

# DANIEL KURZ

T1A3 PRESENTATION

# APPLICATION OVERVIEW

- The application I am developing is a code library. It will supply the user with on demand code snippets to the search enquiry they input. At first it will only have a library of CSS code, but future development will include code snippets from other technologies. The application works by the user first setting up an account. From there they can enter key search words to lookup relevant code snippets relating to the search. The user can find the list of keywords by searching the help menu. Once the user has found what they believe to be useful code, they can save it to their favourites list, or simply print it to the screen.
- The application is populated by the Admin user. The Admin user has full CRUD (Create, Read, Update, Delete) capabilities, as well as also being able to search for code.

# MAIN FEATURES

- - Have 2 types of users, the Admin that can add, remove or edit code. And a General user, that can search for code, and save the code they find most useful.
- - Display code snippets that are relevant to the users search requests
- - Store the information the user selects into a favourites file

# CODE OVERVIEW

- - Login screen that verifies the User and whether they are a General or Admin user
- - Menu screen that allows the user to navigate throughout the app
- - Admin features that allow the admin to add/edit/remove code

# Login Screen

```
puts "Please enter your username"
input_username = gets.chomp
puts "Enter your password"
input_password = gets.chomp
user = list_of_users.find { |user| user.username == input_username }
if user.username == input_username && user.password == input_password
```

# ADMIN MAIN MENU

```
loop do
  prompt = TTY::Prompt.new
  display_options_menu = prompt.select('Please select an option:') do |menu|
    menu.choice 'Add Code'
    menu.choice 'Edit Code'
    menu.choice 'Remove Code'
    menu.choice 'Search'
    menu.choice 'Help'
    menu.choice 'Exit'
  end

  loop do
    case display_options_menu
    when "Add Code"
      puts "Please enter a code Title"
      input_title = gets.chomp
      puts "Please enter a code Description"
      input_description = gets.chomp
      puts "Please enter a code snippet"
      input_snippet = gets.chomp
      code = {title: input_title, description: input_description, code_snippet: input_snippet}
      File.open("code.yml", "a") { |file| file.write(code.to_yaml) }
      break

    when "Remove Code"
      puts "Please enter the title of the code snippet you wish to remove"
      remove_title = gets.chomp

    when "Search"
      prompt = TTY::Prompt.new
      search_options = prompt.select('Please select a category:') do |menu|
        menu.choice 'Image Manipulation'
        menu.choice 'Text Manipulation'
        menu.choice 'Flexbox'
        menu.choice 'Grid'
        menu.choice 'Back'
        menu.choice 'Exit'
      end
    end
  end
end
```

