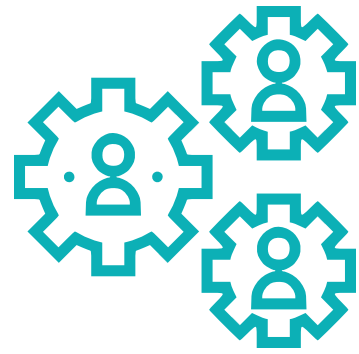


# T1A3 TERMINAL APP ASSIGNMENT

JORDAN ATTFIELD

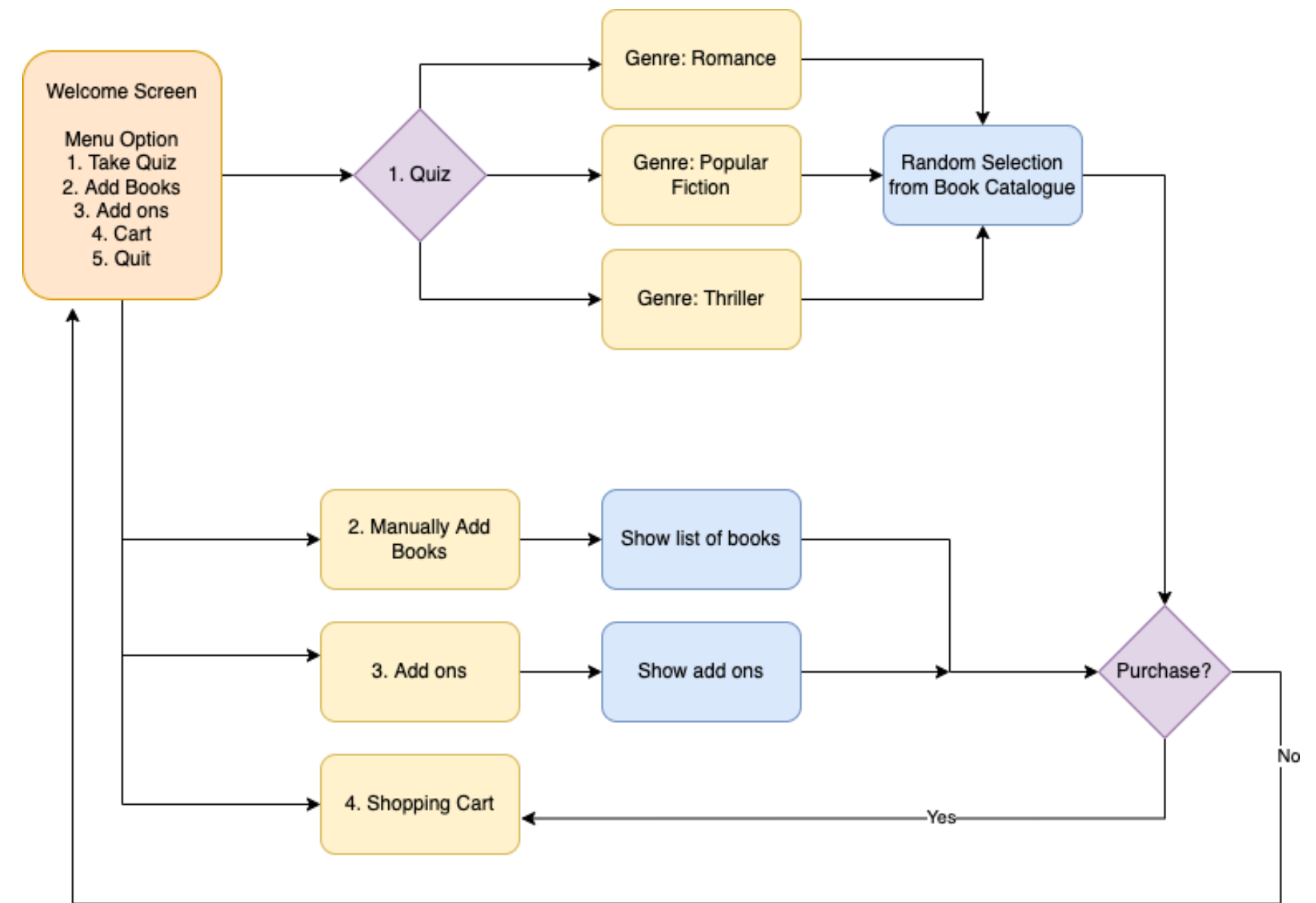




# APP OVERVIEW

My terminal application is called 'Date with a Book'. It's an app that allows users to take a quiz and based on their outcome of the quiz, a book will be randomly selected for them to purchase.

- Users will also be able to view entire catalogue of books and purchase any book they choose
- Add ons option allows users to add additional products to their cart





# Welcome Menu

```
Terminal_App — Python app.py — 80x24

Welcome to Blind Date with a Book!

Please begin by completing our quiz to determine your reading preferences.
Alternatively you can select a book yourself.

1. Take Quiz
2. I'd like to choose a book myself
3. Extra goodies to go with your new books
4. View Cart
5. Quit
Please select your option (1 - 5): █
```

- Simple Welcome menu that allows users to input options:
  - Take quiz
  - View Catalogue and add books manually
  - Add on extra products
  - View shopping cart
  - Goodbye message

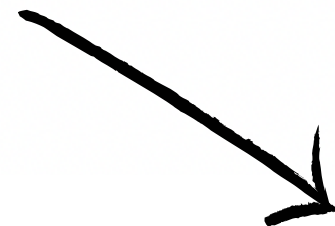
# Features: Quiz & Book Selector



```
Terminal_App — Python app.py — 80x24
What kind of movie is your favourite?
(a) Romance – The Notebook is a favourite!
(b) I honestly couldn't pick – I love a broad range of everything!
(c) Suspenseful movies with creepy twists

b
What's your ideal setting?
(a) Somewhere warm and cozy where I can curl up with a book
(b) Somewhere that's just asking to be explored!
(c) Somewhere creepy and secluded

c
Who would you like to have dinner with?
(a) Augustus Waters
(b) Hermione Granger
(c) Sherlock Holmes
