# Netflix roulette
Project assessment report

## Quick links

GitHub repo

Javadoc

Demo

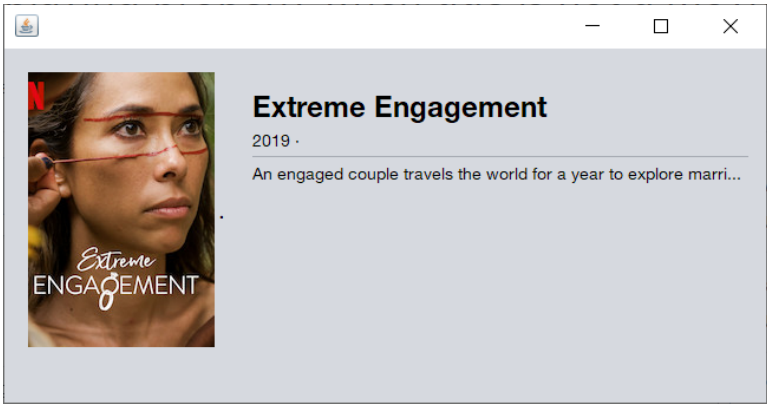
UX assessment

*For more information on user interactions, please view the UX assessment report.*

# Bugfix and improvements

The following is a non-exhaustive list of improvements made during the development of the program. This list only highlights some of the notable changes relating to optimizations that were made to the application during its development. Please also see the Feedback collection and management section for some of the improvements made following user feedback. For the list of all changes made to the program, please check out the commit log.

## Text wrapping on synopsis does not work



Extreme Engagement

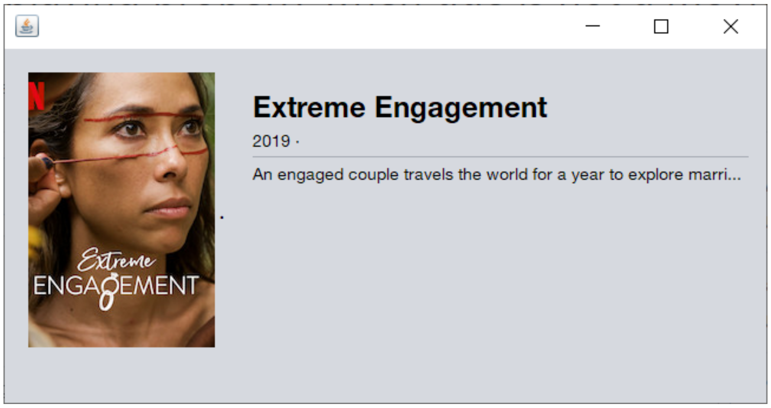
2019 ·

An engaged couple travels the world for a year to explore marri...

Closed by cd951f5.

Fixed by enclosing text in HTML tags. Text in JLabels automatically wraps when marked as an HTML element.

## Subtext not displaying properly when title is not a movie



Extreme Engagement

2019 ·

An engaged couple travels the world for a year to explore marri...

Closed by 3dcf403.

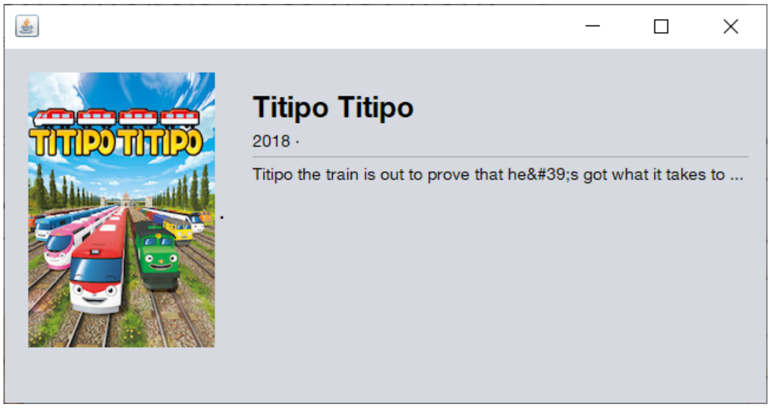
When selected title is a series, subtext of the title name has a trailing interpunct, without runtime. Likely because its runtime is not

applicable and is not returned by the API.
Fixed by adding a case to handle it. Refer to commit for diff.

## HTML codepoints are not unescaped when being displayed in the GUI

Closed by cd951f5.

Wrapping text them in HTML tags also fixed this issue. Sequences are automatically unescaped.



Titipo Titipo

2018 ·

Titipo the train is out to prove that he&#39;s got what it takes to ...

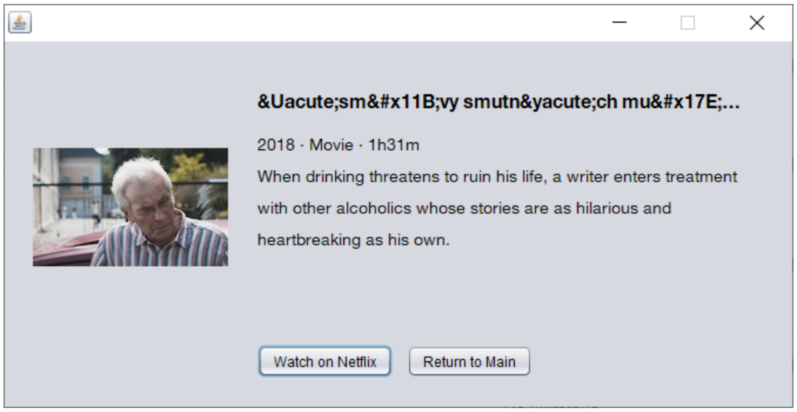
## Handle NullPointers when expected keys are not present in the JSON response

Closed by 2146859.

Initially just prints to console acknowledging error.
Added dialogue box to inform user and request API key when not found.

## Special cases of hexadecimal NCRs are not decoded properly

Closed by 3bec193.


&Uacute;sm&#x11B;vy smutn&yacute;ch mu&#x17E;...

2018 · Movie · 1h31m

When drinking threatens to ruin his life, a writer enters treatment with other alcoholics whose stories are as hilarious and heartbreaking as his own.

## Caching responses

Responses are cached so that it can be used later, for two main reasons:

- uNoGS API has a relatively high response time
- to reduce the consumption of API calls
    - there is a limit to the number of calls that can be made before it starts charging my credit card!

By saving the responses, when the user requests the data with the same parameters, the application can use the responses that were saved instead of sending another query to the API, alleviating wait times and the strain in my wallet.

## Parsing JSON vs instantiating classes

Initially, classes were created to represent the Netflix titles. User feedback said that there were significant delay between pressing the button and awaiting a response.

Aside from the high response time due to latency between the API server and the application, the instantiation of classes for each Netflix title from the response took up a significant amount of time, with the response usually returning up to 100 titles at a time.

The class has been removed and the representation of the JSON response were used to parse the information directly instead. See dab311e.

Because the response from the API server were already being parsed into a class (JSONObject) through the Unirest library, extracting the data from the JSON response object itself to create *another* class for *each* of the responses appears to have created a significant delay.

## Avoiding direct user inputs

All user input are controlled as much as possible. The program is designed to avoid parsing strings manually inserted by users. All inputs are done via fields, such as buttons and drop-down lists, so that users do not have to manually enter values themselves

Default values are also set in place, in case the user neglects those fields or are unsure how to proceed.

Due to the nature of the program, it has been decided that the user only really needs to provide one input, that is the Netflix region. Because not all titles are available in all regions, the user must provide the region that they are in, so that the selected title can actually be viewed from their location.

Instead of a text field for the user to type in a drop-down box ("combo box") has been used so that the user can simply select their region. This avoids having to deal with bad inputs, escaping characters, converting cases, etc. The displayed list is derived from the list of available Netflix regions returned from the uNoGS API, or its saved version. This ensures that the input will always be valid, no matter which item the user chooses to use. The default value of "All regions" is chosen, in case the user does not choose a region.

The only way the list may not be valid is if the response from the uNoGS server is not in an expected format. An error message will be displayed if it cannot be extracted properly. See the Sustainability section for further.

Another input that the user could provide is the selection of genres. This is implemented using checkboxes. Again, to ensure that the given input is valid, the checkboxes will be used to map the selection of the available genres, which will then be used to construct a query string.

The addition of checkboxes comes from user feedback that they wanted a way to narrow down the type(s) of titles that they are looking for, despite the initial judgement. Again, instead of providing a way for user to directly input their options, the checkboxes eliminates the need to check for input.

Radio buttons to select the title type (movie or series) and sliders to select rating values have also been added, following feedback suggestions. They were able to be added easily to work with the existing logic, as the code has been kept separate to complete distinct tasks. The only value that had to be modified were the construction of query string, which is used to request titles.

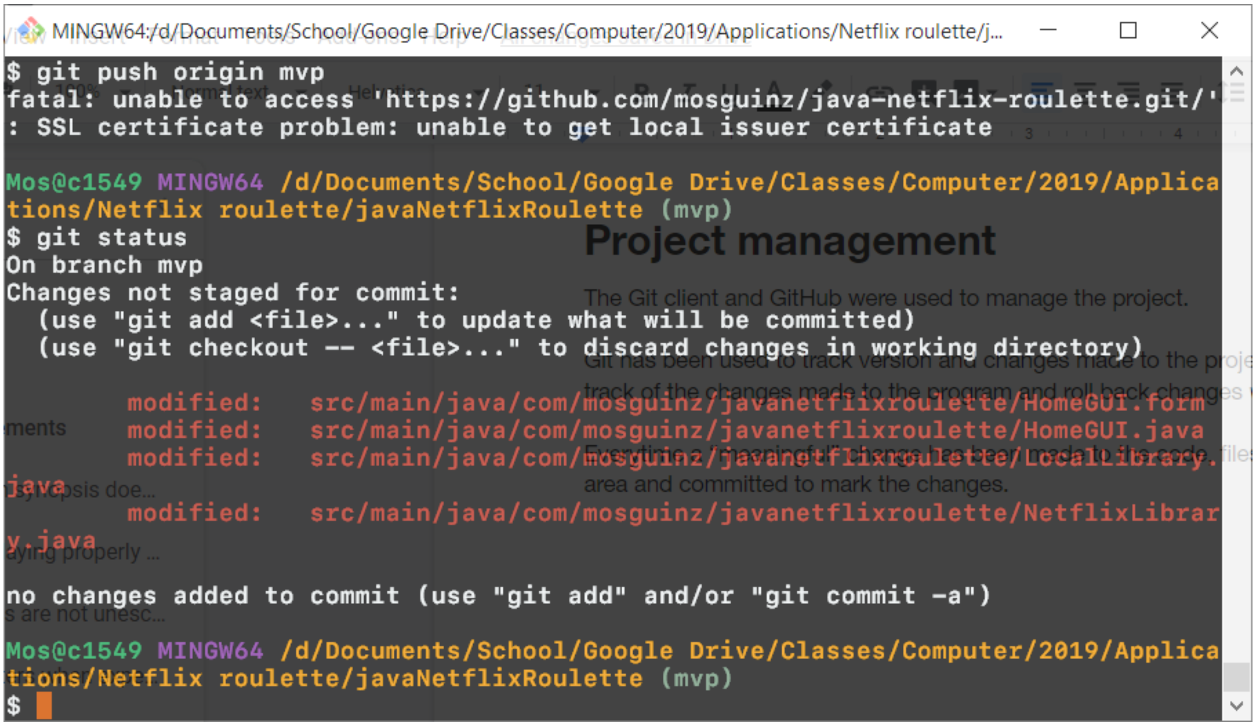
# Project management

## Version control: local Git client and GitHub

The Git client and GitHub were used to manage the project.

Git has been used to track version and changes made to the project. This allowed me to keep track of the changes made to the program and roll back changes when needed. Using Git Bash allowed me to efficiently manage my project, rather than using a cumbersome GUI interface.

Everytime a “meaningful” change has been made to the code, files are staged and committed to the branch to mark the changes made.



