

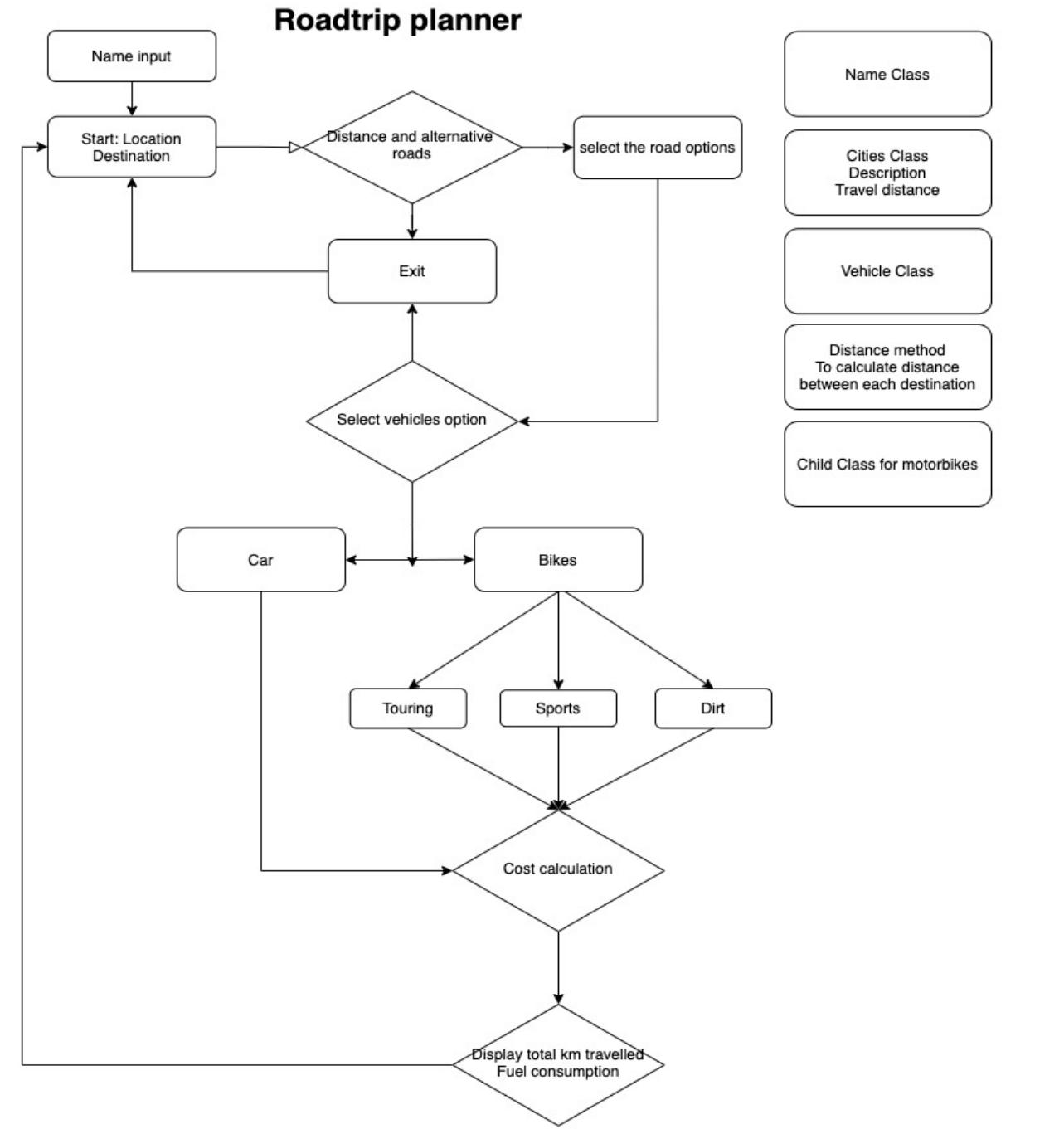
ROAD TRIP PLANNER V1.0

BY DUNG DUC NGUYEN (JOHNNY)

APPLICATION OVERVIEW

- ROAD TRIP PLANNER ALLOWS USERS TO SELECT THE TRAVEL DESTINATION
- PROVIDES A BRIEF DESCRIPTION ABOUT THE DESTINATION
- CALCULATES DISTANCE THAT NEEDS TO TRAVEL
- GIVES A RANGE OF VEHICLE OPTIONS
- SUMMARIES THE TOTAL FUEL COST OF CHOSEN VEHICLES FOR EACH TRIP