# Search Screen

```

when "Search"
  prompt = TTY::Prompt.new
  search_options = prompt.select('Please select a category:') do |menu|
    menu.choice 'Image Manipulation'
    menu.choice 'Text Manipulation'
    menu.choice 'Flexbox'
    menu.choice 'Grid'
    menu.choice 'Back'
    menu.choice 'Exit'
  end

  case search_options
  when "Image Manipulation"
    system("clear")

    puts "
    puts "
    puts "
    puts "
    puts "
    puts "
    puts "

    image_table = TTY::Table.new(["Description", "Code Snippet"],
    [
      image_height,
      image_width,
      responsive_image,
      image_border,
      opacity
      #image_file = YAML.load(File.read("code.yml"))
    ])

    puts image_table.render(:unicode, multiline: true, alignments: [:left, :left], padding:[1,1])
  end
end

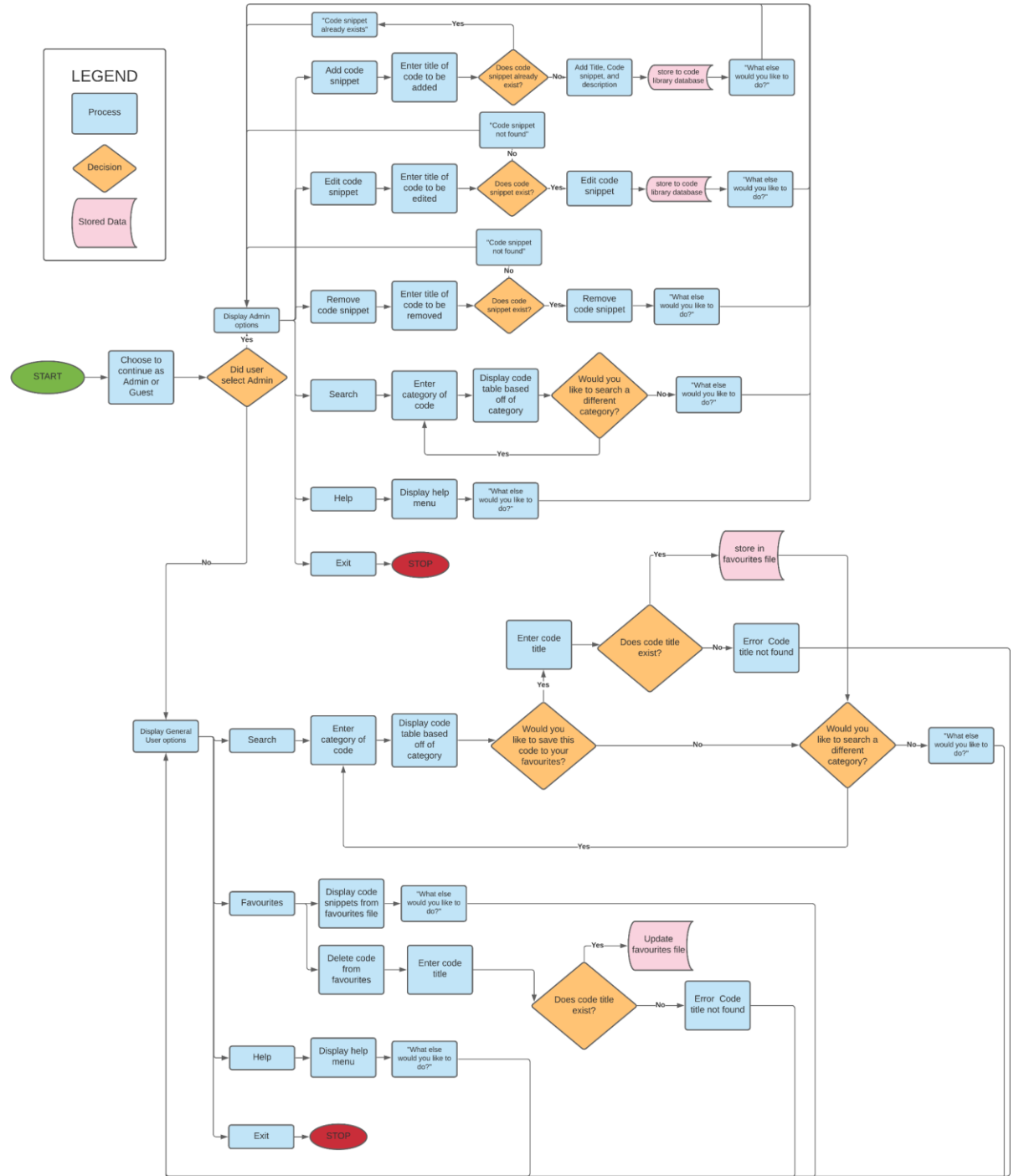
```

# DEVELOPMENT/BUILD PROCESS

- Favourite parts
  - Discovering interesting Ruby Gems and implementing them
  - Small wins whilst coding
- Challenges
  - Making my code dynamic, for example the add/remove code feature, still struggling to make that work how I want to
- Ethical Issues
  - For my login screen I'm using the same login process that we worked on for the library project.



