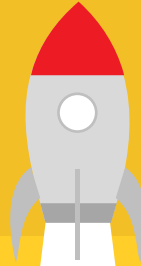# Moon Trip Booking System

Phuong Anh Dang (Olivia)
Fast Track Sydney, 2021

Moonlight Travel

Welcome!!

Fly you to the Moon

Moonlight Travel's mission is to empower your dream about space travel, making it come true

Agenda Style

01

Overview

Overall structure of Moon Trip Booking System

02

Development Process

From idea to application design and deployment

03

Features

Main features of Moonlight Travel application
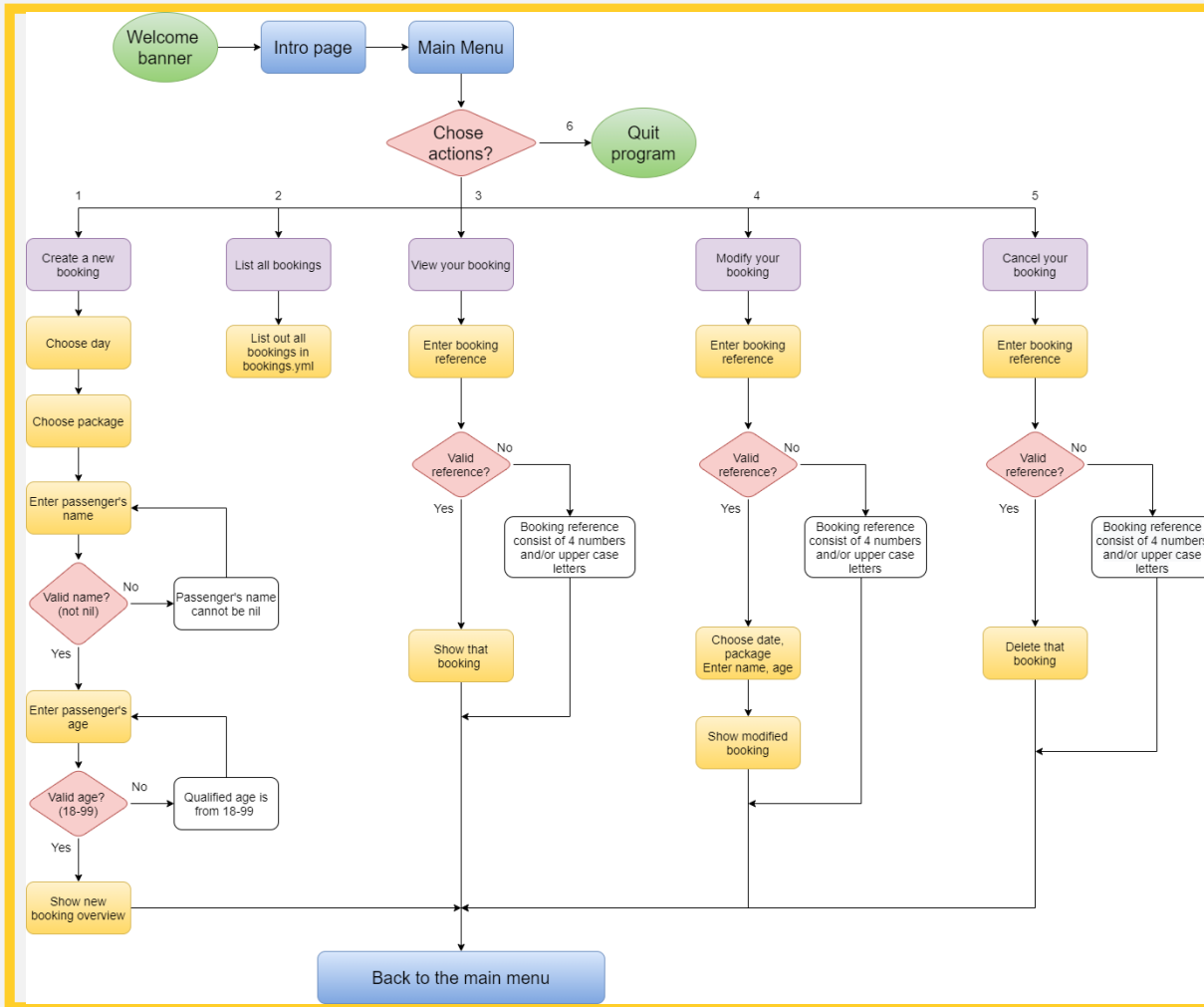
04

