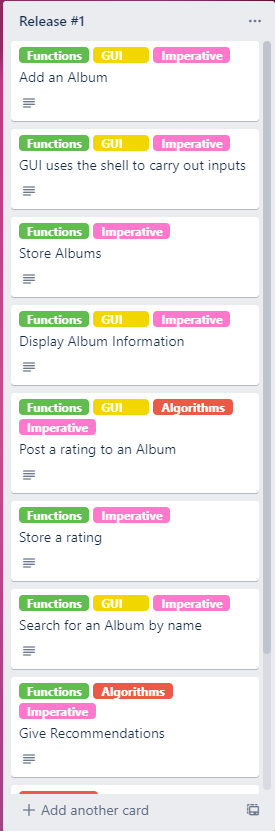
* ***using recognised and appropriate project management tools and techniques to plan the development of a digital technologies outcome***

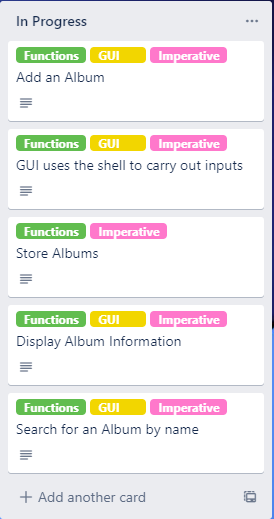
## Planning

*What are you going to work on in this sprint?*

29/06/20 – **1st Sprint:** I want to try to start my program by printing out a collection of information about an album in the shell that was manually entered into the hashmap. This is my first trial. I will specifically print out the artist for an album that is input into the text field for searching. However this is after I implement an adding album feature.

*Provide evidence (screenshot / photo) of your project management tool(s) being used to plan the development of your outcome at the beginning of your sprint here*

For the first Sprint, I want to focus on getting my first release started, this means that I should prioritise the major parts of the code. I will start with making the ability to add an album a reality, this process includes the ability to use the shell to carry out inputs. Since you are inputting an album into the hashmap, I need to store them. This knocks out 3 of the starting parts of my project in a relatively short amount of time. Making sure that the hashmap works is important so I will manually implement an album and print out information via a different class to prove the hashmaps ability to store information.

I decided what my course of action is for this sprint by taking my first release pieces and putting them in my “In Progress” list. Here is what that looks like as of now:

I want to at least be able to make it so I have a button ready on the GUI and I am able to add an album to the hashmap via said button.

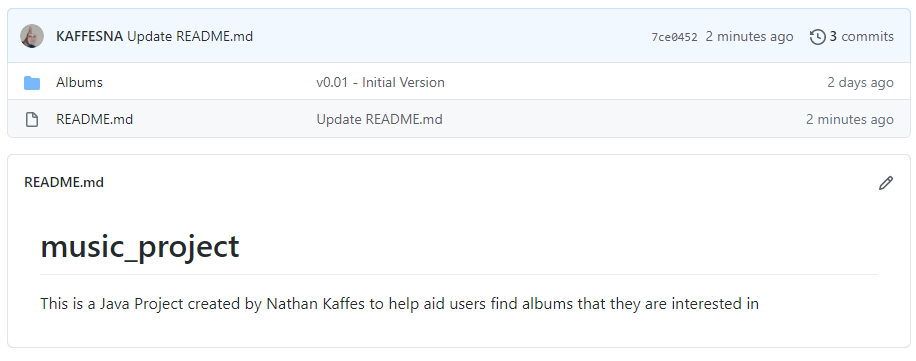
## Development

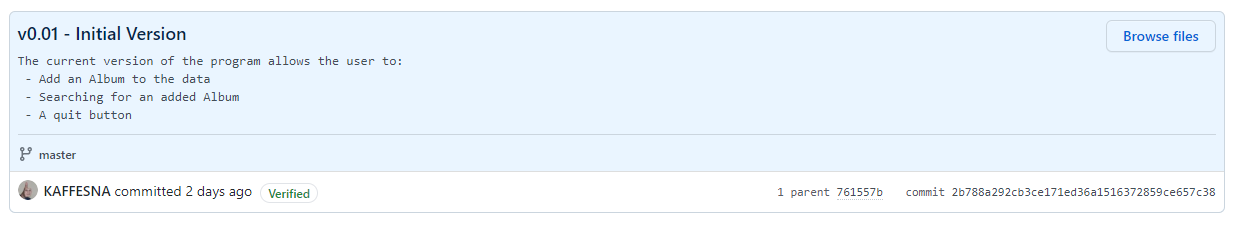
*What components are you going to trial?*

* Implementing the Hashmap
* Calling data from the hashmap via a method.
* Inputting an album into the hashmap

*Provide evidence (screenshot) of your version control*

I have decided to use GitHub for my version control, it’s easy to use and lets me look back at previous versions while recording what features each version has. At this point in time I have only committed one version to the repository, here is what that looks like:





## Feedback

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | Adding an Album | | |
| **Name** | Matthew Holdaway | **Date** | 24/07/20 |
| **Feedback** | The outcome is functional and accurate. You need to add some error checking for the search function and inputs in the UI. Also have some try and catch statements so that users only put in the right type of data without being given an error | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | Searching for an album | | |
| **Name** | Matthew Holdaway | **Date** | 24/07/20 |
| **Feedback** | It works correctly however it’s a bit weird to use enter to run the method rather than a separate button connected to the text field. You could possibly have the user click a button to continue rather than just pressing enter in the text field | | |

*What is the outcome of this feedback?*

It has given me some ideas on how to change the usability of my project. Having a button prepared for the text field would prove more useful and obvious as a method of input rather than just assuming a user would press enter in the text field.

Also Matt’s feedback has reminded me to work on making sure that boundary and invalid inputs are accounted for, I will need to work on this in the coming sprints.