```



Looks like you're in the mood for a Thriller book!  
Please see the book we have selected for you below:

The House Across The Lake by Riley Sager

Would you like to purchase? All new releases are \$18

Please confirm your purchase by typing 'yes' or 'no'



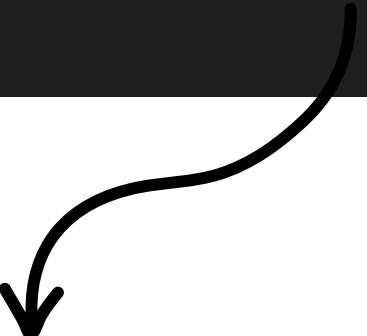
- 3 simple questions that determine the genre preference of the user. A score is tallied and is used to determine which category they prefer from 3 genres; romance, popular fiction or thriller.
- Once genre preference is determined, a random book selection is made from the book catalogue and is presented to the user. They can then choose if they wish to purchase.

# Features: Quiz

## Logic & Code

- Quiz uses OOP to create a Question class and iterates through list of questions.
- If/else statements keep running tally of score, then determine which function should be called.
- Random module selects a random book from the book catalogue dictionary.

```
def determine_genre(score):  
    if score <= 4:  
        romance()  
    elif (score >= 5) and (score <= 7):  
        popular_fiction()  
    elif (score >= 8) and (score <= 9):  
        thriller()
```



```
def popular_fiction():  
    os.system('clear')  
    print(  
        "Looks like you're in the mood for a Popular Fiction book!\nPlease see the book we have selected for you below:\n")  
    chosen_book = random.choice(book_catalogue["fiction"])  
    print(chosen_book)  
    print("\nWould you like to purchase? All new releases are $18")  
    checkout(chosen_book, cart)  
    return chosen_book
```



# Features: View Catalogue



Here is our selection of books:

