

T1A3 - Terminal App

Mitchell Painter

Terminal Application - Overview

Woofz - A Dog Walkers Management App

- The assumed target application user is a dog walker.
- The app usage is from a dog walkers perspective, within a two-sided marketplace for the purpose of managing, organising and selecting dog walking jobs.
- The app is designed and built to help manage, create, read, update, and delete records of previous, current, and future dog walking jobs.
- The app also allows the dog walker to find, choose, and confirm current dog walking jobs with the following information:
- Name, Breed, Age, and Walk time

- The user is greeted with a “Hello! Welcome to Woofz. That prompts the user to select one of three menu options

```
mitchellpainter@Mitchells-MacBook-Pro MitchellPainter_T1A3 % ruby main.rb
Hello! Welcome to Woofz. To begin, please select one of the options below:
For Current Available Jobs, Enter: A
For Future Jobs, Enter: B
For Past Jobs, Enter: C
█
```

- Depending on the response received by the terminal, the app then prompts the user for their name, asks if they are ready and moves on.
- If ‘A’ is selected it allows the user to view today's dog walking jobs, select and confirm which dog they would like to walk and find out how much they will be paid for completing that walk.
- If ‘B’ is selected it allows the user to view future dog walking jobs and make any changes or cancellations required.
- If ‘C’ is selected, it allows the user to view an archive of any past, completed dog walking jobs and see when they occurred and who walked which dog.

- If 'A' is selected the terminal displays "Hello #{@name} the user is prompted to "return/enter for a list of dogs that need walks today".
- This list includes information such as:

Name	Breed	Age	Walk-time in mins
------	-------	-----	-------------------

```
1. Buster is a Labrador, is 4 years old and needs a 30 min walk.
2. Juno is a Border Collie, is 2 years old and needs a 60 min walk.
3. Fluffy is a Groodle, is 3 years old and needs a 45 min walk.
4. Bindi is a Springer Spanial, is 2 years old and needs a 60 min walk.
return/enter
```

- The user is then prompted with "which dog would you like to walk?"

```
Which dog would you like to walk?
To Walk Buster, Enter: 1
To Walk Juno, Enter: 2
To Walk Fluffy, Enter: 3
To Walk Bindi, Enter: 4
To Walk Juno & Bindi, Enter: 5
```

- Once the user has made a selection the app will prompt the user to confirm with a Y/N
- If the user confirms with 'Y' it will display a message stating the final confirmation of the job, including the dog's Name, the walk distance required and finally the amount the walker will be paid.
- The user will then be prompted to "return/enter to finish" and the message "Thank You, Have A Nice Day!" Will be displayed.
- If the user selects 'N' the app will redirect through the use of a loop back to the beginning.

Code:

- The app is utilizing the Rainbow Gem throughout, in order to accentuate and differentiate certain lines and words to make them more clear and readable.
- The app begins with multiple puts statements, with a hello message followed by three menu options named 'A', 'B', and 'C'
- Followed by an if statement: if user_input == 'A'

```
require 'rainbow'
puts Rainbow("Hello! Welcome to Woofz. To begin, please select one of the options below: ").orange
puts Rainbow("For Current Available Jobs, Enter:").blue + Rainbow(" A ").orange
puts Rainbow("For Future Jobs, Enter:").blue + Rainbow(" B ").orange
puts Rainbow("For Past Jobs, Enter:").blue + Rainbow(" C ").orange

user_input = gets.chomp.capitalize
if user_input == 'A'
  puts Rainbow("To get started, please enter your name: ").orange
  name = gets.chomp
  puts "Hello #{name.capitalize}, Are you ready to find a dog to walk?:" + Rainbow(" Y/N ?").orange
  ready = gets.chomp
  puts "Okay, #{name.capitalize}. Let's choose a dog to walk! "
  puts Rainbow("return/enter ").orange + "for a list of dogs that need walks today! "
  dog = gets.chomp
end
```


- Followed by a class called Dog

```
19 class Dog
20   attr_reader :age, :breed
21   attr_accessor :name
22
23   def initialize(name, breed, age)
24     @name = name
25     @breed = breed
26     @age = age
27     @walk = []
28   end
29
30   def walk(time_in_mins)
31     @walk << time_in_mins
32   end
33
34   def walk_time
35     return @walk.sum
36   end
37
38   def display_walk
39     puts Rainbow("#{@name} ").green + "is a " + Rainbow("#{@breed}").red + ", is #{@age} years old and needs a #{@walk_time} min walk. "
40   end
41 end
42
43 #Main
44 dog1 = Dog.new("1. Buster", "Labrador", 4)
45 dog1.walk(30)
46 dog1.display_walk
47 dog2 = Dog.new("2. Juno", "Border Collie", 2)
48 dog2.walk(60)
49 dog2.display_walk
50 dog3 = Dog.new("3. Fluffy", "Groodle", 3)
51 dog3.walk(45)
52 dog3.display_walk
53 dog4 = Dog.new("4. Bindi", "Springer Spanial", 2)
54 dog4.walk(60)
55 dog4.display_walk
```

```
57  # Note: Juno and Bindi both need 60 min walks and can be walked together!
58  b = "return/enter"
59  puts Rainbow(b).orange
60  gets.chomp
61  puts "Note: " + Rainbow("Juno").green + " and " + Rainbow("Bindi").green + " both need 60 min walks and can be walked together! ""
62  puts Rainbow(b).orange
63  gets.chomp
64
65  ~
66  puts "Which dog would you like to walk? "
67  a = "To Walk "
68  b = ", "
69  c = "Enter: "
70  puts a + Rainbow("Buster").green + b + c + Rainbow("1").orange
71  puts a + Rainbow("Juno").green + b + c + Rainbow("2").orange
72  puts a + Rainbow("Fluffy").green + b + c + Rainbow("3").orange
73  puts a + Rainbow("Bindi").green + b + c + Rainbow("4").orange
74  puts a + Rainbow("Juno & Bindi").green + b + c + Rainbow("5").orange
```


- If 'B' is selected from the menu options it will allow the user to view any future/upcoming walking jobs that have been requested.
- Allowing the user to view things like the job date, time, name, age, breed and walk duration.
- Make any changes to these bookings as required, E.g. changes to the booking date/time or walk duration.
- Make any cancellations/deletions to any future bookings.
- It will also allow the user (dog walker) to add any new walking jobs or data that may be required and not yet input.

- If 'C' is selected it allows the user/dog walker to view archived data of any past dog walking jobs.
- Allowing the user to view the dogs' information,
- What day/time the dog was walked,
- Who walked the dog, and
- The walk duration
- It also displays the dog owners' contact information, such as Names, Addresses and Phone Numbers.
- It also allows the user to view a rating/5 the customer made, based on the service they felt they received.