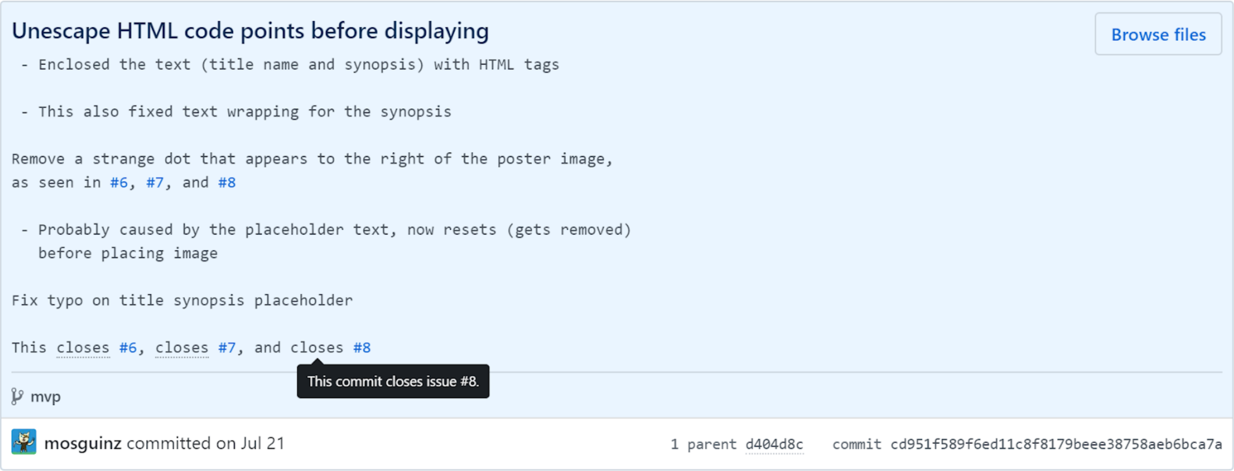
MINGW64:/d/Documents/School/Google Drive/Classes/Computer/2019/Applications/Netflix roulette/j...

```
$ git push origin mvp
fatal: unable to access 'https://github.com/mosguinz/java-netflix-roulette.git/'
: SSL certificate problem: unable to get local issuer certificate

Mos@c1549 MINGW64 /d/Documents/School/Google Drive/Classes/Computer/2019/Applica
tions/Netflix roulette/javaNetflixRoulette (mvp)
$ git status
On branch mvp
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   src/main/java/com/mosguinz/javanetflixroulette/HomeGUI.form
        modified:   src/main/java/com/mosguinz/javanetflixroulette/HomeGUI.java
        modified:   src/main/java/com/mosguinz/javanetflixroulette/LocalLibrary.
java
        modified:   src/main/java/com/mosguinz/javanetflixroulette/NetflixLibrar
y.java

no changes added to commit (use "git add" and/or "git commit -a")

Mos@c1549 MINGW64 /d/Documents/School/Google Drive/Classes/Computer/2019/Applica
tions/Netflix roulette/javaNetflixRoulette (mvp)
$ 
```

The most important part of commiting changes are the commit messages. They are short, but detailed messages outlining the changes made to the code. These messages are crucial in identifying the changes made, when an issue arises.

To make identifying issues easy, changes are also committed often, rather than making huge changes in the code and committing them all at once. The commit messages helps in identifying the problem with the commits. And before each commit, tests are run to ensure that they produce the intended outputs.

Commit messages are also used to note how a change was made to address a certain issue. Alongside the issue tracker on GitHub, which is used to keep track of things to do for the project, including bugs and issues from user feedback, commit messages also references the corresponding issue numbers, so that they can be closed.

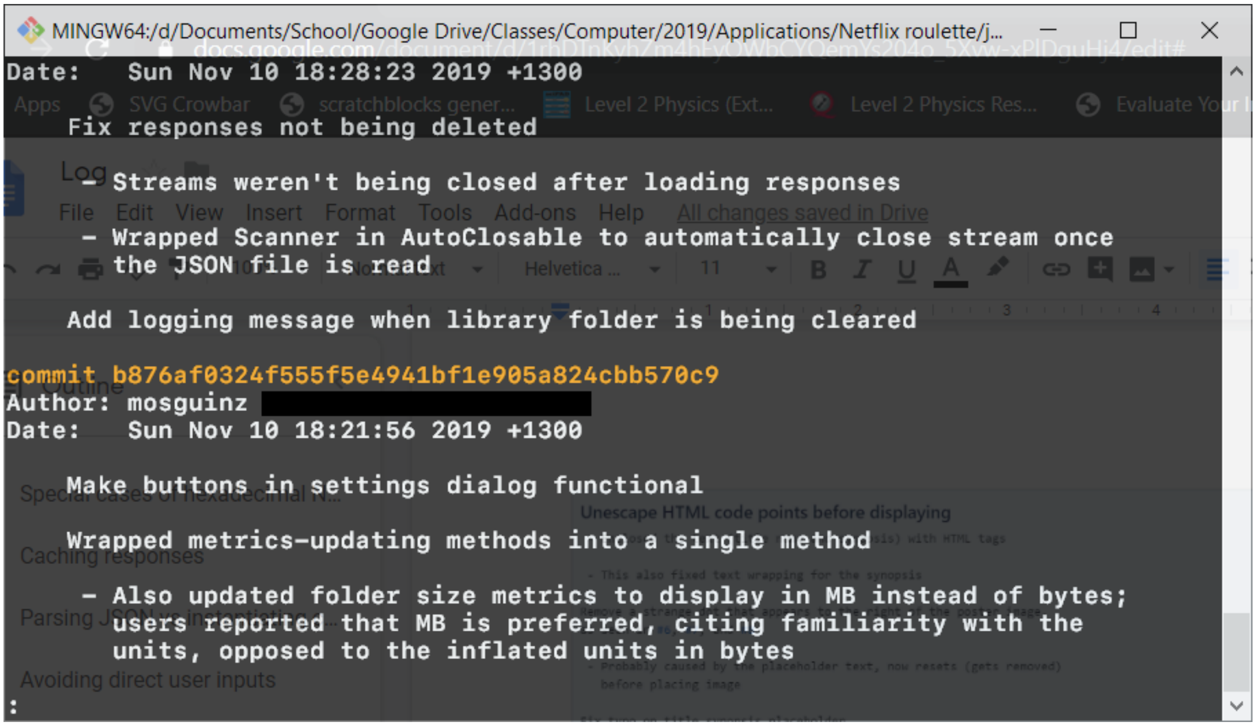


Unescape HTML code points before displaying

```
 - Enclosed the text (title name and synopsis) with HTML tags

 - This also fixed text wrapping for the synopsis

Remove a strange dot that appears to the right of the poster image,
as seen in #6, #7, and #8

 - Probably caused by the placeholder text, now resets (gets removed)
   before placing image

Fix typo on title synopsis placeholder

This closes #6, closes #7, and closes #8
```

mosguinz committed on Jul 21 1 parent d404d8c commit cd951f589f6ed11c8f8179beee38758aeb6bca7a

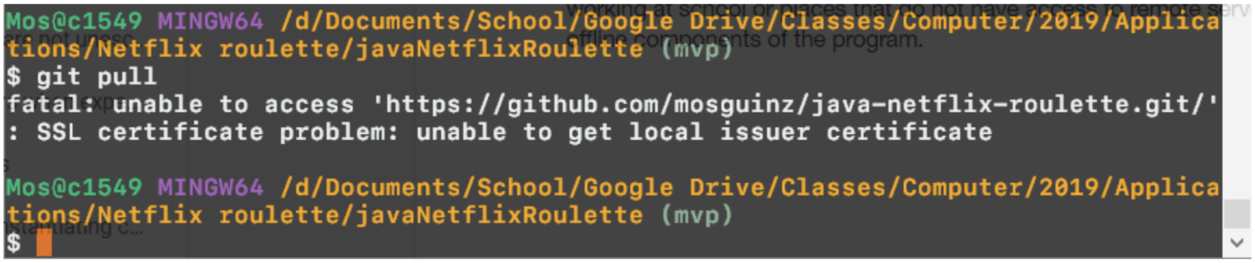
These changes can be checked at any time using the **git log** command, or on the GitHub.


MINGW64:/d/Documents/School/Google Drive/Classes/Computer/2019/Applications/Netflix roulette/j...

```
Date:   Sun Nov 10 18:28:23 2019 +1300

    Fix responses not being deleted

      - Streams weren't being closed after loading responses

      - Wrapped Scanner in AutoClosable to automatically close stream once
        the JSON file is read

    Add logging message when library folder is being cleared

commit b876af0324f555f5e4941bf1e905a824cbb570c9
Author: mosguinz [redacted]
Date:   Sun Nov 10 18:21:56 2019 +1300

    Make buttons in settings dialog functional

    Wrapped metrics-updating methods into a single method

      - Also updated folder size metrics to display in MB instead of bytes;
        users reported that MB is preferred, citing familiarity with the
        units, opposed to the inflated units in bytes
:
```

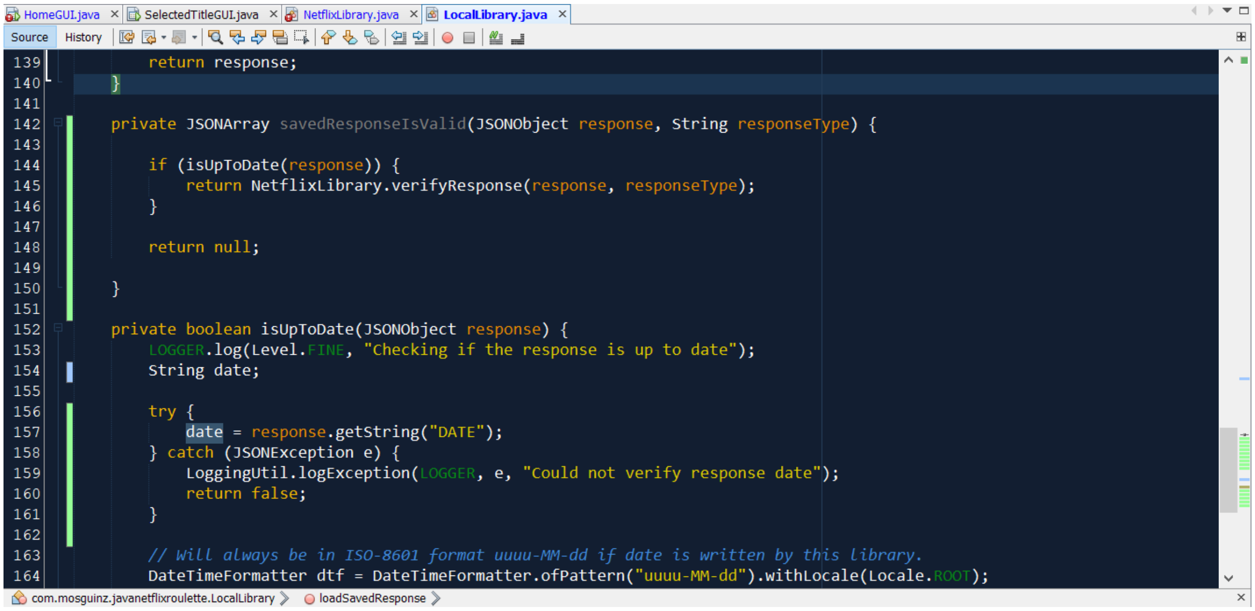
Usually after two or three commits on the local repository, they are pushed to the remote branch (GitHub), so that changes can be saved online. This allows the project to be managed across different computers and provides a backup, should my laptop become inaccessible.

