FINAL PROJECT

Giuseppe Gómez & Edgar Torres

Computer and Systems Engineering Department, Universidad del Norte

NRC 3493: Object-Oriented Programming

José D. Posada Aguilar

June 1, 2022

CONTENT

I. PROBLEM DEFINITION

Many anime and manga lovers like to organize and keep a record of the mangas and animes they are watching or have watched. In the same way, planning is a factor this particular group takes into account for organizing themselves to watch or read this kind of content in the future, well be because of seasonal content, trends, and recommendations, among others. In addition, this community loves giving reviews of the content they consume and likes to share those perspectives with others.

Taking into account the previous, there is a clear necessity for the existence of a simple and functional way that facilitates them to organize and keep a record of their activities, as well as share opinions of the content they watch/watched to share them with others.

It is proposed to implement a complete, friendly, simple, and easy-to-use agenda/tracker application that helps manga and anime lovers track and organize the different kinds of content they consume, along with rating particular contents saved in the agenda/tracker.

II. SIMILAR SOLUTIONS IN THE MARKET

Kitsu: It provides its users a tracking feature to show other users what shows and manga they’ve seen.

Anime-Planet: It keeps track of what the users have seen in their anime and manga list. It allows them to write blogs and make friends, like a social network.

AniDB: It is a non-profit anime database that is open freely to the public. This means that all users can add or edit information.

MyAnimeList: It is an anime and manga social networking and social cataloging application website. It provides its users with a list-like system to organize and score anime and manga.

AniKore: It is a website that measures anime ratings and anime evaluation ranking specifically in Japan

III. SCOPES

Target audience: People who show interest in keeping track of the animes and mangas they watch/read.

Technologic platforms:

- **Devices:** Only computers
- **Operating systems:** Windows and Linux (Arch and Ubuntu distributions)
- **Programming languages:** Python, Javascript
- **markup languages:** CSS and HTML
- **Libraries:** certifi , charset-normalizer , idna, proxy-tools, pycairo, PyGObject, pywebview, requests, urllib3, json, os, shutil, datetime, sqlite3

Functionalities

Functional Requirements:

- **Search for a specific Title:** For starting an agenda or tracker, users want to look and see if a specific anime, manga, or movie is available.
- **Keep a record of the user's activities:** Users would add, organize and have a record of the current animes, movies, or mangas they have or are watching/reading.
- **Similar content recommendations:** Users might enable the option to get recommendations on similar content based on tags from the finished series or custom categories they are looking for.

- **Rating the added shows:** Once added the series, the user might want to rate (by stars) their shows saved on their agenda. This would affect the recommendation system. They might even add an opinion of the show in the corresponding record.
- **Export/Import the user´s agenda to share with others:** Users might like to share their agenda with other people or keep it saved in a file. There is nothing better for making this than being able to export or import an agenda, as it includes the user´s shows, ratings, etc.
- **Used technologies:** The application source code is written in Python. It uses the AniList GraphQL API to access the anime and manga entries.

Non-Functional requirements:

- **Legal requirements:** Using appropriately and fulfilling the condition´s uses of the selected API.
- **Navigation:** The application will be displayed in a very simple and user-friendly graphic interface to ensure a better user experience.
- **Platforms:** The application will only be able to work on desktops.
- **Performance:** A fast connection between the application and AniList API.
- **Connectivity:** An internet connection will be needed when adding a manga, anime, or movie to the agenda. The same goes for recommendations.

IV. SOLUTION PRESENTATION

POSSIBLE OPTIONS

Backend development:

- A Python-based backend
- A Java-based backend
- A JavaScript backend

Frontend development:

- A python-based frontend using Tkinter library
- A python + Flask solution based on Heroku.
- A python application with Electron framework.
- A python application with pywebview framework.

SOLUTION CHOOSE CRITERIA

- The graphic interface must be complete, dynamic, and easy to use.
- The backend solution must fulfill the object-oriented paradigm.
- The intercommunication between the backend and the frontend must be easy to handle.

SOLUTION REVIEWS

Taking into account the criteria:

- Tkinter would not be an option for developing the frontend, as it is more likely to offer a static result.
- Javascript would not be an option to develop the backend, as it does not support multiple inheritances.

- Java is a language that fulfills the object-oriented programming requirements, but taking into account that does not meet enough requirements to handle SQLite databases by itself, it would not be an option.
- A python + Flask solution based on Heroku would not be an option as it is necessary to keep a running server.
- A python application with Electron framework would not be an option because the NodeJS modules required by Electron are so heavy for this project.

CHOSEN SOLUTION

Taking into account the solution reviews, the chosen solution is to develop a python-based backend combined with a pywebview framework frontend. Python is a programming language that fulfills all the established criteria. pywebview is a framework that allows us to create GUI for any Python program with HTML, CSS and JS.

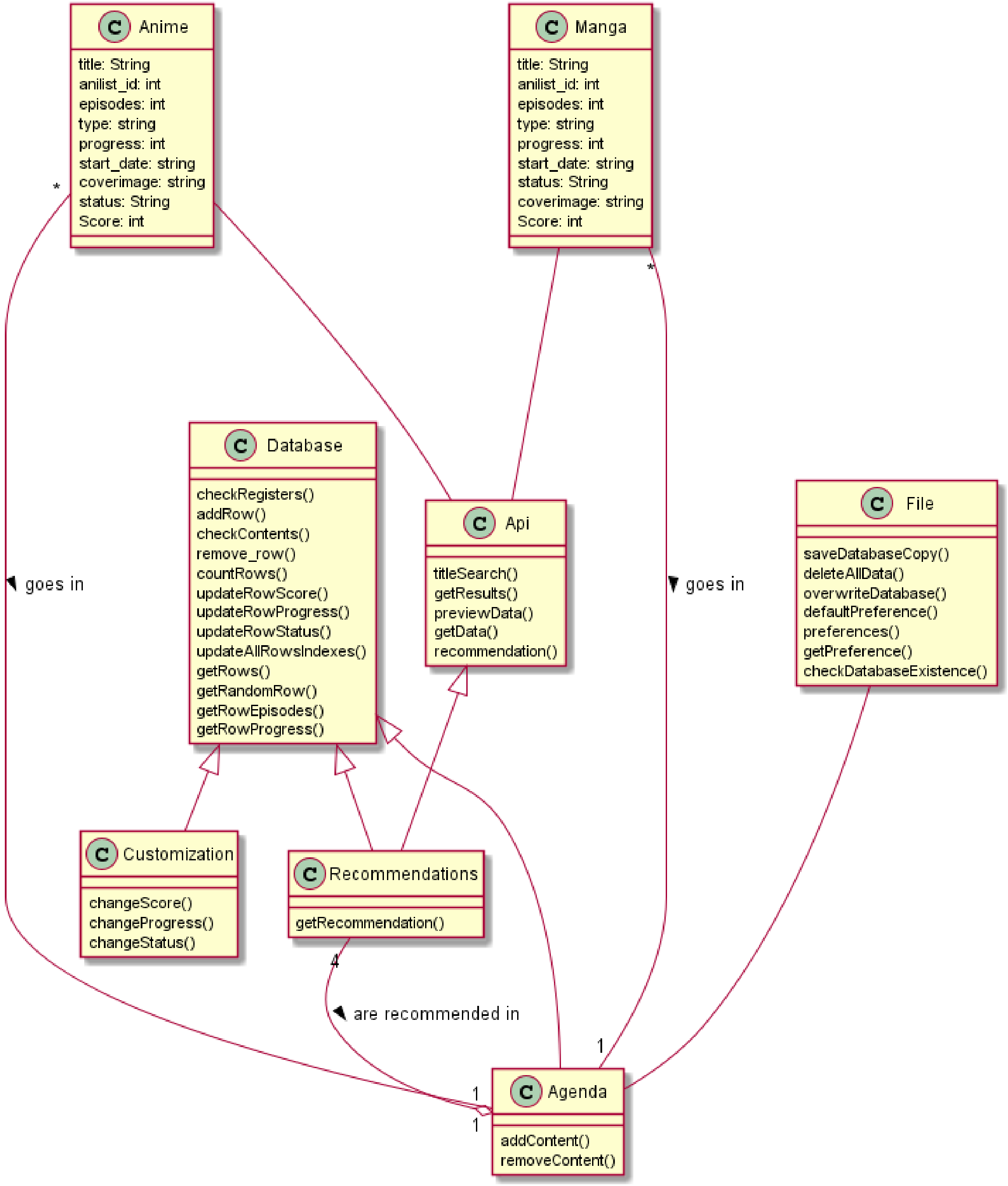
V. ROLES

Frontend (Giuseppe Gomez): Was in charge of building the graphic interface and creating the API to establish the connection between the Javascript and python programs.

Backend (Edgar Torres): Was in charge of making the python program and creating the functions and methods for the Javascript API class.

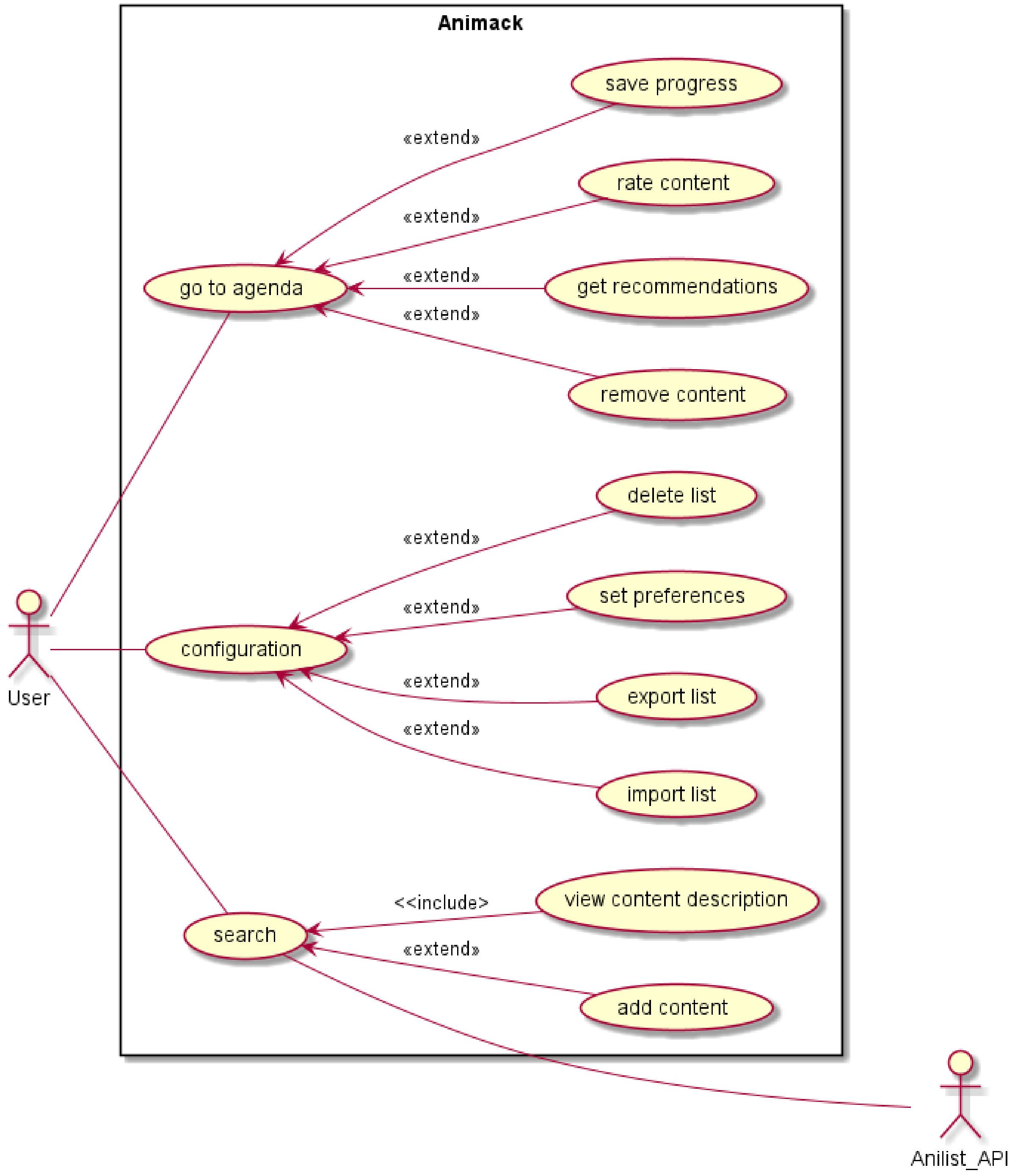
VI. SOLUTION DESIGN

Class Diagram



Our main program consists of 8 classes, being the API and Database the most important ones. The API class is in charge of giving the user the content they are looking for to start working on their agenda. The database class plays an important role in storing the user´s activities, and it is the main core of the customization class, as this last class methods can only work with what is stored in the database.

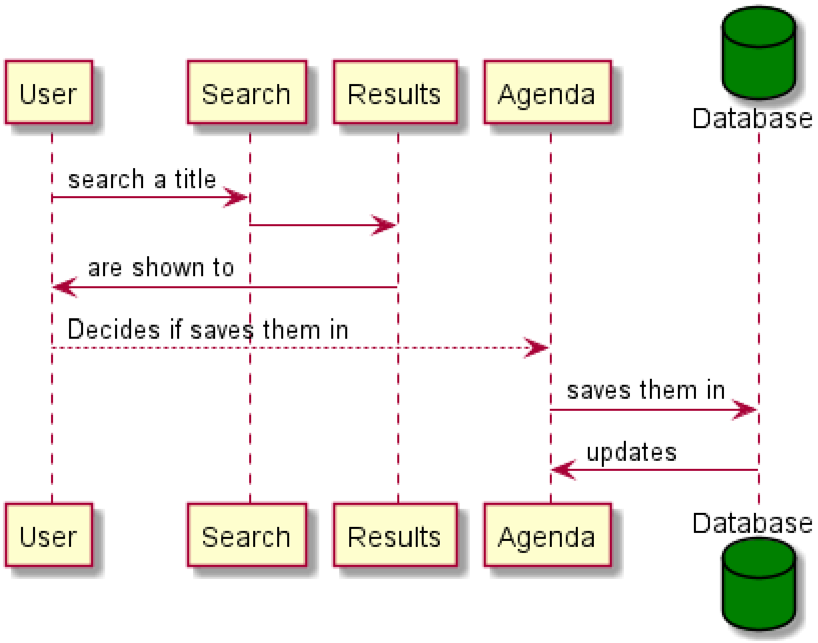
USE CASE



In this diagram is visible the interactions the user can have with the Animack App. There are three main cases, which are: go to agenda, configuration, and search. Each one mostly extends optional interactions. In the search interaction can be found an actor called "Anilist_API", this actor is in charge of establishing the connection between the Animack app and the Anilist database to give the user results they might be looking for.

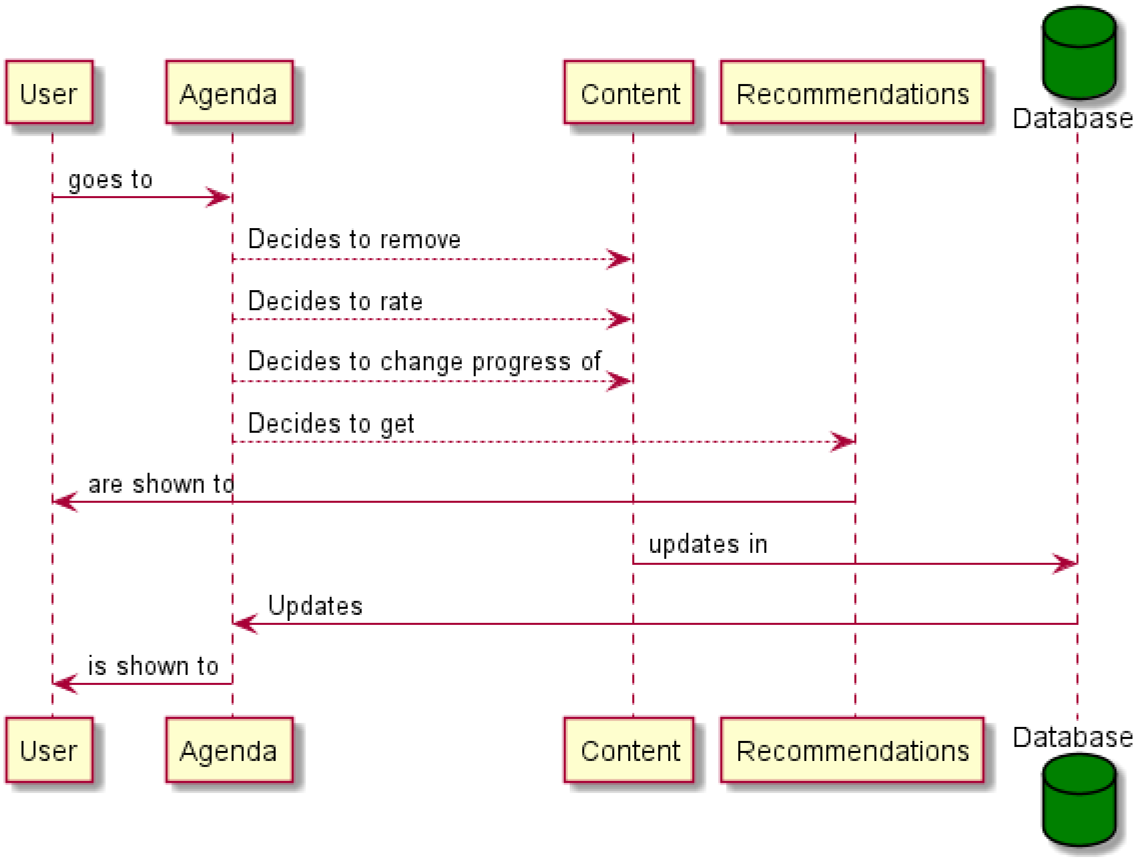
SEQUENCES DIAGRAM

For search:



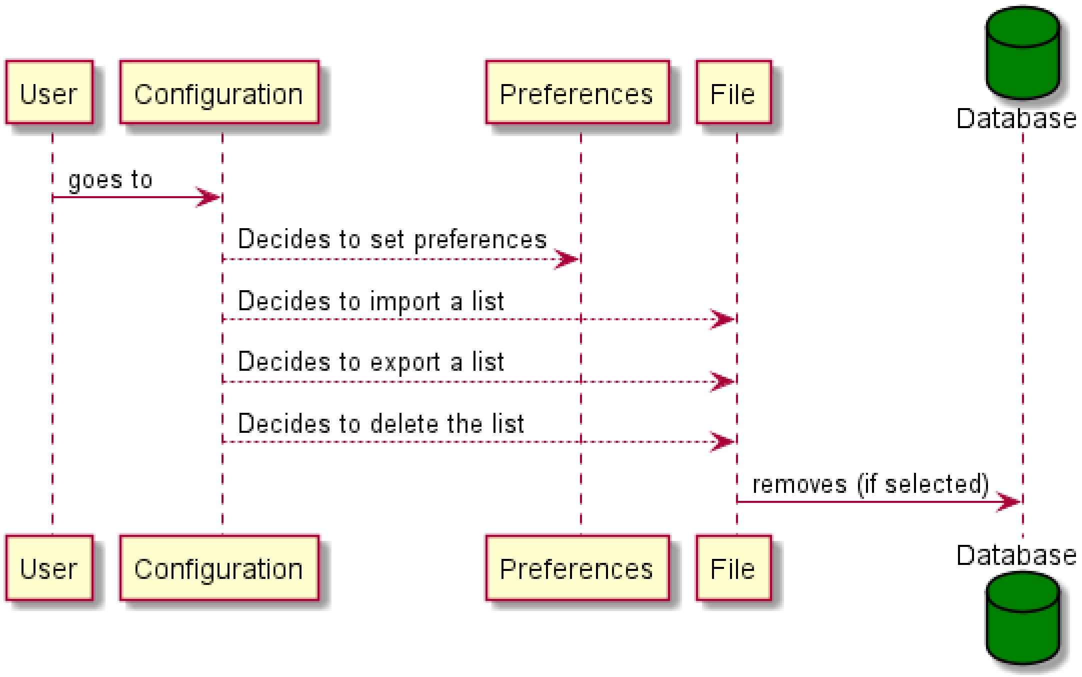
the user search for a title and the results are shown. they might decide if save a result in their agenda, which is, in reality, being saved in a database.