# CONTROL FLOW

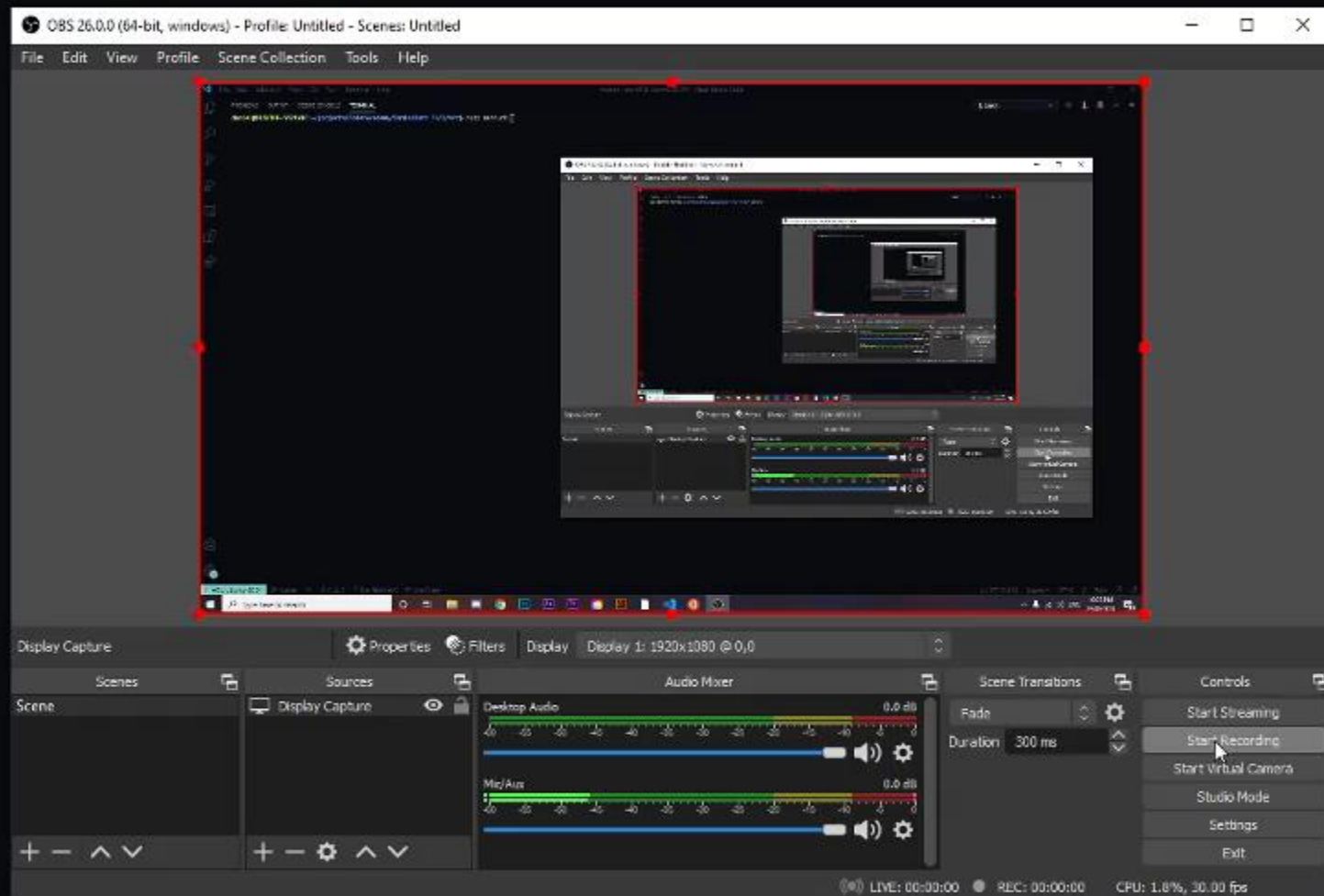


**VIDEO**

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
daniel@DESKTOP-5421VBC:~/projects/CoderAcademy/DanielKurz_T1A3/src$ ruby main.rb
```

1: bash +



**QUESTIONS?**

**END**