Having a local Git client also allowed me to make commits to the local Git repo before pushing them to the remote branch. It also allowed changes to be committed when I am working at school (where access to the remote repository is restricted, thanks to the school's proxy), allowing me to work on offline components of the program.



```
Mos@c1549 MINGW64 /d/Documents/School/Google Drive/Classes/Computer/2019/Applications/Netflix roulette/javaNetflixRoulette (mvp)
$ git pull
fatal: unable to access 'https://github.com/mosguinz/java-netflix-roulette.git/': SSL certificate problem: unable to get local issuer certificate

Mos@c1549 MINGW64 /d/Documents/School/Google Drive/Classes/Computer/2019/Applications/Netflix roulette/javaNetflixRoulette (mvp)
$
```

The local Git repo also integrates nicely with NetBeans, which displays modified/changed/deleted lines in the editor -- kind of like a live **git diff**, which allows me to easily spot the changes I've made, so that they can be outlined in commit messages. And also serves as a reminder for me to stop making changes and start testing, so that those changes can be staged, opposed to making massive changes that can become hard to track.



```java
        return response;
    }

    private JSONArray savedResponseIsValid(JSONObject response, String responseType) {

        if (isUpToDate(response)) {
            return NetflixLibrary.verifyResponse(response, responseType);
        }

        return null;

    }

    private boolean isUpToDate(JSONObject response) {
        LOGGER.log(Level.FINE, "Checking if the response is up to date");
        String date;

        try {
            date = response.getString("DATE");
        } catch (JSONException e) {
            LoggingUtil.logException(LOGGER, e, "Could not verify response date");
            return false;
        }

        // Will always be in ISO-8601 format uuuu-MM-dd if date is written by this library.
        DateTimeFormatter dtf = DateTimeFormatter.ofPattern("uuuu-MM-dd").withLocale(Locale.ROOT);
```

## Feedback collection and management

Since the project has a GUI, I have created mockups on Adobe XD of what different versions of the program may look like and mimicking its results. Feedback from most respondents have said that the design of the program was "simple" and "pretty straightforward" to use.

However, unlike the prototypes on XD, which can be easily shared and viewed on browsers, the program itself could not be easily shared via something like a URL for users to test. In order for users to run the compiled program, their systems must have Java runtime environments installed, which is less common among users to have, and is therefore harder to get them to test on their systems.

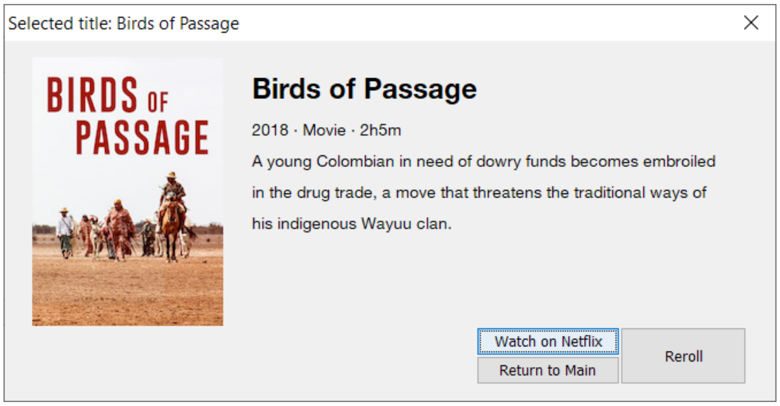
Furthermore, due to unresolvable issues with the compiler, a runnable JAR file could not be created for distribution, even for systems that meets the requirements. During testing, it appeared that the compiler had failed to include the dependencies required to run the program.

However, when the program was able to be run through the IDE normally. So, feedback was only gathered from users interacting with the program from my computer, mainly my CS classmates.

Any feedback, especially bugs and issues, are added noted on the GitHub Projects board and turned into issues, so that they can be referenced in commits. Below are some of the notable changes made as a result of user feedback. Also refer to the list of bugfix and improvements from above.

## Reroll button

A user suggested that a reroll button should be added to the interface, so that users would be able to easily find another title without closing the popup. See f69575b.



Selected title: Birds of Passage

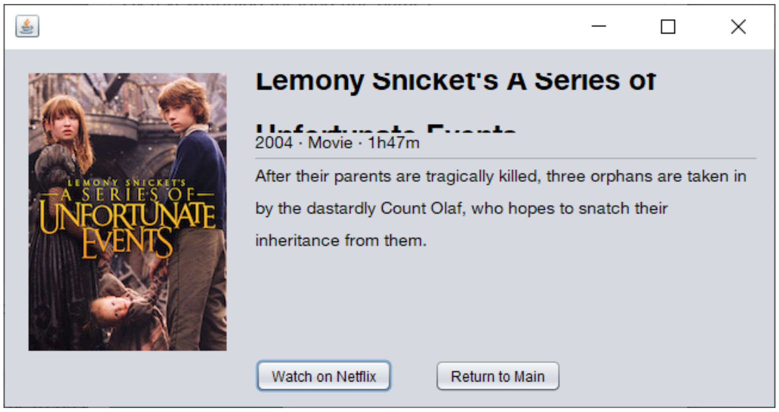
Birds of Passage

2018 · Movie · 2h5m

A young Colombian in need of dowry funds becomes embroiled in the drug trade, a move that threatens the traditional ways of his indigenous Wayuu clan.

## Text wrapping/resizing for text fields

Users reported that texts in the synopsis area was not being displayed properly. This was addressed multiple times at 3dcf403 for subtext labels, cd951f5 for synopsis text, and 6b6e2ed for title names.



Lemony Snicket's A Series of

Unfortunate Events

2004 · Movie · 1h47m

After their parents are tragically killed, three orphans are taken in by the dastardly Count Olaf, who hopes to snatch their inheritance from them.

## Genre and title type selection

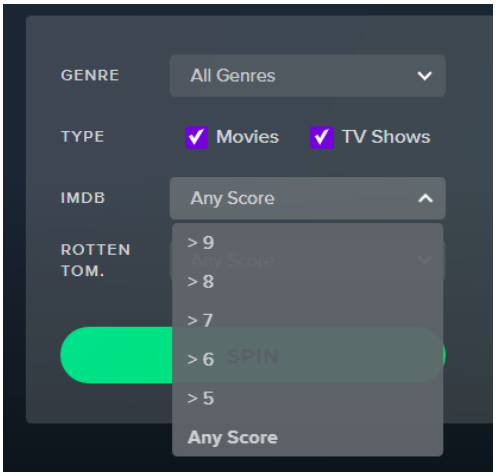
Users have reported that they wanted a way to be able to select specific types of genres to be returned. This was implemented, despite initial decision against it due to repeated requests for this feature. See 7c1e13e and 8f23275.

Aside from this request, the first iteration of the application was said to be “simple to use and effective” and “good 👌.” Users also requested for a reroll button, which was addressed later on.

## Rating filter

Rating filter was added as an addition to the list of filters that users could use. It was initially planned to be implemented as a combo box, where users may choose ratings in increments of two, where each would return titles that have ratings greater than the selected values. For example, > 6.0, > 8.0, > 9.0, etc.

This idea initially came from the implementation on [reelgood.com](reelgood.com), where a list was used. However, a user said that there should be a way to specify both the minimum and maximum rating values, rather than just a “greater than *x*” selection, reasoning that some users may want to look for “bad movies.”



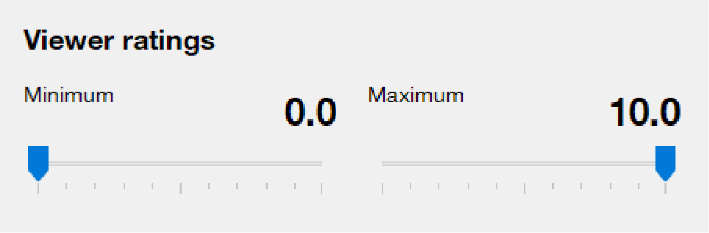
GENRE All Genres

TYPE ☑ Movies ☑ TV Shows

IMDB Any Score

ROTTEN TOM.

This idea was implemented using sliders instead, where users can select the upper and lower bound values of title ratings. See [060bc99](060bc99).



Viewer ratings

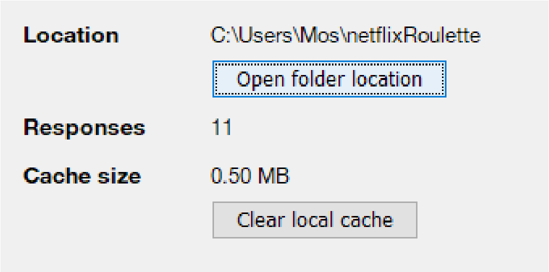
Minimum 0.0

Maximum 10.0

After the implementation, the user also pointed out a potential issue, where a user could select a minimum rating that is greater than the maximum rating value. While this issue was handled, as the API would not return a title, this feedback allowed me to implement a validation logic, so that the request would not have to be sent in the first place. Thus, saving API consumption costs and avoids the user having to wait for the request to be sent. See [0516b08](0516b08).

## File size units

Originally, file size was displayed in bytes. However, users suggested that the number should be in megabytes instead, citing the familiarity with the usage of the unit in everyday life. See [b876af0](b876af0).



Location C:\Users\Mos\netflixRoulette

Responses 11

Cache size 0.50 MB

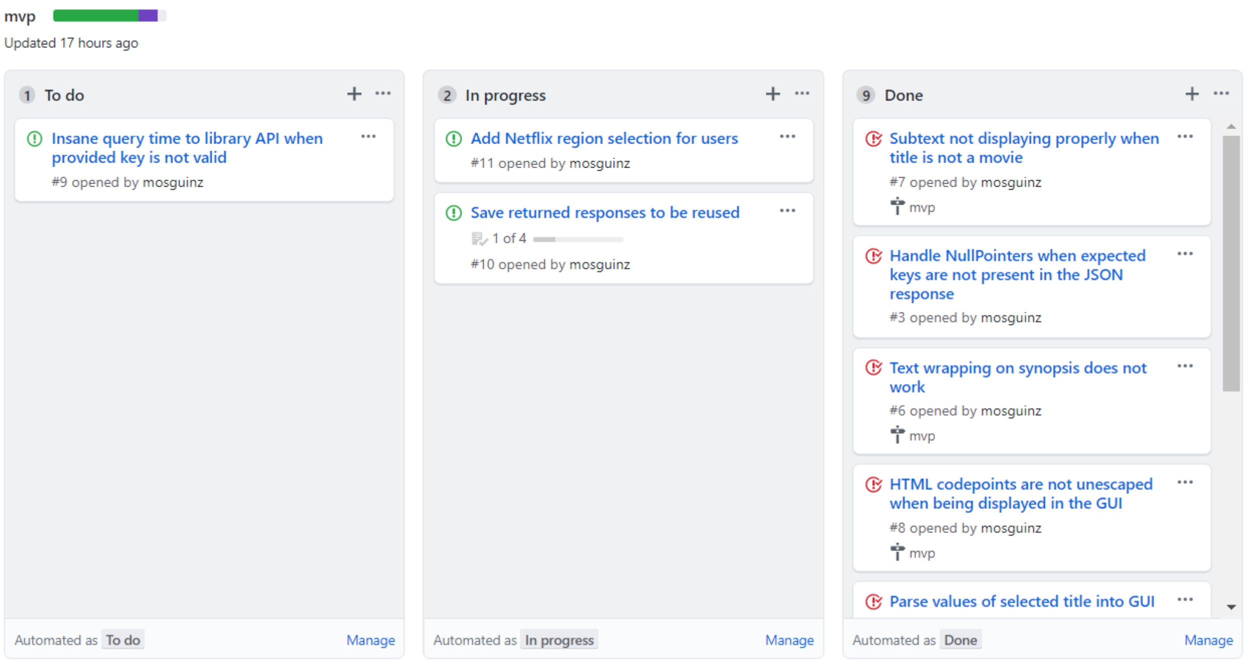
### Other changes

The above list only highlights the most notable changes made from feedback. There are many more minor changes that were made from feedback that are not listed. For a complete list of changes made due to user feedback, please refer to the commit messages.

## Workflow and task management

The GitHub Projects board was used to manage tasks and plan out the development of the program. The board is an implementation of the kanban board, much like Trello.

See the board here.



mvp
Updated 17 hours ago

1 To do

Insane query time to library API when provided key is not valid
#9 opened by mosguinz

Automated as To do — Manage

2 In progress

Add Netflix region selection for users
#11 opened by mosguinz

Save returned responses to be reused
1 of 4
#10 opened by mosguinz

Automated as In progress — Manage

9 Done

Subtext not displaying properly when title is not a movie
#7 opened by mosguinz
mvp

Handle NullPointers when expected keys are not present in the JSON response
#3 opened by mosguinz

Text wrapping on synopsis does not work
#6 opened by mosguinz
mvp

HTML codepoints are not unescaped when being displayed in the GUI
#8 opened by mosguinz
mvp

Parse values of selected title into GUI

Automated as Done — Manage

Given that it is directly implemented on the GitHub platform, it can be easily integrated into the development of my program. Cards on the board can be turned into issues and referenced in commits, making it easy to see the changes that addresses the issues.