## Testing

***Brief Description of what you are testing***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Type of test**  **(E, B, I)** | **Method to Test** | **Value(s) to enter** | **Expected result** | **Actual result (screen snip / time stamp)** | **Comments/ changes needed** |
| Expected | addAlbum() | Bangarang  Skrillex  2004  Dubstep | It should return a statement that repeats all the information you just input in order |  | When I am doing the finishing touches to my code I will make it look cleaner, but at the moment the code works and displays the correct information |
| Expected | printChosen() | Thriller | It should return a statement that puts all of the information involving Thriller in context |  | I will also touch up this part of the code as the statement being in context doesn’t make much sense, however it’s clear that the code works as I input a key into the text field and it returned the corresponding values |

## Evaluation

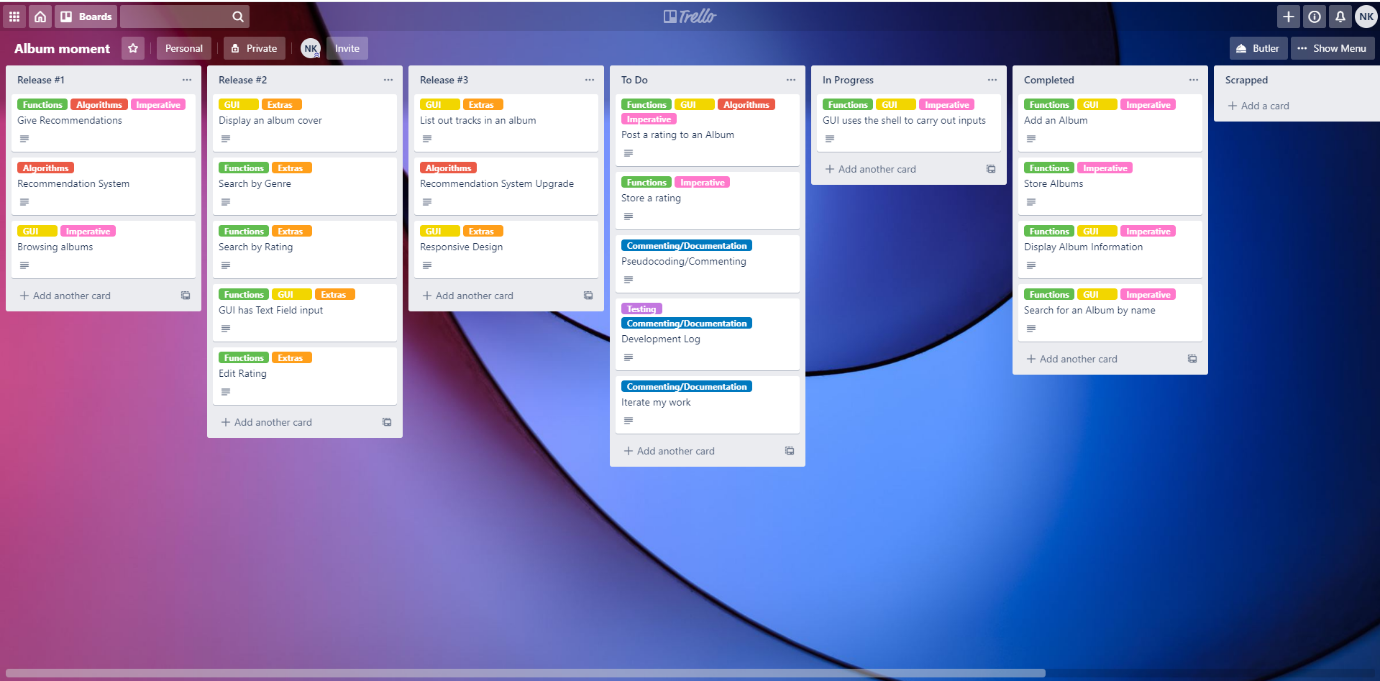
*Sprint reflection and summary*

I believe that I have successfully completed my first sprint, I set my expectations and bounds low for the first sprint so that I didn’t instantly get lost behind schedule. However I believe that I need to increase my workload for the next sprint if I want a shot at being able to finish this in time

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

I have managed to implement a GUI where you can add an album to a hashmap and then search for that album via the key using a text field.

*Provide evidence (screenshot / photo) of your project management tool(s) being used to manage the development of your outcome at the end of your sprint here*



At the end of the first sprint I have completed a majority of the parts that I set out to make, I am still working on creating the GUI as there are more things to implement therefore the part that requires me to use the shell for inputs isn’t fully completed. If I want to get my first release done by the end of the second sprint I need to get to work on the recommendation system and the rating system. I am planning to work on the rating system in the next sprint but for now I believe I am on track.

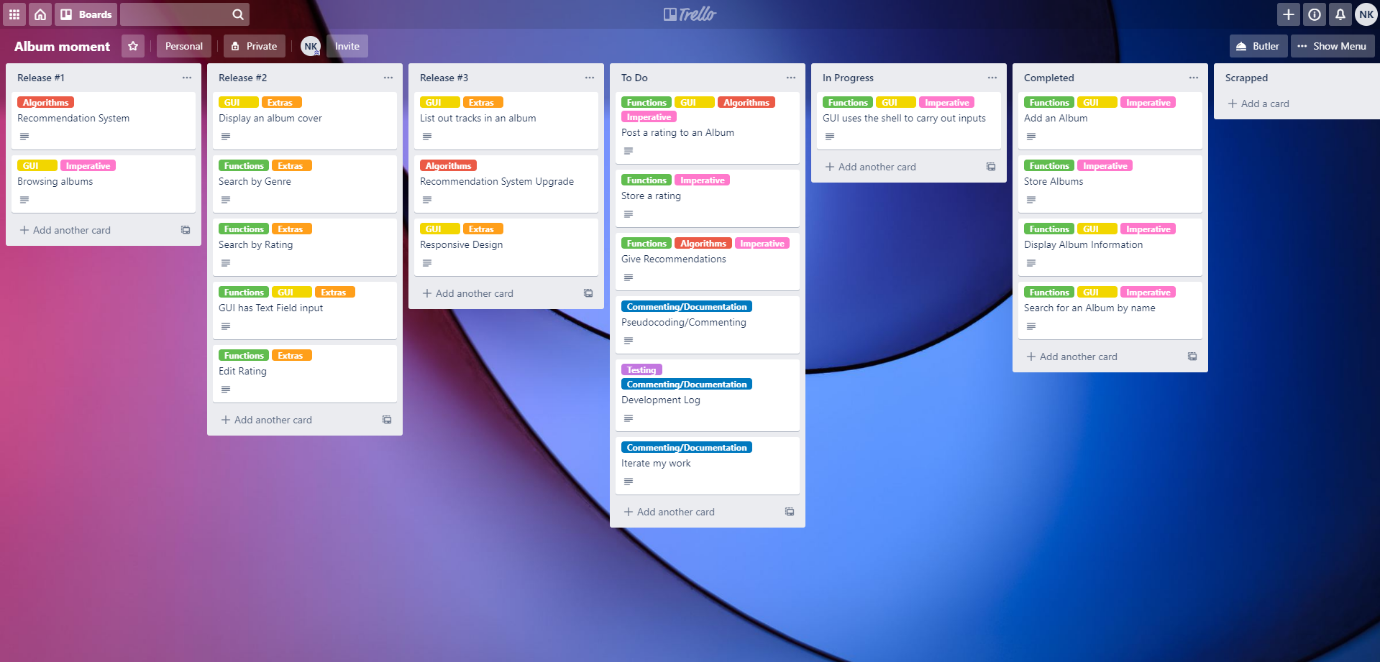
# Sprint Tracking | Sprint 2

|  |  |  |
| --- | --- | --- |
| **Sprint Number** | **Start Date** | **End Date** |
| 2 | 27/07/20 | 10/08/20 |