For the Agenda:



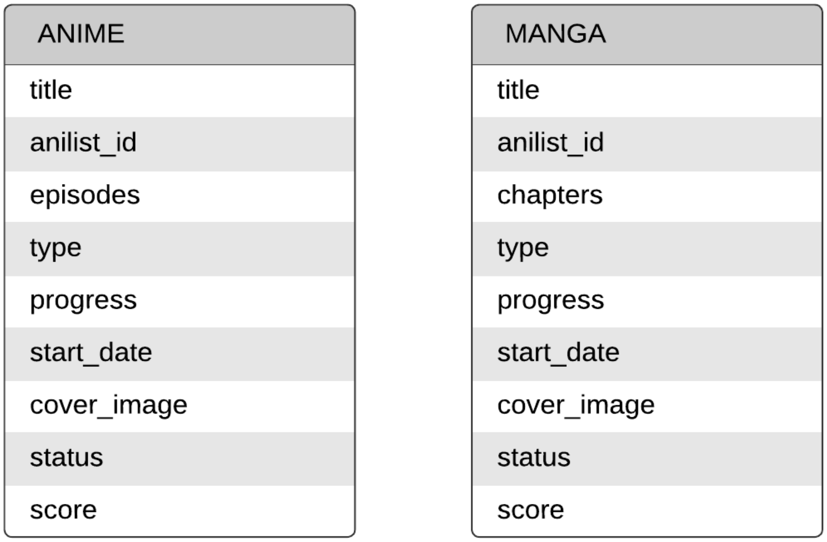
In their agenda, the user can decide to remove, rate, or change the progress of several contents, as well as get recommendations. Each change the user does to content updates its data in the database.

For configuration:



In the configuration window, the user can set preferences for filtering search results. He might also choose to import, export, or deletes a list. Deleting a list will remove the current list database file.

Database relational diagram



ANIME
title
anilist_id
episodes
type
progress
start_date
cover_image
status
score

MANGA
title
anilist_id
chapters
type
progress
start_date
cover_image
status
score

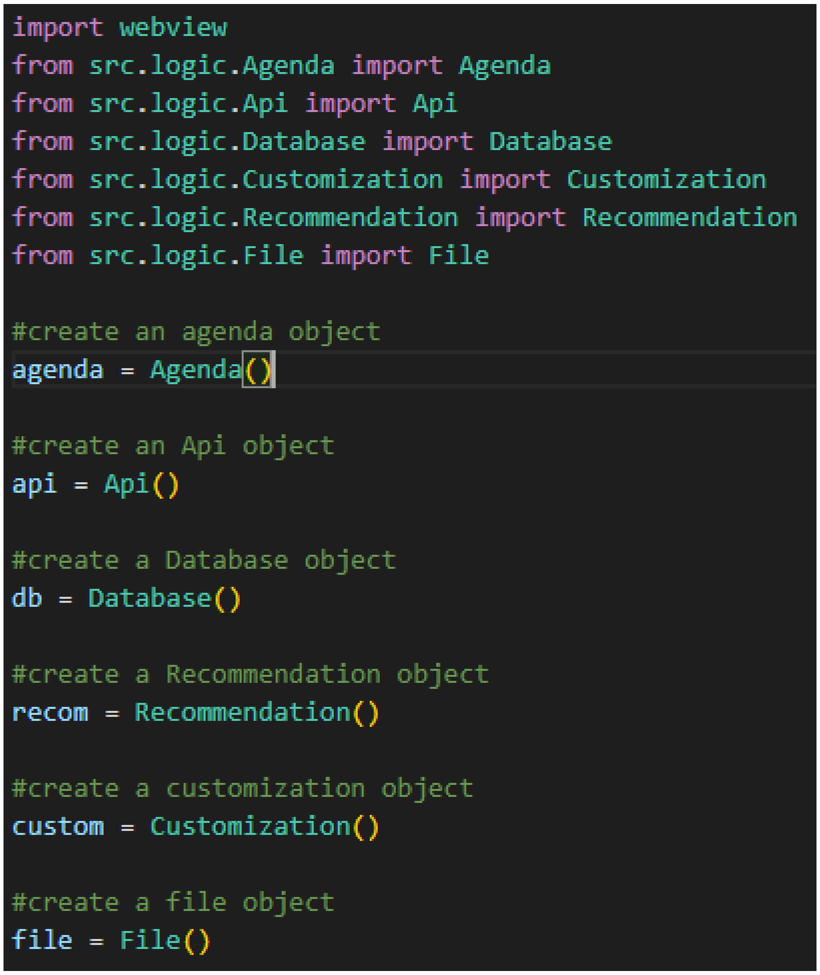
The database is composed of 2 tables that do not share any connection. Each table represents a list in the agenda section.

VII. PROJECT DOCUMENTATION

CODE DOCUMENTATION:

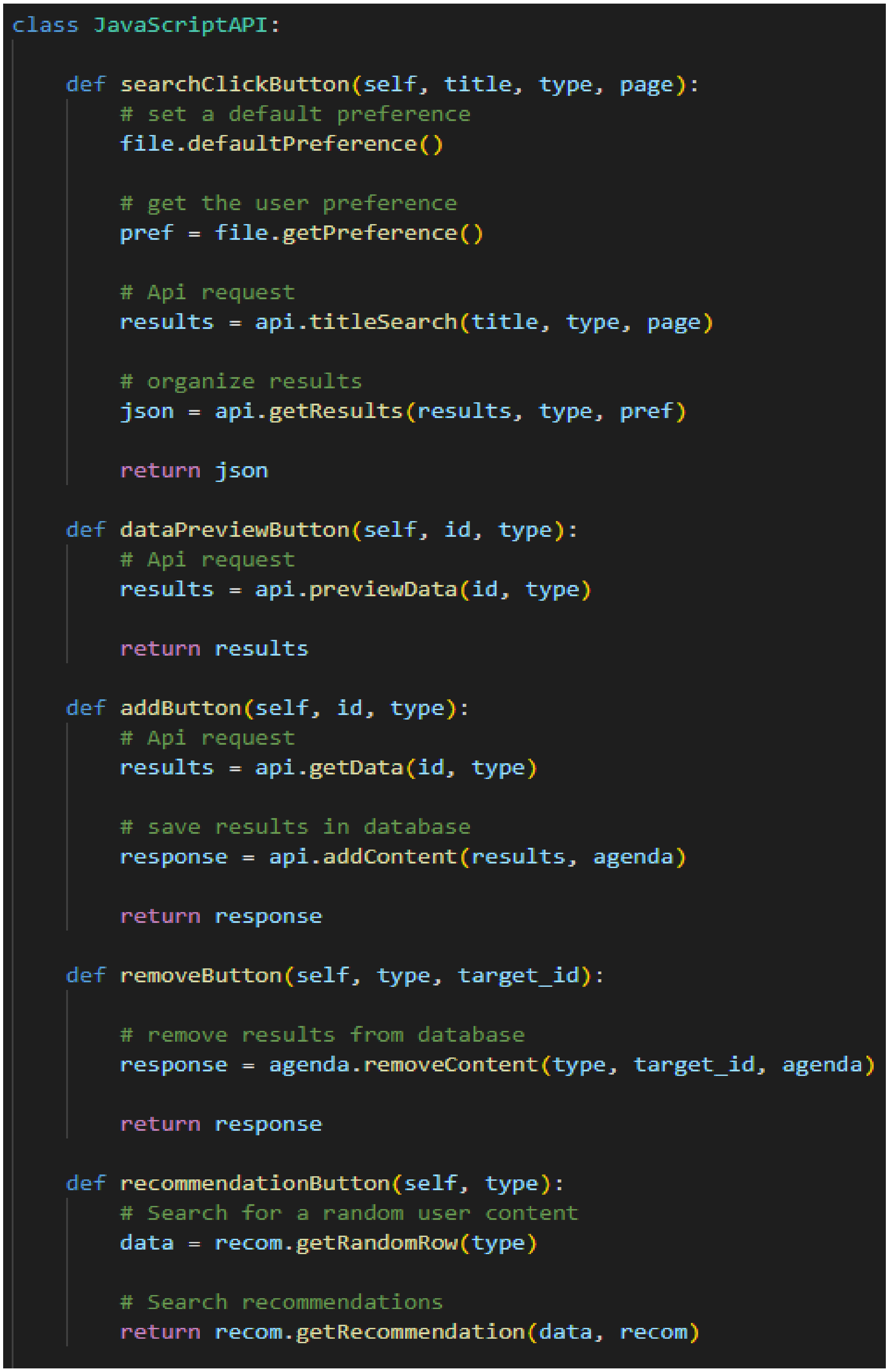
As the code is insanely extensive, it is going to be explained only what is happening in the main.py file.

In this file, the necessary modules are imported and the required objects are created.

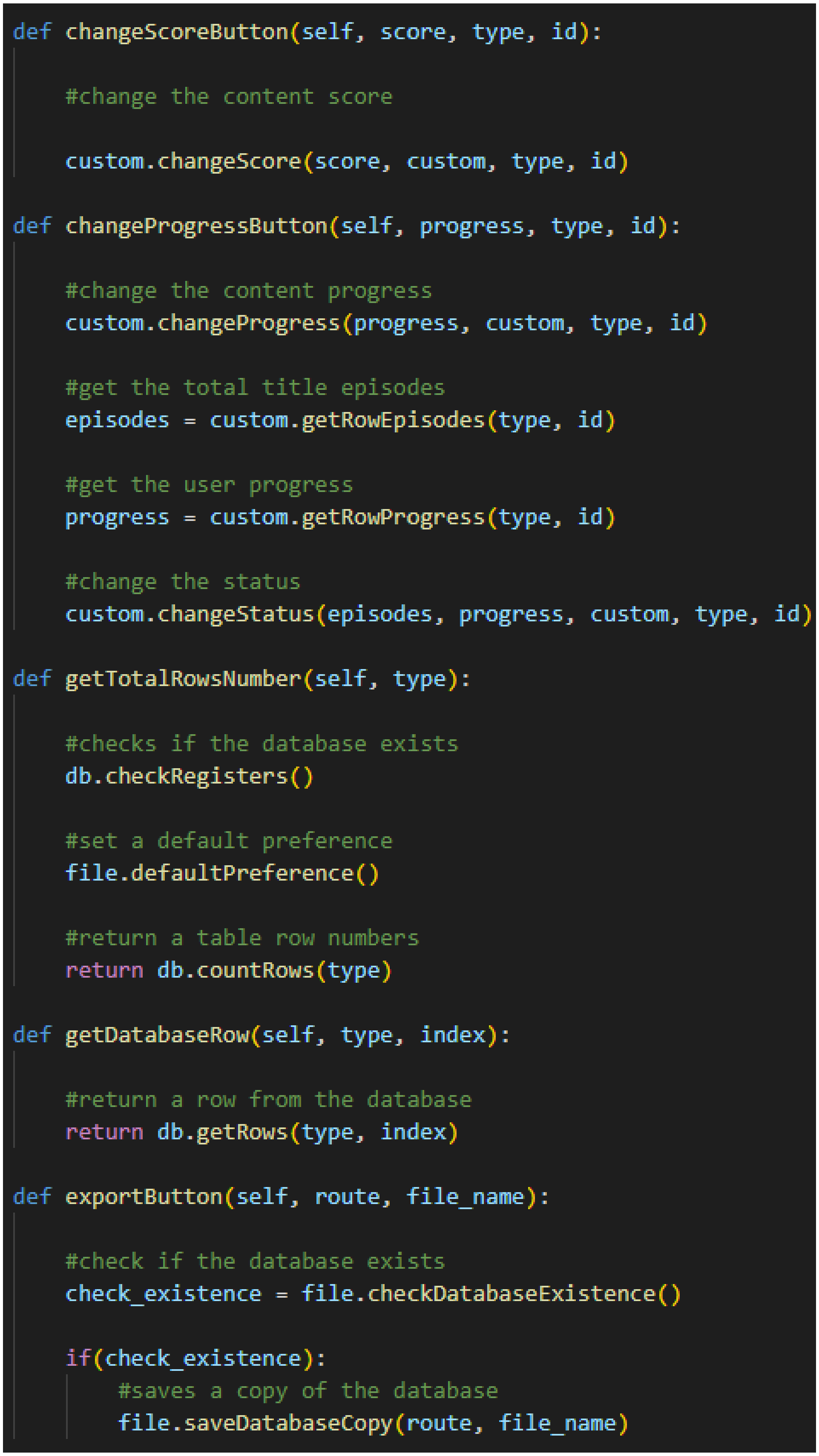


```
import webview
from src.logic.Agenda import Agenda
from src.logic.Api import Api
from src.logic.Database import Database
from src.logic.Customization import Customization
from src.logic.Recommendation import Recommendation
from src.logic.File import File

#create an agenda object
agenda = Agenda()

#create an Api object
api = Api()

#create a Database object
db = Database()

#create a Recommendation object
recom = Recommendation()

#create a customization object
custom = Customization()

#create a file object
file = File()
```

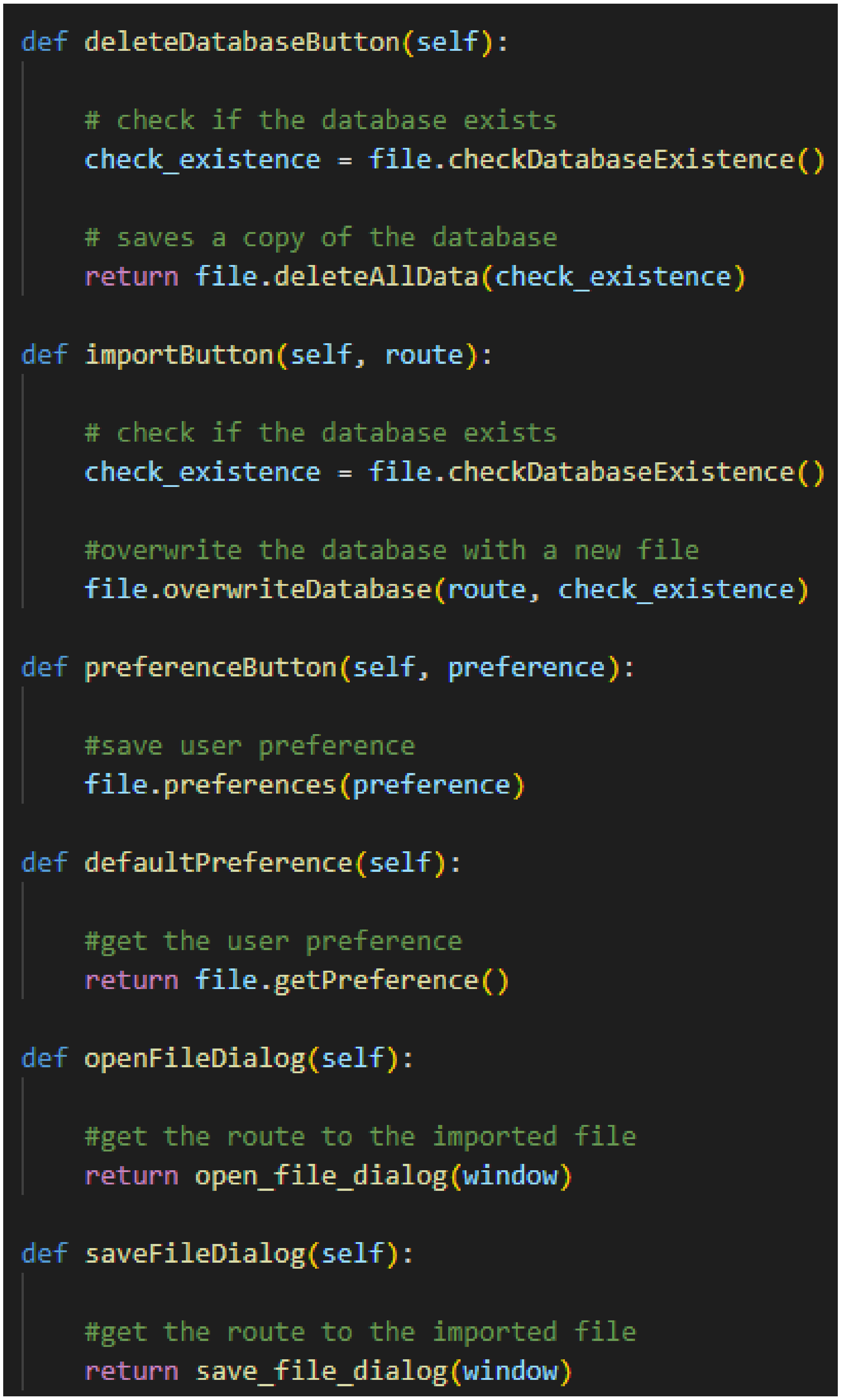
A javascript API class is created. This class contains all the required methods and functions for executing processes in the backend by actions performed in the frontend.



```
class JavaScriptAPI:

    def searchClickButton(self, title, type, page):
        # set a default preference
        file.defaultPreference()

        # get the user preference
        pref = file.getPreference()

        # Api request
        results = api.titleSearch(title, type, page)

        # organize results
        json = api.getResults(results, type, pref)

        return json

    def dataPreviewButton(self, id, type):
        # Api request
        results = api.previewData(id, type)

        return results

    def addButton(self, id, type):
        # Api request
        results = api.getData(id, type)

        # save results in database
        response = api.addContent(results, agenda)

        return response

    def removeButton(self, type, target_id):

        # remove results from database
        response = agenda.removeContent(type, target_id, agenda)

        return response

    def recommendationButton(self, type):
        # Search for a random user content
        data = recom.getRandomRow(type)

        # Search recommendations
        return recom.getRecommendation(data, recom)
```