Having the board set up allows me to easily see the tasks that needs to be done, and order them in levels of priority. For example, low occurrence and less critical issues are addressed last, so that more crucial ones -- i.e., ones that prevents the program from functioning, can be addressed first.

# Relevant implications

## Legal

A disclaimer has been placed in the “About” window to notify end users that the usage of the trademark “Netflix” is only for demonstration purposes only.

The data obtained about titles and catalogue are provided by a third-party API provider, Unofficial Netflix Online Global Search (“uNoGS”) via RapidAPI.

As an API provider on RapidAPI, uNoGS agrees that it adheres to and are responsible for all the support and all claims relating thereto (e.g., product liability, legal compliance or intellectual property infringement). As mentioned in RapidAPI’s Terms of Service (https://rapidapi.com/terms).

This application is not endorsed by, directly affiliated with, maintained, authorized, or sponsored by Netflix, Inc., uNoGS, or RapidAPI. All product names, trademarks, and registered trademarks are property of their respective owners. All company, product and service names used in this application are for identification purposes only. Use of these names, trademarks, and brands does not imply endorsement.

Poster images are owned by their respective publisher or creator of the work depicted. The images used are for demonstration purposes only and not solely for illustrations. These images are of low resolution and may qualify as fair use or fair dealing for educational purposes.

## Sustainability

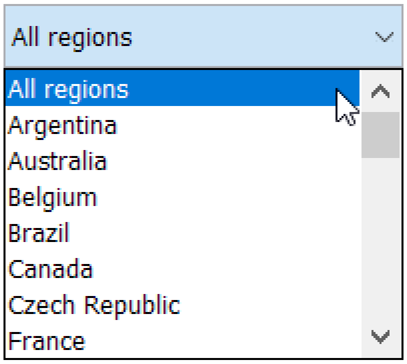
### Response format

The application depends on the uNoGS API and the format of its response. In essence, the way this application works is by sending a query to the catalogue with the selected parameters, and then displaying the response (or a random item from the response) to the end user.

Therefore, for the application to function properly, the format of the response from the API must be in an expected format, otherwise the program may have trouble parsing the information from the JSON and utilizing them properly. This issue is unavoidable, and the only way to ensure that the program uses the right information is by manually modifying the way that the application extract the information.

Appropriate measures have been put in place in the case where the application may receive a response that does not match the format that it expects. When it fails to extract the response (for example, if a key can't be found or if its value cannot be parsed), it will raise an exception which will be logged, so that anyone that may be resolving this issue will be able to identify the issue with the response (see here and here). And it will display an error message, notifying the user that the application may be outdated.

## Response-dependent GUI components



Although the list of Netflix regions and genres have remained consistent in the previous years, they have been designed in a way so that, if a change in those list occurs, the GUI can be adjusted accordingly.

The combo box used for Netflix region selection depends on the list of regions that Netflix is available in. It should be noted that this does not appear to be an exhaustive list of the regions that Netflix is available in. For example, New Zealand is not included in the list, but can be assumed that it likely shares the same catalog with Australia, which is included. The response from the API may indicate that the list may be derived from the result of web scraping, as noted here.

Even though the list is unlikely to change since Netflix's major international expansion in 2016, it is not hard-coded with the combo box itself, but rather uses the list that is returned from the API or its cached version.

The checkboxes also behaves in a similar manner. Instead of being hard-coded with its labels, they are dynamically generated using the list that it has gathered from the API or its cached version. Similar to the list of regions, the list of supercategory genres (see what is meant by this here) also appears to be consistent over the previous years. Its behavior is highlighted below, in the following section.

## Code structure

### Code conventions and documentation

In order for the program to be easily expanded (or graded) by other users, it has been made with the language's convention in mind.

While I do not know much about Java's convention, there are a few that is quite common throughout different languages:

- class names are always in UpperCamelCase,
- constants are always in ALL_CAPS,

- variable and method names are in lowerCamelCase.

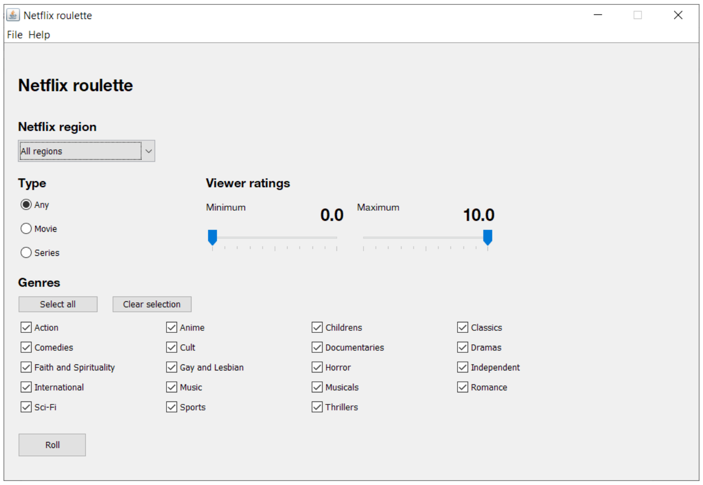
This is done so that the components can be easily identifiable by users that are interpreting the code. Its name are also chosen to describe its purpose, in a way that they are self-explanatory, so that they are immediately obvious to other users, and perhaps myself in a few years time. NetBeans also has a built-in linting tool that makes sure that the code follows a few of Java's styling conventions.

Classes, class variables, constants, and methods, are all documented in order for anyone reading the code to be able to clearly identify its purpose and function -- even if its names are self-explanatory. They are written to work with the Javadoc tool, so that it can be generated in HTML format, viewable here. The format allows for the documents, related methods, and external documentations to be linked together, so that they can be easily accessed.

Comments within the modules are also added where necessary, where the logic behind a method may not be immediately obvious. This allows anyone reading the code, myself included, to better understand how the logic behind a method works, so that changes can be easily made when needed.

## Logic separation

Components that are responsible for constructing GUIs are kept separate from other logical components. For example, this application has two primary GUIs: a "main menu" and an interface to display a Netflix title. The GUI components are named with "GUI" at the end of their filenames, so that they can be easily identified.



Netflix roulette

Netflix roulette

Netflix region
All regions

Type
● Any
○ Movie
○ Series

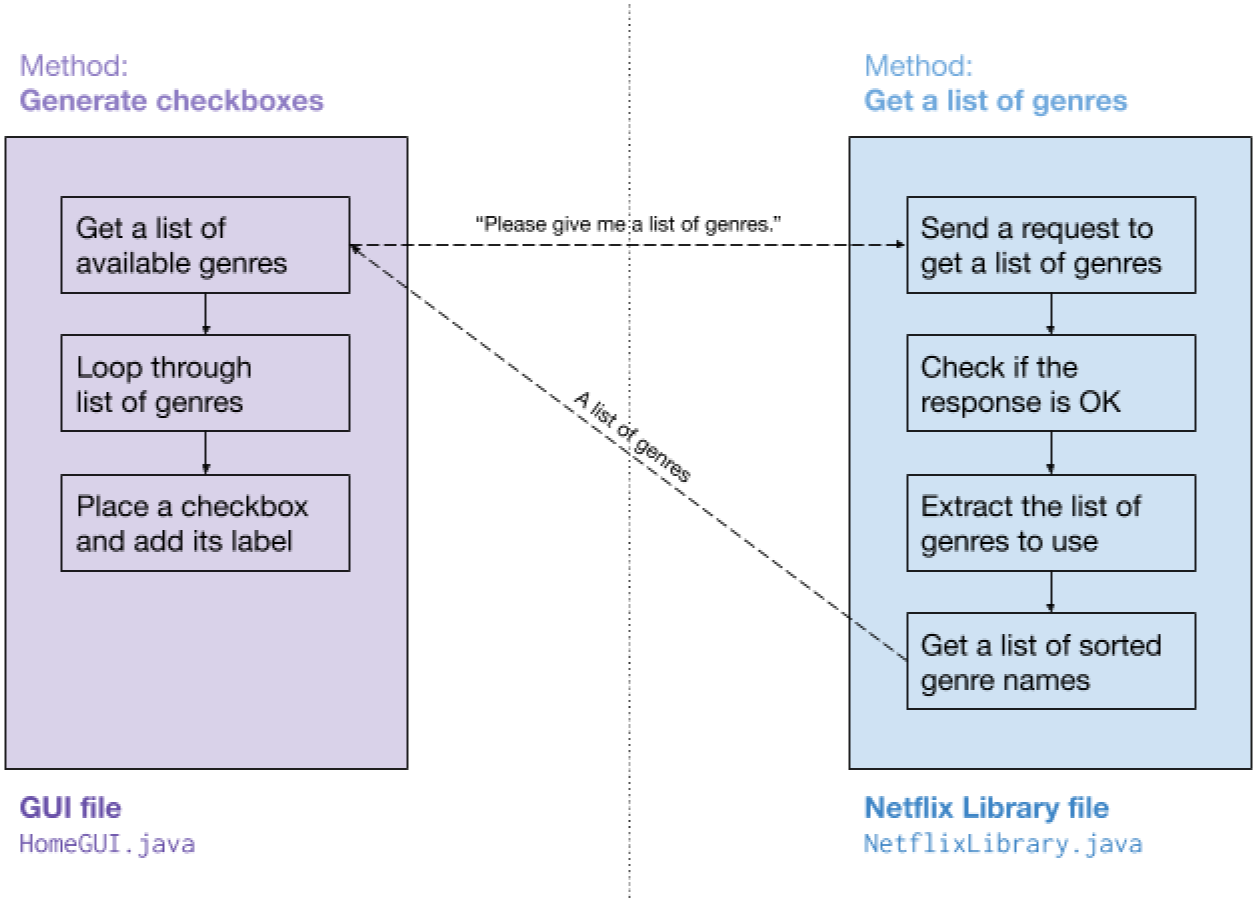
Viewer ratings
Minimum 0.0
Maximum 10.0

Genres
☑ Action ☑ Anime ☑ Childrens ☑ Classics
☑ Comedies ☑ Cult ☑ Documentaries ☑ Dramas
☑ Faith and Spirituality ☑ Gay and Lesbian ☑ Horror ☑ Independent
☑ International ☑ Music ☑ Musicals ☑ Romance
☑ Sci-Fi ☑ Sports ☑ Thrillers

The main menu is responsible for taking user inputs of the filters they would like to apply to the search. So, its file only contains the logic that is responsible for:

- verifying the user's input
  *e.g., checking that at least one genre is selected*
- constructing the interface itself
  *e.g., generating checkboxes, displaying slider values, region lists, etc.*

Anything else that does not directly involve this interface will not be in the file. For example, the checkboxes are dynamically generated using the list of genres returned from the API. The logic that is responsible for generating the checkboxes itself belongs to the class, but the logic that is responsible for getting the list of genres (which involves extracting the response, validating the file, checking the timestamp, etc.) will be placed in other classes. Similar procedures also applies to the drop-down list of Netflix regions.