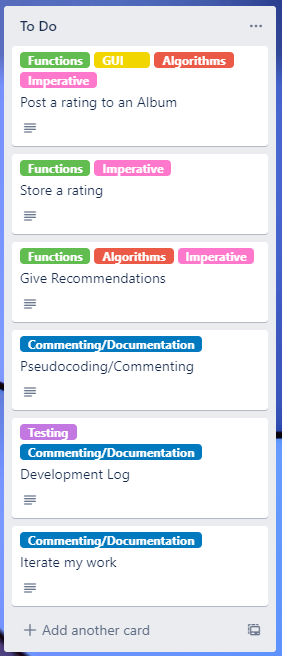
* ***using recognised and appropriate project management tools and techniques to plan the development of a digital technologies outcome***

## Planning

*What are you going to work on in this sprint?*

I need to keep working on getting my project functional, therefore I want to work on other processes like printing all albums in the database out, printing all in one genre. The main thing I want to work on in this sprint is the rating and recommendation system. Here is what my trello board looks like at the moment.

*Provide evidence (screenshot / photo) of your project management tool(s) being used to plan the development of your outcome at the beginning of your sprint here*

The recommendation system is still in an idea stage, the extra card in release #1 is for an update to the recommendation system. I want it to be more complex than show all in one genre if liked or don’t show if disliked. But the things I need to ensure I work on is posting a rating to an album and then storing it. I also will need to work on my recommendation system.

Starting off, I will adapt my hashmap to allow for the rating system to be changed. I will add a rating Boolean to the system. It will be set to 0 stars when you first add the album to the hashmap. I will then have a button which will ask the user to type the name of the album they wish to rate, then it will ask for the star rating out of 3 that they want to rate the album. The recommendation system will use this information to find which albums have been rated 3 first, if none then 2 and so forth. This 3 star rating system is simpler than a 5 star rating system but can still work with a recommendation system.

## Development

*What components are you going to trial?*

* Posting a rating to an album
* Storing the rating
* Recommending different albums based on rating

I found that the searching by genre and searching by name had some bugs due to them being text fields. Clicking off of the text field or pressing enter would run the method linked to the field but sometimes that could be premature. I got an idea from Matthew Holdaway for the text field to store it in a variable and then click a button to grab that variable and use it in a method. This is smarter as there isn’t any unwanted input.

I am not sure if I

*Provide evidence (screenshot) of your version control*

I have 3 options for my rating system.

* Boolean with true or false for “liked” or “disliked”
* Integers from 1 - 3 for a star value
* Integers from 1 – 5 for a star value

I believe that the second option is the better choice. The 1st option is too black and white to allow for a complex recommendation system to revolve around them, having a Boolean also comes with its own problems as numbers are easier to change and work with. The 3rd option is also a good idea, it’s recognisable and universal with many rating systems across the internet. However I believe it would become too complex to work with at this current stage in development. I don’t want to bite off too much and then be stuck with a system that is half unused or redundant compared to a more simple and concentrated scale.

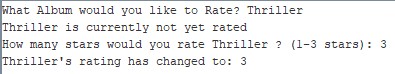
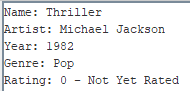
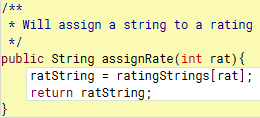
Therefore, I’m going to take the 2nd option and use a range of 1 – 3 stars in my rating system

Changing the rating in the hashmap is interesting as the way to do it is by adding something in with the same name. It seems to overwrite the hashmap and just change what was different.However it could be done differently. I’ll need to explore other ways to change a rating.

I wanted to add a classification to every integer for the rating system. Where:

* 0 = “Not Yet Rated”
* 1 = “Bad”
* 2 = “Average”
* 3 = “Good”

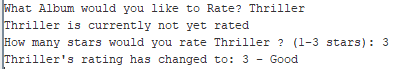
I needed to make a separate method to assign that every time it grabbed a rating for printing.

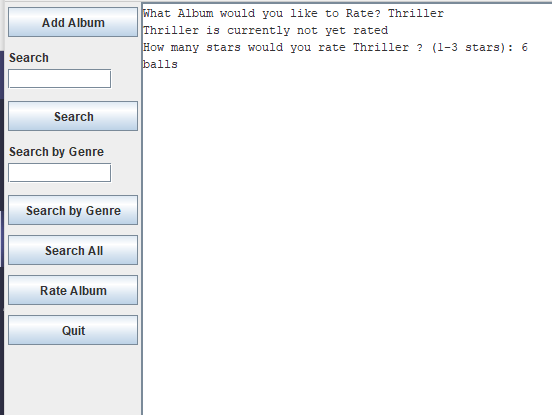


PS: I should actually assign the string here too, it would be good for users to confirm what the rating system is based on when they are rating something.

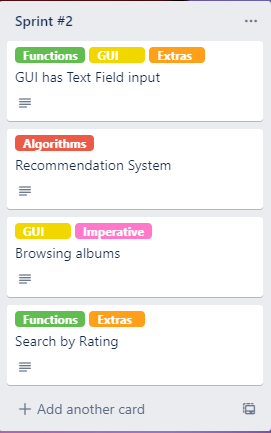


I ended up adding that feature, here is what it looks like in action:





I have also been able to implement something that I didn’t plan for, but was in my trello board



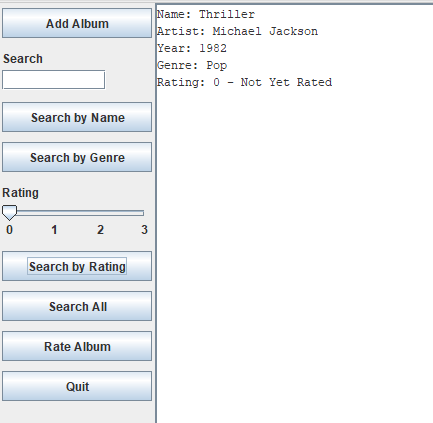
When working with my trello board, I noticed that I pushed some things back to Sprint #2. I ended up pushing back a lot of the rating system and recommendation system things, along with UI things that I haven’t gotten around to. One thing that was pretty easy to implement was the “Search by Rating” feature. I ended up using a similar method to my searching features but instead of a text field, it’s a slider. I felt like a slider would be useful as it sets specific numbers as bounds and is easier to check for as you can’t enter an invalid answer into a slider that has certain bounds

When adding it to my GUI, it looked like this;



Using the button means that any unwanted premature input isn’t done and it makes it so that the GUI is more intuitive and communicates it to the user clearly.

