

BookShelf App

Terminal App by Morgan Rohan

Scope

What is it?

An app that you can think of as a virtual bookshelf, storing books you have read or want to read.

Also be able to provide recommendations based on the results of a quiz.

Who is it for?

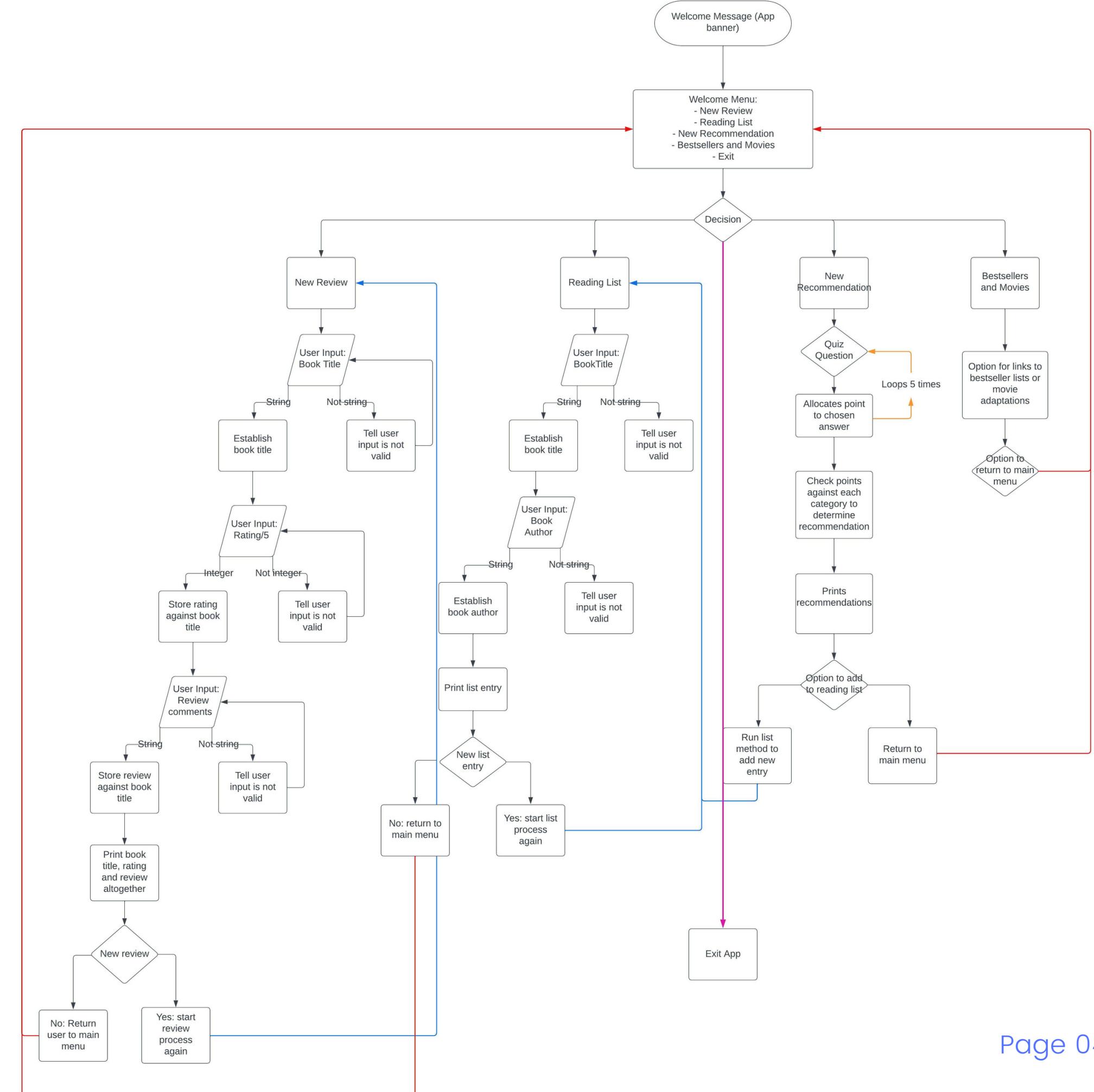
- The book lover who likes to keep track of everything they've read or got coming up
- The new reader who needs a place to rate some reads and get new recommendations
- The casual reader who doesn't want to waste their time re-reading or getting recommendations unsuited to them