Method:
Generate checkboxes
Get a list of available genres
Loop through list of genres
Place a checkbox and add its label
GUI file
HomeGUI.java
"Please give me a list of genres."
A list of genres
Method:
Get a list of genres
Send a request to get a list of genres
Check if the response is OK
Extract the list of genres to use
Get a list of sorted genre names
Netflix Library file
NetflixLibrary.java

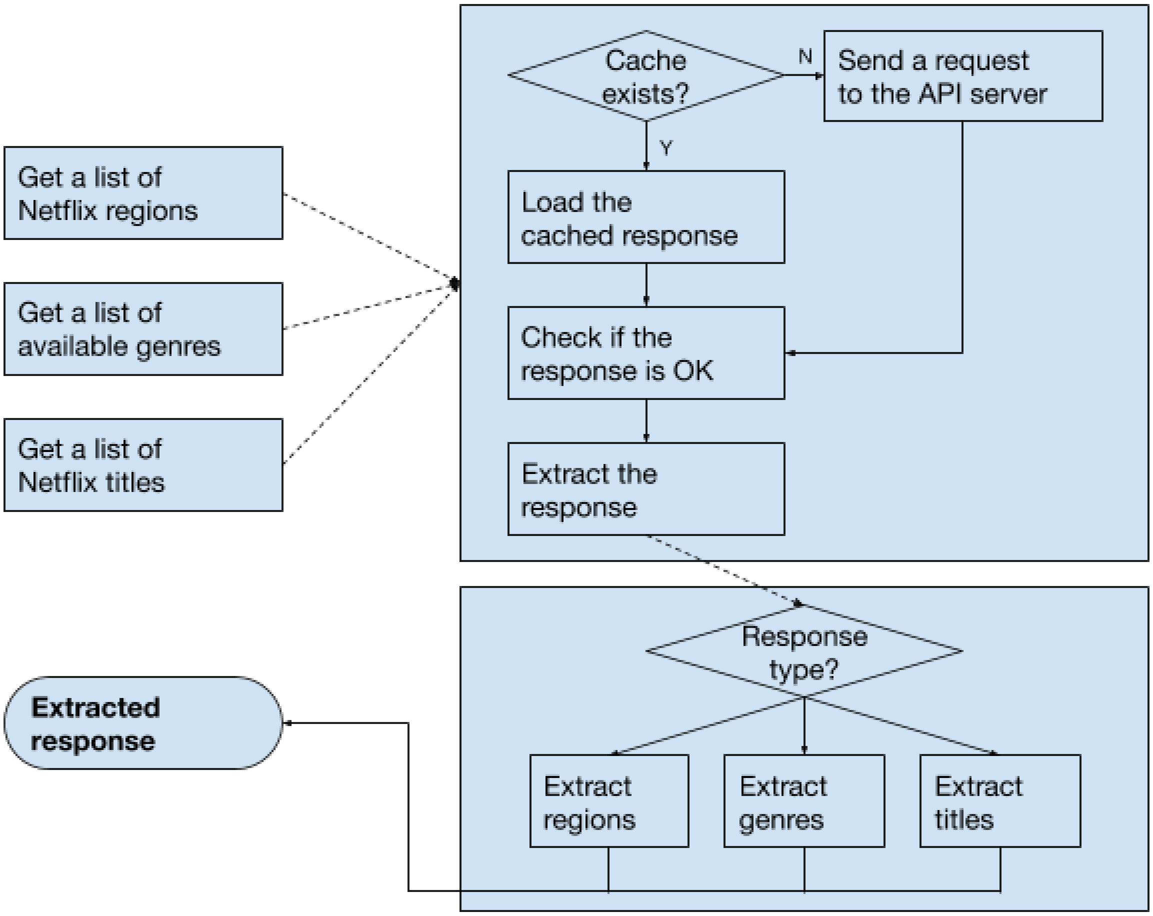
Methods within classes are also kept separately and are designed in a way to complete distinct tasks. This is done to: make the code logic easy to follow, avoid the duplication of code, and allow changes to be made without tearing your hair out.

For example, there are three types of responses: a list of Netflix regions, a list of available genres, and a list of titles. Although, the format of these responses are different, they all use the same method to load the response, by either sending a request to the API or loading a cached response.

First, keeping the module that is responsible for fetching the response separate allows it to be reused for different types of responses, thus avoiding that part of the logic to be duplicated across different types.

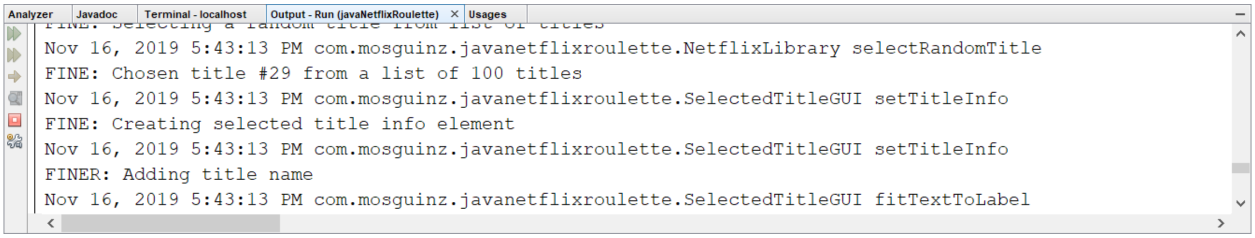
Second, it makes the logical process of getting a response easy to follow. With the logic separated into different methods, you can easily see the process that is behind the loading, extraction, and validation of the responses.

Lastly, changes can be easily made to a specific part of the logic, without making large changes elsewhere and causing a huge headache. For example, if the format of the response from the API has changed and the current extraction logic no longer work as intended, you would only need to modify the logic where the extraction of the response occurs. And the rest of the application can be left as-is.


Get a list of Netflix regions
Get a list of available genres
Get a list of Netflix titles
Cache exists?
N
Send a request to the API server
Y
Load the cached response
Check if the response is OK
Extract the response
Response type?
Extract regions
Extract genres
Extract titles
Extracted response

## Logging

A logging module is also added to the code, allowing the steps that the application takes to be viewed during its runtime. This is especially useful when debugging the program, as it the logger prints the events and exceptions to the console as the application runs into an issue.



```
FINE: Selecting a random title from list of titles
Nov 16, 2019 5:43:13 PM com.mosguinz.javanetflixroulette.NetflixLibrary selectRandomTitle
FINE: Chosen title #29 from a list of 100 titles
Nov 16, 2019 5:43:13 PM com.mosguinz.javanetflixroulette.SelectedTitleGUI setTitleInfo
FINE: Creating selected title info element
Nov 16, 2019 5:43:13 PM com.mosguinz.javanetflixroulette.SelectedTitleGUI setTitleInfo
FINER: Adding title name
Nov 16, 2019 5:43:13 PM com.mosguinz.javanetflixroulette.SelectedTitleGUI fitTextToLabel
```

Log messages are configured to be of different levels, so that the logging output can be controlled. For example, messages with the “FINE” level contains messages that are for detailed tracing. And higher level messages like “WARNING” and “SEVERE” contains messages and tracebacks of events when a serious failure occurs that prevents a normal program execution.

### Structuring limitations

Because the GUI is made using NetBeans, there are a lot of code generated by NetBeans that cannot be refactored or documented. This may decrease the maintainability of the program. However, efforts have been made to contain those elements. All code that contains generated content are within the classes that contains the underlying code for the GUI.

## Security

Given that this application relies on the consumption of an external API, uNoGS, an API key for this API is required in order for this application to function.

API credentials are never, ever hardcoded into the program. Instead, for testing purposes, they are stored in the system environment variables. This prevents the token from being exposed to the public, which will allow anyone to consume the API.

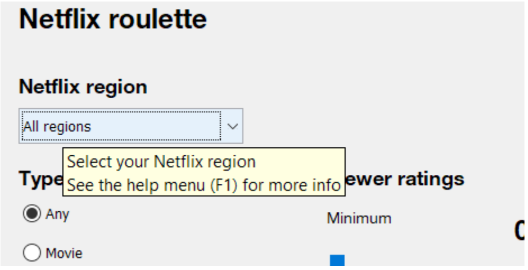
The uNoGS API is a freemium, meaning that there is a quota limit to the number of calls that can be made to the API for free, after which it will start charging my credit card. Committing the token to a public repository exposes it to be used by anyone, which will put a hole in my wallet, which is not nice.

Because of this, the application cannot be distributed for testing, as the end users will not have the API key stored in the system. Instead, feedback is gathered by asking users to directly interact with the program from my computer.

## Accessibility

While the application is not built with accessibility in mind, some measures have been put in place to ensure that most users could use the application.

The sequence of the GUI components have been ensured that they are ordered in a sensible sequence, so that screen readers would read out the text labels and tooltips in the correct order.


Netflix roulette

Netflix region

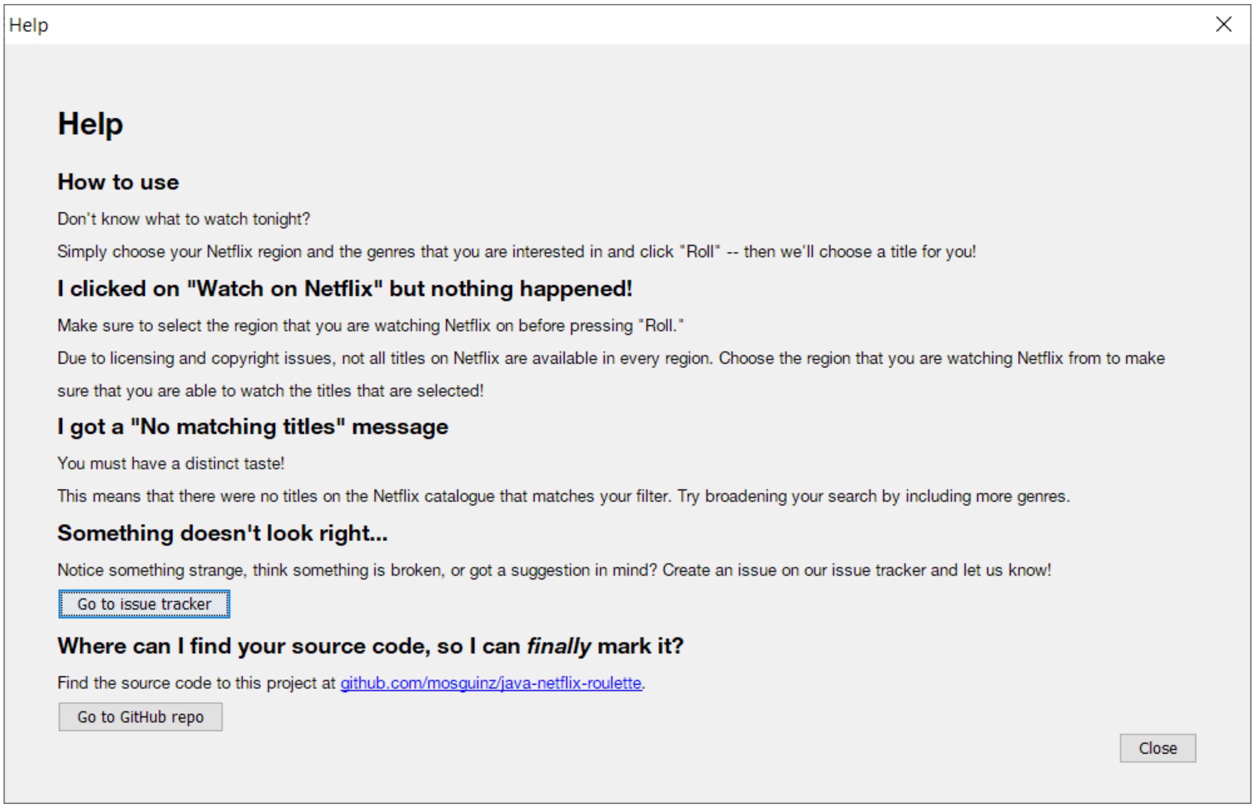
All regions

Select your Netflix region
See the help menu (F1) for more info

Tooltip texts have also been added to buttons and components to describe their purpose. Explanatory texts are added to less self-explanatory labels and interactive components for screen readers, and also simply to guide users.

## End-user considerations

A help dialog is included in the application to guide the user on how to use the application and how to troubleshoot common issues that they may run into. A button to open the issue tracker on GitHub has also been added for users to report bugs.



Help

How to use

Don't know what to watch tonight?

Simply choose your Netflix region and the genres that you are interested in and click "Roll" -- then we'll choose a title for you!

I clicked on "Watch on Netflix" but nothing happened!

Make sure to select the region that you are watching Netflix on before pressing "Roll."

Due to licensing and copyright issues, not all titles on Netflix are available in every region. Choose the region that you are watching Netflix from to make sure that you are able to watch the titles that are selected!