When testing it out, I tried 0 as I hadn’t rated the album present in my hashmap, this is the result;



## Feedback

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | Rating System and Adding Album | | |
| **Name** | Jamie Gordon | **Date** | 10/08/20 |
| **Feedback** | I think it should tell the user they need to click ‘Rate Album’ button again if they put in a wrong number. Also you should work on adding different parameters for adding albums as putting one with the same name breaks things | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | Rating System and text entry | | |
| **Name** | Matthew Holdaway | **Date** | 10/08/20 |
| **Feedback** | It work but when the user inputs an invalid value it should make them input the value again instead of making them click again. And it should account for title, should not be case sensitive | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | Rating System and UI stuff | | |
| **Name** | Leon Dudley | **Date** | 10/08/20 |
| **Feedback** | Two search bars kinda clunky, clear text cool. | | |

*What is the outcome of this feedback?*

I was able to change the way that my searching system works thanks to Leon’s advice, it’s much easier to use and far less clunky. Also, I am thinking of adding a system the works with case sensitive inputs, however I don’t plan to add it at this point in time. I would prefer it to be something added during my final sprint as it is purely a Quality of Life addition rather than a necessity. Also, I do need to add a while loop for the rating system as it is annoying to go back through the inputs to get back to the input you did incorrectly.

## Testing

***Brief Description of what you are testing***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Type of test**  **(E, B, I)** | **Method to Test** | **Value(s) to enter** | **Expected result** | **Actual result (screen snip / time stamp)** | **Comments/ changes needed** |
| Expected | searchAlbum | random | It should return “That album is not in our database” |  | This is done so that any users can get a response if they input an incorrect string. |
| Boundary | rateAlbum | 4 | “Please enter a number between 1 - 3” |  | This ensures that there isn’t any out of bounds answers for the rating. |
| Expected | searchRating | 0 | “There is no album with that rating in our database” |  | The slider can’t be tested for boundary inputs as the boundaries are already set and cannot be passed. |
| Expected | rateAlbum | “Not Thriller” | “That album is not in our database” |  | Works as expected |
| Invalid | rateAlbum | “Three” | “Please enter a number” |  | I believe that the UI has a built in feature that accounts for invalid inputs, the same outcome comes from it but isn’t what I expected |
| Expected | rateAlbum | 3 | “\_\_\_\_\_’s rating has changed to 3 - Good” |  | Stores the album’s rating when updated. Also has a classifier that is applied to the number that helps with the accessibility of the code by using a string to explain the way the rating system works. |
| Expected | searchGenre | “Pop” | “Thriller”  “Pet Sounds" |  | It returns all albums with the desired genre, separate from one another and with the rating |

## Evaluation

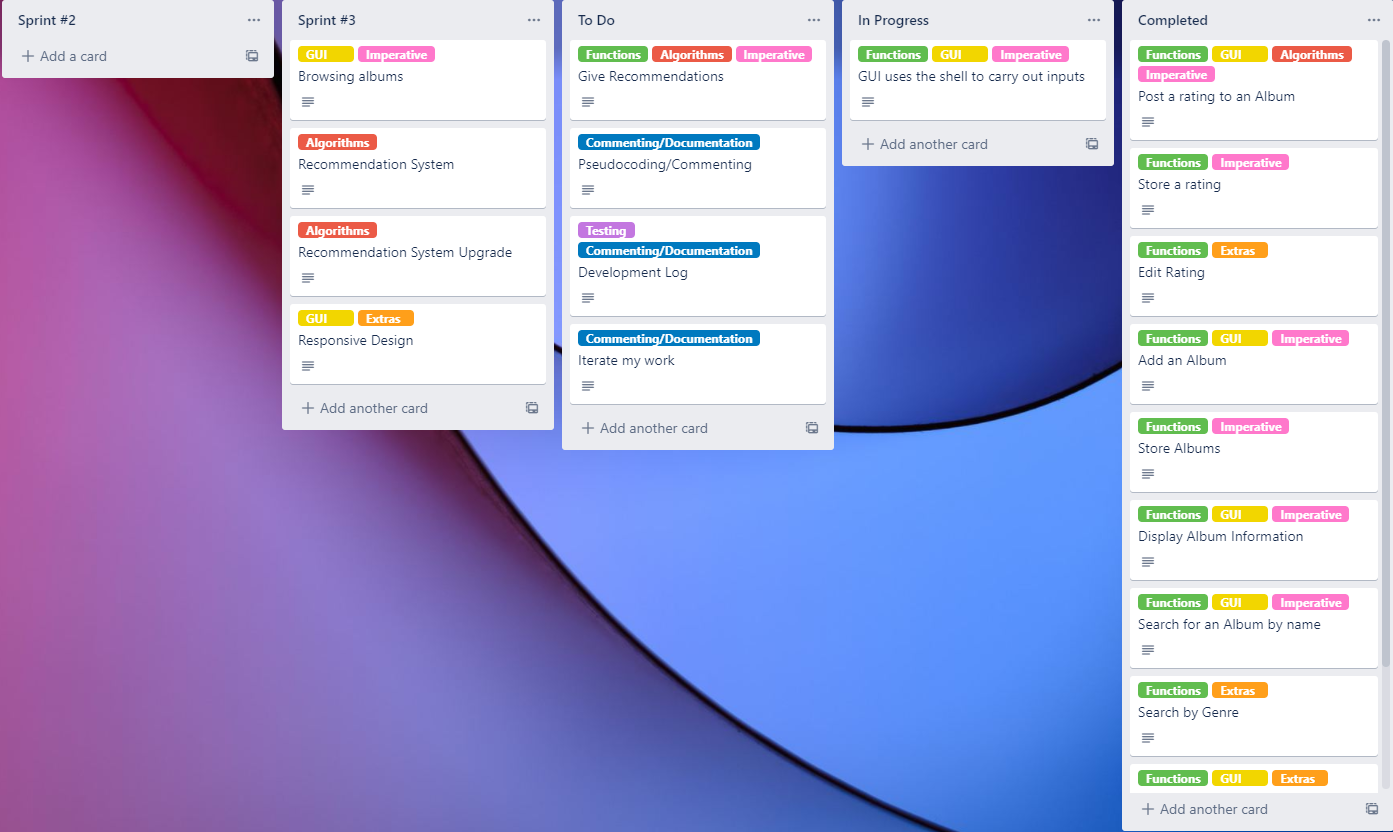
*Sprint reflection and summary*

I believe I have made some good progress during this sprint. I wasn’t able to start my recommendation system however and I really need to pick up the slack and get to work on probably the most important part of this project. But completing the rating system has been really helpful and added some more features to my code. Also, due to some of the feedback I have gotten during this sprint, I was able to make my UI more intuitive and thus make my code work better overall.