Challenges
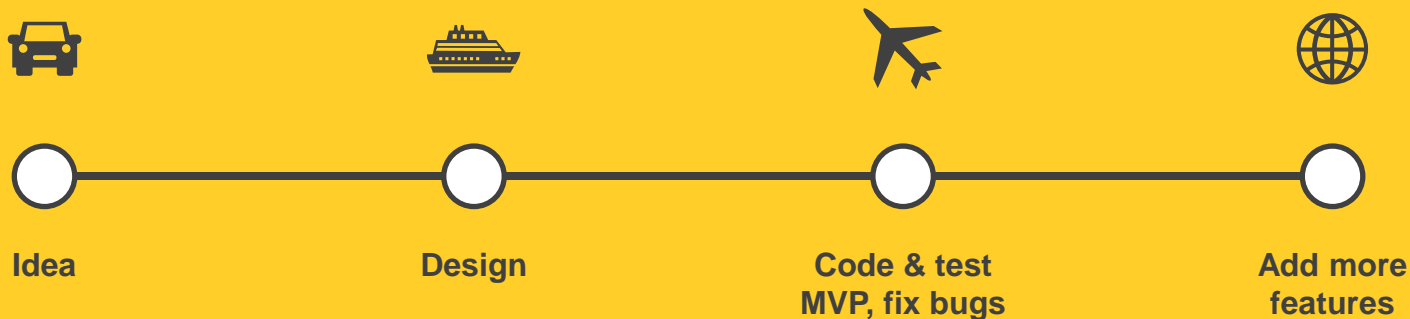
Errors, bugs and difficulties

Overview

Basically, this is a very simple tour booking system.

When the app runs, it will show a welcome banner and then description about the Moon Trip.

The main menu is followed. Users will choose which action to take. Unless users choose to quit, the program will always go back to the main menu after each action is taken.

Development process

Idea

Design

**Code & test
MVP, fix bugs**

**Add more
features**

I know that I'm still in "kindergarten level" when it comes to reading about Objects, class, YAML, testing …, so I'll try to make an "enough-to-pass" MVP first, and then fancier features later, and then refactoring, syntactic sugar or pretty code later.

Idea

When I heard about the terminal app assignment, I kept thinking what I would like to do? I looked up in the night sky and smiled: "Let's go to the moon". And that's how I started making a Moon Trip Booking System.

At first, I want to make a flight booking system to the Moon only, but people can't just take a flight by rocket to the Moon, because they cannot travel by themselves, book an Airbnb elsewhere and find their own food. Therefore, it makes more sense if I offer a Full Package Moon Trip with date and package options.

Features

As a tourism booking application, this app can create a booking, modify a booking, view a booking, delete a booking. Those are basic functions that end-users (tourists) would like to do when booking a tour. I also created "list all bookings" function. This is basically just for the tourism company to manage all bookings.

Target audience: tourism company and tourists (end-user)

```
---------------------------------------------------------------
                      Moonlight Travel
               Fly you to the Moon and back
---------------------------------------------------------------
What would you like to do? (Press ↑/↓ arrow to move and Enter to select)
▸ Create a new booking
  List all bookings
  View your booking
  Modify your booking
  Cancel your booking
  Quit
```

CRUD Feature

Create a new booking

Users can create a booking in the system: Choose date, choose package, enter name, age