Features

- Review Feature
 - The user will be able to input a review for a book
 - Will be prompted to enter a book title, rating and any comments they want to make
 - Review will print to screen
- Reading List
 - The user will be able to add titles to a stored list
- Recommendation Quiz
 - The user can take a quiz and based off the answers will be given a recommendation
- Other Media
 - Includes links to some bestseller lists and information about upcoming book adaptations

Control Flow Chart



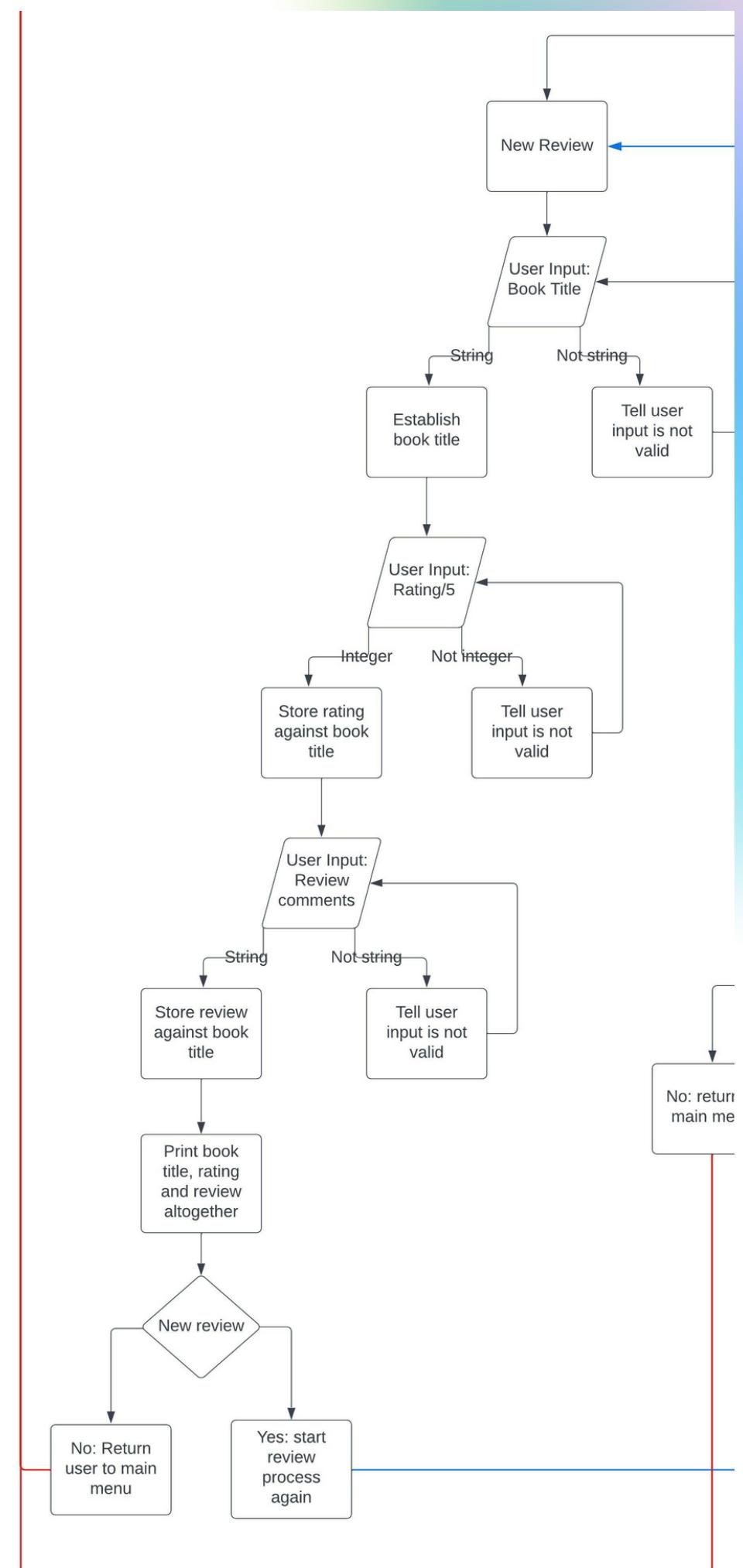
Feature 1

Review

- User selects new review from main menu
- User prompted to enter book title, choose a rating and write any comments
- Review is printed to the screen
- User asked whether they want to start a new review or go back to main menu