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

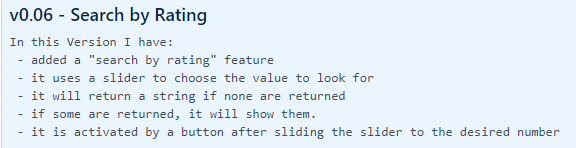
I managed to complete my rating system and my searching features. I might add to them at a later date to make it work with the recommendation system or to transfer outputs into the GUI, but for now I have made my code more intuitive and also completed an integral part of my project.

*Provide evidence (screenshot / photo) of your project management tool(s) being used to manage the development of your outcome at the end of your sprint here*



I have pushed back my recommendation system to Sprint #3 as a result of not being able to finish it in Sprint #2, however everything else in Sprint #2 has been completed and put into Completed. I have set “Give Recommendations” as my most important piece to construct next. This will be done in the GUI and should be a big part of my code that I will be able to make during this sprint, getting this done will get me back on track to finalising my Minimum Viable Product and then moving on to touching up the code and making it more efficient or more aesthetically pleasing.

At this current point in time, my latest commit on Github is v0.06, this is the commit notes on Github:



This version includes a search by rating feature that uses a slider to choose what rating to search by. This was a small feature but acted as the last main feature that I added before this sprint finished.

# Sprint Tracking | Sprint 3

|  |  |  |
| --- | --- | --- |
| **Sprint Number** | **Start Date** | **End Date** |
| 3 | 10/08/20 | 21/08/20 |

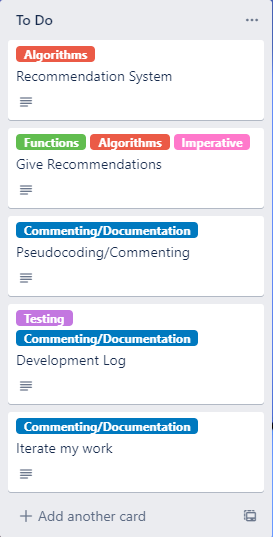
* ***using recognised and appropriate project management tools and techniques to plan the development of a digital technologies outcome***

## Planning

*What are you going to work on in this sprint?*

In this sprint I plan to work on:

* The Recommendation System
* A possible revamp of said system
* GUI printing with information
* GUI input with buttons (eg. Stars for Rating System)

If I manage to complete this in the time I have, this should set me up well for an on time and high-quality completion. The big problem from this sprint is the time constraint, I have had to start my sprint late due to being slow. I was supposed to start this sprint on the 10th of August but I have only begun working on it a week later on the 17th. This is a problem but with some hard work and time at home I should be able to get it done

*Provide evidence (screenshot / photo) of your project management tool(s) being used to plan the development of your outcome at the beginning of your sprint here*

I am prioritising the Recommendation system for this sprint as it is part of my Minimum Viable Product. If I were to focus on implementing GUI input it would be wasted as I already have the same process set up in the Shell. Also, I can adapt my Recommendation system into the GUI input, so working off of my recommendation system will be beneficial for me. My idea is to have it where if a user rates an album, an albums information will show up in the GUI depending on whether it meets the criteria for being recommended to the user. The GUI print will also include 3 stars, which the user can click to store a rating for the recommendation.

For the recommendation system, I have two ideas:

1. Recommend based on rating and Genre
2. Recommend based on rating and Artist

Both of these are valid. I could do something where if a user rates something 3 stars, it will recommend something that is the exact same genre, or the same artist if applicable. If the user rates something 2 stars, it should only recommend it if both the genre and artist match. But, if the user rates something 1 star, it should avoid recommending something that has either the same genre or artist. However, I don’t know if it is easier to do a recommendation system with both or just one of the two options. Both could result in a more accurate recommendation, but could be more clunky and confusing to understand. One could result in more vague recommendations but be easier to produce.

I can either:

1. Create the Recommendation System only factoring in the Rating and Genre
2. Create the Recommendation System only factoring in the Rating and Artist
3. Create the Recommendation System factoring in both
4. Do either 1 or 2 and then update the recommendation system later to be like 3

I believe 4 is the best option here. Doing that will save me time for this sprint and leave it so that my Minimum Viable Product is completed first, then if I have time I can adapt my recommendation system to make it more in-depth and complex. This saves me time where I need it and also allows for me to work more efficiently on my project over this sprint alone. I will only factor in the rating and the genre for the time being. If I have time, I will update it to bring it up to factoring in both.

Looking at the way I want to print out my information on the graphics pane, I have a few ideas, all ranging in time to complete and complexity. I think I need to go for a lower time choice as the last sprint should be used touching up my code, my minimum viable product needs to be done first and that means I should focus on getting that done as soon as possible.

My 2 ideas are:

1. Make a system where it stores all recommendations in an ArrayList and lets you scroll through them using arrows on the side. This would take a while for me to implement and would be quite annoying, you would also be able to rate it via stars on the graphics pane, I’m worried that this would come with a lot of different bugs due to different things in the MouseListener.
2. Make a system where it stores all recommendations in an ArrayList and it will display each one after you rate the last by clicking stars in the graphics pane. This would take far less for me to implement and would be still effective at showing all the recommendations and letting you rate each one. It would also come with less bugs as it would be purely a for loop looping through the ArrayList until there are no more left in the ArrayList to loop through.

I think I should do 2. Considering the time constraint surrounding the project I should go for an easier but still effective method of going through recommendations. This shouldn’t take as long as the other one, If I had more time then I would’ve been able to make a more complex system and put it in place. But I’m not about to bite off more than I can chew and not be able to produce my Minimum Viable Product.

