

Coder Academy

T1A3 - *Terminal Application.*

puts "Bone Death Academy"

CMP1041 + PRG1002



01

Welcome to



Skateboard Academy

You're
Invited!

Mathew Lawton - CAS022107.

Some images in this presentation come under the free to use licence by Canva.



99 ~~Problems~~

IDEAS

WHAT'S THE ~~PURPOSE?~~

~~PROBLEM?~~ → USE CASE?

- Bone Death Skate Academy is a world renowned boot camp for turning beginner level skaters into industry leading professionals but they currently do not have a way of booking in new skaters to their campus online.



- 1.
- 2.
- 3.
- 4.

Problem - Access to a dedicated academic sports facility for everyone, from anywhere.

Skateboarders all across the globe have a lack of facilities that encourage education towards their desired industry as a potential career.

User Case Approach

THE BONE DEATH UTOPIA.

"Learn how to skate, break bones, crush n00bs and become a shred lord! Enrol in bone death academy and just like a career in computer science, we will shred the future together."

- Po tay toe, CEO of BDS Academy.



Approach 1

How can an aspiring student get access to making a booking and becoming a shred lord from anywhere?

Approach 2

Pat - Wants to enrol but is not sure which campus is best for his chosen career path.

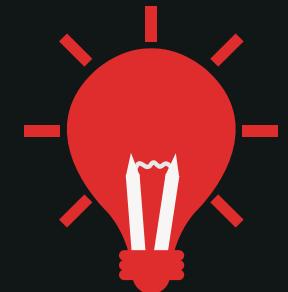
Approach 3

Mary - Has a child who accessed the booking system but did not meet the age requirements nor had parental consent.



