

AJ RECORDS: Terminal App

Created by Angela Johnson, October 2021 | [Link to Github](#)

1. Install

Steps to install app Dependencies required System/hardware requirements

2. Statement of Purpose & Scope

Overview: Endless lockdowns getting in the way of all your record-buying dreams? Well look no further! AJ Records is a simple CLI app designed to simulate an in-store record-buying experience. Users can visit the store and look through hundreds of records, narrow their search by searching for whatever genre/album/price/artist they want, add them to their cart and go and check out. Don't want to buy anything? Try your luck with the blonde at the counter (disclaimer: you will be disappointed).

Why? As a vinyl-obsessed audiophile myself, this app hits close to home. My uncles opened up record stores in the 90s in LA, while my dad opened up one in Brisbane. I've grown up surrounded by music and pop culture, having worked at the store for a long time. In lockdown, I miss that feeling of wandering around a record store, listening to music, and endlessly digging through to find a good title. So essentially, the app looks to solve the problem of endless lockdown boredom in the MOST exciting way possible -- a terminal app. 😊 As this is my Ruby debut, I wanted to ensure I was building something that was intriguing to me personally, so I could stay engaged while learning as much as I could in the process.

Target audience With the recent resurgence of vinyl among the younger generation, this app is aimed at all ages. Given its record library of just shy of 500 different titles, those with a keen interest in music or pop culture will likely be most interested.

Usage AJ Records is very simple to navigate. The user can enter the store, have a look around, go up to the counter and have a chat to the girl there, or go and look at the vinyl. Once they are in the vinyl section, they are able to choose between searching by Genre, Artist, Album or Price. The first three involve typing any number of keywords, and the program will return all of the records that contain those keywords according to the respective search category. Or, if the user is keeping to a budget, they can search by price, which will allow them to type a maximum price they are willing to pay, and the program will only display records less than or equal to that value. They can review their cart, make changes, or go up to the counter and interact with the girl. Once at the counter, they can choose to pay, exit, look around again, or ask her out on a date. Future updates to the game will include a Memorabilia section, Games section and a Listening Station (where users can listen to music!).

3. Features

F1: Store Navigation

- The user is welcomed to the store by a banner bearing the store name, AJ Records, as well as some introductory text. At this point, they can decide whether they'd like to enter or not.
- If they do enter, they are shown a Store Navigation menu that allows them to choose between 4 options:

1. Cruise over to the vinyl (includes feature 2)
 2. Review what's in your cart (includes feature 3)
 3. Head over to the counter (includes feature 4)
 4. Turn around and leave (exit)
- This Store Navigation module's menu functions as the convergence point for the entire program. It is able to be returned to at various places in the program, based on the user's decisions (i.e. once a user has finished searching for records, or if they go up to the counter then change their mind, and so on).

F2: Product search filter

- If the user has chosen option 1 from the Store Navigation menu, they are presented with yet another multiple choice menu, that gives them the option of sorting the wide range of records AJ records has in stock via the following tags:
 1. By genre
 2. By artist
 3. By album
 4. By price
 5. Go back (takes them back to Store Navigation menu)

Products/stock

- The store's stock is located within a json file (data/vinyl.json). The data is an array of 498 hashes (i.e., records), that each contain the following keys:
 - Catalogue number
 - Price
 - Year
 - Album
 - Artist
 - Genre
 - Subgenre
- All include their own unique value data pertaining to each key (e.g. The Beatles, Revolver, etc).

Search by genre/artist/album

- If the user decides to filter the products by either genre, artist or album, they are prompted to input text of their choosing into the program.
- When the user searches for a particular keyword, their input is manipulated so it matches the format of the json file and can be compared effectively. Then, the program loops through the data and prints out any matches that it finds. For example:
 - *Genre search:* 'rock' would print out anything with Genre: Rock, or Subgenre: Rockabilly, Psychedelic Rock, Rock & Roll, etc.
 - *Artist search:* 'angela johnson' would return no results & display a message to the user with some hints on what/how to search.
 - *Album search:* 'bad' would print out Albert King - Born Under a Bad Sign & Michael Jackson - Bad

Search by maximum price

If the user is on a budget, they can choose to search by typing in a maximum amount they are willing to pay. Similarly to the above, the program will loop through the vinyl.json data and print out any records that are equal to or less than the user input. If the input is invalid, it will display a message and a helpful hint for the user and allow them to try again.

F3: Cart

Add to cart

- When the product search is complete and successfully returns displays a list of matching records to the user's screen, the user is asked whether they would like to add any items to their cart, or not.
- If they choose to add items, they will be prompted to type the respective catalogue number of the record that they want. If the number is valid, it will be added to the cart. If it is not, an error and hint message will display and they can try again.