## Development

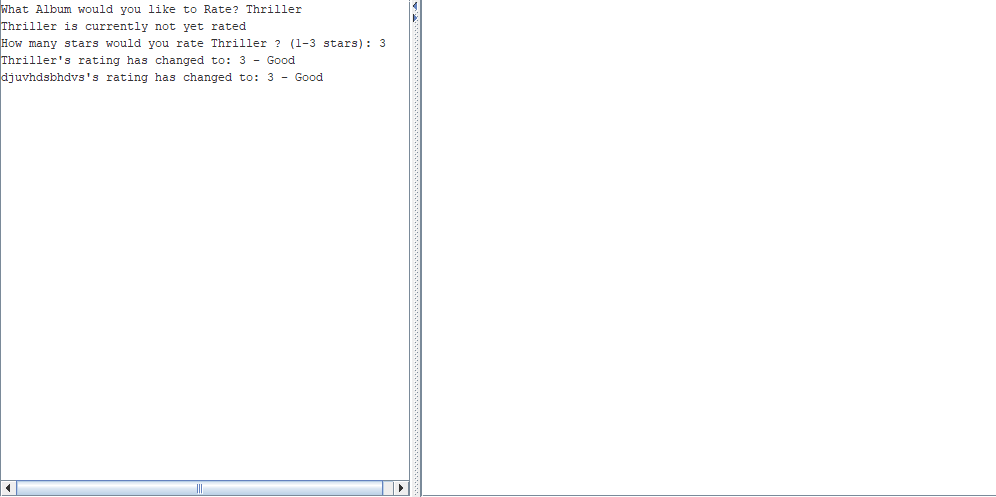
*What components are you going to trial?*

In this sprint I will trial:

* Recommending an album to a user
* Using the GUI to print information
* Rating an album via the GUI
* Rating an album via a Recommendation

*Provide evidence (screenshot) of your version control*

I have a problem with the MouseListener, when setting this function, there doesn’t seem to be a way to remove it, when clicking on the stars, the code will clear the graphics pane, but the MouseListener is still active. I don’t know how to unset the MouseListener



The stars and the text disappears from the graphics pane but the MouseListener is still active, it also still works, clicking on certain parts of the graphics pane after clicking the stars will cause the album in the recommendation to be rated once again.

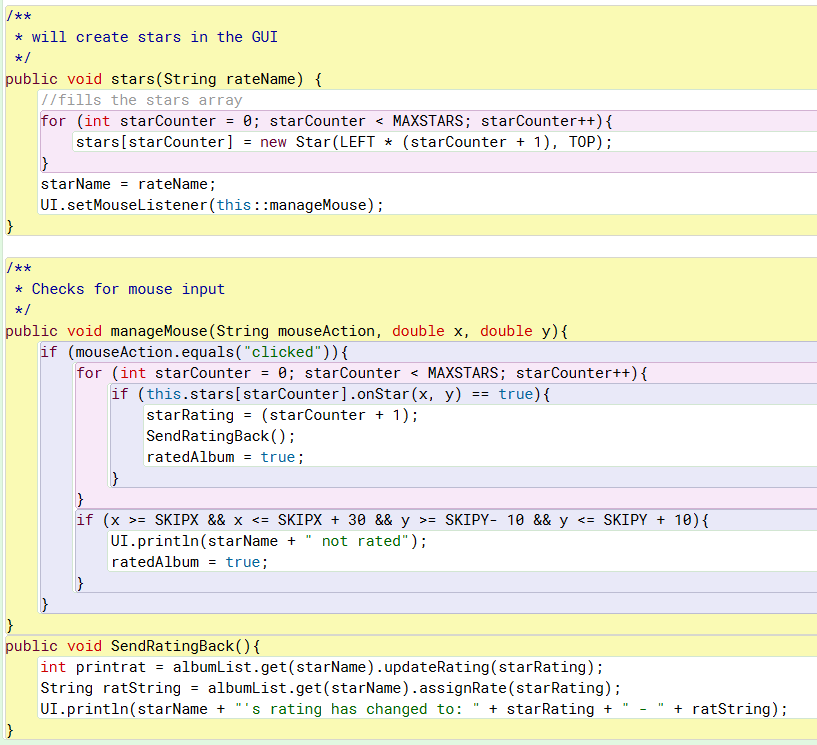
I had a problem where every time I finished rating recommendations it would throw an error and display an error in the albumRating method, I found the problem with this, when using the printArr for the bounds for a for loop, I forgot to indicate that it should be 1 less, because of how the size of an ArrayList is defined as opposed to how Arrays work, starting at 0. I managed to fix it by simply changing the for loop condition to this:



\*printArr is a variable storing the size of the AlbumKeys ArrayList.

Changing this was simple, making it so that the condition for the iterating number was 1 less means that its in line with the ArrayLists bounds rather than always returning a message at the end due to being out of the ArrayLists bounds.

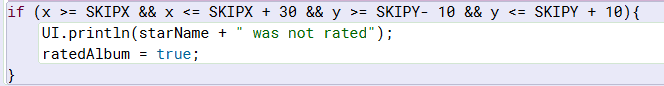
I originally had the MouseListener in a class called Stars, it would link to Star to create and check the coordinates of each star. I ended up changing it as linking to another class while being able to change the rating and grab things from the hashmap and the ArrayList would be hard. I ended up moving the stars array into the Playlist class:



This is right underneath the recommendAlbum method. This makes it easier for me to do things in less methods and classes, streamlining the objects in my code.

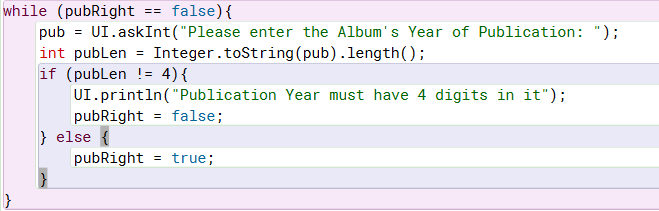
I have added a skip album button, you click on the words “Skip Album” and then it will skip it in the graphics pane, if there are more recommendations it will display the next one, it will also say that the Album in question has been skipped in the shell. I printed the words skip album right underneath all of the album information like this:



After printing it, I found the co ordinates of it on the graphics and then put a conditional for it in the manageMouse class, so that if clicked on, it will skip the current album. Here is what the code looks like:

Clicking on the words will skip the album and clear the graphics pane.

I have also added a condition to the Year entry for albums, the only part of entering an album that should be considered conditional is the Year. I have added a condition where the length of the int must be 4 digits, which means no years any higher than 9999 or lower than 1000. I didn’t know what bounds to put on music as its quite ambiguous. Leaving this entry without a condition wouldn’t be very smart but I can’t limit it too much otherwise it would defeat the usability of my code in the future, especially if the code limited album publication years too much.