List all bookings

Users can see all bookings stored in bookings.yml as a persistent database

View your booking

Users can enter booking reference to see a specific booking

Modify your booking

Users can update a booking using booking reference

Cancel your booking

Users can delete a booking using booking reference

Ruby Gems

tty-font

Ruby Gems

tty-prompt

```ruby
puts "Please keep in mind that this is a non-refundable booking. \
date = TTY::Prompt.new.select ("Choose departure date") do |menu|
    menu.choice "10 October 2022"
    menu.choice "11 November 2022"
    menu.choice "12 December 2022"
end
```

```ruby
age = TTY::Prompt.new.ask("How old is the passenger? ") do |q|
    q.validate (/^(1[89]|[2-9]\d)$/)
end
```

tty-table

```ruby
def self.index(bookings)
    return puts("No one has booked yet") unless bookings.length>0
    table = TTY::Table.new(
        ["Reference", "Departure date", "Package", "Passenger's name", "Age"],
        list_for(bookings)
    )
    puts table.render(:ascii, padding: [0, 1, 0, 1])
end
def self.list_for(bookings)
    bookings.map do |booking|
        [booking.reference, booking.date, booking.package, booking.name, booking.age]
    end
end
```

Persistent storage

YAML


```
bookings.yml
1    ---
2  - !ruby/object:Booking
3      name: Olivia
4      age: '28'
5      date: 12 December 2022
6      package: Adventurer
7      reference: 0NTL
```

Challenge

YAML error

It took me 3 days to code "create function". The most difficult part is using YAML as a database. I thought of .txt, but when I did research on the internet, professionals don't usually use file.txt to store data. So I tried to play around with YAML, create an instant called BOOKINGS:

```
BOOKINGS = YAML.load(File.read('bookings.yml')) rescue []
```

CRUD worked fine. However, whenever I "quit" app and run again, the storage file "booking.yml" cannot load existing data at start and kept overwriting with new bookings. This is very annoying and no longer permanently store data as a database.

I knew the problem but couldn't fix it. I watched Michael's lecture again and again, from the basic design until he successfully moved the database to YAML. Following his steps, he said we shouldn't use a constant to store data. Maybe this is my problem.
I didn't fully understand in class but after many failures, I finally nailed it.

Challenge

Add more features

I want the app to show availability of each trip and each travel package, but I still don't know how to.

I also want the app to create Users, but this is too tough can I'm afraid I don't have enough time to learn and code.
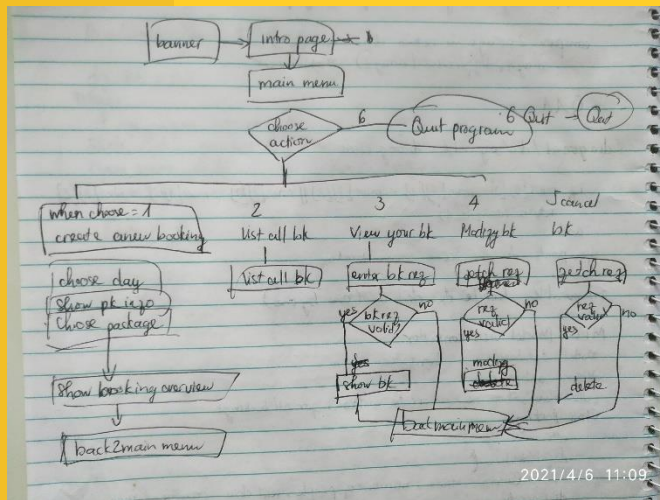
Errors Handling

I spent 10 hours doing challenges.
I didn't understand what is standardError, what's not the difference between raise and rescue.
Now I got it. 15 attempts for the Invalid Words challenge and 19 attempts for Invalid Division.

BUT I still cannot handle all possible errors, for example nil name input.

Challenge

It was not until yesterday (Tuesday) that I listen to the TDD videos. I can understand them and understand Michael's lecture, but I cannot actively use rspec to write tests for now. Maybe too much new technology to learn so I was overwhelmed.

Although it seems like I happened to follow Red/Green/Refactor when I code. I wrote out the app idea and started making a simple MVP. I always test along the way whenever I moved code snippets to a new file or new module, change the variables' name, after creating a new method that may break existing function.

Thank you

See you at our Space Station