CONCEPT

Features



AGE
REQUIREMENT
FOR ACCESS
TO CREATING
A BOOKING



CREATE,
EDIT
&
VIEW
BOOKINGS



VIEW
CAMPUS
LOCATIONS
CURRENT
&
FUTURE



CUSTOM
DISPLAYS
USING
TTY-FONT
+
TTY-PROMPT



Gems



tty-prompt



tty-box



tty-progress bar



tty-font

MENUS

RECEIPT

BOOKING

TITLES

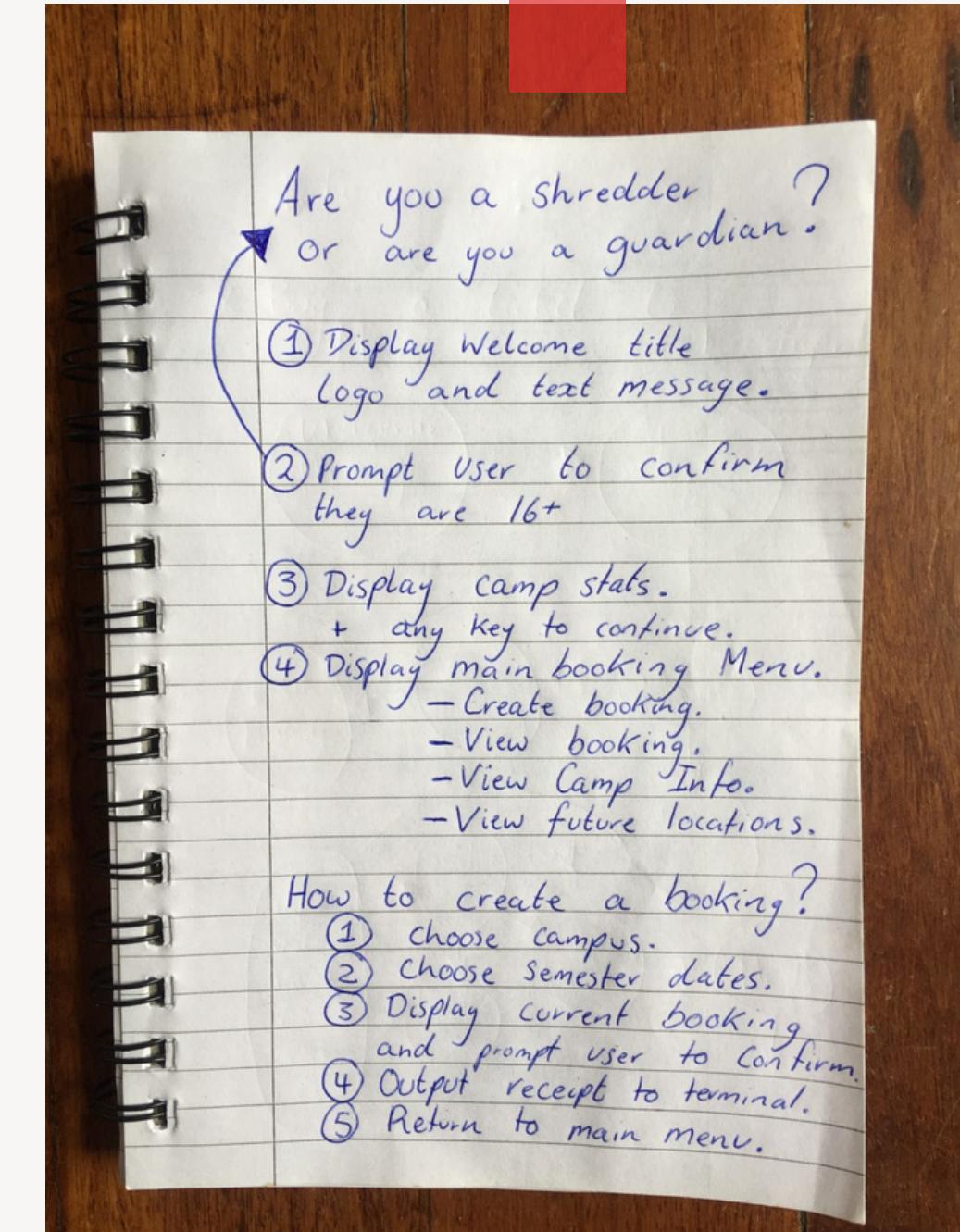
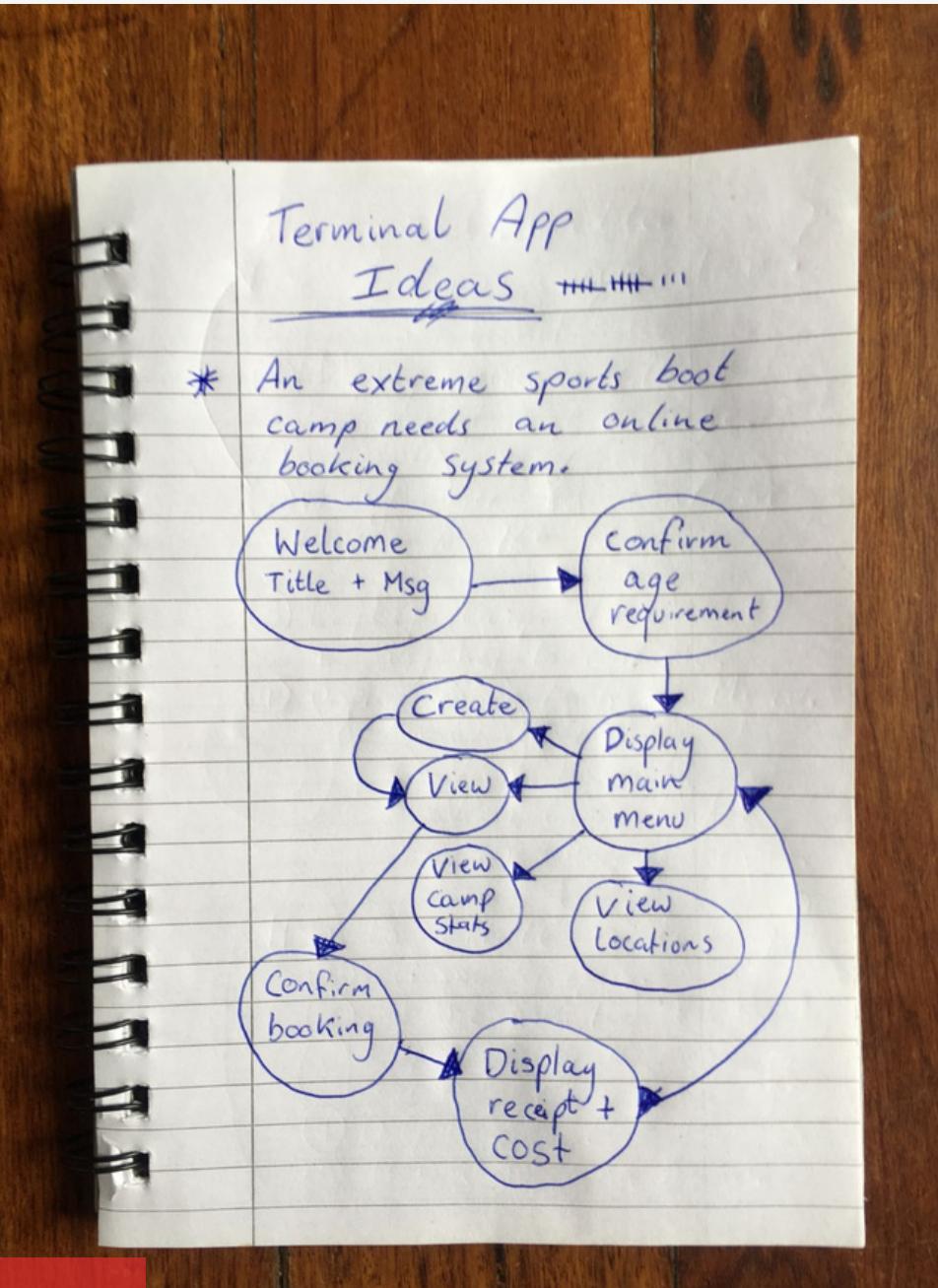
C. R. U. D