Error handling

- User input is given as correct type
- Yes/no menu at the end of process loops each time



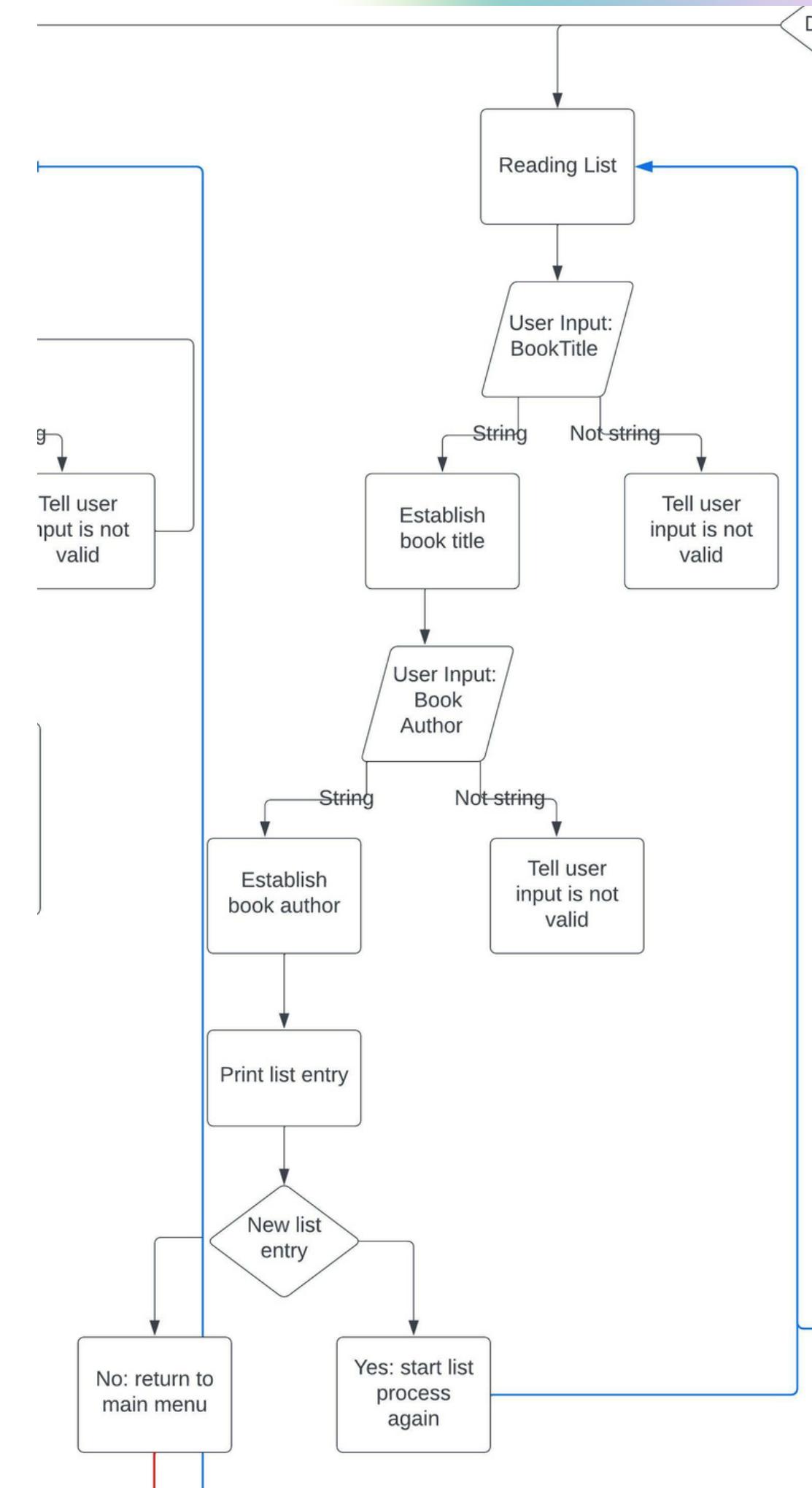
Feature 2

Reading List

- User selects reading list from main menu
- User prompted to enter book title and author
- Entry is printed to screen
- User asked whether they want to make another entry or go back to main menu