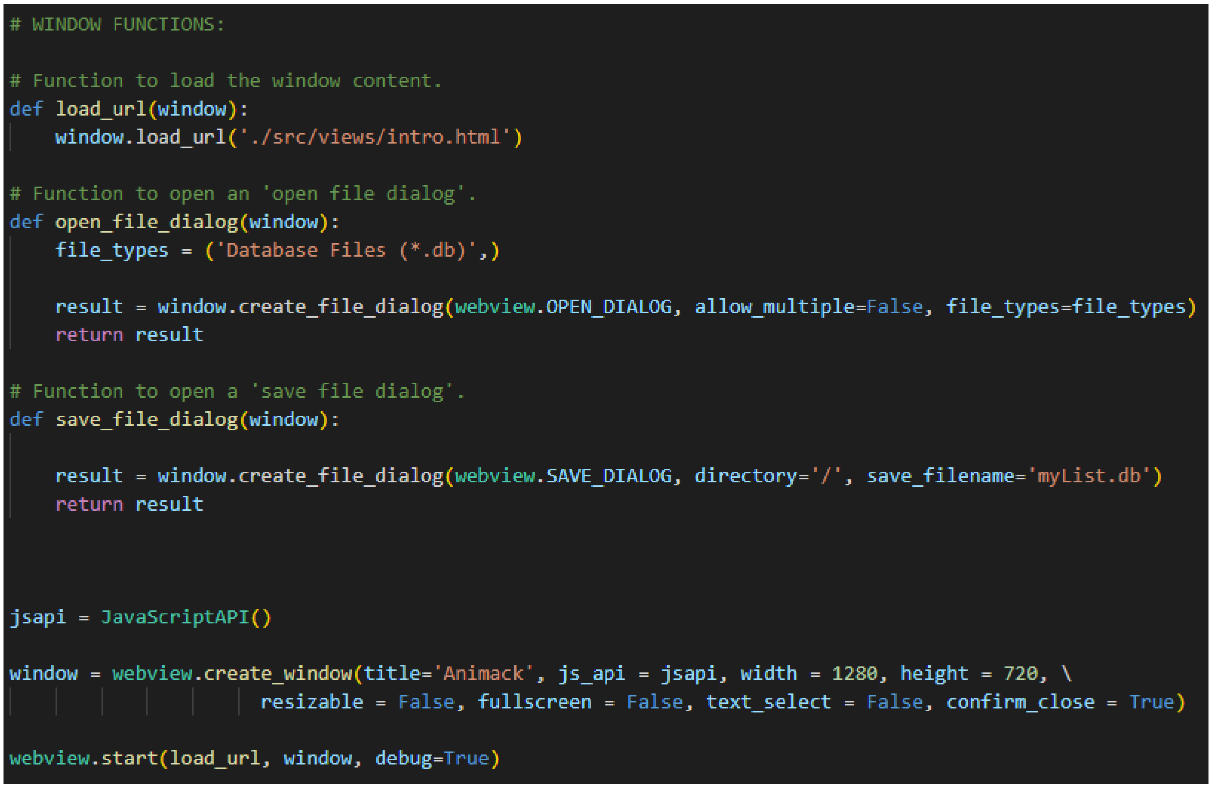


```
def changeScoreButton(self, score, type, id):

    #change the content score

    custom.changeScore(score, custom, type, id)

def changeProgressButton(self, progress, type, id):

    #change the content progress
    custom.changeProgress(progress, custom, type, id)

    #get the total title episodes
    episodes = custom.getRowEpisodes(type, id)

    #get the user progress
    progress = custom.getRowProgress(type, id)

    #change the status
    custom.changeStatus(episodes, progress, custom, type, id)

def getTotalRowsNumber(self, type):

    #checks if the database exists
    db.checkRegisters()

    #set a default preference
    file.defaultPreference()

    #return a table row numbers
    return db.countRows(type)

def getDatabaseRow(self, type, index):

    #return a row from the database
    return db.getRows(type, index)

def exportButton(self, route, file_name):

    #check if the database exists
    check_existence = file.checkDatabaseExistence()

    if(check_existence):
        #saves a copy of the database
        file.saveDatabaseCopy(route, file_name)
```



```
def deleteDatabaseButton(self):

    # check if the database exists
    check_existence = file.checkDatabaseExistence()

    # saves a copy of the database
    return file.deleteAllData(check_existence)

def importButton(self, route):

    # check if the database exists
    check_existence = file.checkDatabaseExistence()

    #overwrite the database with a new file
    file.overwriteDatabase(route, check_existence)

def preferenceButton(self, preference):

    #save user preference
    file.preferences(preference)

def defaultPreference(self):

    #get the user preference
    return file.getPreference()

def openFileDialog(self):

    #get the route to the imported file
    return open_file_dialog(window)

def saveFileDialog(self):

    #get the route to the imported file
    return save_file_dialog(window)
```

Finally, in this part of the code, functions can be executed to open the main windows, as well as open the file explorer to import or export a list.



```
# WINDOW FUNCTIONS:

# Function to load the window content.
def load_url(window):
    window.load_url('./src/views/intro.html')

# Function to open an 'open file dialog'.
def open_file_dialog(window):
    file_types = ('Database Files (*.db)',)

    result = window.create_file_dialog(webview.OPEN_DIALOG, allow_multiple=False, file_types=file_types)
    return result

# Function to open a 'save file dialog'.
def save_file_dialog(window):

    result = window.create_file_dialog(webview.SAVE_DIALOG, directory='/', save_filename='myList.db')
    return result



jsapi = JavaScriptAPI()

window = webview.create_window(title='Animack', js_api = jsapi, width = 1280, height = 720, \
                      resizable = False, fullscreen = False, text_select = False, confirm_close = True)

webview.start(load_url, window, debug=True)
```

USER MANUAL:

Functionalities:

1. Search for titles
2. Preview Titles
3. Add titles to the agenda
4. Remove titles from the agenda
5. rate contents
6. change the progress of contents
7. set preferences (filter adult content)
8. export list

9. import list

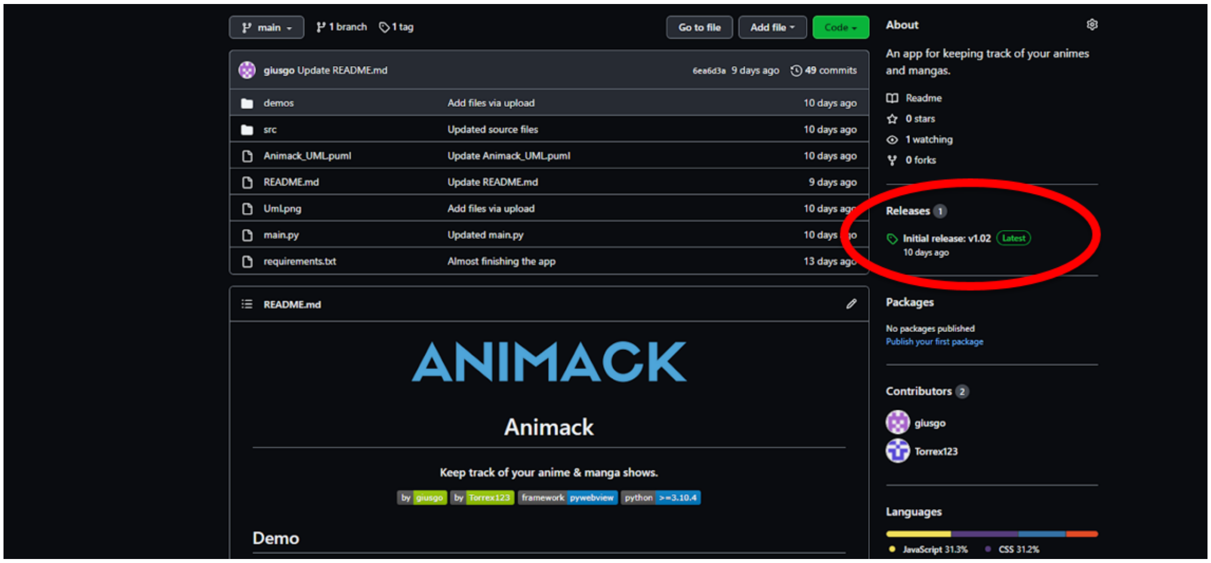
10. delete list

Installation guidance:

Step 1: Go to the official python page: https://www.python.org/ and download the latest version of python.

Step 2: Go to the official Animack GitHub repository: https://github.com/Torrex123/Animack

Step 3: On the right side of the page, locate the "release" area and click on it.



giusgo Update README.md 6ea6d3a 9 days ago 49 commits

demos	Add files via upload	10 days ago
src	Updated source files	10 days ago
Animack_UML.puml	Update Animack_UML.puml	10 days ago
README.md	Update README.md	9 days ago
Uml.png	Add files via upload	10 days ago
main.py	Updated main.py	10 days ago
requirements.txt	Almost finishing the app	13 days ago

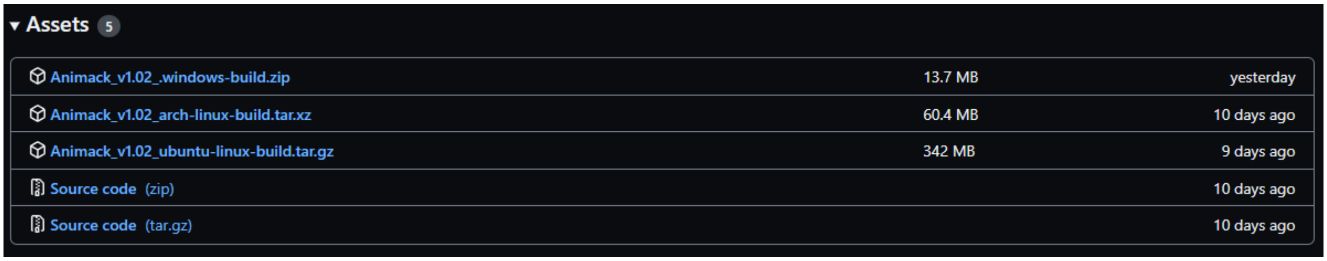
An app for keeping track of your animes and mangas.

Animack

Keep track of your anime & manga shows.

Step 4: Once in "initial release", click on one of the built-in files that correspond with your operating system. You can also download the source code of the application.

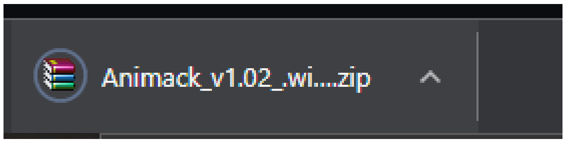
Note: Only built-in files are available for Windows and Linux (Arch and Ubuntu distributions)



Assets 5

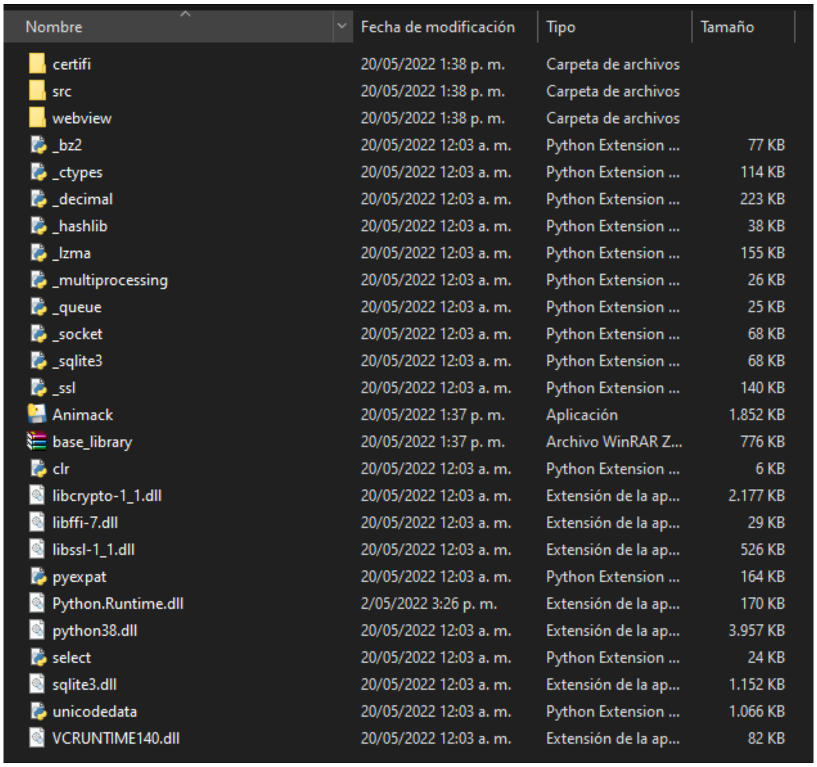
Animack_v1.02_windows-build.zip	13.7 MB	yesterday
Animack_v1.02_arch-linux-build.tar.xz	60.4 MB	10 days ago
Animack_v1.02_ubuntu-linux-build.tar.gz	342 MB	9 days ago
Source code (zip)		10 days ago
Source code (tar.gz)		10 days ago

Once clicked, a compressed file will be automatically downloaded in your system



Animack_v1.02_.wi....zip

Step 5: Extract the zip file. A folder with the app will be decompressed.

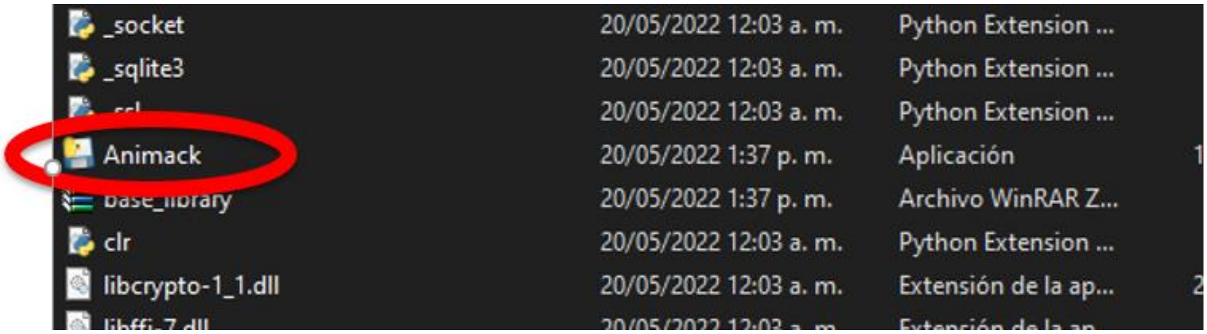


Nombre	Fecha de modificación	Tipo	Tamaño
certifi	20/05/2022 1:38 p. m.	Carpeta de archivos	
src	20/05/2022 1:38 p. m.	Carpeta de archivos	
webview	20/05/2022 1:38 p. m.	Carpeta de archivos	
_bz2	20/05/2022 12:03 a. m.	Python Extension ...	77 KB
_ctypes	20/05/2022 12:03 a. m.	Python Extension ...	114 KB
_decimal	20/05/2022 12:03 a. m.	Python Extension ...	223 KB
_hashlib	20/05/2022 12:03 a. m.	Python Extension ...	38 KB
_lzma	20/05/2022 12:03 a. m.	Python Extension ...	155 KB
_multiprocessing	20/05/2022 12:03 a. m.	Python Extension ...	26 KB
_queue	20/05/2022 12:03 a. m.	Python Extension ...	25 KB
_socket	20/05/2022 12:03 a. m.	Python Extension ...	68 KB
_sqlite3	20/05/2022 12:03 a. m.	Python Extension ...	68 KB
_ssl	20/05/2022 12:03 a. m.	Python Extension ...	140 KB
Animack	20/05/2022 1:37 p. m.	Aplicación	1.852 KB
base_library	20/05/2022 1:37 p. m.	Archivo WinRAR Z...	776 KB
clr	20/05/2022 12:03 a. m.	Python Extension ...	6 KB
libcrypto-1_1.dll	20/05/2022 12:03 a. m.	Extensión de la ap...	2.177 KB
libffi-7.dll	20/05/2022 12:03 a. m.	Extensión de la ap...	29 KB
libssl-1_1.dll	20/05/2022 12:03 a. m.	Extensión de la ap...	526 KB
pyexpat	20/05/2022 12:03 a. m.	Python Extension ...	164 KB
Python.Runtime.dll	2/05/2022 3:26 p. m.	Extensión de la ap...	170 KB
python38.dll	20/05/2022 12:03 a. m.	Extensión de la ap...	3.957 KB
select	20/05/2022 12:03 a. m.	Python Extension ...	24 KB
sqlite3.dll	20/05/2022 12:03 a. m.	Extensión de la ap...	1.152 KB
unicodedata	20/05/2022 12:03 a. m.	Python Extension ...	1.066 KB
VCRUNTIME140.dll	20/05/2022 12:03 a. m.	Extensión de la ap...	82 KB

The installation is finished.