The Design Process



Mud Map V1

"not ment to make sense"



Mud Map V1.0.2

"supposed to make sense"



Target Audience

Who can benefit from this application?

- Anybody interested in skateboarding.
- Industry enthusiasts looking to further their career.
- Content creators.
- Extreme sports enthusiasts.



How to manage the load?

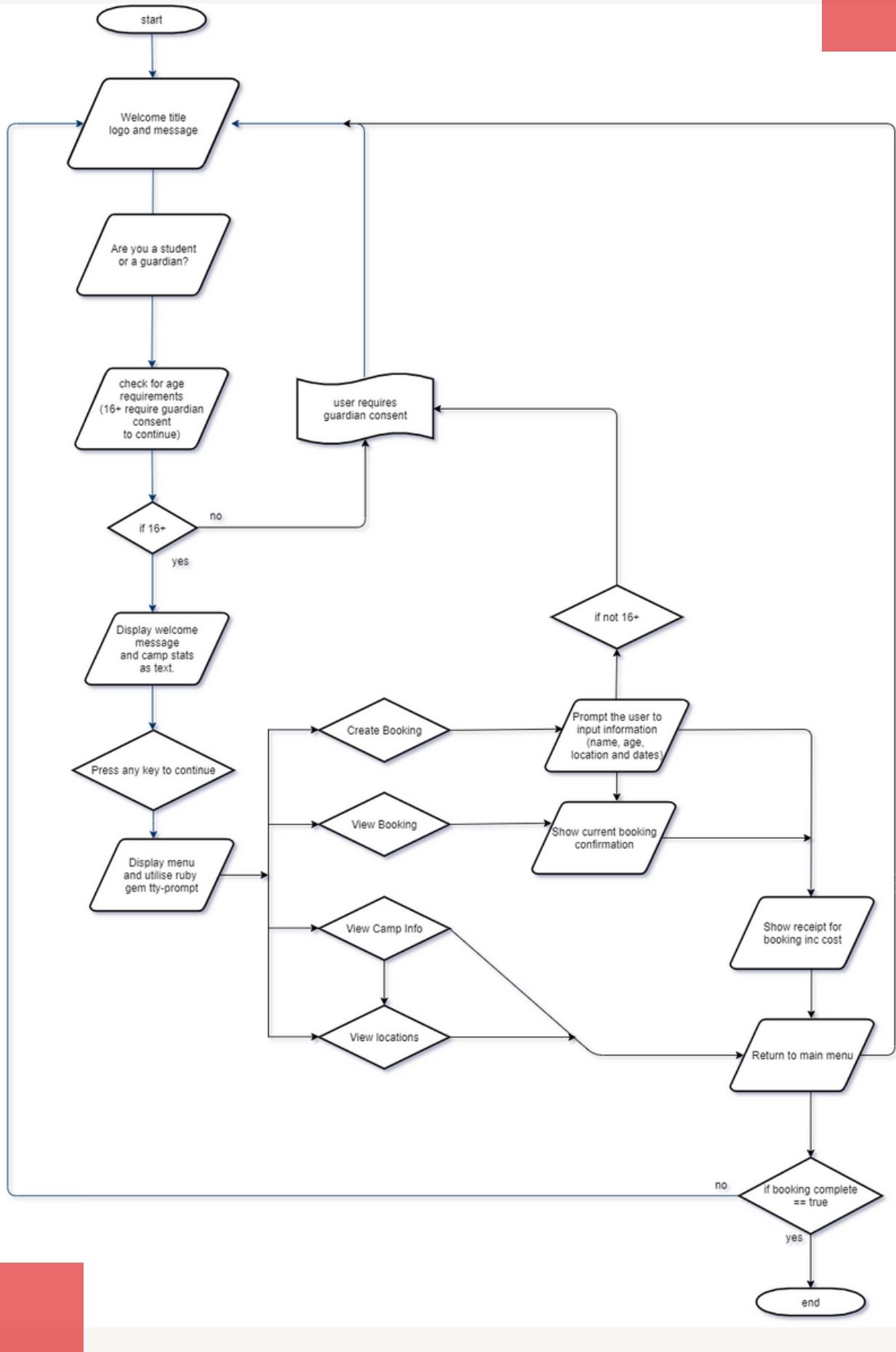
Project Management = Trello + Kanban template + Power-ups(priority) + Canva.com.