FLOWCHART



FEATURES

- ALLOWS USERS TO INPUT THEIR NAMES UP TO THREE TIMES
- DISPLAY TRAVEL OPTIONS, CITIES IN PARTICULAR
- WHEN DESTINATION SELECTED, DISPLAY THE INFORMATION ABOUT THE CITY AND PROVIDES DISTANCE IN KM
- AFTER SELECTING TRAVEL OPTIONS, USERS COULD THEN SELECT THEIR MEANS OF VEHICLE SUCH AS CARS, OR MOTORBIKES.
- DISPLAY THE FUEL CONSUMPTION OF EACH VEHICLE CHOSEN
- PRINT OUT THE SUMMARY OF TOTAL COST FOR EACH SELECTED TRIP
- BACK TO MAIN MENU, AND EXIT BUTTON TO QUIT THE PROGRAM

LOGIC & CODE

Name input

- While loops
- Conditional statement - IF
- User class

```
1 require "tty-prompt"
2 class User
3   attr_accessor :name
4
5   def initialize(name)
6     @name = name
7   end
8
9 end
```

```
puts "Roadtrip Planner! v.101"
puts "Wanting to have a roadtrip? You have come to the right place!"
puts "Please enter your name: "

user_name_count = 0
while user_name_count < 3
  user_name = gets.strip.capitalize
  if user_name != ""
    break
  else
    user_name_count += 1
    if user_name_count == 3
      puts "\n Hi there!"
      user_name = "Mysterious!"
    else
      puts "Please enter your name: "
    end
  end
end

rider = User.new(user_name)
puts "Let's get going #{rider.name}!"
```

LOGIC & CODE

Travel options

- While loops
- Iterating through a hash with cities and their destination
- Case statement to display each city
- Exit method

```
while true
  selection = TTY::Prompt.new.select("Where would you like to travel today? Please select from the following options:")
  places.choice('Melbourne', 1000)
  places.choice('Sydney', 2000 )
  places.choice('Perth', 3000)
  places.choice('Cairns', 400)
  places.choice('Exit', 0)

  case selection
  when 1000
    # puts "You have selected Melbourne"
    puts
    melbourne = Cities.new("Melbourne","This is the most liveable city in the world!", 6000)
    melbourne.display_distance
    puts
    vehicles
    # call choose vehicle method, assign to variable
    # puts "your trip is #{ride[:Melbourne]} kms long"
  when 2000
    puts
    sydney = Cities.new("Sydney","Here is where you would see the Opera House!", 2000)
    sydney.display_distance
  when 3000
    puts "You have selected Perth"
  when 800
    puts "You have selected Cairns"
  when 0
    puts "Thank you #{user_name}!"
    puts "I hope to see you on the road soon!"
    exit
  end
end
```

LOGIC & CODE

Cities & vehicles

- Cities class
- Vehicles class
- Cost calculation method

```
require "tty-prompt"

def vehicles
  prompt = TTY::Prompt.new
  prompt.select("How would you like to travel?", %w(Car Motorbike))
  if prompt.to_s == "Car"
    puts "Cars consume 0.7 litre per 100 kms"
  else
    puts "Motobikes consume 0.2 litre per 100 kms"
  end
end
```

```
class Car
  attr_accessor :capacity

  def initialize(capacity)
    @capacity = capacity
  end

  def consume
    return "Choosing #{@type} offers #{@description}"
  end

end

def choose_vehicle
  # choose car or motorbike
  # return choice
end

car = Vehicle.new("Cars","comfort & safety for your trip")
puts car.to_s
```

DEVELOPMENT REVIEW

DECISION MAKING

- SIMPLE, FULFILL BASIC REQUIREMENTS
- EASY TO MAINTAIN THE CODE LATER

CODES & SYNTAXES

- VARIABLES, LOOPS, CONDITIONS
- METHODS, CLASSES, GEMS
- ERROR HANDLING
- TESTING

TRELLO BOARD

The image shows a Trello board with five columns:

- To Do**: Contains cards for "To Do", "Read me file", "Error handling", and "Script file". A plus icon "+ Add another card" is at the bottom.
- Doing**: Contains cards for "Doing", "Install gems", "Conditional statement", "Git push, commits", "Slide deck", and a plus icon "+ Add another card".
- Code Review**: Contains cards for "Code Review", "Cities class", "Vehicle class", "Name class", and "Cost calculation method". The "Vehicle class" card is highlighted with a yellow background.
- Testing**: Contains cards for "Testing", "Flowchart", and a plus icon "+ Add another card".
- Done**: Contains cards for "Done", "Trello boards", "Loops", "Create variables", "Ideas for terminal app", "[Completed task] (24 Jan 2020)", and a plus icon "+ Add another card".

DEVELOPMENT REVIEW

FAVORITE PARTS

- ITERATING THROUGH HASH
- GEMS – TTY
- PROBLEM SOLVING

CHALLENGES

- FIRST TERMINAL APP
- LACK OF EXPERIENCE
- CODING PRACTICE
- PROGRAMMING TOOLS
- LANGUAGES & TIME MANAGEMENT

THANK YOU FOR LISTENING!