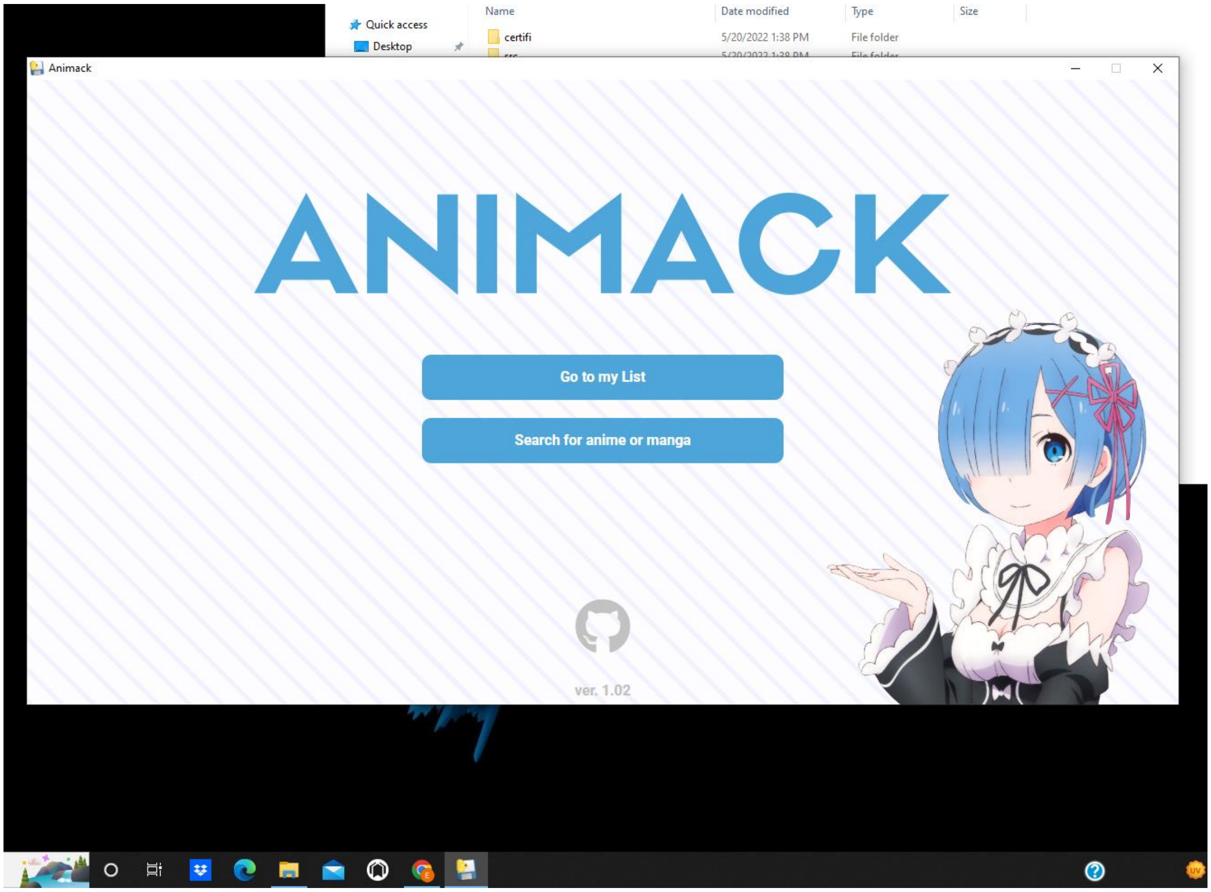
Execution guidance:

Once downloaded and extracted the Animack compressed file, go to the folder and double click the Animack file.



_socket	20/05/2022 12:03 a. m.	Python Extension ...	
_sqlite3	20/05/2022 12:03 a. m.	Python Extension ...	
_ssl	20/05/2022 12:03 a. m.	Python Extension ...	
Animack	20/05/2022 1:37 p. m.	Aplicación	1
base_library	20/05/2022 1:37 p. m.	Archivo WinRAR Z...	
clr	20/05/2022 12:03 a. m.	Python Extension ...	
libcrypto-1_1.dll	20/05/2022 12:03 a. m.	Extensión de la ap...	2
libffi-7.dll	20/05/2022 12:03 a. m.	Extensión de la ap...	

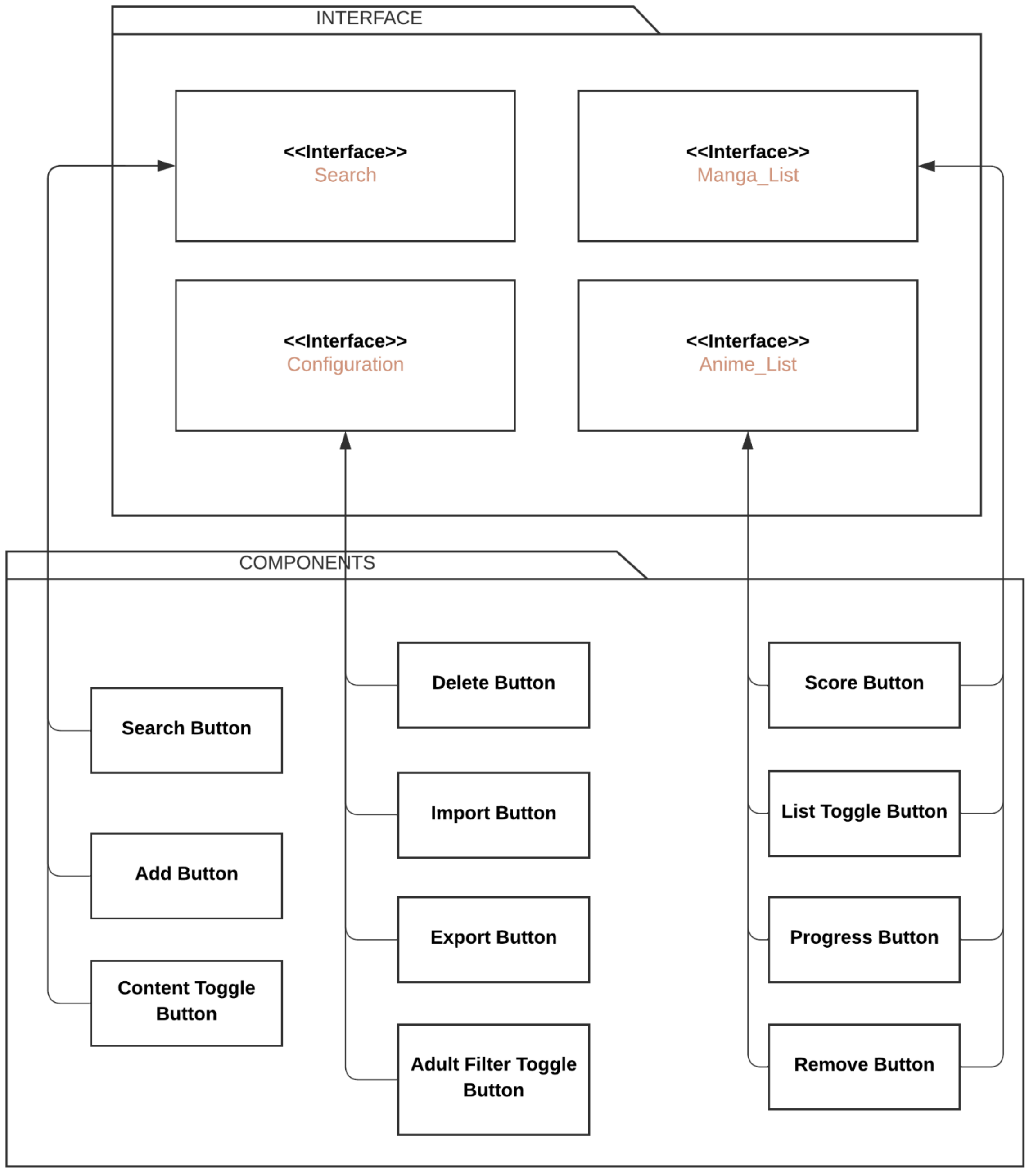
The program will now execute:



Name	Date modified	Type	Size
certifi	5/20/2022 1:38 PM	File folder	

VIII. USER EXPERIENCE AND GRAPHIC INTERFACE

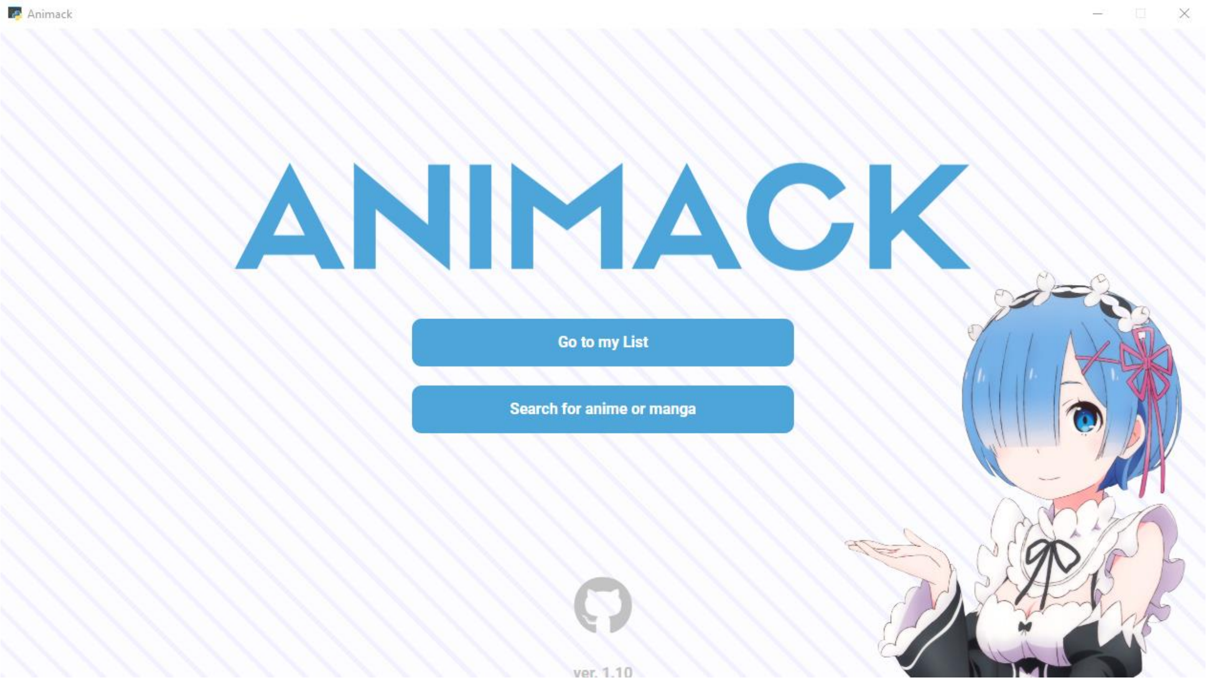
Design process:



The frontend is composed of 4 windows: the search window, the manga list window, the anime list window, and the configuration window. Each window has several components and their relations can be seen in the UML.

Navigation + Graphic component descriptions

Start Window

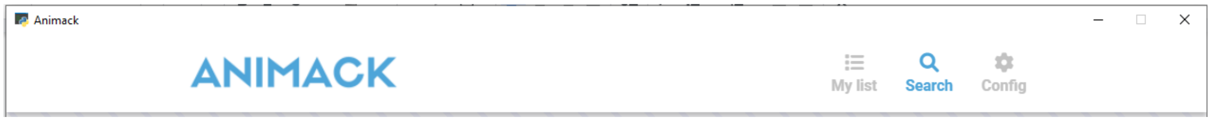


The start window is composed of 2 buttons:

- **Go to my list:** where, if clicked, redirects the user to the lists windows.
- **Search for anime or manga:** where, if clicked, redirects the user to the search window.

Additionally, there is a GitHub icon button that, if clicked, redirects the user to the official Aniamck Github repository.

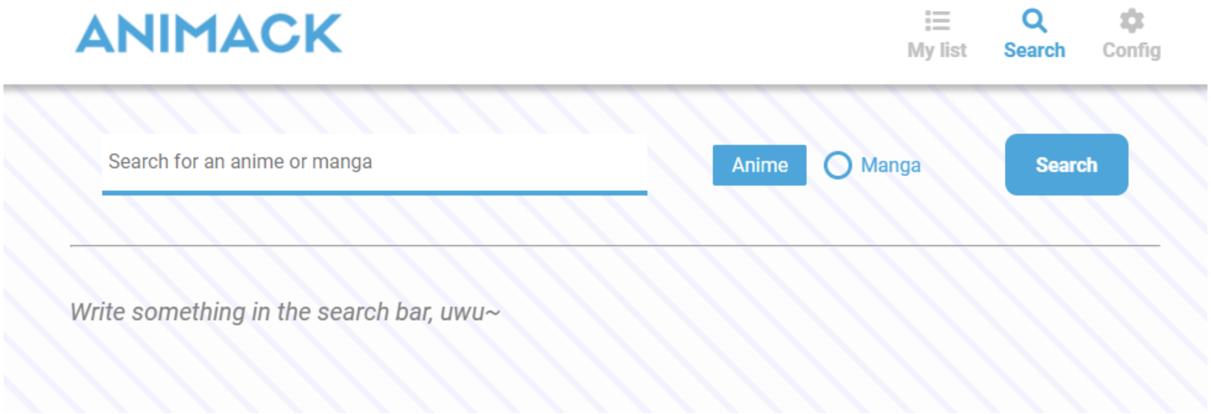
Navigation Window



This navigator is present in the list, search, and configuration windows. It helps you to easily navigate between windows. It consists of three buttons:

- **My list:** to go to the list window
- **Search:** to go to the search window
- **Config:** to go to the configuration window

Search Window

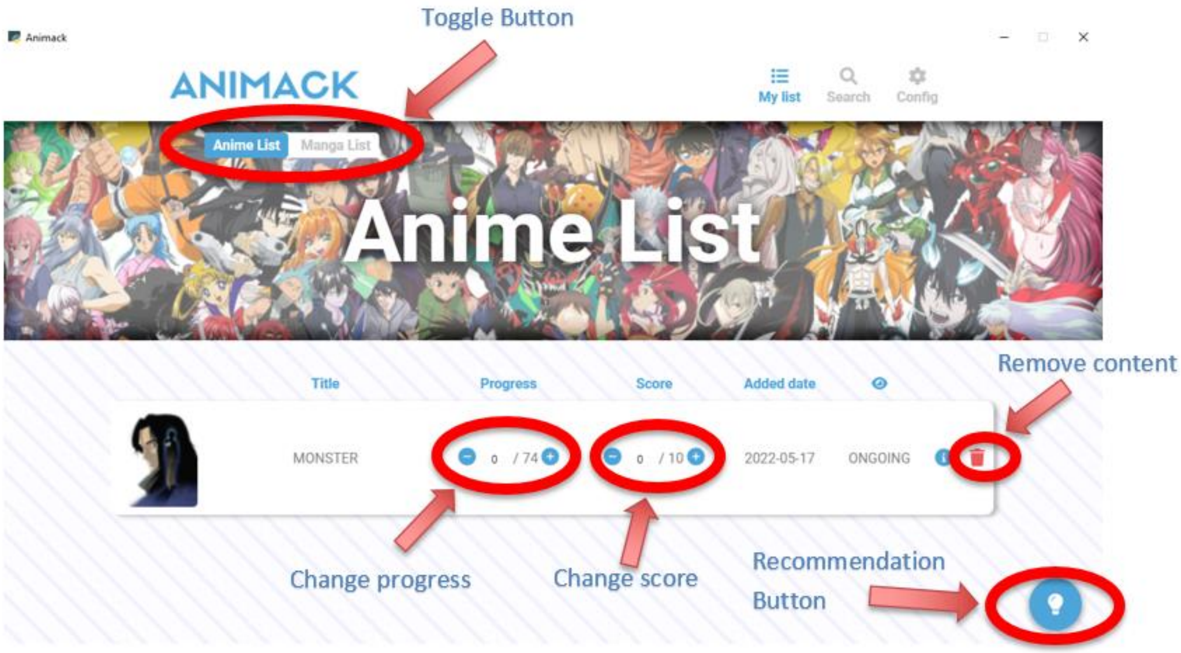


Write something in the search bar, uwu~

this window is composed of several widgets:

- **Toggle button:** the user can filter the search results by manga or anime
- **Textfield:** the user input goes there.
- **Search button:** Once clicked, it will throw results based on the user input.