This will check if the length of the string is 4 digits, if it is not, it will repeat the question, otherwise it continues.

## Feedback

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | Recommendations and Skipping a recommendation | | |
| **Name** | Matthew Holdaway | **Date** | 4/09/2020 |
| **Feedback** | What if you don’t want to rate one of the albums, you should be able to skip it. Maybe a button to click that lets you skip a recommendation | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** |  | | |
| **Name** |  | **Date** |  |
| **Feedback** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** |  | | |
| **Name** |  | **Date** |  |
| **Feedback** |  | | |

*What is the outcome of this feedback?*

After taking Matthews feedback into consideration, it’s a very smart idea. Some people may not want to rate something immediately and instead will want to jot them down and then listen to them and come back later. Plus, giving users more options is always a better idea, I have decided to add this feature. I will do this in the future, probably after fully refining my recommendation system.

## Testing

***Brief Description of what you are testing***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Type of test**  **(E, B, I)** | **Method to Test** | **Value(s) to enter** | **Expected result** | **Actual result (screen snip / time stamp)** | **Comments/ changes needed** |
| E | recommendAlbum | Clicking the 1st, 2nd and 3rd star. | Clicking the 1st should return a 1 star rating, 2 star for the 2nd and 3 for the 3rd |  | Each star returns an expected value, looking at the searchAll feature, I can also reinforce this rating by looking at the different ratings stored in the hashmap. |
| E | manageMouse | Clicking the skipAlbum button | Clicking skip album should skip the album in recommendations and let you continue with recommendations, should also display this in the shell. |  | Clicking the button skips the album in the graphics pane and lets you continue on with the recommendations or skips it and finishes the recommendations. It also prints this string in the shell. |
| E | recommendAlbum | Having two different Genres stored in the hashmap | I will add two albums, I already have a pop album stored, one Pop and one Rock, they will be called the same as the genres in question, one should show up in the graphics pane to be rated, the pop one, the rock one shouldn’t | This part of the if statement will check if the genre of the album you rated matches any in the hashmap and then store it in the ArrayList for recommendations. | As expected, the recommendation only displayed the Pop album, the rock album was not recommended as it is a different genre, since Thriller is the same genre and I rated that, the recommendation focused on Pop albums. |
| E | recommendAlbum | Rating an album a 1 even with a pop album in the hashmap | Should not show anything in the graphics pane, no recommendation should show if the rating is low | This part of the code has a conditional where it will only add albums to the ArrayList if the rating of the album in rateAlbum is above 1. If not, this wont run and therefore nothing will be printed in the graphics pane | Nothing appears in the graphics pane because of the rating, this is intended as if a person doesn’t like the album that they rated, then you shouldn’t recommend the same genre as they most likely wouldn’t like it. |
| E | recommendAlbum | Having no recommendations available in the hashmap. | Should say “no recommendation available in the hashmap” | The printArr is the length of an ArrayList that stores all of the albums in the hashmap that match the conditions to be a recommendation comparing to the rated album, if there is nothing in the ArrayList, the Try() will throw an error  Here is the error:    This Exception will only occur if the amount of things in the ArrayList is 0. | This works well as it easily shows the user that there are no recommendation in the hashmap. That’s all it needs to tell you. |
| B | addAlbum | 5 digit year and 3 digit year in entry for an Album Publication | Should return an error and let me enter the year again | This code will check the length of the integer by converting it to a string and checking the length of the string, it stores this string in a separate variable. | The condition works well, using more or less than 4 digits in the publication year entry will return an error and loop through. I decided not to put any parameters on the string entries as string entries can be far more ambiguous than integers. |
| B | rateAlbum | An album that is not in the hashmap | Should return “That album is not in our database” |  | This correctly throws the right exception and prints the string in the shell, then does not continue the method. |
|  |  |  |  |  |  |