The image shows a Trello board titled "MatLawton - T1A3" with the following columns and card details:

- Backlog**:
 - BACKLOG (Yellow emoji)
 - Make regular checks of my backlog a priority (2 cards)
 - R4 - Provide a link to your source control repository. (Oct 3, High priority)
 - R6 - Develop a list of features that will be included in the application. (Sep 29, High priority)
 - R7 - Develop an outline of the user interaction and experience for the application. (Sep 29, High priority)
 - R13 - Begin implementing design process into Terminal app project files. (Oct 3, Medium priority)
- Design & Development To Do**:
 - DESIGN & Development (Yellow emoji)
 - This is my Design & Development to do list. (2 cards)
 - R8 - Develop a diagram which describes the control flow of your application. (Sep 29, High priority)
 - Create a PDF Presentation of my development process (High priority)
- README To Do**:
 - TO DO List (Yellow emoji)
 - This is my README.md to do list. (2 cards)
 - R5 - Describe at a high level what the application will do. (Oct 3, High priority)
 - R5 - Identify the problem it will solve and explain why you are developing it (Oct 3, High priority)
 - R5 - Identify the target audience. (Oct 3, High priority)
 - R5 - Explain how a member of the target audience will use it. (Oct 3, High priority)
 - R10 - Design help documentation which includes a set of instructions which accurately describe how to use and install the application. (Nov 3, High priority)
- In Progress**:
 - IN Progress (Yellow emoji)
 - In Progress (2 cards)
 - R18 - Use Git for Source Control. (Sep 29, High priority)
 - R14 - Apply DRY coding principles to all code in my project. (Oct 3, Medium priority)
 - R15 - Apply all style and conventions for Ruby consistently to all code. (Oct 3, Medium priority)
 - Design a Development Plan (High priority)
 - R9 - Develop an implementation plan which: (Oct 3, High priority)
- Testing**:
 - TESTING (Yellow emoji)
 - Testing (2 cards)
 - R17 - Design Test One (Oct 2, Low priority)
 - R17 - Design test Two (Oct 2, Low priority)
- Done**:
 - COMPLETED Tasks (Yellow emoji)
 - Done (3 cards)
 - Submit ideas to educators (Sep 23, Low priority)
 - Brainstorm concept ideas for my terminal app (Sep 23 - Sep 23, Medium priority)