I got a "No matching titles" message

You must have a distinct taste!

This means that there were no titles on the Netflix catalogue that matches your filter. Try broadening your search by including more genres.

Something doesn't look right...

Notice something strange, think something is broken, or got a suggestion in mind? Create an issue on our issue tracker and let us know!

Where can I find your source code, so I can *finally* mark it?

Find the source code to this project at github.com/mosguinz/java-netflix-roulette.

# Summary

The GitHub Projects board and the issue tracker on GitHub has played the biggest part in organizing and planning the development of the application. These tools have allowed me to easily visualize and organize tasks. Most importantly, it allowed the prioritization of tasks to be clearly noted, for those that are more critical to the development of the program.

More critical bugs can be placed at the top of the Projects boards, allowing me to focus on fixing those before moving on to tasks of a lower priority, such as feature requests and GUI appearance.

The board is used in conjunction with the issue tracker on GitHub, where tickets can be created in order to track the bugs and tasks that needs to be addressed. These issues are arranged on the Projects board in order of their priority status.
Those issues are also addressed and cross-referenced in commit messages, allowing me to refer to the changes that were made to address the issue later on. Once a particular issue is addressed, they can also be marked as closed in the commit messages. Then, when a pull request has been made and the branches are merged, the corresponding issue is then automatically closed on GitHub, and its corresponding card on Projects would also be automatically moved to the “Done” section.

While the logistics of gathering user feedback has not been ideal, due to security risks, among other things. Utilizing the integration between the Projects board, issue tracker, and the local Git client has created a seamless workflow, allowing me to focus on building the application, and creating a better outcome. *Wow, this last bit sounds like an ad, lol…*