Error handling

- User input is given as correct type
- Yes/no menu at the end of process loops each time
- New entry is correctly added to list



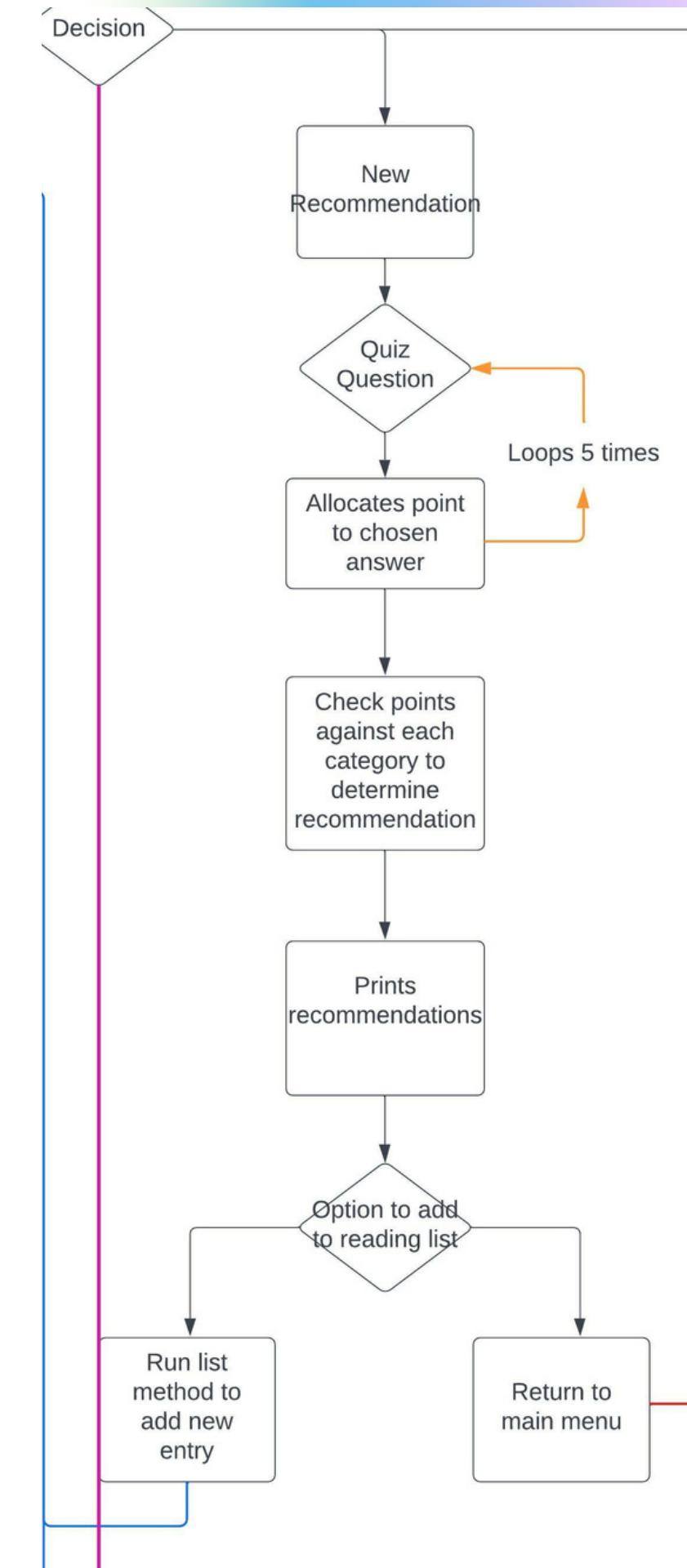
Feature 3

Recommendation Quiz

- User selects recommendation quiz from main menu
- User prompted to answer each question (loops through for 5 questions)
- Final score printed to screen along with message and recommendation list
- User asked whether they want to get a new recommendation, go to reading list or go back to main menu