The Design Process



Flow Chart - V1

Control Flow Diagram

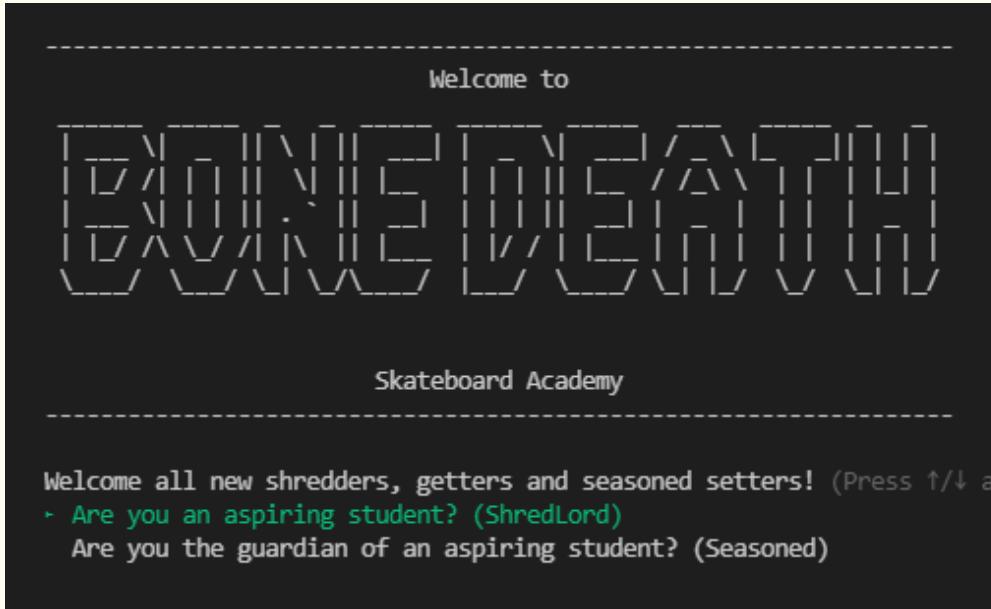


Overview



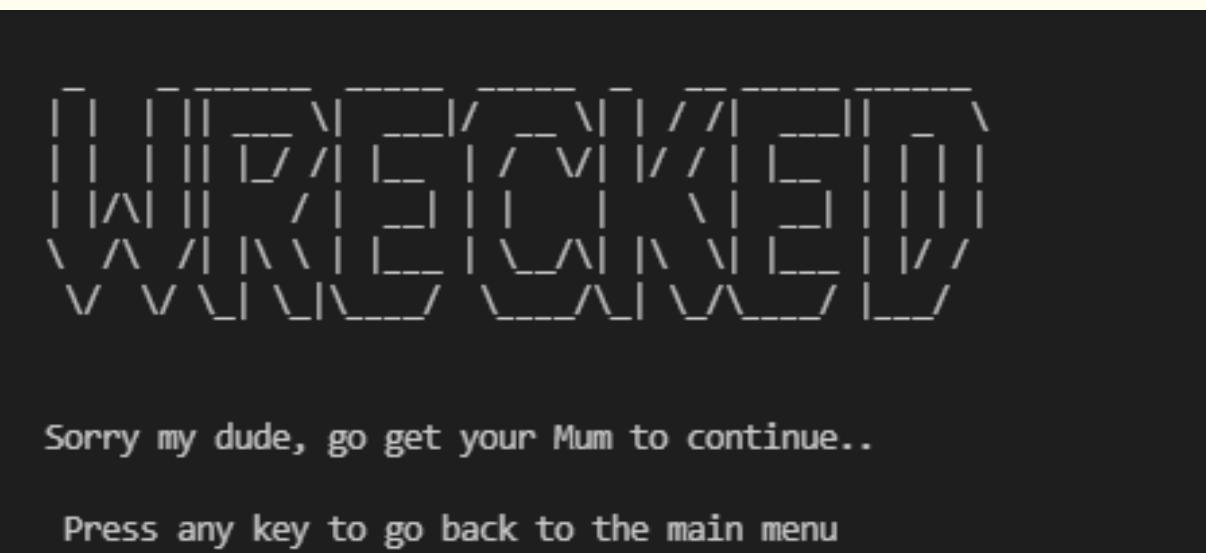
1 - Start program by executing main.rb in the terminal.