Review cart

From the Store Navigation menu, if the user selects option 2, they can review all the items in their cart. It will display their catalogue numbers, artist name, album name and price. If their cart is empty, however, it will just return to the Store Navigation menu with a message letting them know their cart is empty. While reviewing their cart, they are asked if they would like to remove anything.

Remove from cart

If they choose to remove something, they can type the respective catalogue number to remove the item. If the number is valid, it will be removed from the cart. If it is not, an error and hint message will display and they can try again if they want, or if not, they can return to the Store Navigation menu.

F4: Checkout

- From the Store Navigation menu, if the user selects option 3, they will be greeted by the girl at the counter. A menu will be displayed, and the user can select between multiple options:
 1. I'd like to pay for these
 2. Errr actually, on second thought. I might have a second look around
 3. (NOTE: only displays if cart is empty) "I changed my mind and I'm just gonna head out."
 4. (NOTE: only displays if cart is not empty) "I changed my mind, and I'm going to head out. Can I just leave all these records with you?"
 5. Actually... I wanted to see if you were busy later....want to grab a drink?"

Pay for items

- When the user opts to pay for their items (option 1), two scenarios can happen. If their cart is empty, the girl will be confused by the user and tell the user to come back when they have some records. If their cart is not empty, a table of the user's order summary (courtesy of the terminal-table gem) will print out on the screen. The girl will ask the user to confirm. If they aren't ready to pay, a message will be displayed from the girl and the user will return to the Store Navigation menu. If they choose to pay, payment confirmation messages will print out, the girl will thank you and the program will exit.

Second look around

- If the user chooses option 2, a message from the girl will display and the user will be redirected back to the Store Navigation menu

Changed mind, want to leave

- Only one of Option 3 and 4 is displayed on the screen, 3 if the cart is empty, and 4 if the cart is not empty.
- If the option 3 is selected, a friendly message from the girl will print and the program will exit.
- If 4 is chosen, however, the girl will be passive aggressively frustrated with you because she has to sort your records. A message will display and the program will exit.

Ask on a date

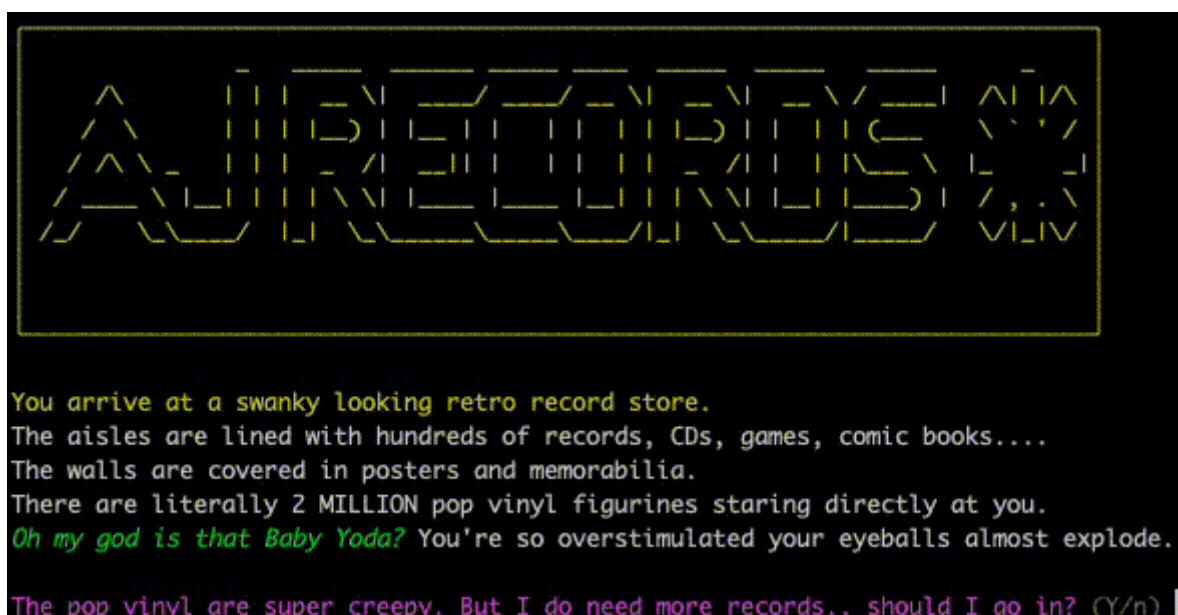
- If the user decides to take the leap and ask the girl out (option 5), any 1 of 4 separate rejection messages will print out, and the user can then choose another counter menu option.

4. User Interaction & Experience

How the user finds out how to interact with each feature

- The user interaction in AJ Records is purposefully straightforward, in order to mimic a real in-store experience. The user will find out how to interact with each of the four features after being directly prompted by the program at various points by displaying multiple choice menus (as described through section 3. Features).