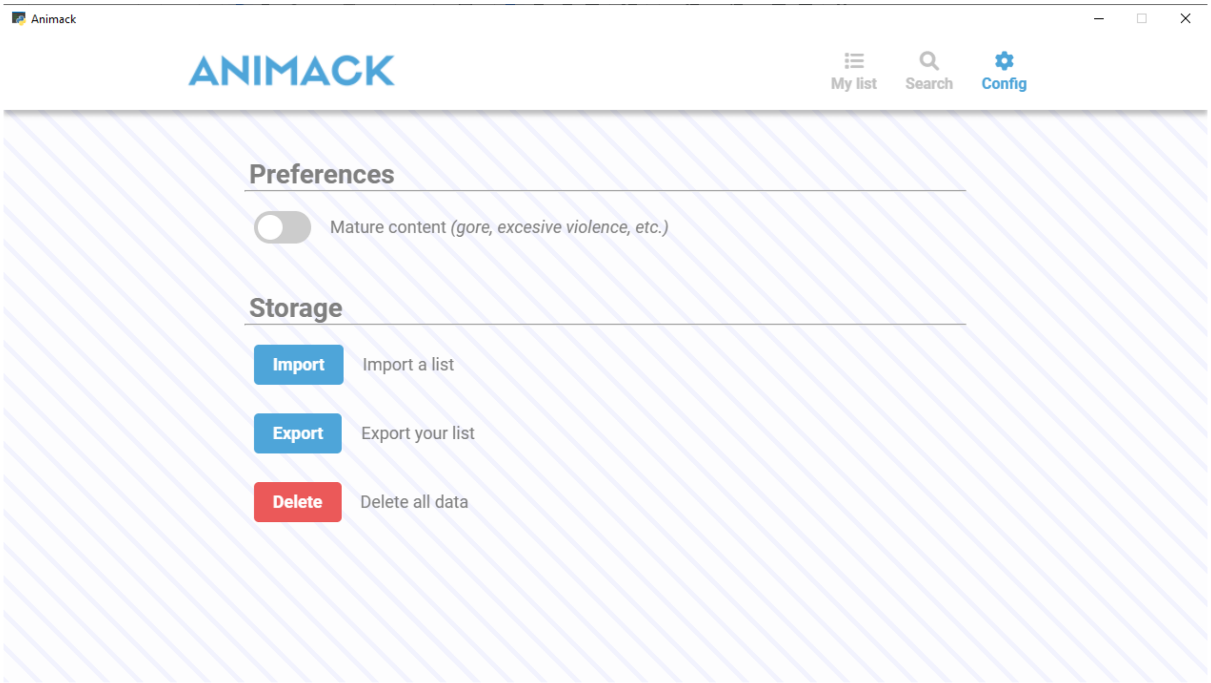
List Window



This window contains several buttons:

- **Toggle button:** the user can switch between manga and anime lists
- **Add and rest buttons:** the user can change the progress and score of a content
- **Recommendation Button:** the user can see recommendations based on their contents
- **Remove content**: the user can decide to delete a content

Configuration Window:



Preferences

Mature content *(gore, excesive violence, etc.)*

Storage

Import — Import a list

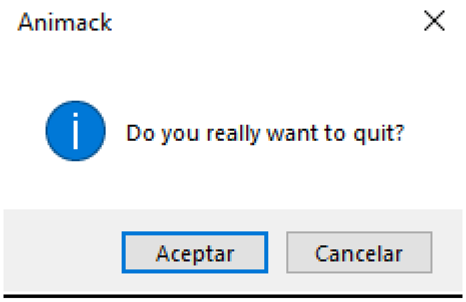
Export — Export your list

Delete — Delete all data

This window contains several buttons:

- **Toggle button:** the user can switch between filtering or not the adult content
- **Import button:** the user can import an external list to the application
- **Export button:** the user can export the current app list to the desired location on their system
- **Delete content**: the user can decide to delete the current app, perhaps for starting a new one.

Exit Confirmation Window:

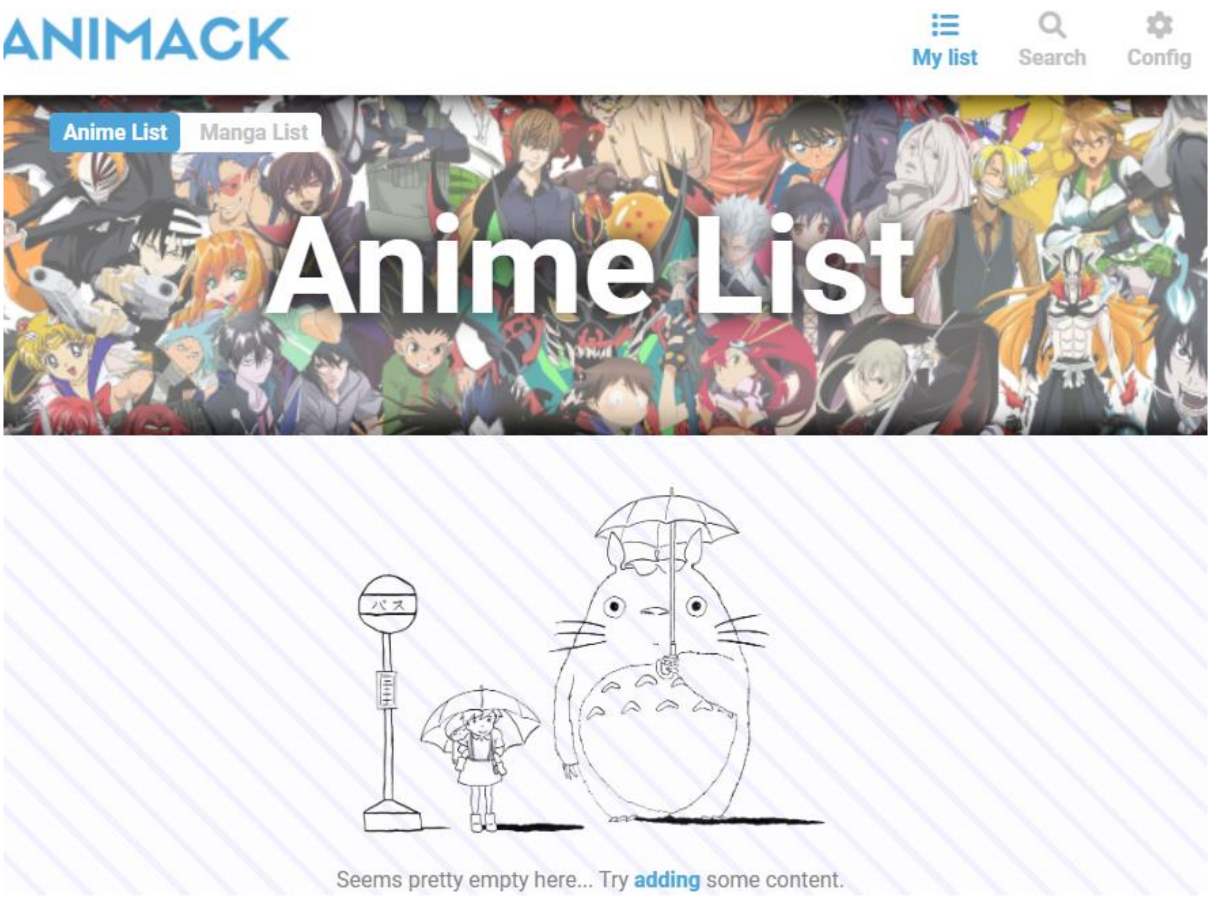

Animack

Do you really want to quit?

this window is composed of two buttons: One for exiting the app and one for staying.

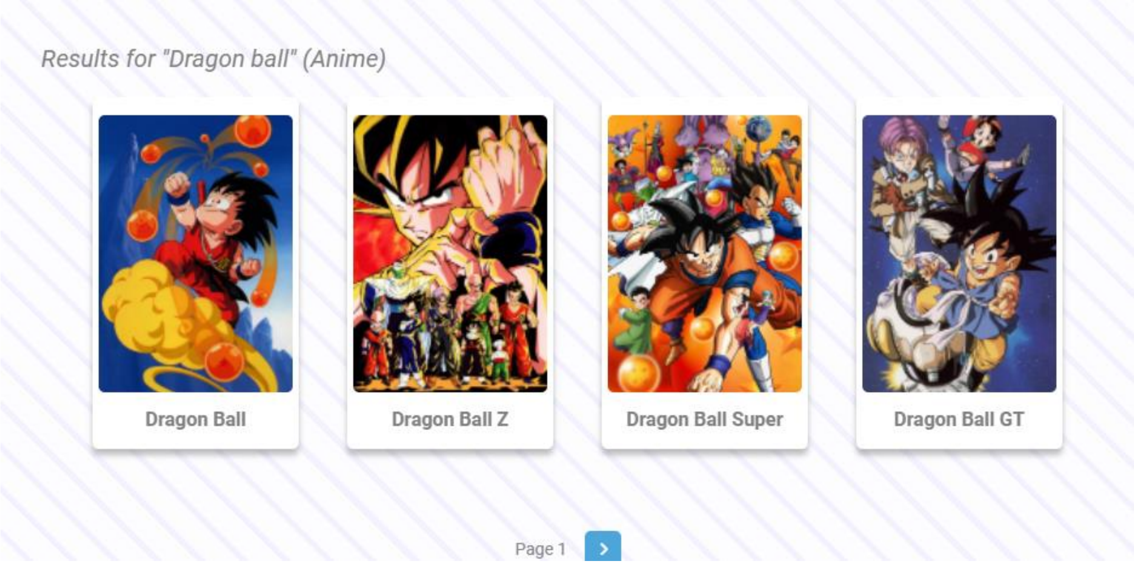
IX. APP TEST

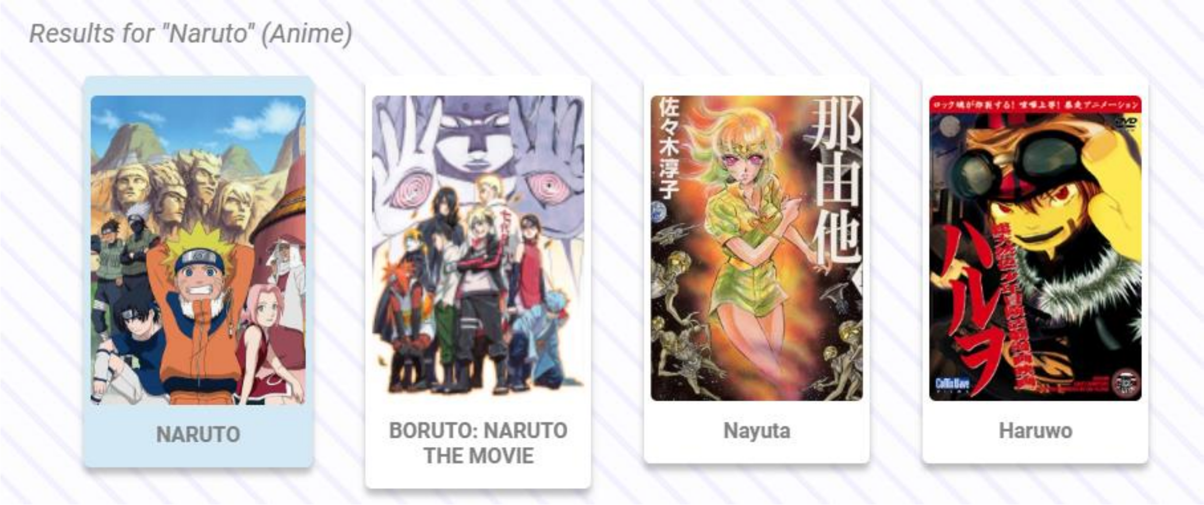
For running the test, it must be ensured that the list is empty.

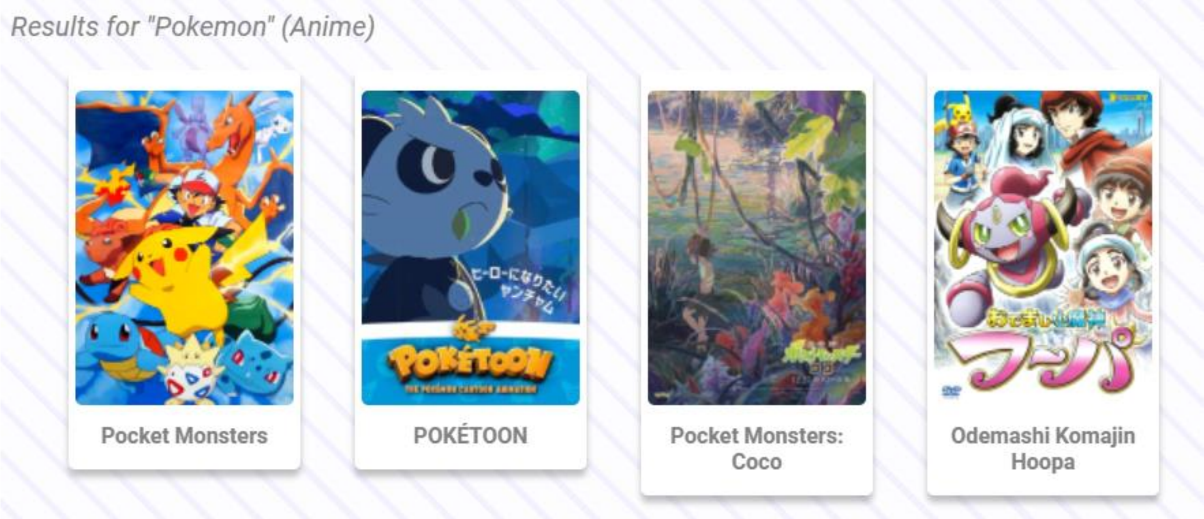


Seems pretty empty here... Try adding some content.

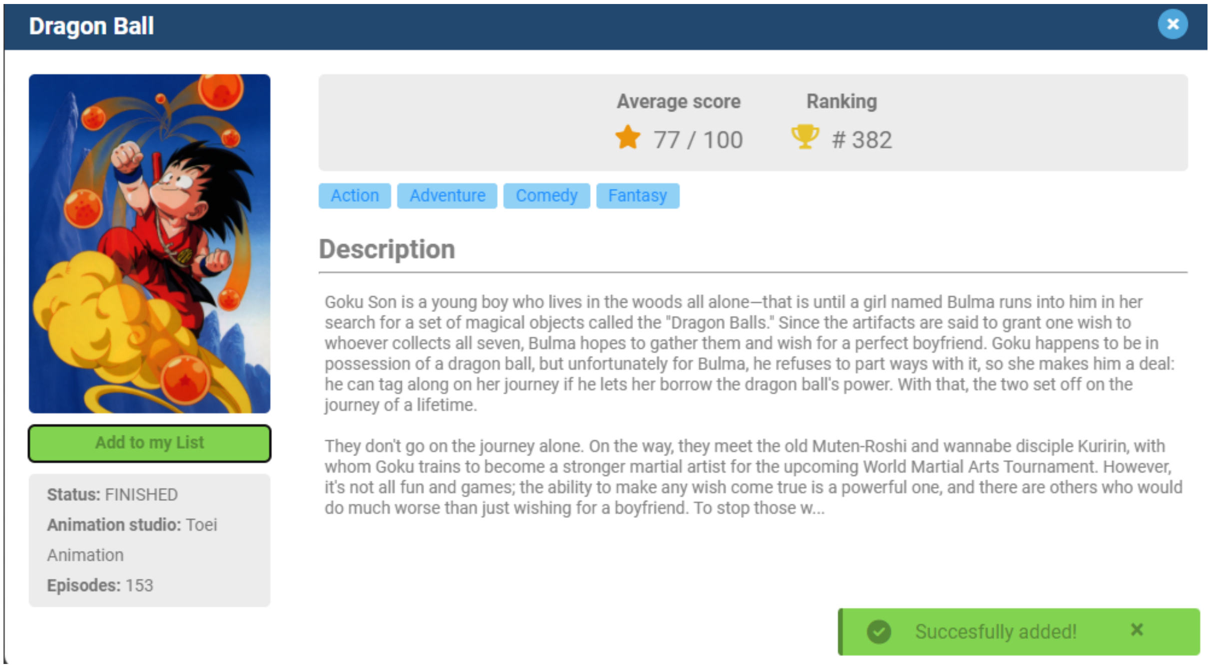
In the search window, it is going to be searched for Dragon Ball, Naruto, and Pokemon.







Once seen that the search window is working, it is going to proceed to add one of those results to our Anime list.



Dragon Ball

Average score ★ 77 / 100

Ranking 🏆 # 382

Description

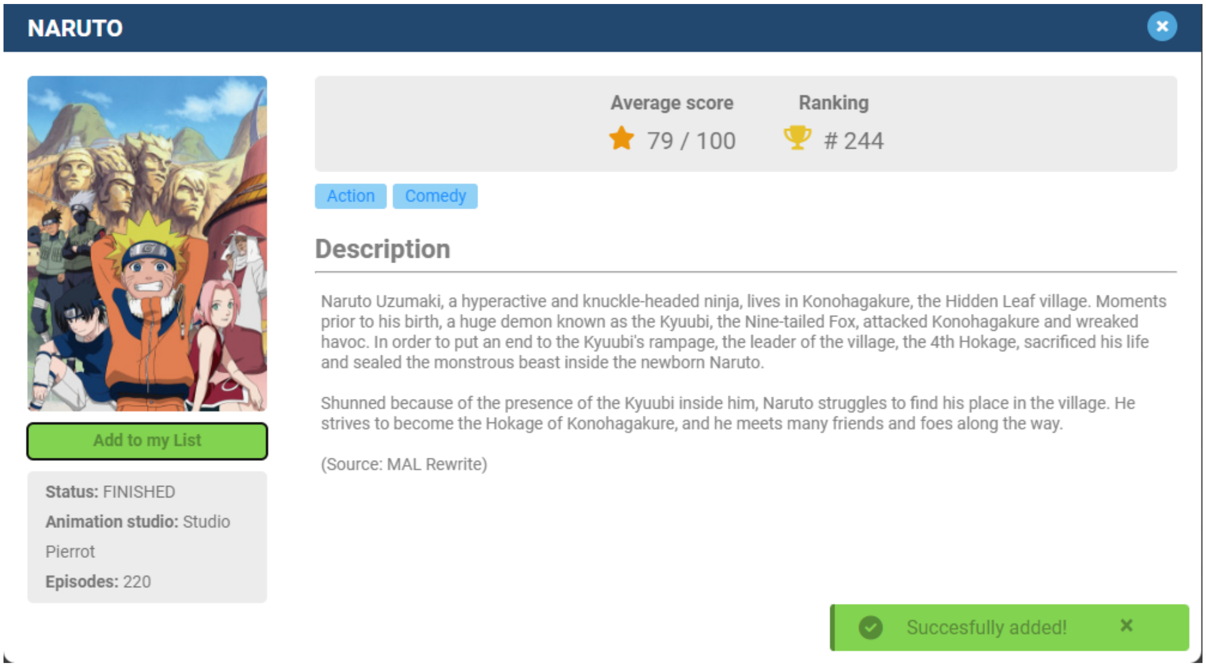
Goku Son is a young boy who lives in the woods all alone—that is until a girl named Bulma runs into him in her search for a set of magical objects called the "Dragon Balls." Since the artifacts are said to grant one wish to whoever collects all seven, Bulma hopes to gather them and wish for a perfect boyfriend. Goku happens to be in possession of a dragon ball, but unfortunately for Bulma, he refuses to part ways with it, so she makes him a deal: he can tag along on her journey if he lets her borrow the dragon ball's power. With that, the two set off on the journey of a lifetime.

They don't go on the journey alone. On the way, they meet the old Muten-Roshi and wannabe disciple Kuririn, with whom Goku trains to become a stronger martial artist for the upcoming World Martial Arts Tournament. However, it's not all fun and games; the ability to make any wish come true is a powerful one, and there are others who would do much worse than just wishing for a boyfriend. To stop those w...