The Seven Husbands of Evelyn Hugo by Taylor Jenkins Reid  
Anxious People by Fredrik Backman  
The Four Winds by Kristin Hannah  
The Lies I Tell by Julie Clark  
Things We Do In The Dark by Jennifer Hillier  
The House Across The Lake by Riley Sager  
The Family Upstairs by Lisa Jewell  
People We Meet on Vacation by Emily Henry  
The Unhoneymooners by Christina Lauren  
Every Summer After by Carley Fortune  
One Italian Summer by Rebecca Serle

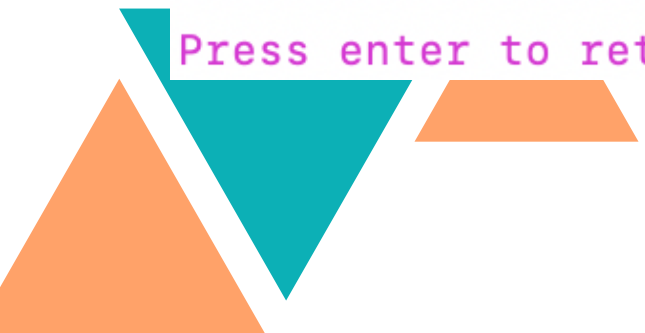
Please type the name of the book you'd like to purchase: People We Meet on Vacation by Emily Henry

Please confirm your purchase by typing 'yes' or 'no'  
yes

Here is your shopping cart:  
['People We Meet on Vacation by Emily Henry']

Press enter to return to the main menu

- Users can select to view the entire book catalogue and purchase any book they choose.
- Prompt asks user to input book selection
- Check user input is actually in catalogue, and if so allows book to be added to cart.



# Features: View Catalogue

## Logic & Code

```
# Option from main menu that lists entire book catalogue and gives option to add book
def add_book():
    print("Here is our selection of books: \n")
    flatlist = reduce(lambda a,b:a+b, book_catalogue.values())
    print(*flatlist, sep="\n")
    while True:
        chosen_book = input("\nPlease type the name of the book you'd like to add: ")
        book_choice = False
        for i in flatlist:
            if chosen_book == i:
                book_choice = True
        if book_choice:
            checkout(chosen_book, cart)
            break
        else:
            print("We don't have that book in our selection. Please try again.")
    return chosen_book
```

- Book catalogue stored in a dictionary with nested lists for each genre.
- Keys = genre
- Values = list of books
- Needed to flatten dictionary using a python module to search if item was in the list

# Features: Add ons

Here's a selection of some of our goodies you can purchase:

Penfolds Shiraz,  
Enjoy a glass of red wine while you dive into your book, \$30.0  
Haighs Chocolates Small Hamper Box,  
Indulge in a selection of Haighs most popular chocolates, \$15.0  
Book Lovers Candle,  
Enjoy the aromas of dusty tomes and leather bound books while you  
Byron Bay Gourmet Drinking Hot Chocolate,  
Sip a mug of rich, velvety hot chocolate, \$8

Please enter the product you'd like to purchase:

- Users can view a selection of add ons to purchase along with their book
- I used OOP to create an Addons class. Each Addon object consists of the product name, description and price.
- These items are displayed to the user, and then asks the user to input the product they wish to purchase. This input is then checked against the list of objects and added to cart.



# Features: View Cart

Here is your shopping cart:

People We Meet on Vacation by Emily Henry  
The House Across The Lake by Riley Sager

Press enter to return to the main menu

- The shopping cart keeps track of any purchases made either as a result of the quiz or from manually adding books from the catalogue or the Add on selection.
- Users can view their cart at any time from the main menu.

# Wrap-up

---



## Challenges

- Getting everything done in very short time frame!
- Encountering so many coding issues - I originally wanted a login/registration feature but I was spending too much time trying to figure it out - so I scrapped it

## Favourite Parts

- Feel much more confident with Python now. Encountering and working through issues is the best way to learn!

**Thank You!**