2 -



3 - Upon answering this question the user is prompted to declare if they are over the age of 16 or not

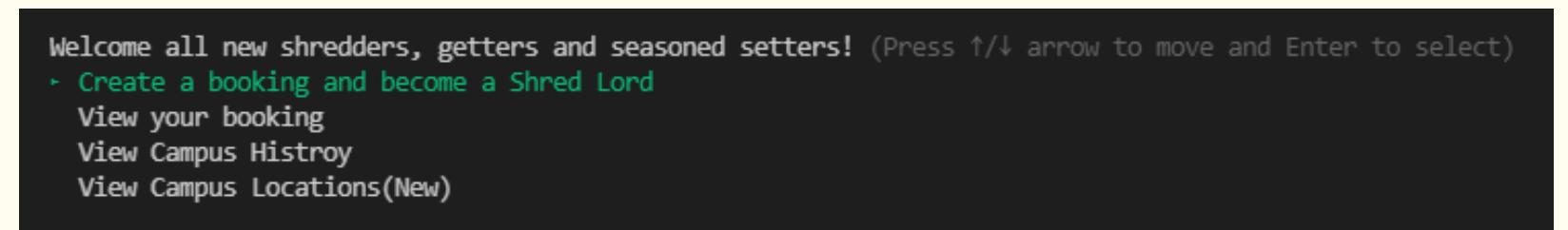
4 - If not over the age of 16 then,



6 - If over the age requirement and/or Guardian, then a brief welcome message and a list of stats appear in the terminal such as broken bones, current students, n00bs turned pro, previous students etc. These will be variables stored in an array.

7 - If the show reel of stats has ended, the user will be prompted to press/enter any key to continue.

8 -



9 - If the user selects create a booking, then a list of options in a new menu list will be displayed showing location(only two locations available), semester dates(only two semesters per year), personal info (name, age, email) show the price of course and prompt the user to confirm booking or cancel and return to main menu. Use of tty-prompt and tty-progress bar will be used here.

Overview - Continued

10 - Upon confirmation by the user, a receipt will be displayed to the terminal with all the information provided by the user, a thank you message and the booking menu will be displayed again. Use of tty-prompt and tty-box will be used here.

11 - User will be prompted to press any key to return to main menu or complete booking and end program.

12 - If the user selects main menu and chooses the view a booking option in the menu then their receipt of their previous booking will be displayed.

13 - User will be prompted to press any key to return to main menu or complete booking and end program.

```
# This is my main menu welcome message
def welcome()
  font = TTY::Font.new(:doom)
  puts
  puts "---" *22
  puts " " *28 + "Welcome to"
  puts font.write("BONE DEATH")
  puts " " *24 + "Skateboard Academy"
  puts "---" *22
  puts
end
```

```
# This is how I can clear the terminal screen when and if required
def clear_term
  | puts "\e[2J\e[f" # Found this block of code on ruby-forums.com
end

# If a user is not 16 and above then.
def not_age
  font = TTY::Font.new(:doom)
  puts clear_term
  puts font.write("WRECKED")
  puts "Sorry my dude, go get your Mum to continue.."
  puts "\n Press any key to go back to the main menu"
  gets.chomp
  # $stdin.getch
  clear
end
```



Challenges - Some of my challenges included having great ideas but realising I could not yet transfer my ideas into my skillset with Ruby. Deciding on the correct method for storing input and outputs.

Ethical Issues - Some of the ethical issues that I considered during the design process were things such as the ethical ramifications of collecting and storing an individuals personal information, a minor gaining access to the booking system without permission and the legality of taking payments via a credit card through the application. I also considered that if this was a real world business that a disclaimer for admission and booking would need to be implimented.

Challenges, Ethical Issues and favourite parts.

```
def favourites = [  
    "Ruby for breakfast!", +  
    "Supportive peers!", +  
    "Design process!", +  
    "I'm learning!"  
]  
puts favourites
```





*Tacos for
everyone!*

- **Thank you.**
Thanks for hanging in there with me!
- **Did I miss something?**
Please let me know if I did!
- **Good Luck!**
We got this!