Accessing Feature 1: Store Navigation:



Accessing Feature 2: Product Search Filter:

```
Where should I go? (Press ↑/↓ arrow to move and Enter to select)
> Cruise over to the VINYL
  Review what's in your CART
  Head over to the COUNTER
  Turn around and leave. (EXIT)
```

Accessing Feature 3: Cart:

```
Where should I go? (Press ↑/↓ arrow to move and Enter to select)
> Cruise over to the VINYL
  Review what's in your CART
  Head over to the COUNTER
  Drop your records to the girl at the counter and leave (EXIT)
```

Accessing Feature 4: Checkout:

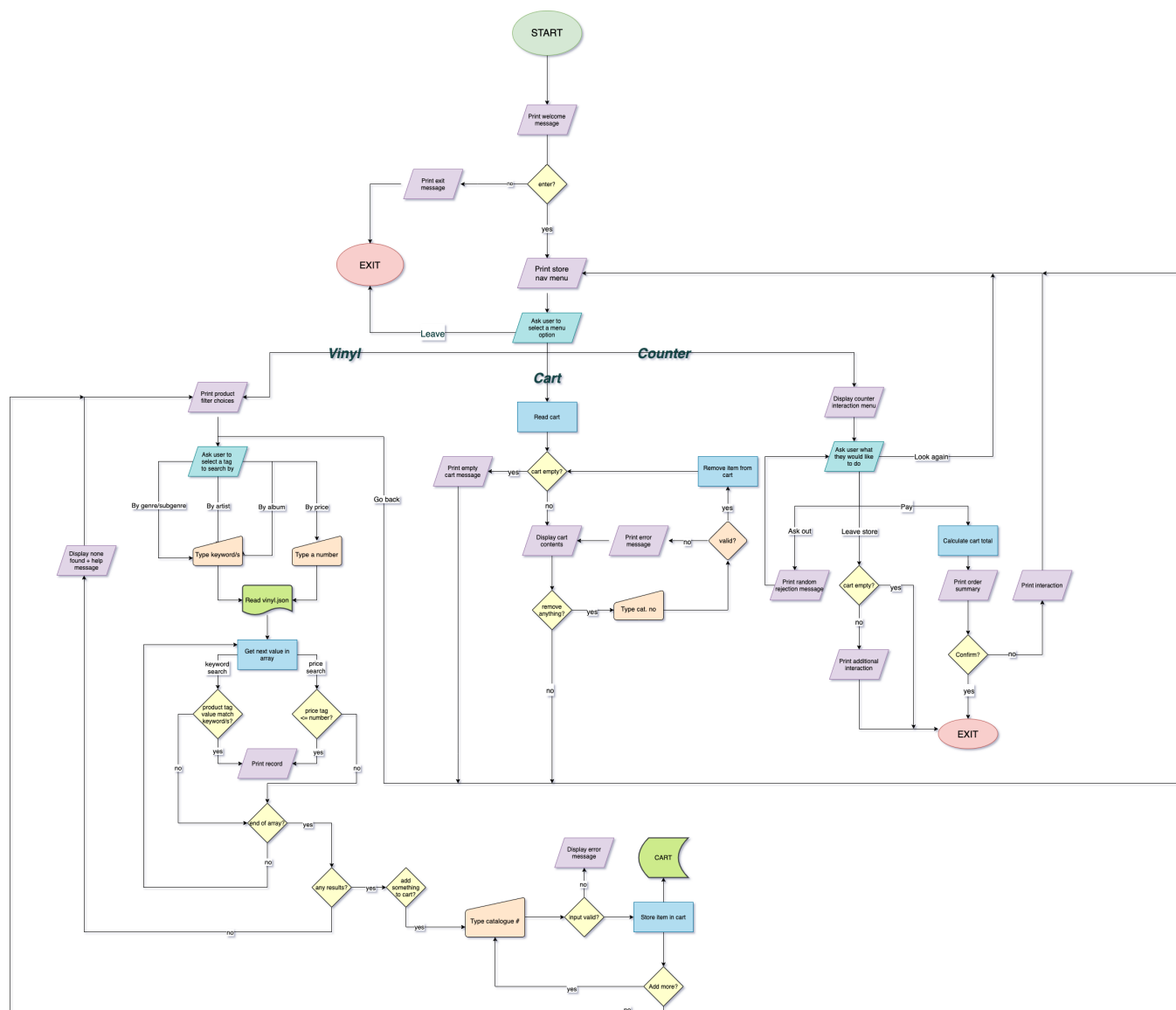
```
Where should I go?
> Cruise over to the VINYL
  Review what's in your CART
  Head over to the COUNTER
  Turn around and leave. (EXIT)
```

2. How the user interacts with features

Feature 1 Interactions:

3. How errors are handled by app and displayed to user

Control Flow Diagram



Full size diagram located here:

[./docs/ControlFlowDiagram.png](#)

Testing

Manual testing was performed throughout the development process. The testing spreadsheet can be found here:

[./docs/ManualTests.pdf](#)