Error handling

- User input is given as correct type
- Sub menu at the end of process loops each time



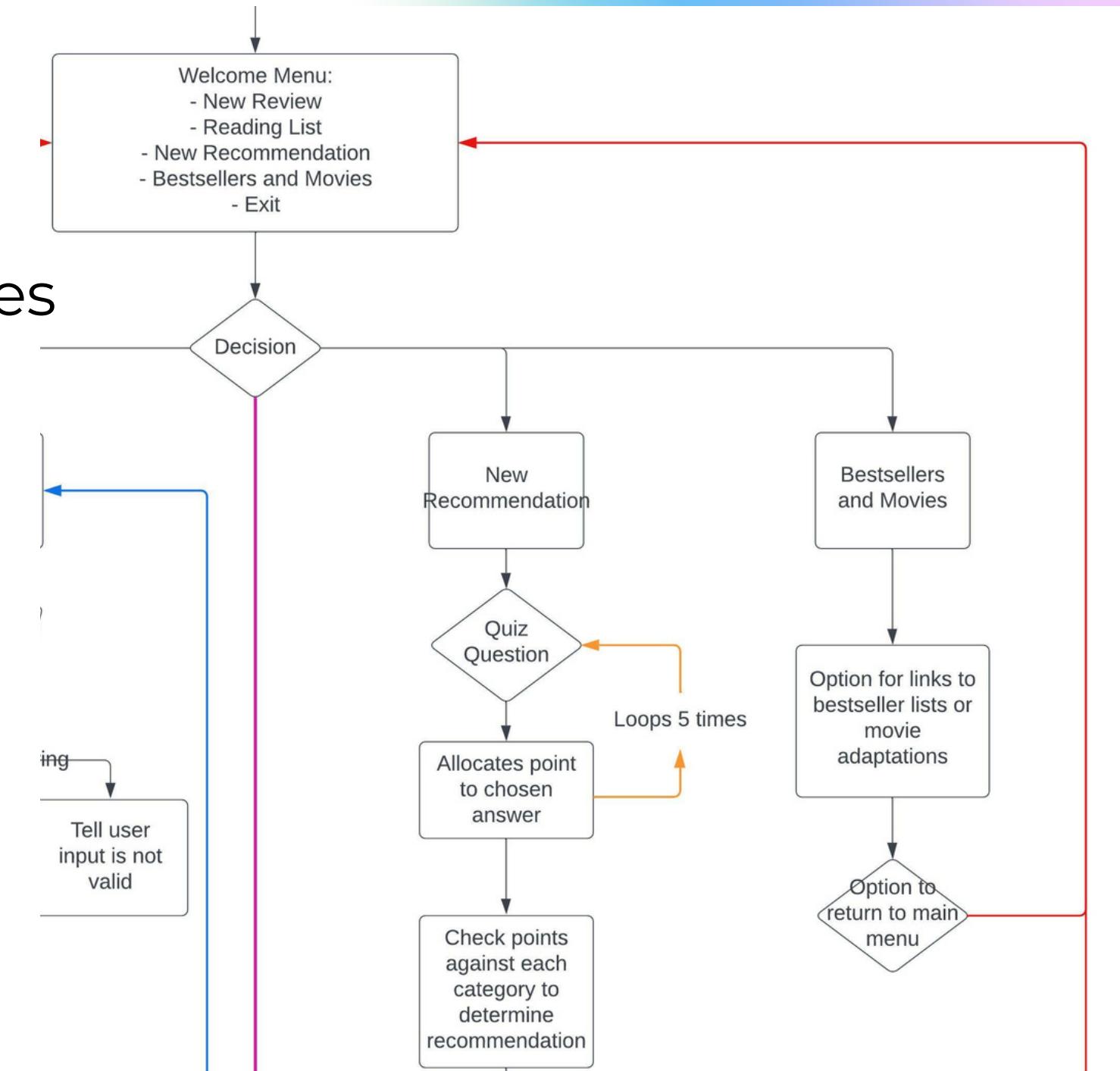
Feature 4

Other Media

- User selects other media from main menu
- User provided with links to different online resources for more up to date bestseller lists and screen adaptations
- User returned to main menu