Status: FINISHED
Animation studio: Toei Animation
Episodes: 153

✓ Succesfully added! ×

The right bottom green message confirms the content "Dragon Ball" was added successfully



NARUTO

Average score ★ 79 / 100

Ranking 🏆 # 244

Description

Naruto Uzumaki, a hyperactive and knuckle-headed ninja, lives in Konohagakure, the Hidden Leaf village. Moments prior to his birth, a huge demon known as the Kyuubi, the Nine-tailed Fox, attacked Konohagakure and wreaked havoc. In order to put an end to the Kyuubi's rampage, the leader of the village, the 4th Hokage, sacrificed his life and sealed the monstrous beast inside the newborn Naruto.

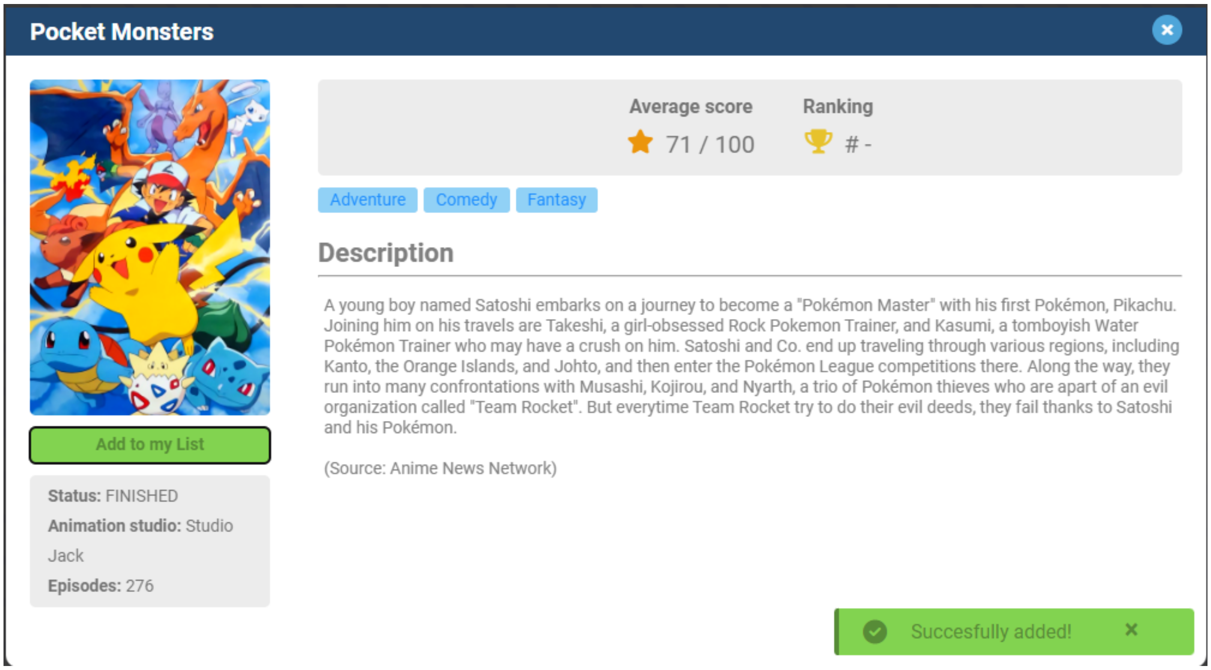
Shunned because of the presence of the Kyuubi inside him, Naruto struggles to find his place in the village. He strives to become the Hokage of Konohagakure, and he meets many friends and foes along the way.

(Source: MAL Rewrite)

Status: FINISHED
Animation studio: Studio Pierrot
Episodes: 220

✓ Succesfully added! ×

The right bottom green message confirms the content “Naruto” was added successfully



Status: FINISHED
Animation studio: Studio Jack
Episodes: 276

Description

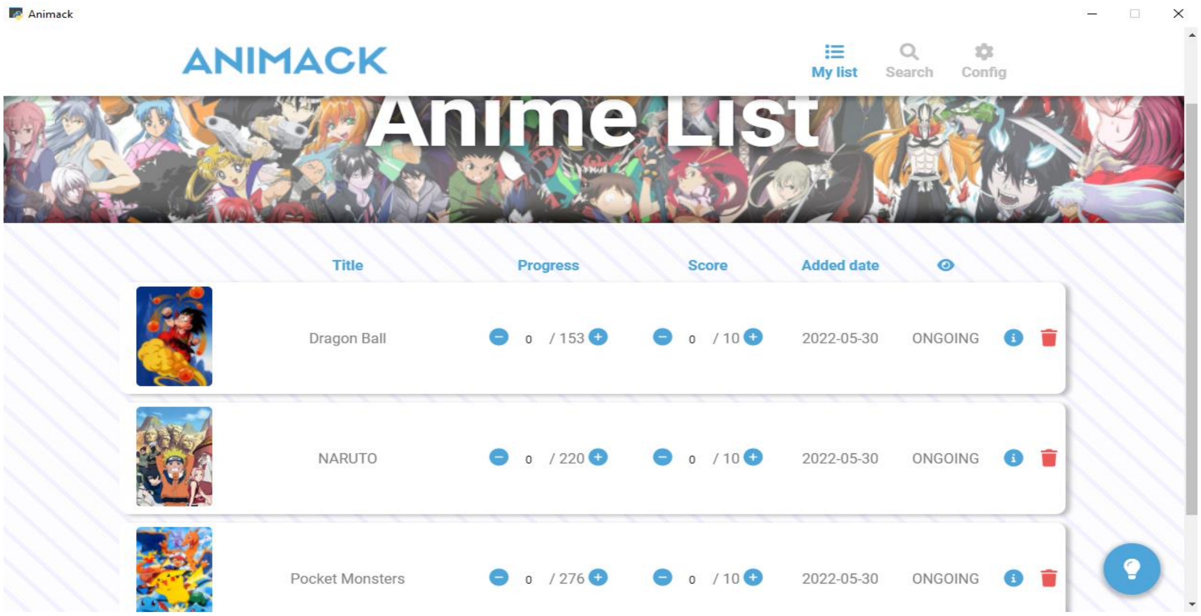
A young boy named Satoshi embarks on a journey to become a "Pokémon Master" with his first Pokémon, Pikachu. Joining him on his travels are Takeshi, a girl-obsessed Rock Pokemon Trainer, and Kasumi, a tomboyish Water Pokémon Trainer who may have a crush on him. Satoshi and Co. end up traveling through various regions, including Kanto, the Orange Islands, and Johto, and then enter the Pokémon League competitions there. Along the way, they run into many confrontations with Musashi, Kojirou, and Nyarth, a trio of Pokémon thieves who are apart of an evil organization called "Team Rocket". But everytime Team Rocket try to do their evil deeds, they fail thanks to Satoshi and his Pokémon.

(Source: Anime News Network)

Succesfully added!

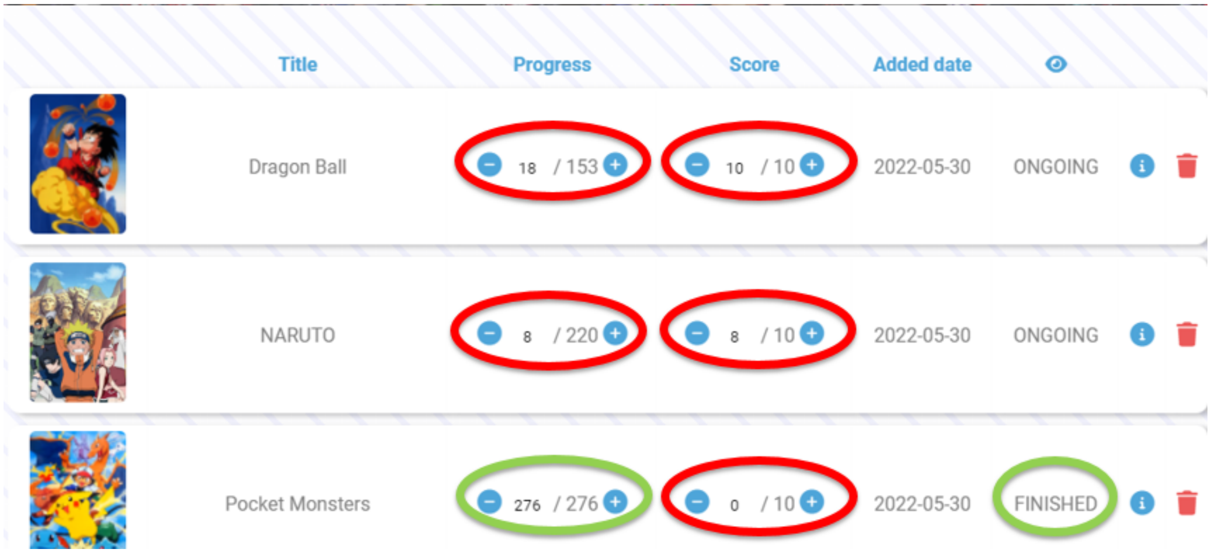
The right bottom green message confirms the content “Naruto” was added successfully

Now, it is going to be confirmed if those three contents were successfully added to the list by going to the anime list section:



Title	Progress	Score	Added date	
Dragon Ball	0 / 153	0 / 10	2022-05-30	ONGOING
NARUTO	0 / 220	0 / 10	2022-05-30	ONGOING
Pocket Monsters	0 / 276	0 / 10	2022-05-30	ONGOING

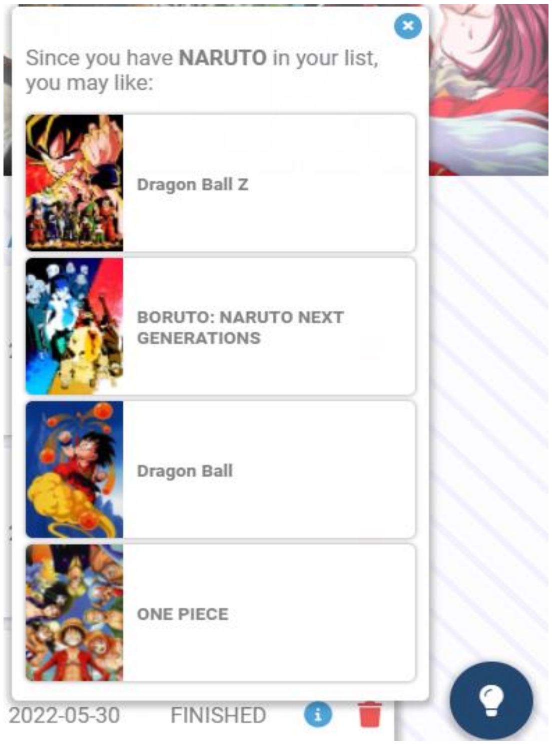
It can be confirmed that the contents are on the list. Now is going to be modified the progress and score of each one of the contents to see if the buttons really work



	Title	Progress	Score	Added date	👁
	Dragon Ball	18 / 153	10 / 10	2022-05-30	ONGOING
	NARUTO	8 / 220	8 / 10	2022-05-30	ONGOING
	Pocket Monsters	276 / 276	0 / 10	2022-05-30	FINISHED

The values were successfully modified. Notice how when the progress corresponds with the maximum episode of content, the status changes from “ONGOING” to “FINISHED”.

Now it is going to be clicked on the recommendation button to see if it works:



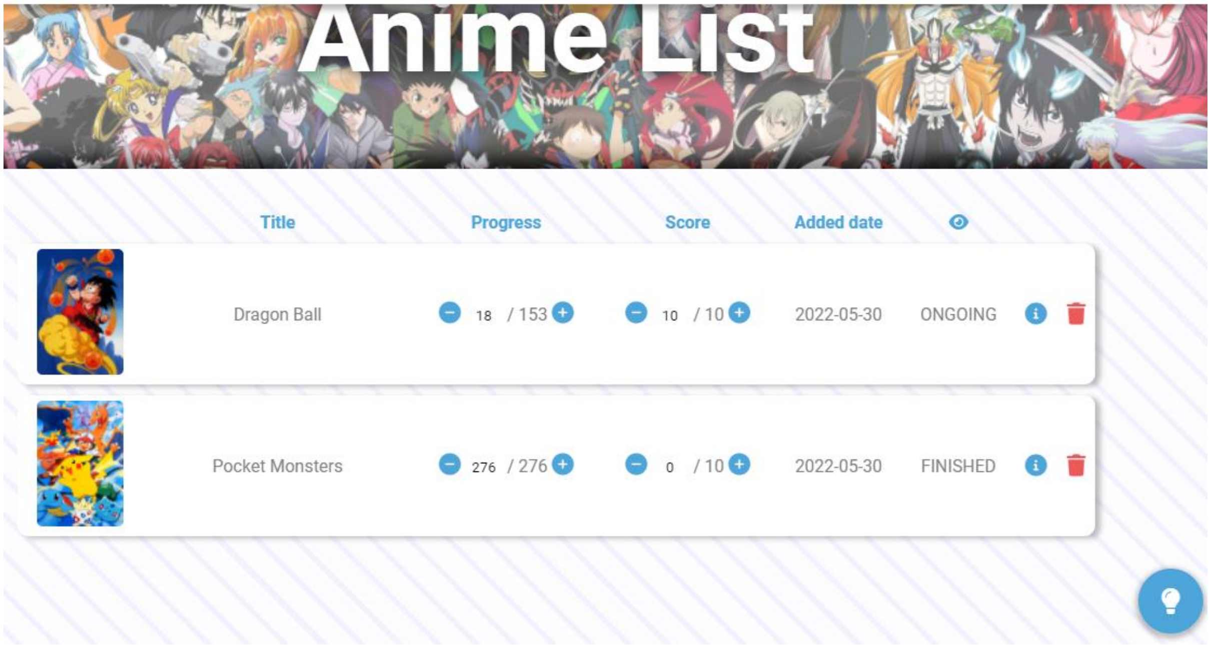
Since you have **NARUTO** in your list, you may like:

- Dragon Ball Z
- BORUTO: NARUTO NEXT GENERATIONS
- Dragon Ball
- ONE PIECE

2022-05-30 FINISHED

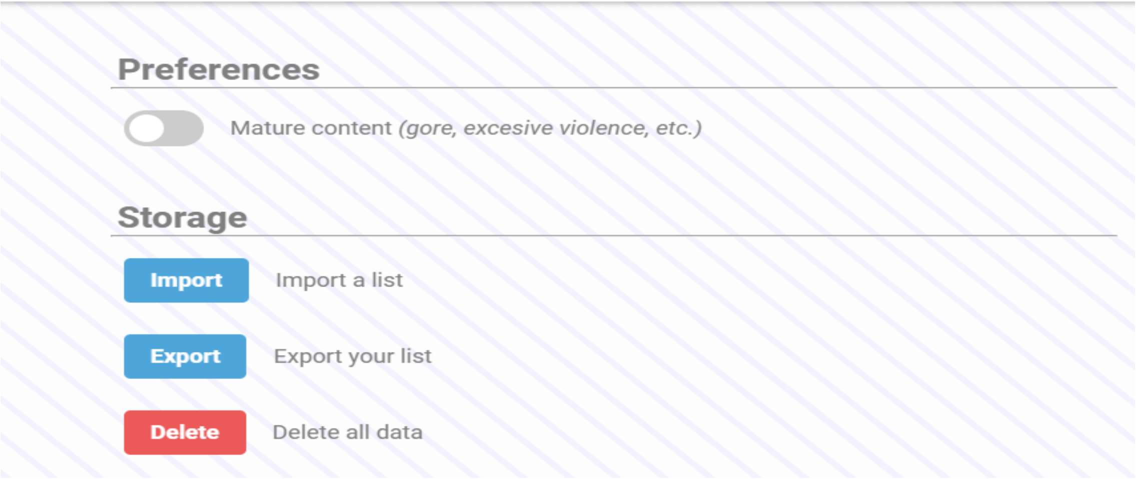
It can be confirmed that the recommendation button works. In this case, it is recommending similar content based on Naruto.

Now it is going to be tried the remove button. For this test, the content “Naruto” is going to be removed from the list, so the contents that should remain must be “Dragon Ball” and “Pokemon”.