## Evaluation

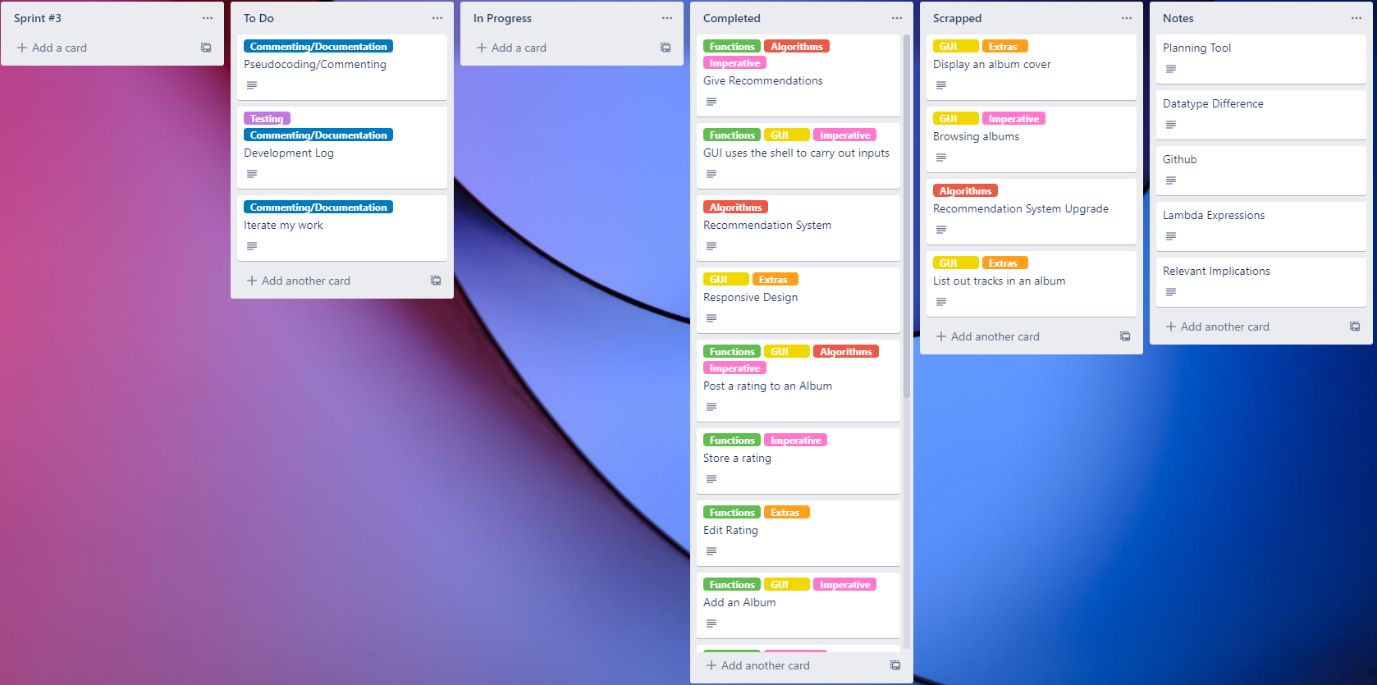
*Sprint reflection and summary*

As my last sprint, I did a lot of work in this sprint, the main amount of work in this sprint was spent doing the recommendation system and the different things involving the graphics pane. Looking back at this sprint, I spent a lot of it making different things rather than fixing things up for my finished product. Due to the time constraint this entire project has been on, I haven’t been able to fully realise my ideals of what this project was meant to be, I finished all of the imperative tasks on my Trello Board and met all the requirements for my Minimum Viable Product but it is in no way refined and clean. If I had more time or had managed my time better, I could’ve refined my code a whole lot more and made it much nicer to work with, but for now it works well and does all that it needs to plus a little more. I also believe that all the different conditionals that I made for some previously added features are really useful as they make the user experience much better and better communicated.

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

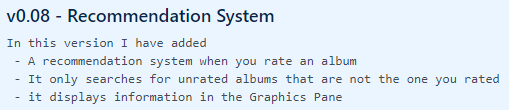
During this sprint I managed to fully complete my recommendation system, I didn’t get to upgrade it like I had planned at the start of the sprint but I managed to make a system that recommends albums based on rating and the genre of the album that was rated. I also managed to adapt this system into the Graphics pane so that the user may choose to rate any of the albums that they were recommended. The user can click stars to rate any recommendations and can click skip album to skip the recommended album, it also allows for multiple recommendations and will not display anything if the rating is low or if there are no recommendations to be stored in the ArrayList.

*Provide evidence (screenshot / photo) of your project management tool(s) being used to manage the development of your outcome at the end of your sprint here*

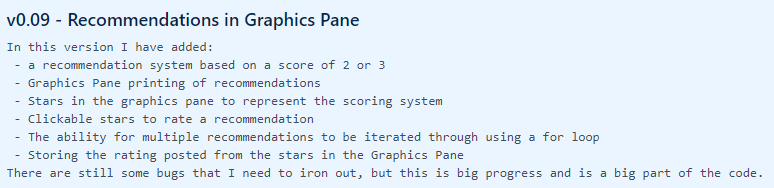


Looking at my trello board compared to at the start of the sprint, everything has been removed out of Sprint #3 as I have either finished said tasks or scrapped them, the two tasks that I have scrapped from Sprint #3 was the Recommendation System Upgrade and the Browsing Albums. I decided to scrap these two as the time constraint on this project meant that I couldn’t see any way to upgrade the recommendation system and not make it completely convoluted and annoying. I also couldn’t see any way to fully make all albums browsable as I hadn’t had made persistent storage and would need multiple albums present in the hashmap from the start to be able to browse them, also I sort of got this working by making the recommendations store inside an ArrayList and then iterating through them in the graphics pane. However, you can’t browse through them but instead iterate through them every time you either rate them or skip them. I’d say this idea was half-completed but wasn’t fully realised into what I envisioned at the start of this project. I have moved “Give Recommendations”, “Responsive Design” and “Recommendation System” into completed. This is because of the recommendation system that I created and also the way to give them to the user in the Graphics Pane. I also included Responsive design as clicking the stars or clicking Skip Album will do different things, it responds to inputs and allows for the user to do different things, letting them have choices instead of being forced into rating a recommendation.

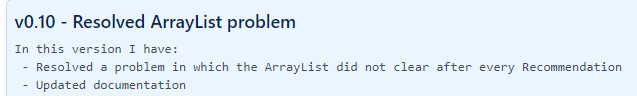
For this sprint I ended up changing up my Github, I created a branch called “Recommendations” and committed directly there until the latest version where I committed it to the master branch. These are the different commits I did during this sprint:



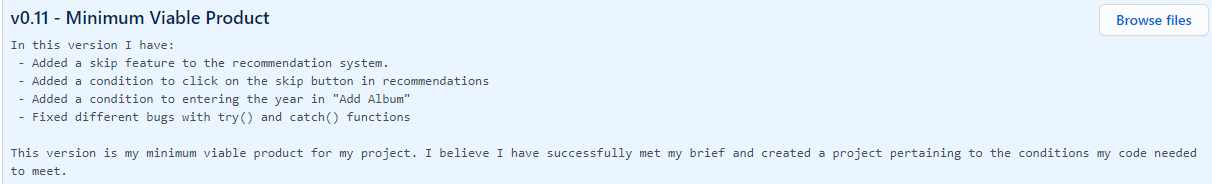
This was the start of the recommendation system, at this point in time there was no way to rate albums in the graphics pane and it would recommend every unrated album stored in the hashmap.



For this commit, I added a lot, I made it so the recommendation checked if the rating you gave the main album was a 2 or 3, it also printed things in the graphics pane and made stars in the graphics pane for you to click that would let you rate the recommended album. Also, this is where it let you iterate through multiple recommendations in the graphics pane. At this point I hadn’t made the “Displaying” message, so the user didn’t know how many recommendations there was at this point in the code’s history. Also, there were still many bugs that I had to iron out, like the bug with the catch in rateAlbum and the problem with printArr. This was a commit with big progress towards finishing Sprint #3.



This was a very small commit, this was when I fixed the printArr problem and updated the documentation, this commit was mainly for me to be able to download it on my laptop at home as at this point in time it was on my USB and only at school.



This is the latest version of my code at this point in time. This version includes a skip feature where you click the words;” Skip Album” to skip a recommendation and not rate it, this idea was given to me from Matthew Holdaway. I also added a condition to adding an album where the publication year must be 4 digits to make it not as unrealistic. I also fixed a few try() and catch() bugs. This was an averagely sized commit; this was the end of the recommendations branch and was then merged with master as to indicate that my code is complete in my eyes. Not fully complete but up to standard that I am happy with presenting.

# Project Summary

* ***addressing relevant implications.***

*How did you address the relevant implications in the development of this outcome?*

* ***synthesising information gained from the planning, testing and trialling of components***

*How did you use the tools, techniques and process of each sprint inform the development of this outcome?*

* ***discussing how this information led to the development of a high-quality digital technologies outcome.***

*How did the process help to shape the development of your outcome? Provide evidence.*