# Commits

https://github.com/mosguinz/java-netflix-roulette/commits

```
git log --since="last year" --pretty=medium > log.txt
```

```
commit 04c8b66405d24fd915e65922f516245cef639218
Date:   Mon Nov 18 21:26:47 2019 +1300

    Fix roll button not working when genre is invalid

     - Roll button freezes when genre selection is invalid (no generes
       selected)

     - Separate input validation logic for genres from setting query values

     - Add flag to verify if both ratings and genre selection values are
       valid

commit defe0d841cf06b3f93473d1263126f214837a3e3
Merge: 4724513 1b3e129
Date:   Tue Nov 12 11:57:12 2019 +1300

    Merge pull request #32 from mosguinz/javadoc-update

    Javadoc update

commit 1b3e1297ca7652fad2c7adcf9e8ce7c4d27ba02a
Date:   Tue Nov 12 11:42:24 2019 +1300
```

```
    Remove old javadoc folder

commit b82998a580277906018a1c304c0fc610feaaedb1
Date:   Tue Nov 12 11:40:45 2019 +1300

    Change javadoc directory to default

    Update .gitignore

commit 472451379b3f62a3808660bf15d95f390ae57208
Merge: 3cde982 f821c51
Date:   Mon Nov 11 20:15:42 2019 +1300

    Merge pull request #31 from mosguinz/javadoc-update

    Update javadoc directory

commit f821c514e8230381705cbeae756fcd7b2e05ea9a
Date:   Mon Nov 11 20:14:24 2019 +1300

    Update javadoc directory

    Update .gitignore

commit 3cde982ec6460760d4bd23faba724689e0c1d06a
Merge: 6b311f1 d1b7f52
Date:   Mon Nov 11 20:02:08 2019 +1300

    Merge pull request #30 from mosguinz/javadoc-update

    Javadoc update

commit d1b7f52cb6d62bdb495d4c903049e2cebb3c3b8b
Date:   Mon Nov 11 20:00:31 2019 +1300

    Add javadoc

    Update .gitignore to include javadoc file

commit de0827d6581239d31c640c38399fe41426fafe98
Date:   Mon Nov 11 19:48:53 2019 +1300
```

```
    Fix javadoc references to params

commit 6b92751a2efbebe4cd559fff4338fe457360f144
Date:   Mon Nov 11 18:30:55 2019 +1300

    Update javadoc for methods

    Add description for class variables

commit 6b311f166d56c6502ce1881ad29df2c0a3d823c8
Date:   Mon Nov 11 17:21:56 2019 +1300

    Organize imports

commit f063b2bed5ba42d2fa63e45b07bb9102f5ab5291
Merge: f9a1036 0516b08
Date:   Mon Nov 11 16:58:54 2019 +1300

    Merge pull request #29 from mosguinz/verify-rating

    Reject invalid rating value selection

commit 0516b081c933040c74a8caac62cfd58192056ec2
Date:   Mon Nov 11 02:01:01 2019 +1300

    Reject invalid rating value selection

     - Reject input if the maximum rating < minimum rating

     - Will not request for titles when rejected

commit f9a1036e9d5723573cec624bf4bb131b5687a7fb
Merge: 073fb74 f69bf0b
Date:   Mon Nov 11 01:29:31 2019 +1300

    Merge pull request #27 from mosguinz/convert-to-dialog

    Convert to selected title card to JDialog

commit f69bf0b341f006596b0f02ac3483231981a29a3f
Date:   Sun Nov 10 23:44:19 2019 +1300

    Fix reroll button not working
    Remove frame-disabling logic when title is displayed

     - Title is now displayed as a modal JDialog; no disabling needed

commit 279f2e3ba030bc4a21b1eefbd9ec309f2fd058ef
Date:   Sun Nov 10 19:52:21 2019 +1300

    Fix modal dialogs not appearing in center

     - Set dialogs location in cetner before setting to visible

     - Also fix issue where metrics for save file are not updated when
       the settings dialog is open for the first time

commit 5b64b58fc1f2c20dc3f818477cd0f1ca66ce8318
Date:   Sun Nov 10 19:48:49 2019 +1300

    Covert SelectedTitleGUI to JDialog

     - Was JFrame

    Make all dialog modal

commit 073fb74c22e855df5f9b19f97ef70989153923d6
Merge: e4733cc f2309aa
Date:   Sun Nov 10 19:40:22 2019 +1300

    Merge pull request #26 from mosguinz/settings

    Settings dialog

commit f2309aa3eb76b6c94fad0104d85a6c65b7221a64
Date:   Sun Nov 10 18:28:23 2019 +1300

    Fix responses not being deleted

     - Streams weren't being closed after loading responses

     - Wrapped Scanner in AutoClosable to automatically close stream once
       the JSON file is read

    Add logging message when library folder is being cleared
```

```
commit b876af0324f555f5e4941bf1e905a824cbb570c9
Date:   Sun Nov 10 18:21:56 2019 +1300

    Make buttons in settings dialog functional

    Wrapped metrics-updating methods into a single method

     - Also updated folder size metrics to display in MB instead of bytes;
       users reported that MB is preferred, citing familiarity with the
       units, opposed to the inflated units in bytes

    Add method to clear library folder in LocalLibrary

commit 21ffda9b68e5da91370748971d007a138196ccd7
Date:   Sun Nov 10 18:07:17 2019 +1300

    Add logic to count files and file size in library

    Add logic to display file counts in the settings dialog

commit d275cb2ca0969b2f9863899d9fe97e275713db82
Date:   Sun Nov 10 17:30:51 2019 +1300

    Create public getters for library path

     - Rename methods that create paths: get[...]Path -> create[...]Path

     - Methods that create the paths will remain private for the class
       to use upon instantiation

     - The getters for the file paths will return the File object of the
       class instance pointing to the file locations respectively

commit 9496d25ba3995faf8f19272519dd9d1b1bc1a472
Date:   Sun Nov 10 17:04:03 2019 +1300

    Resolve naming conflict for components

     - Fix component names to make more sense

     - Names were conflicting between the help dialog and settings dialog
    Fix buttons in the help dialog

commit ab29f53b024520741cb74f5f0b909a9d2b6c352d
Date:   Sun Nov 10 16:50:54 2019 +1300

    Add settings window

commit e4733cc76023c61aace4a141f756cd753ab28f02
Merge: 60d5c42 b1027f5
Date:   Sun Nov 10 01:43:34 2019 +1300

    Merge pull request #25 from mosguinz/rating-filter

    Rating filter

commit b1027f55138c06edd29d207386dc789e0109663e
Date:   Sun Nov 10 01:33:33 2019 +1300

    Rearrange GUI components order

commit 44ce49efe644f8691363f1635732dab9e5b9e15a
Date:   Sat Nov 9 22:49:54 2019 +1300

    Add logic to use selected values in q-string

     - Actually requests titles with the selected rating values

     - Also updated javadoc for the getTitlesQueryString method

commit b29f217a16071abcde98b97a46817230800d807c
Date:   Sat Nov 9 22:37:02 2019 +1300

    Add logic to get the selected value from sliders

    Add setters for query values to be used from user input

commit 0097f925ce74415a2cb601822bb5758343d2c244
Date:   Sat Nov 9 22:23:46 2019 +1300

    Increase rating accuracy

     - Users can now select ratings to the nearest tenth; this is also
       the nearest decimal unit that is returned from the AP
```

```
commit 060bc998a9c8e9de58a92cd26a7232567ccde240
Date:   Sat Nov 9 22:16:57 2019 +1300

    Add label to display rating slider value

commit 4f320a8cccf2dd2e8aa5787b6c4d57ea00e31fa8
Date:   Sat Nov 9 21:54:25 2019 +1300

    Add sliders for rating selection

commit 60d5c42849777da3cd44fd1cab70e26a054794df
Merge: 2552df2 a9bee3c
Date:   Sat Nov 9 21:14:39 2019 +1300

    Merge pull request #24 from mosguinz/gui-enhancements

    GUI enhancements

commit a9bee3cd60db2a2998e2528dc0adf7bc01de1d1e
Date:   Sat Nov 9 21:08:14 2019 +1300

    Font consistency

     - Label texts and non-interactive elements will use the Helvetica
       font, or sans-serif alternatives (such as Arial)

     - Interactive components, such as buttons, combo box, check boxes,
       etc. will use Tahoma

commit 8cb4d09df48d7478ef868f96d900cacfbd7c5375
Date:   Sat Nov 9 20:55:16 2019 +1300

    Add visual indication when fetching titles

     - Change label tenses to indicate that a process is happening
       "Roll" -> "Rolling...", etc...

     - Also disabled buttons when fetching titles to prevent effects of
       spamming

commit 04fc5e99fa90f094d9959211c34160753b832abc
Date:   Sat Nov 9 20:20:54 2019 +1300
```

```
    Add tooltip texts and window name for titles

commit a616f4ef4ecdd47f9dd7970964dd5ab3759cc835
Date:   Sat Nov 9 19:59:33 2019 +1300

    Make the reroll button functional

    Change the object type of the parent frame referenced in the selected
    title window

     - JFrame -> HomeGUI

     - So that the class can reference its method; and not just its
       JFrame attributes

commit 8ab030850ad9ba35945983aad0cdb2615c878b1c
Date:   Sat Nov 9 19:53:09 2019 +1300

    Add logic to disable home window

     - Set to disable the home window when a title is found

     - This is to prevent users from making changes to the fiters selected

     - Users must close the selected title window to re-enable the home
       window

     - This is so that the reroll button uses the same value, rather than
       trying using a filter that may be different

commit f69575b91cc9f716e98b994afb0a42b1b82aab84
Date:   Sat Nov 9 19:31:12 2019 +1300

    Add reroll button to selected title window

    Add logic to pass instance of home window to selected title window

     - To implement the function of the reroll button

       - The home window will be disabled until the selected title is
         closed, this is designed to mimick the modal JDialogs
    Wrap title fetching logic into a method: getNetflixTitle

    Refactor parts of HomeGUI to make this a little more readable

commit 58855bcc58413e5ce7efd583a3395dec5cad4936
Date:   Sat Nov 9 18:15:07 2019 +1300

    Add tooltip text for some buttons

commit 1284b3057bd73b4f0bccd07a4e04005ee2e52edf
Date:   Sat Nov 9 17:59:25 2019 +1300

    Remove File/Settings menu

     - Unused

commit 04a66e45299c84bca2c637bdad215d8bce7392df
Date:   Sat Nov 9 17:48:40 2019 +1300

    Add help menu

commit 97cc9041076764c37b27289d7d00f5d90de0efd8
Date:   Sat Nov 9 16:15:59 2019 +1300

    Make Windows as default UI look and feel

     - Was Nimbus

commit 2552df20e45fd884ade12fd139a57202ec46e2b2
Merge: 4a4b302 75ae2d2
Date:   Sat Nov 9 16:04:16 2019 +1300

    Merge pull request #23 from mosguinz/type-filter

    Add filter for title type selection

commit 75ae2d2b03fbdf1d5771bd10d57051bc8f52857d
Date:   Sat Nov 9 16:02:44 2019 +1300

    Update javadoc for getTitlesQueryString method

commit b42486360fb11a5eb49968e4eaee30745b877a3d
Date:   Sat Nov 9 15:48:32 2019 +1300
```

```
    Make type selection button functional

     - Add logic to insert the selected title type in the query string

commit 0bbda553974815e324fee8a3c57be49f7157d74b
Date:   Fri Nov 8 23:17:36 2019 +1300

    Fix radio button not returning action command

     - ActionCommand of the radio button does not appear to work properly
       unless set manually via the setActionCommand method

    Fix javadoc for query setters

commit 183211cc90a062c140c8e2e429c83513ff542b53
Date:   Fri Nov 8 21:45:20 2019 +1300

    Add logic to get selected value for title type

    Wrapped input getters from GUI into one method: setQueryValues

    Renamed references of title types to be more verbose: type -> titleType

     - This is to prevent confusion when used in conjunction while
       referring to queries

commit c367a5bb99a7ee417b7dbae1079f10d64678f105
Date:   Fri Nov 8 21:12:36 2019 +1300

    Add radio buttons for title type selection

    Rename and reorder GUI components to be meaningful

commit 4a4b3020a07cca9ba0e1d55a00f21347a65656eb
Date:   Fri Nov 8 20:22:00 2019 +1300

    Remove unused dependency

commit 2d15a0eb3159f9f0fab7512c8eca6e935b36f371
Merge: d691af2 a3476e7
Date:   Fri Nov 8 18:07:11 2019 +1300
```

```
    Merge pull request #22 from mosguinz/query-content

    Query content

commit a3476e7ffb2bc39ceb9e2f8c86163b4a2aa6e7f9
Date:   Fri Nov 8 17:59:54 2019 +1300

    Add logic to look check query string in responses

     - Also add logic to write query string into saved responses for titles

     - Now checks if a saved response for titles matches the given query
       string; so that returned titles from saved responses actually
       matches the parameters that are provided

    Fix verifySavedResponse signature

     - queryType no longer needed to verify responses

    This commit address and fixes #20

commit 1e33e20f98c07d89ff266ee989c090cc007d5fcf
Date:   Thu Nov 7 19:14:59 2019 +1300

    Update return type for loadSavedResponse

     - Also updated docstring

     - Now returns JSONObject to represent the response file (and not just
       its content)

     - Removed validation logic

       - To be added back to getSavedResponse instead; responses should only
         need to be validated when it is actually the desired one

commit b337a8ad8c2cfa879643cbff5128d0d88643dd7d
Date:   Thu Nov 7 18:46:09 2019 +1300

    Rename loadSavedResponse -> getSavedResponse

     - New method with the name loadSavedResponse
     - "get" and "load" logic will now be separated

       - loadSavedResponse will load the specified JSON file into memory
         and parse it as a JSONObject

       - getSavedResponse will return the response content of the JSON file
         as a JSONArray object

         - This method will call loadSavedResponse to obtain the JSONObject
           of the file to check whether the file contains the desired
           content; this is used for titles requests

     - These operation have been separated to prevent a JSON file from
       being loaded and parsed more than once for a single lookup

    Fix signatures for getResponseFilename method

commit d4243040d4420a6eeafd33ff248b938e4038fd02
Date:   Thu Nov 7 17:23:01 2019 +1300

    Change file-naming logic for saving responses

     - No longer appends query string for the response to the file name

     - Will simply add a number to the end of the filename to prevent
       overwriting existing ones for titles responses

    Fix javadoc for methods referencing query types

commit 6323a5e3f7236bee824ba8a9f46228d12f978cc2
Date:   Tue Nov 5 11:59:33 2019 +1300

    Restructure file reference system

     - Now use File object to reference file paths instead of Strings

commit d691af28980c55473b93502b0ac949be3f958633
Merge: 1164028 2b209c1
Date:   Tue Nov 5 11:22:01 2019 +1300

    Merge pull request #21 from mosguinz/genre-filter

    Add filters for genre selection
```

```
commit 2b209c172c38ef8ca6e21c55f334653aeaed46a4
Date:   Tue Nov 5 00:29:50 2019 +1300

    Move validation logic for genres

     - For cases where no genres are selected; this is moved back to the
       GUI component to handle instead of the library

    Add some annotations for clarifications

commit 8f232756e9d10979c0e6f1b96c9bd026038698d9
Date:   Tue Nov 5 00:11:36 2019 +1300

    Make genre checkbox functional

     - Add logic to lookup respective genre IDs for use to request
       APIs for titles associated with the selected genres

    Add getter for query genres

    Remove unused default values for query region and genres

     - Previously fallbacks for failed requests/lookup; this is now handled
       by response validation

    Add javadoc for getRegionID

commit eab9c41acc2b4d4d8edf49b5b362cf7e58176054
Date:   Mon Nov 4 19:22:44 2019 +1300

    Add select all/clear all button for genres

     - For ease in selecting and unselecting genres

    All genres are selected by default

    Rename checkbox names for avoid ambiguity

commit 7c1e13e9e77582571ebb5769e9531ee4bfdb1535
Date:   Mon Nov 4 00:33:00 2019 +1300

    Add checkboxes for genre selection
```

```
commit a39454cb154c4c3fa6877dd74c429af26cfc606d
Date:   Sun Nov 3 19:00:35 2019 +1300

    Change HomeGUI layout to use GridBayLayout

commit 1164028b63ee393bef4807df70df521899c25b71
Merge: 3bec193 ad06a50
Date:   Sun Nov 3 18:48:01 2019 +1300

    Merge pull request #19 from mosguinz/disclaimer

    Add "About" menu to display disclaimer

commit ad06a500e9f0fd88a6e50b5093288b9906c97388
Date:   Sun Nov 3 18:46:09 2019 +1300

    Add "About" menu to display disclaimer

    Add menu bar for other options

     - Other features to be implemented later on

commit 3bec193e0db4554a00f0562a670f9f2608bd8ec1
Date:   Sat Nov 2 21:26:37 2019 +1300

    Fix NCR hex codes not being uncoded

    Add JSoup as a dependency to address this issue

    Close #18
    See also #8

commit 22321aec2a273adf0f90e7e7c7ae42100e9abd34
Merge: b37f3f3 6b6e2ed
Date:   Sat Nov 2 20:29:09 2019 +1300

    Merge pull request #17 from mosguinz/mvp

    MVP application completed

commit 6b6e2ed52d24b01c84c3bbb6851677a3c7bd32a7
Date:   Sat Nov 2 20:22:03 2019 +1300
```

```
    Fix text wrapping for title names being displayed