Error handling

- Sub menu at the end of process loops each time



Implementation Plan (Trello)

The image shows a Trello board titled "Terminal App" with the following structure:

- General To Do** (Column 1):
 - Create opening message
 - Create menu
 - Create exit option
 - Plan control flow diagram for overall structure
 - Create HELP documentation (download info etc)
 - Install colorize gem to customise colours for text
 - Install TTY prompt to help with menu selection
- Feature 1 - To Do** (Column 2):
 - Review class - take book title input
 - Review class - confirm valid input (if not print error)
 - Review class - create way of storing review next to book title
 - Review class - check if inputs are valid at each step before moving on
 - Review class - method for choosing star rating
 - Review class - input any review comments
 - Review class - confirm valid input (if not print error message)
 - Review class - print final review (title, star rating and additional comments)
 - Review class - confirm final review
- Feature 2 - To Do** (Column 3):
 - Previous Review class - menu to order previous reviews by A-Z or star rating
 - Previous Review class - option to print review to screen
 - Previous Review class - option to save review to file
 - Previous Review class - if no reviews, prompt to return to previous menu and create new review or get a recommendation to review
 - Previous Review class - return to review menu
 - Previous Review class - option to return to main menu
- Feature 3 - To Do** (Column 4):
 - Recommend class - select genre for recommendation
 - Recommend class - option to get more info about recommendation
 - Recommend class - option to add recommendation to "to be read" list
 - Recommend class - option to exit and get new recommendation
 - Recommend class - prompt if user wants to save list as a file
 - Recommend class - check if list has entries, if no entries print message to add to list
 - Recommend class - save list in a file
- Feature 4 - To Do** (Column 5):
 - Other media class - option to view bestseller list
 - Other media class - option to view recent media adaptations
 - Other media class - option to edit "to be read/watched" list
 - Other media class - take user input and add to "to be read/watched" list
 - Other media class - option to save list in file
 - Other media class - check if list has entries, if no entries print message to add to list
 - Other media class - save list in file

Code for Reviews and Reading List

```
● ● ●  
1 # Book Reviews  
2 # Class: reviews  
3 # Objects: title, rating, review  
4  
5 class Reviews  
6   attr_accessor :title, :rating, :review  
7   def initialize(title, rating, review)  
8     @title = title  
9     @rating = rating  
10    @review = review  
11  end  
12  
13  def to_s  
14    puts "-----"  
15    puts " Your Book Review"  
16    puts "-----"  
17    puts "Book Title: #{@title}"  
18    puts "-----"  
19    puts "Rating: #{@rating}/5"  
20    puts "-----"  
21    puts "Review: #{@review}"  
22  end  
23 end
```

```
● ● ●  
1 # Reading List  
2 # Class: list  
3 # Objects: title, author  
4 class List  
5   attr_accessor :title, :author  
6   def initialize(title, author, books, list)  
7     @title = title  
8     @author = author  
9     @books = books  
10    @list = list  
11  end  
12  
13  def to_s  
14    return "#{@title}: by #{@author}"  
15  end  
16  
17 end
```

Code for Recommendation

```
● ● ●

1 # Recommendation Quiz
2 # Class: Quiz
3 # Objects: prompt, score
4 class Quiz
5   attr_accessor :prompt, :score
6   def initialize(prompt)
7     @prompt = prompt
8     @score = score
9   end
10
11  q1 = "Which setting sounds the best?\n (a) On a planet in an alternate reality, maybe there's spaceships zooming by\n (b) Historical, like Regency Era London, with carriages"
12  q2 = "If you were a main character, who would you want to be?\n (a) A lone gunslinger type, out for retribution\n (b) A Viscount looking to find a wife"
13  q3 = "You have to pick only one:\n (a) A Journey \n (b) Romance"
14  q4 = "Which of these is your favourite colour?\n (a) Black\n (b) Pink"
15  q5 = "Do you prefer:\n (a) An epic story lasting multiple books\n (b) A story that has a happy ending and is all wrapped up in one book"
16
17  $questions = [
18    Quiz.new(q1),
19    Quiz.new(q2),
20    Quiz.new(q3),
21    Quiz.new(q4),
22    Quiz.new(q5)
23  ]
24
25  questions = $questions
26
27  def run_quiz(questions)
28    answer = []
29    score = 0
30    for question in questions
31      puts question.prompt
32      answer = gets.chomp.to_s
33      if answer == "a"
34        score += 1
35      end
36      if answer == "b"
37        score += 0
38      end
39    end
```