Title	Progress	Score	Added date	👁		
Dragon Ball	18 / 153	10 / 10	2022-05-30	ONGOING	ℹ	🗑
Pocket Monsters	276 / 276	0 / 10	2022-05-30	FINISHED	ℹ	🗑

It can be confirmed that the remove button works. Now it is going to be proceed to the configuration window to try its features:



Preferences

○ Mature content *(gore, excesive violence, etc.)*

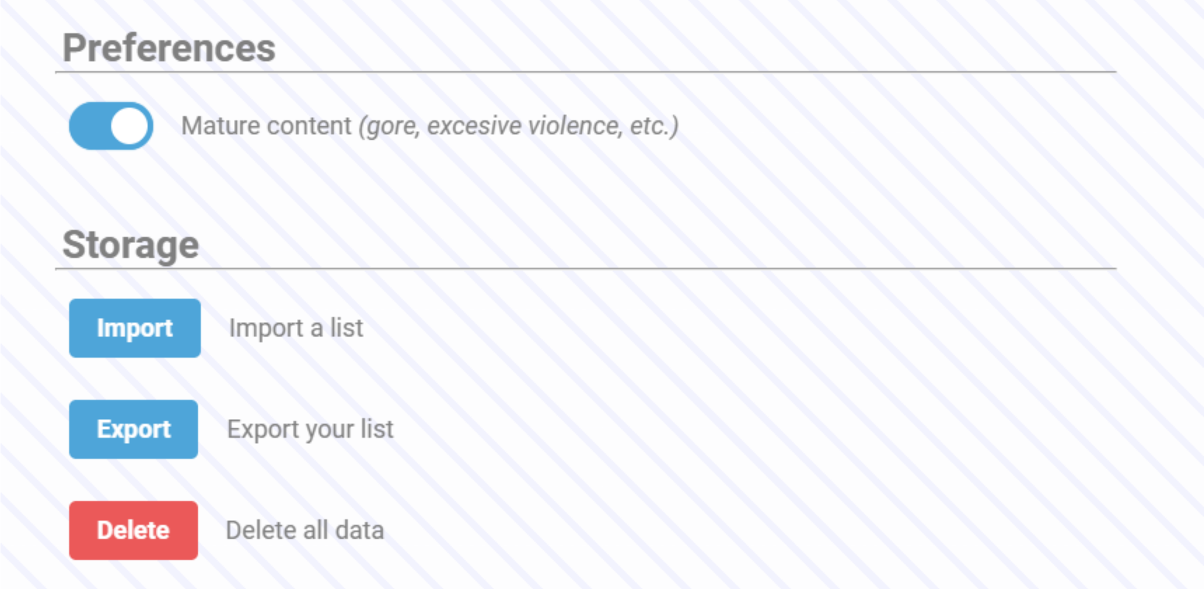
Storage

Import a list

Export your list

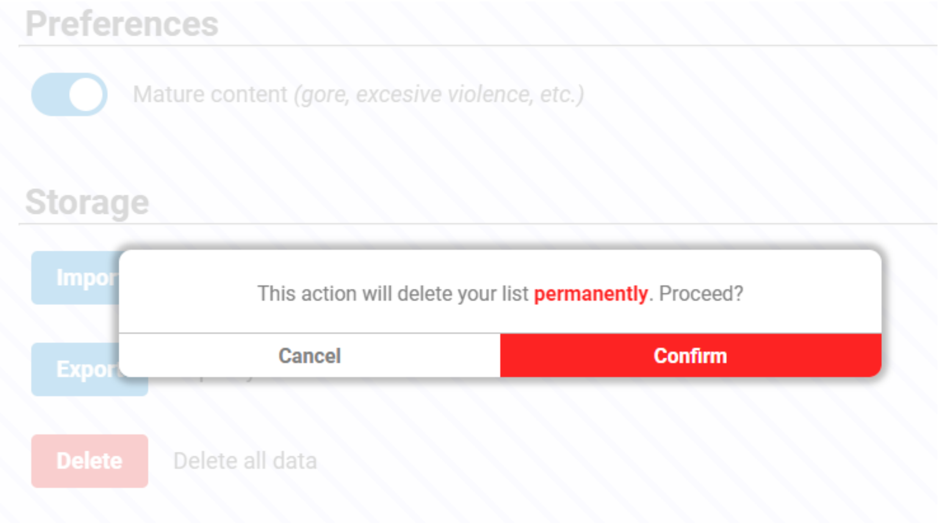
Delete all data

It is going to be clicked on the toggle button to see if it works:

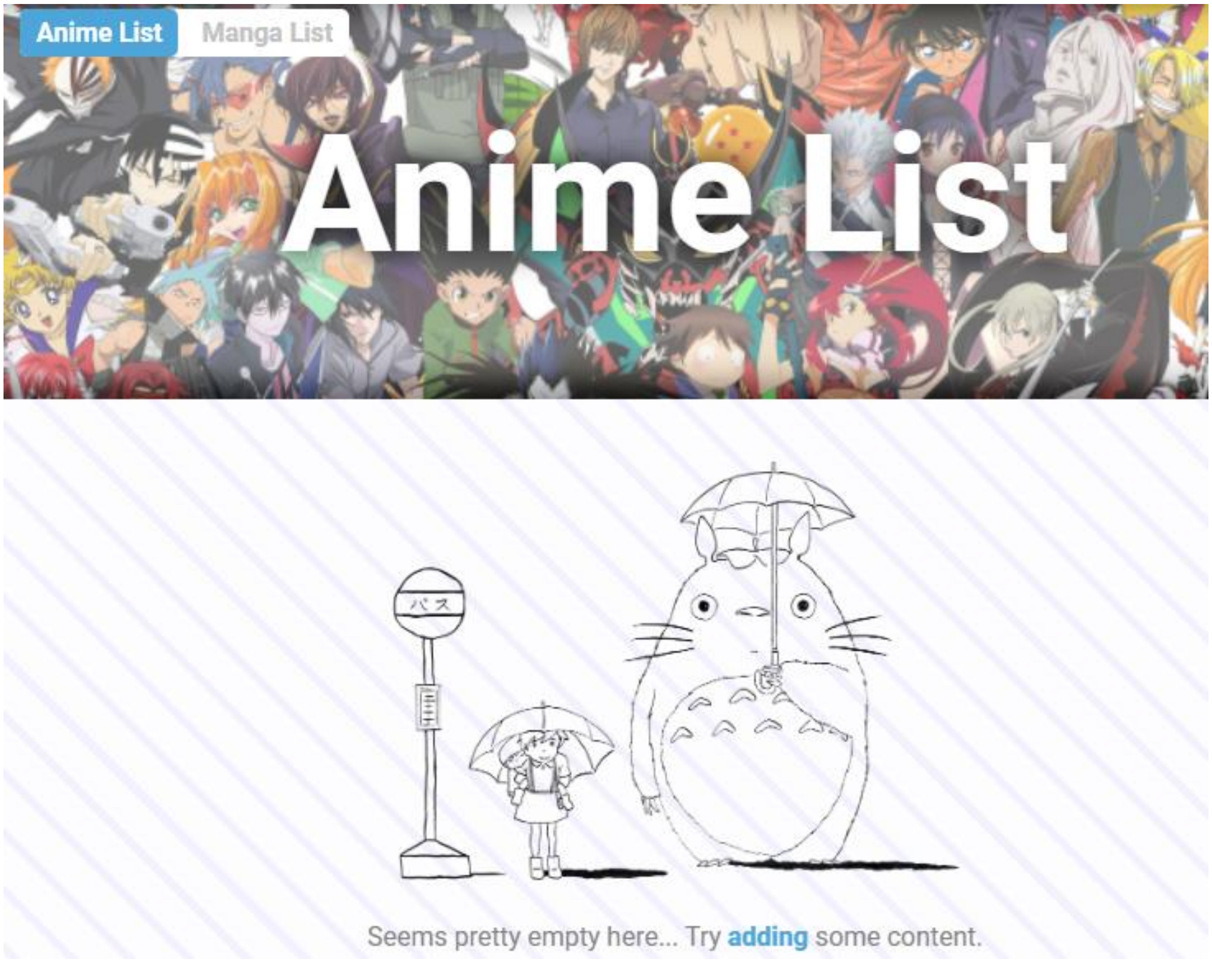


the toggle button is now on so it allows adults results in the search window.

Now it is going to be tried the delete button. By clicking on it , taking into account our anime list, the contents “Dragon Ball” and “Pokemon” should be deleted from the list.

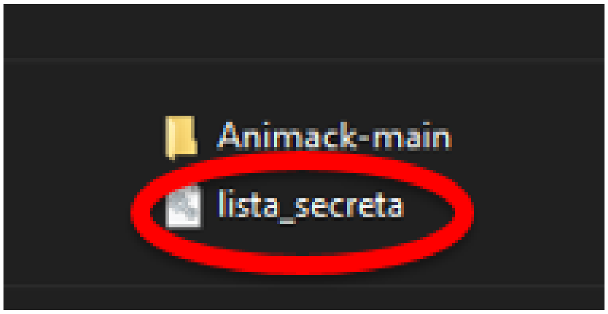


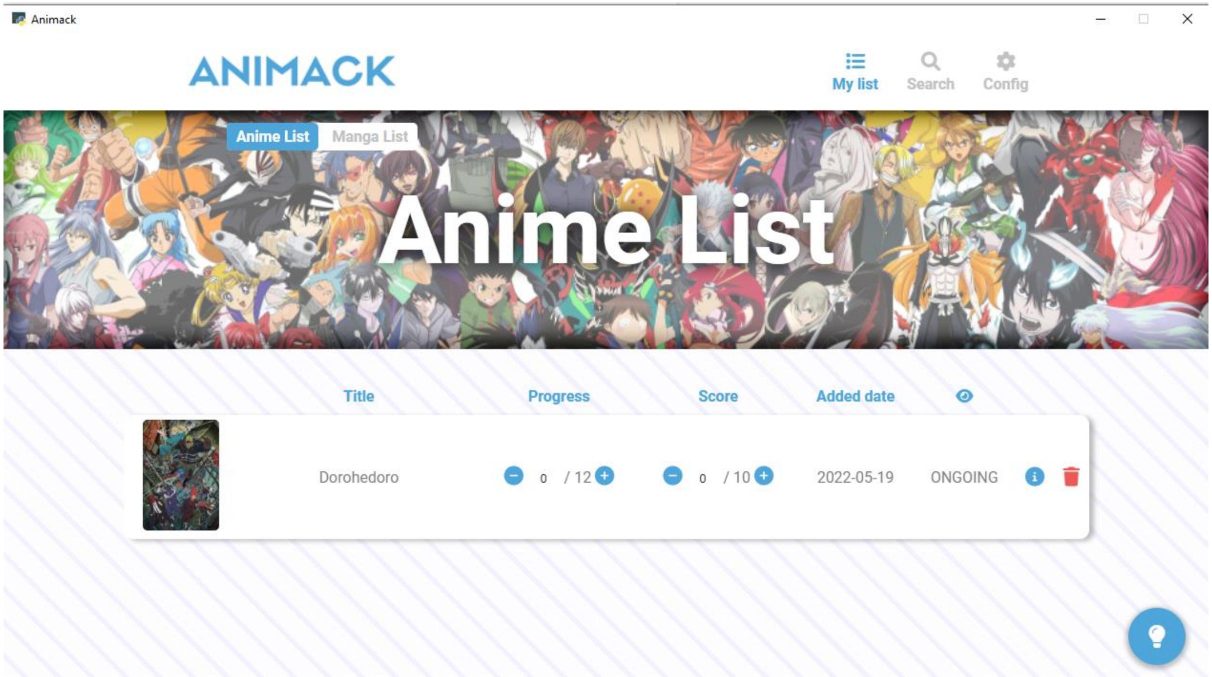
This action will delete your list **permanently**. Proceed?



Seems pretty empty here... Try adding some content.

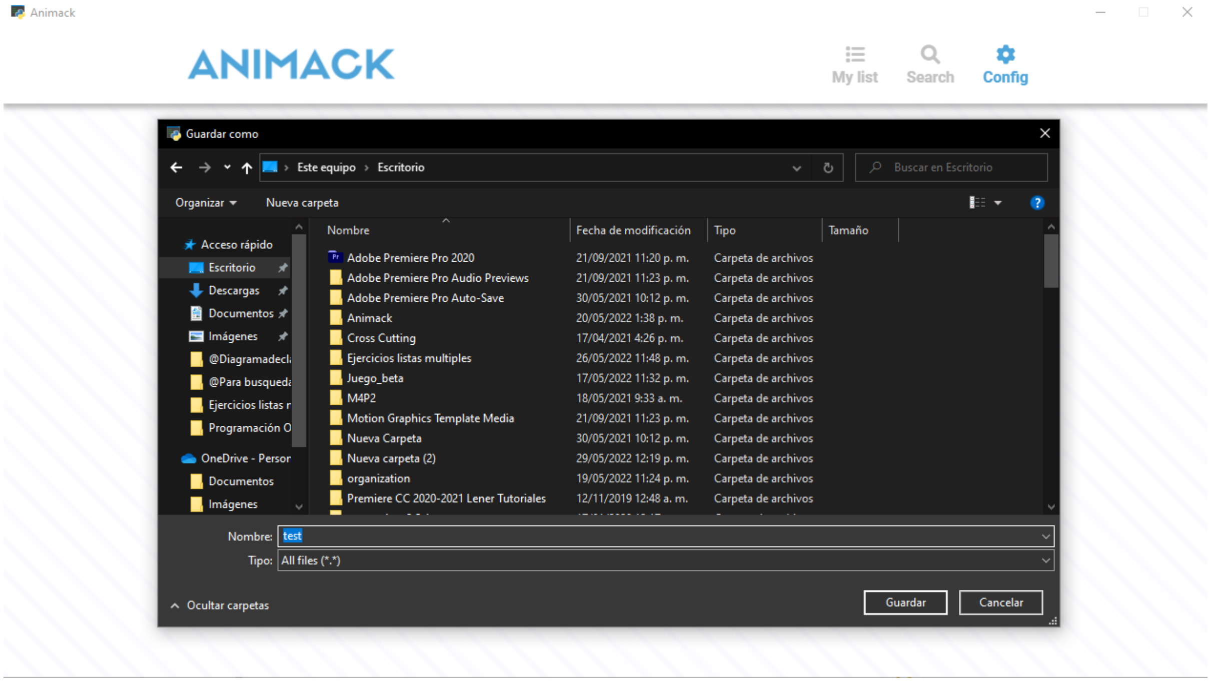
The list is effectively empty. This confirms that the delete button works perfectly. It is going to proceed to test the import button. For running this test, it is going to be used an external file called “lista_secreta” that contains “Dorohedoro”. It is going to be confirmed if this content appears in the anime list once imported the file.





Title	Progress	Score	Added date	
Dorohedoro	0 / 12	0 / 10	2022-05-19	ONGOING

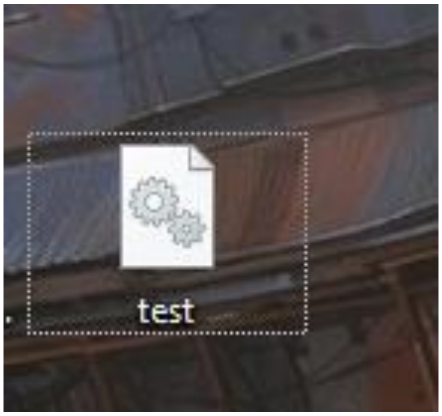
The list was successfully imported and the results can be seen in the list windows. Now it is going to proceed to test the export button. For this test it is going to be exported the current list as "test" to the desktop:



Nombre	Fecha de modificación	Tipo	Tamaño
Adobe Premiere Pro 2020	21/09/2021 11:20 p. m.	Carpeta de archivos	
Adobe Premiere Pro Audio Previews	21/09/2021 11:23 p. m.	Carpeta de archivos	
Adobe Premiere Pro Auto-Save	30/05/2021 10:12 p. m.	Carpeta de archivos	
Animack	20/05/2022 1:38 p. m.	Carpeta de archivos	
Cross Cutting	17/04/2021 4:26 p. m.	Carpeta de archivos	
Ejercicios listas multiples	26/05/2022 11:48 p. m.	Carpeta de archivos	
Juego_beta	17/05/2022 11:32 p. m.	Carpeta de archivos	
M4P2	18/05/2021 9:33 a. m.	Carpeta de archivos	
Motion Graphics Template Media	21/09/2021 11:23 p. m.	Carpeta de archivos	
Nueva Carpeta	30/05/2021 10:12 p. m.	Carpeta de archivos	
Nueva carpeta (2)	29/05/2022 12:19 p. m.	Carpeta de archivos	
organization	19/05/2022 11:24 p. m.	Carpeta de archivos	
Premiere CC 2020-2021 Lener Tutoriales	12/11/2019 12:48 a. m.	Carpeta de archivos	

Nombre: test

Tipo: All files (*.*)



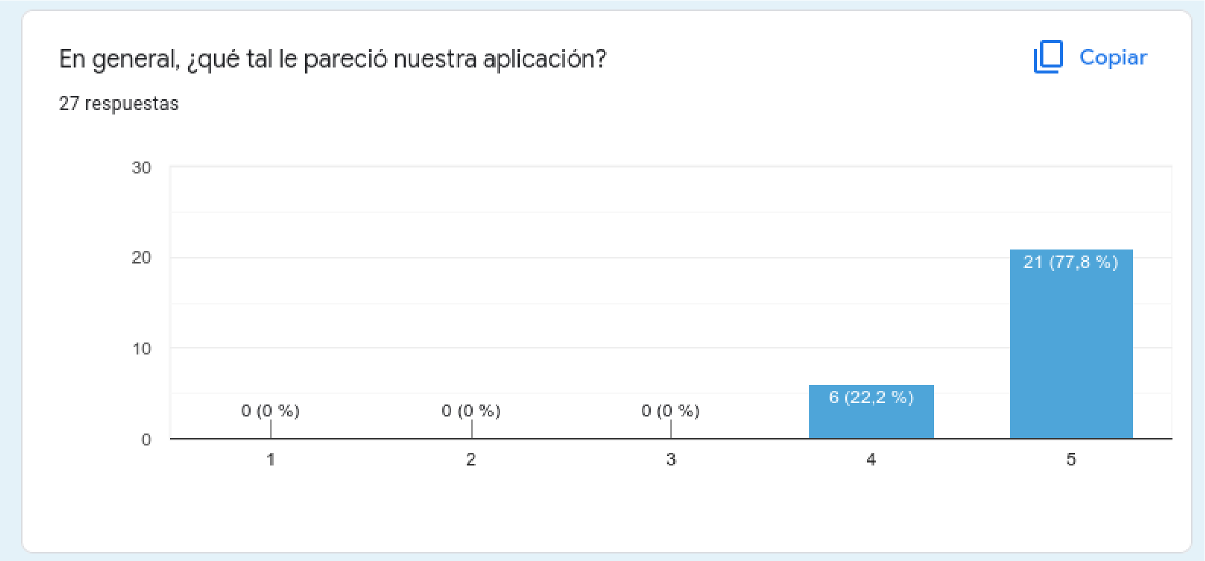
The file was successfully created on the desktop. This confirms that the export button works.