     - Title names will be resized (and truncated, if necessary) to fit the
       JLabel

     - Closes #15

    Move visibility logic for the SelectedTitleGUI to its own class

     - GUI will be displayed if its title info is updated

    Change SelectedTitleGUI layout to use GridBayLayout

     - Was AbsoluteLayout, this is to enable the layout of the GUI to be
       more consistent with more components to be added in the future

commit ec103aea9ae47b7e6470e2c131b93c6c0282d61e
Date:   Thu Oct 31 15:23:44 2019 +1300

    Add call to check and make local library directory

    Update javadoc on Throwables

commit 1fd684ea4b58d4c0ad43c70de3e129bc8981ad1d
Date:   Thu Oct 31 01:05:39 2019 +1300

    Add dialogue boxes to display error message

     - Also handled cases where query returns no titles (see #16)

     - Handle case where query result may be in an unexpected format

    This commit closes #16

commit 721ac97114efd98ba6283e2f24681a21605b0a9b
Date:   Tue Oct 29 23:54:34 2019 +1300

    Add logic to name responses filename accordingly

     - Responses are now named using their response type

       - And for titles, their filename is also appended with the query
```

```
         string used to send request to the uNoGS API

       - Saved responses can now be properly reused instead of sending a
         new request, whenever the same parameters are passed

    This commit closes #10

commit 2ceb44b0f3edbbd3494a60d474f2bed5b508fce8
Date:   Tue Oct 29 22:54:17 2019 +1300

    Make region selection functional

     - Actually looks up the region ID and requests API for titles that are
       available in the selected region

     - See #11

    Fix ID value for "All regions"

     - ID is "all" not "All"

     - Also removed key-pair for this value to be written to file or used

       - This value does not need to be saved to file

       - Also to maintain consistency with how genres are saved; values for
         an "all genres" selection is not saved

     - Made sure "All regions" option is always displayed at top of list

commit c1bcc204cab8133ca1133ed3e7fb1af195596588
Date:   Sun Oct 27 23:14:21 2019 +1300

    Add region selection

    Closes #11

commit 4d0b344bce5525025806d8dfd39b095a69783998
Date:   Sun Oct 27 18:08:46 2019 +1300

    Remove "COUNT" key in saved responses

     - Now redundant
```

```
commit 9fa807636a5c4546167922c460f98fec26c4ece9
Date:   Sun Oct 27 18:06:07 2019 +1300

    Change the data model for list of genres

        - Instead of being a JSONArray of single-entry JSONObject(s)
          containing each genre with its IDs, it is now a JSONArray with a
          single entry of a JSONObject with multiple entries of the genres
          with its IDs as an array

         - The JSONObject is still wrapped under JSONArray to preserve
           consistency for validation

        See #14

commit 0a8c7217ebffde0e99448e93a7c1d6f8ab10cf8c
Date:   Sun Oct 27 17:46:01 2019 +1300

    Change the data model for list of available region

     - Instead of being a JSONArray of single-entry JSONObject(s)
       containing each region, it is now a JSONArray with a single entry
       of a JSONObject with multiple entries of the available regions and
       its IDs

      - The JSONObject is still wrapped under JSONArray to preserve
        consistency for validation

     See #14

commit 8cad956eef6499c78753d291d94d9ebebbe2ab09
Merge: 53a5a5c a87480c
Date:   Sun Oct 27 16:39:59 2019 +1300

    Merge pull request #13 from mosguinz/javadoc

    Create javadoc for classes

commit a87480c7187fcfe3d52e96191d9a8c2dcfc6c5d7
Date:   Sun Oct 27 16:37:21 2019 +1300

    Add javadoc for class constants
```

```
commit 3b183bd08dcdeecc708abff721e3711319e8673a
Date:   Sun Oct 27 15:38:52 2019 +1300

    Update javadoc

     - for LocalLibrary and LoggingUtil

     - Also updated links to constant, methods, and classes in the javadoc
       for all other classes... even though some of them don't even have
       any documentation... ahem... org.JSON...

commit 1df2d2f9dcea343563a279fcdfa3034c2670bfa4
Date:   Sun Oct 27 13:54:02 2019 +1300

    Update javadoc

     - for HomeGUI, SelectedTitleGUI, and NetflixLibrary

commit 2637be07defbfb559ccdf7c1699f6ce8b733071a
Date:   Sat Oct 26 21:56:02 2019 +1300

    Update license

commit 59d23d87729c9876bd47ea509d0327e481b3c22c
Date:   Sat Oct 26 21:54:39 2019 +1300

    Rename LICENSE to LICENSE.md

commit a5357f1a8e17a40c06d738f5e1c2bfd61cd234c7
Date:   Sat Oct 26 21:53:40 2019 +1300

    Create LICENSE

commit 53a5a5c65e60713e117356028d40c099ccf2a2dd
Date:   Sat Oct 26 21:31:21 2019 +1300

    Swapped region key-pair value

commit 56e6eead439f7f034c4870e01667f35dda1b7ebc
Date:   Sat Oct 26 21:27:06 2019 +1300

    Refactor SelectedTitleGUI
```

```
     - No longer instantiates the class every time this window is
       displayed... not sure what I was thinking beforehand

      - This has massively improved wait time between "Roll" button press
        and displaying the selected title -- almost instantly using saved
        responses

     - Also fixes logger instance being created everytime a title was
       displayed, bombarding the console

commit 3d728e3a77f7d8f327acd9351f77a959f2b0843e
Date:   Sat Oct 26 20:32:23 2019 +1300

    Refactor response validation logic

     - Separate VALIDATION and EXTRACTION logic

     - Because saved responses only needs to be validated, but does not
       need to be extracted as it will already be saved in an appropriate
       format

     - VALIDATION involves checking if the JSON response contains the key
       "ITEMS"; all valid responses from uNoGS will contain that key, and
       any valid saved copies will also contain them for consistency sake

     - EXTRACTION involves extracting useful data from the response sent
       from uNoGS, into a format that makes more sense for this
       application's use case

    Refactor data fetching logic

     - All requests that COULD'VE been sent to the uNoGS server are checked
       for its cached responses on the local machine first

     - Only sends a request to uNoGS if the saved response is not valid or
       does not exists

    Add constructor for LocalLibrary

     - no wonder why logs weren't coming out of this class...

commit f5d5add80b689b9920fe85e75ada4771b5ad7065
```

```
Date:   Sat Oct 26 17:14:50 2019 +1300

    Add GUI for Netflix region selection

    Add logic to save responses with its queryType name

    Fix response verification for saved files

     - Ensure that correct verification method is used for its respective
       response type

     - Add logic to check the age of the response

    Update .gitignore

commit b6d67289f3c23d97e82a062c8807239600140d0e
Date:   Wed Oct 23 16:19:54 2019 +1300

    Add logic to extract available Netfix regions

    Rename validation and extraction method names to avoid ambiguity

     - To clarify between "fetch[Titles/Regions/Genres]"

     - getSupercategoryGenres -> extractSupercategoryGenres

     - getAvailableRegions -> extractAvailableRegions

commit 0ed41083da66a58dc5ad3cf0957b7e9f7e88847d
Date:   Tue Oct 22 22:16:51 2019 +1300

    Add and fix logging consistency

commit 125b3f5cac6ea041bb9e827df66ddaeef8691359
Date:   Tue Oct 22 11:15:51 2019 +1300

    Add date info to saved responses from uNoGS API

     - This is in the JSON file, under the key "DATE"

     - Date is in the ISO ISO-8601 format uuuu-MM-dd

    Add logic to verify the age of the response
```

```
     - To verify that the response is fresh enough to be used

     - Maximum days is set as MAX_RESPONSE_AGE constant

    Add logic to look for saved resoinses with the same request parameters

commit de480e2e3efb3a02bebc764750b8e8672d794a83
Date:   Sun Oct 20 16:37:56 2019 +1300

    Add logic to get genre list from library API

    Refactor file-writing logic

     - Moved calls for writing responses in sendQuery method;
       responses should only be saved when sending a query is required --
       becomes redundant elsewhere

    Add endpoint URL for genres

    Add logic to extract list of supercategory genres from API

    Rename saveTitles -> saveResponse

     - Method will be used for both writing list of titles and genres from
       response

    Add logic to create file names for writing responses

    Update .gitignore

     - Exclude export log files for commit history

commit f51a4241e665ab7cc1d780d6f93b99ddd8653fc1
Date:   Fri Oct 11 17:00:53 2019 +1300

    Rename LocalLibraryReadWriter -> LocalLibrary

commit 76ecd9e2cb8174e44cccf9f32ca42a10e0c3455e
Date:   Thu Oct 10 03:27:49 2019 +1300

    Separate logic to verify response content for library API
    Also remove some testing code in GUIs

commit c50659c4b5ace8aff7d7766c23ecd4635ef3d4df
Date:   Wed Oct 9 00:49:13 2019 +1300

    Refactor logging facilities

     - Methods defined specifically for reporting thrown exceptions

    Fix response validation

     - Made sure to only attempt writing savefiles if the given responses
       are valid

     - TODO: requests with invalid API keys takes ages (like a whole
       minute) to respond for some reason... find a way to validate key
       and mitigate wait times when API keys can't be found or provided
       ones aren't valid

commit 5948796f769e9e18e4a72f16fd190811b4acd354
Date:   Tue Oct 8 17:37:52 2019 +1300

    Set up logging

commit 8747072f477936466c2eb60b9ddf340bcb248051
Date:   Mon Oct 7 18:23:01 2019 +1300

    Refactor classes

commit 8f4b73579069c42cfcda0ecb96a626ae189e829b
Date:   Sun Oct 6 17:38:17 2019 +1300

    Add method to write JSON response

commit 473ae082ea0233fe08e12d7c6ba4dcfb33bb7401
Date:   Thu Oct 3 19:30:37 2019 +1300

    Add LocalLibraryReadWriter

     - This library will be responsible for reading and writing JSON
       responses from the API

     - Will be used to mitigate wait times from API server and reduce
```

```
      number of calls needed to API server

commit dab311eacfed8cbf94f48c20a18f14f883c22571
Date:   Sat Sep 21 18:18:16 2019 +1200

    Remove NetflixTitle class

     - NetflixTitle is no longer used to represent a title from JSON
       responses

       - Instantiation of the class with multiple items takes too long

       - Causes significant delay, according to user feedback

     - Instead, data are parsed directly from the JSON response to display
       on to GUIs

    Update pom.xml and .gitignore files

     - Added dependecies and set up .gitignore in prepartion for added
       test classes

commit 90e51d83b500a54962c1ebc1642595f88ca71074
Date:   Fri Sep 20 11:47:51 2019 +1200

    Temporary fix for failed requests

     - Mainly to handle requests made in coperate/school networks that
       fails SSL handshake

     - Temp fix implemented as a popup to notify user of error

commit c43f421d25b4e26fe376f9a7b79494075e311256
Date:   Sun Jul 28 19:17:19 2019 +1200

    Update RapidAPI proxy URL

     - Mashape legacy URL is deprecated

commit 3428475b6ca77bb65cdbb526a4e19a24c8853235
Date:   Tue Jul 23 19:26:50 2019 +1200

    Specify main class name in pom.xml
```

```
     - Fix typo for Netbeans rep(o) name

commit 021a55aa00d49ffd42d03c48ac31a0476577b244
Date:   Mon Jul 22 22:20:57 2019 +1200

    Update pom.xml to include AbsoluteLayout

commit 925ec4862449c37d4e3393ccb04d0e1eb8888ee0
Date:   Sun Jul 21 23:25:05 2019 +1200

    Reject empty API keys and null returns

     - Handle cases where user presses "Cancel" or closes dialogue when
       requested for the Netflix library API key

commit b331a8912c2d91809ea0b30f98d954e86e56d034
Date:   Sun Jul 21 23:02:12 2019 +1200

    Refactor API key request and validation

     - Dialogue box to request Netflix library API key are now in a method,
       so that different message can be used

     - Dialogue now also appears when the response is invalid -- i.e.,
       cases where user may input an invalid key and a request has
       been made

     - Now rejects any key with whitespaces

commit 2146859ae932607d9046ea7041e81b3e4309e83e
Date:   Sun Jul 21 22:02:25 2019 +1200

    Add dialogue box to request API key when not found

    Also deleted a strange pair of brackets that made it in somehow...

commit 3dcf403ee38f26807f56837d8c1a0b0dce9d8023
Date:   Sun Jul 21 20:19:15 2019 +1200

    Fix hanging interpunct for series

     - Selected title subtext now says if it is a movie or a series
     - Series will not have a runtime, as it is not applicable and is
       returned as blank from the API

    Actually resolves #7 this time

commit a984cbaeecd42c00f6a817ff735520dc72cb5849
Date:   Sun Jul 21 20:14:05 2019 +1200

    Add buttons to return to Main and open in Netflix

commit cd951f589f6ed11c8f8179beee38758aeb6bca7a
Date:   Sun Jul 21 19:29:44 2019 +1200

    Unescape HTML code points before displaying

     - Enclosed the text (title name and synopsis) with HTML tags

     - This also fixed text wrapping for the synopsis

    Remove a strange dot that appears to the right of the poster image,
    as seen in #6, #7, and #8

     - Probably caused by the placeholder text, now resets (gets removed)
       before placing image

    Fix typo on title synopsis placeholder

    This closes #6, closes #7, and closes #8

commit d404d8c19822c26c08979315f36e6811ac1b1a1f
Date:   Sun Jul 21 17:15:26 2019 +1200

    Parse all info about selected title into GUI

     - Closes #4

commit df6733f2ca264db297a307656cd4fc2f3c7f2983
Date:   Sun Jul 21 02:57:56 2019 +1200

    Rename variable name to be more concise

    Refactor instantiation code for selected title
```

```
commit 722a9528b0032bf94a12a5d7ead7041872e931fe
Date:   Sun Jul 21 02:21:06 2019 +1200

    Parse image URL to display on GUI upon selection

commit 22640f165413c6f83805ed9c0c54483902a38a19
Date:   Sat Jul 20 22:43:38 2019 +1200

    Renamed class to reduce ambiguity

     - GetNetflixTitles -> NetflixLibrary

     - Also renamed methods to align with this change and make
       them more concise

commit 3aeea8c6c8bde988d975c43136317d837198574a
Date:   Sat Jul 20 22:22:32 2019 +1200

    Separate fetching and randomizing logic of titles

     - All objects in the JSONArray of the response body are converted into
       a NetflixTitle object

     - Each NetflixTitle object are appended to an ArrayList, which is
       returned at the end of the method

     - New function, selectRandomTitle, now handles the randomization logic

     - Add GUI window for displaying selected title as a placeholder -- for
       now

commit 8b8f4fa0e65a00ffd031456a6d23cf68c90e8179
Date:   Mon Jul 8 14:41:43 2019 +1200

    Basic logic of roulette complete

    - Add logic to select random titles from response body
    - Create instance of NetflixTitle to represent selected title

commit 1de33b5466f1b66cda8a5c5ec613084de76f6241
Date:   Sun Jun 30 23:40:51 2019 +1200
```

```
    Add method to send request to uNoGS API

commit 71695b464fd308d110506cefd6d7eec3c492f0aa
Date:   Wed Jun 26 23:27:05 2019 +1200

    Migrate to Maven

    - This is to better facitiate unit testing and dependencies handling

    - Update pom.xml to include unirest as a dependency

commit 387213345a8876b205f43211b86ed5dd4114107e
Date:   Sun Jun 23 22:59:39 2019 +1200

    Add class to represent Netflix title

commit b37f3f3a6594906adf01c511e7b0b16ff125f2d9
Date:   Sun Jun 23 19:45:37 2019 +1200

    Update .gitignore

commit 568bd9946563ccf7b14ac55a9a3b16d70243667b
Date:   Sun Jun 23 19:42:08 2019 +1200

    Update .gitignore

commit 315dbf9234814c37f04e3bb3db3d826ee799cb04
Date:   Sun Jun 23 19:30:33 2019 +1200

    Remove even more untracked files

commit 3c39e9b37d19daebcc0eb2829b1b23db54fd6269
Date:   Sun Jun 23 19:29:35 2019 +1200

    Update .gitignore

commit e1f2bc08ba2e20a17565cd7e9d285f1dea7224c2
Date:   Sun Jun 23 19:23:08 2019 +1200

    Remove files in .gitignore

commit bab4df76dce8d6a33ad1856ee9fe59493a450afc
Date:   Sun Jun 23 19:22:14 2019 +1200
```

```
    Update .gitignore

commit b6617e3717071d074ff0c5c6ddc46b1f2a24f9b8
Date:   Sun Jun 23 19:11:51 2019 +1200

    Remove files in .gitignore

commit 7dbe1cff2ec37beca6d0704205d6ed1bb778c53b
Date:   Sun Jun 23 18:34:05 2019 +1200

    Add .gitignore

commit d46ff49fdf30f6cadfa8b25742f9fa39546126a7
Date:   Sun Jun 23 18:31:24 2019 +1200

    Initial commit
```