Code for Main App



```
1 #Main app document that brings all pages together
2 #Gems required:
3 require "tty-prompt"
4 require "colorize"
5
6 #Local files required:
7 require_relative "./review.rb"
8 require_relative "./recommendationquiz.rb"
9 require_relative "./readinglist.rb"
10
11 #Define a method to get input for new review
12 #Print format of review defined in review class
13 def new_review
14     puts "Your book review details"
15     sleep 1
16     puts "Enter your book title: "
17     title = gets.chomp
18     sleep 1
19     puts "Enter your rating (out of 5): "
20     rating = gets.chomp
21     sleep 1
22     puts "Enter your review: "
23     review = gets.chomp
24     sleep 1
25     new_review = Reviews.new(title, rating, review)
26 end
27
28
```



```
1 #Define a method to open the reading list
2 #Run new list item method from readinglist file
3 #Take user inputs and print result
4 def view_list
5     puts "Your Current Reading List!"
6     puts "Let's add some titles..."
7     sleep 1
8     puts "What is the book title?"
9     title = gets.chomp.to_s
10    puts "What is the book author?"
11    author = gets.chomp.to_s
12    list = [title, author]
13    books = list.push(title, author)
14    new_list = List.new(title, author, books, list)
15 end
16
17 #Define a method to start recommendation quiz
18 #Welcome to the quiz statement and input specification
19 def recommendation_quiz
20     puts "Welcome to the book recommendation quiz!"
21     puts "Please answer the following questions by typing 'a' or 'b'"
22     new_quiz = Quiz.new($questions)
23 end
24
25 $prompt = TTY::Prompt.new
26 def main_menu
27     answer = $prompt.select("Please select an option:", ["New Book Review", "Reading List", "Take Recommendation Quiz", "Exit App"])
28     answer
29 end
```

Code for Main App

```
● ○ ●
1 system "clear"
2 puts "Welcome to the Book App"
3 option = ""
4 while option != "Exit App"
5   option = main_menu
6   case option
7     when "New Book Review"
8       review = new_review
9       puts review.to_s
10    def review_menu
11      answer = $prompt.select("Would you like to add another review?", ["Yes", "No"])
12      answer
13    end
14    while option != "No"
15      option = review_menu
16      case option
17        when "Yes"
18          review = new_review
19          puts review.to_s
20        else
21          next
22        end
23      end
24    when "Reading List"
25      list = view_list
26      puts list.to_s
27    def list_menu
28      answer = $prompt.select("Would you like to add more list items?", ["Yes", "No"])
29      answer
30    end
31    while option != "No"
32      option = list_menu
33      case option
34        when "Yes"
35          books = view_list
36          puts books.to_s
37        else
38          next
39        end
40    end
```

```
● ○ ●
1 when "Take Recommendation Quiz"
2   quiz = recommendation_quiz
3   puts quiz.run_quiz($questions)
4 else
5   puts "Sad to see you go. See you next time!"
6   next
7 end
8 print "Press enter to continue..."
9 gets
10 system "clear"
11 end
```

Challenges

- Logic:
 - Upon writing the code I realised that I needed to go and review some of the content, especially around methods to ensure I was actually writing and invoking them efficiently.
- Time:
 - I did the implementation planning and control flow chart first but ended up going in a slightly different direction, which honestly just wasted some time
 - Had some time taken up by work issues
- The Blank Page Complex
 - Even with the rubric, the actual scope of the app was quite wide and I find that I get caught up in how big it feels.
- Still need to:
 - Implement styling and error handling

Thank You