SUMMARY

FUNCTIONALITY	¿DOES IT WORKS?
Search contents	yes
Add contents to the list	yes
Remove content from the list	yes
Change progress	yes
Change Score	yes
Preview contents	yes
Get recommendations	yes
Activate adult filter	yes
Import list	yes
Export list	yes
Delete list	yes

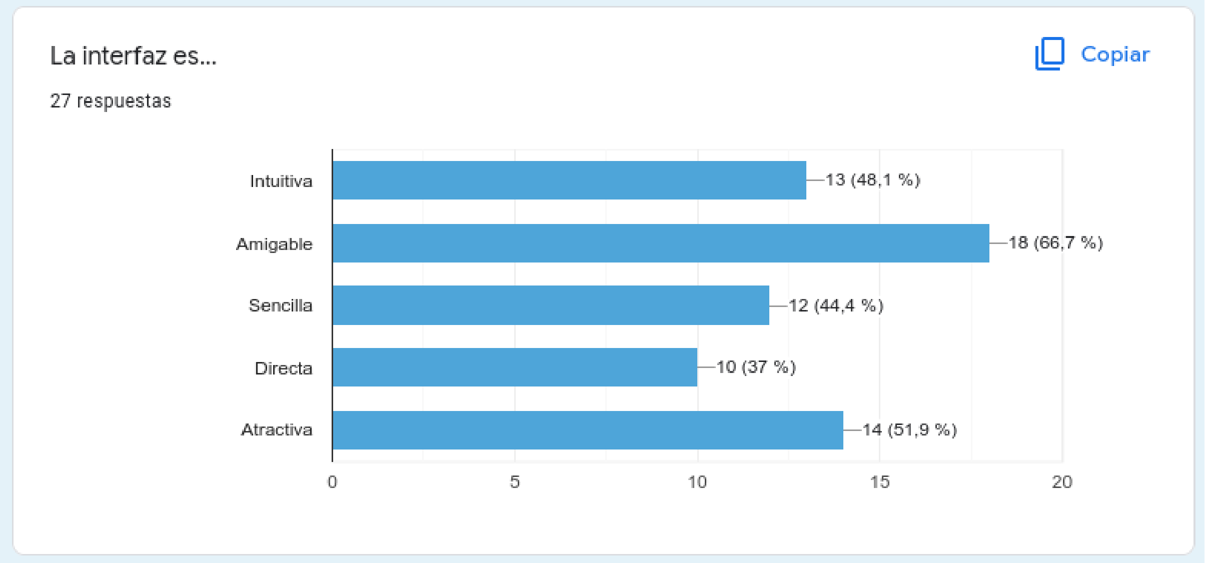
X. USABILITY SURVEY

After the app development was completed, a short survey to know the satisfaction of people using Animack was made. The results are shown in the following images:



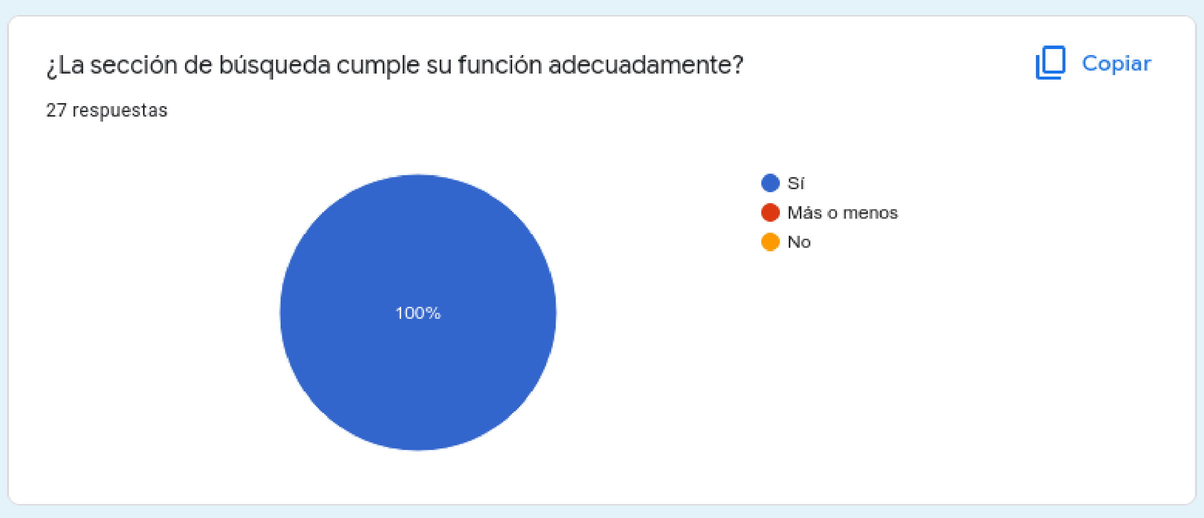
En general, ¿qué tal le pareció nuestra aplicación?

27 respuestas



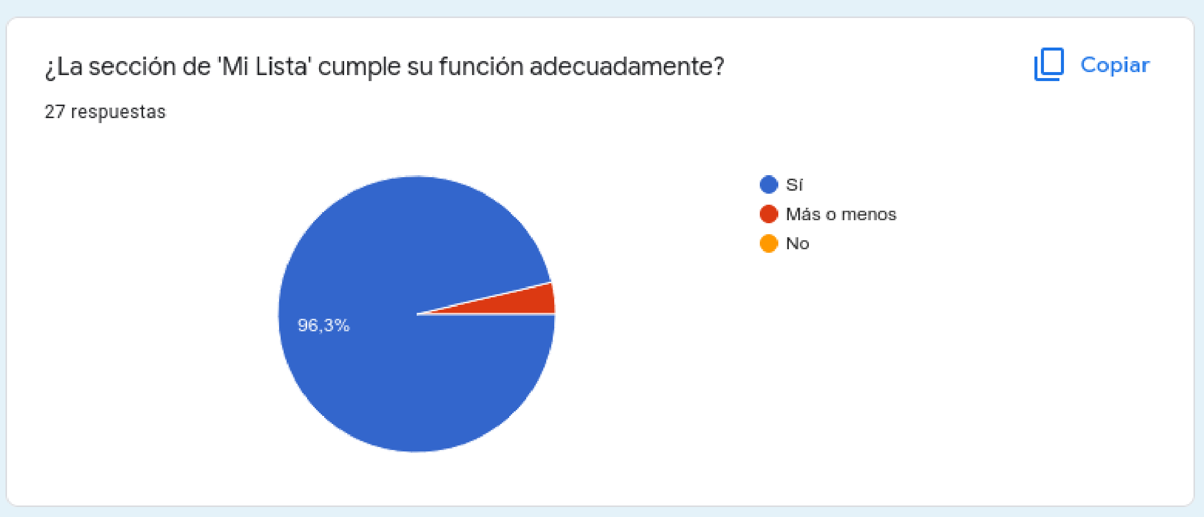
La interfaz es...

27 respuestas



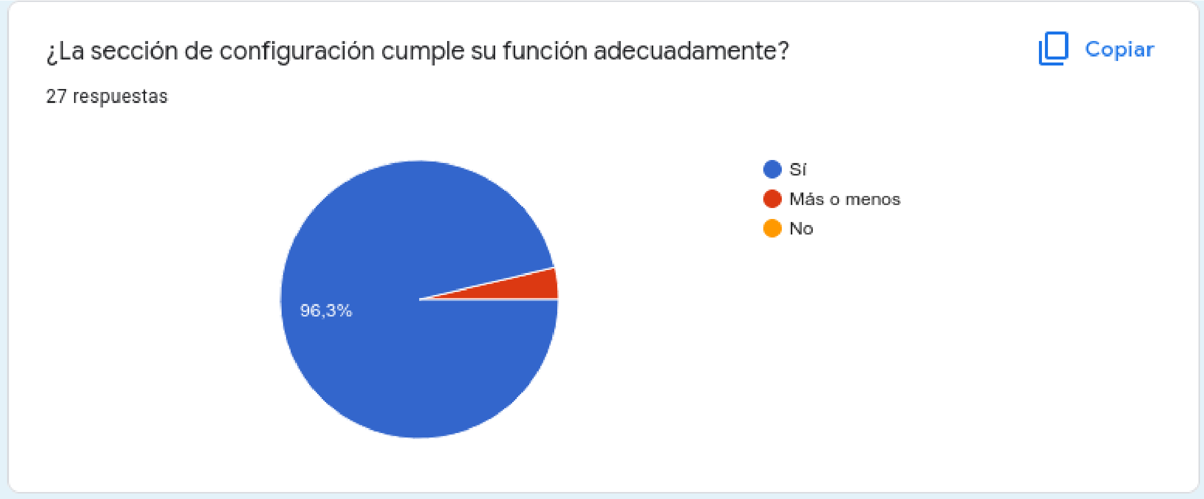
¿La sección de búsqueda cumple su función adecuadamente?

27 respuestas



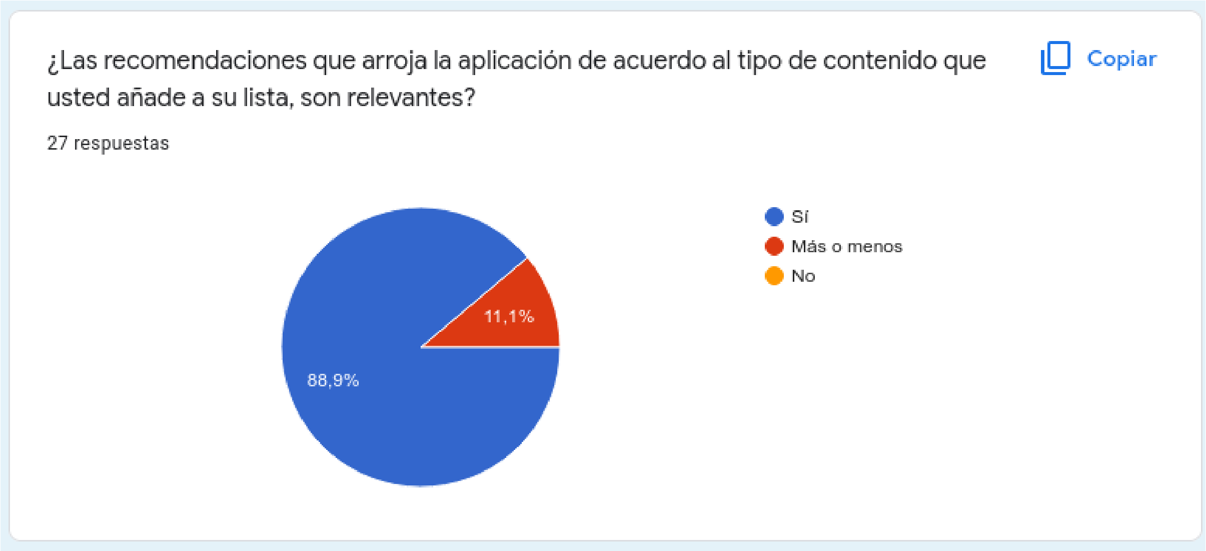
¿La sección de 'Mi Lista' cumple su función adecuadamente?

27 respuestas



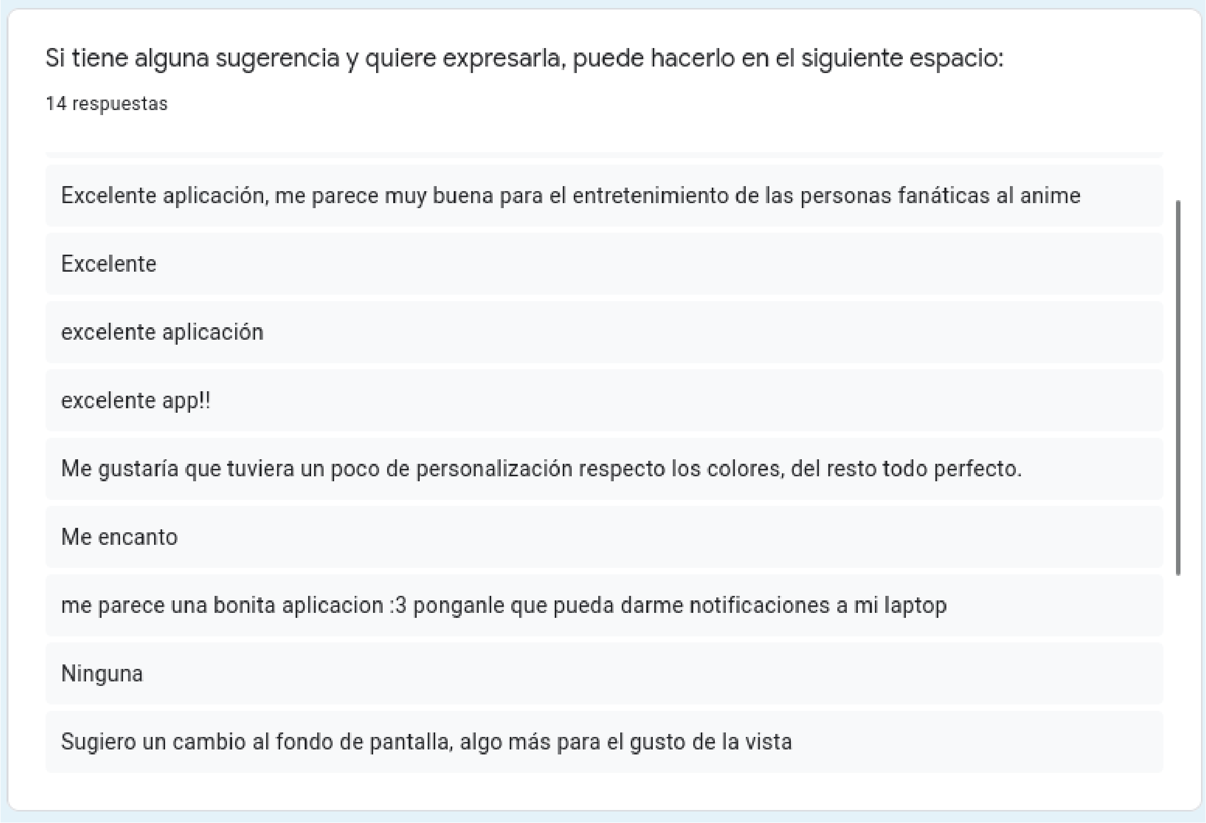
¿La sección de configuración cumple su función adecuadamente?

27 respuestas



¿Las recomendaciones que arroja la aplicación de acuerdo al tipo de contenido que usted añade a su lista, son relevantes?

27 respuestas



Si tiene alguna sugerencia y quiere expresarla, puede hacerlo en el siguiente espacio:

14 respuestas

Excelente aplicación, me parece muy buena para el entretenimiento de las personas fanáticas al anime

Excelente

excelente aplicación

excelente app!!

Me gustaría que tuviera un poco de personalización respecto los colores, del resto todo perfecto.

Me encanto

me parece una bonita aplicacion :3 ponganle que pueda darme notificaciones a mi laptop

Ninguna

Sugiero un cambio al fondo de pantalla, algo más para el gusto de la vista

XI. COMPLIANCE LOG

ACTIVITY	DATE	ACHIEVED?
PROJECT FEEDBACK	April 6, 2022	YES
PRELIMINARY PROJECT	April 18, 2022	YES
GITHUB REPOSITORY UPDATE	April 21, 2022	YES
GITHUB REPOSITORY UPDATE	May 4, 2022	YES
GITHUB REPOSITORY UPDATE	May 11, 2022	YES
GITHUB REPOSITORY UPDATE	May 18, 2022	YES
DEMO PRESENTATION	May 26, 2022	YES
FINAL APP RELEASE	May 30, 2022	YES
FINAL PROJECT DOCUMENT	June 1, 2022	YES
APP PRESENTATION	June 1, 2022	YES

XII. CONCLUSIONS

Main problems faced and their solutions:

1. **Problem:** Searching for an anime and manga database.

   Solution: was solved by investigating throughout the internet, recopilating options, reading many websites' API terms, comparing the options, and selecting the most suitable one.

2. **Problem:** Route management for the import and export functionalities.

   Solution: was solved using the Os and Shutil python libraries.

3. **Problem:** saving the user preferences and actions in the application.

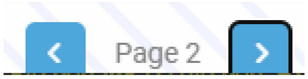
   Solution: for the preferences, a .cfg file was created, and for saving the user actions in their lists, a database was created using sqlite3 python library.

Learned Lessons:

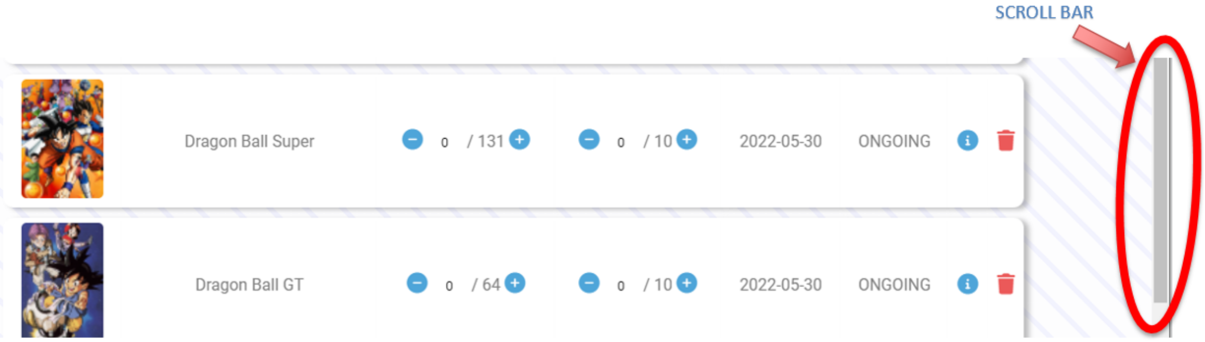
- Communication and organization are essential for reaching success, achieving big projects, avoiding setbacks, and increasing performance.
- Making a multilanguage functional app requires commitment.
- UML and pre-designing facilitate and help to clary how the implementation of the solution will be at the moment of start coding.

Possibles improvements:

In the search page, the user can pass different pages to see more results, as they sometimes do not fit in just one window.



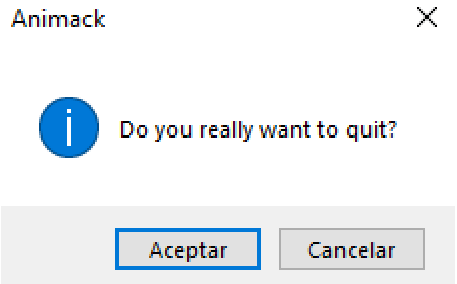
The same happens in the list, but in this case, the user does not pass through pages but scrolls:



Dragon Ball Super	0 / 131	0 / 10	2022-05-30	ONGOING
Dragon Ball GT	0 / 64	0 / 10	2022-05-30	ONGOING

It might have been better to keep just one format.

In windows, for some reason, in the exit confirmation window, the message is in English but the options are in Spanish:


Animack

Do you really want to quit?

It would be better to keep the language format to just English for this case.

XIII. REFERENCES

AniList API (Version 2). (2013). [AniList API V2 GraphQL Documentation]. AniList. https://anilist.gitbook.io/anilist-apiv2-docs/

Tanaka, S. (Executive Producer). (2016-?). Re: Zero, Starting Life in Another World [TV series]. Kadokawa, Media Factory, White Fox.