James Xu - PokeDex

T1A3 Terminal Application

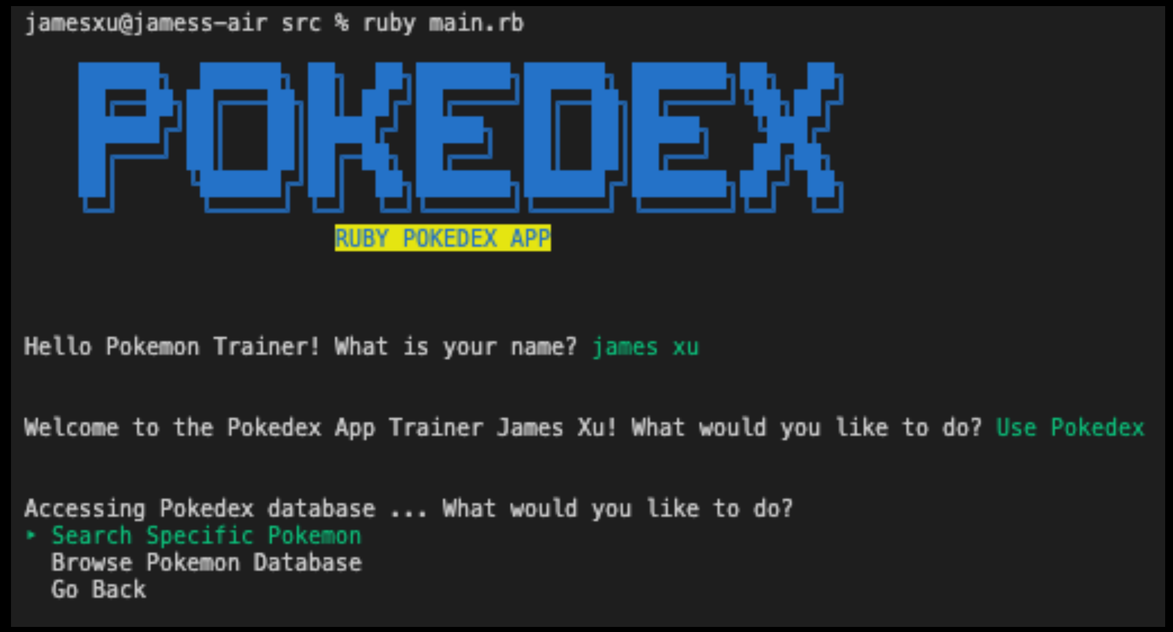
What to cover

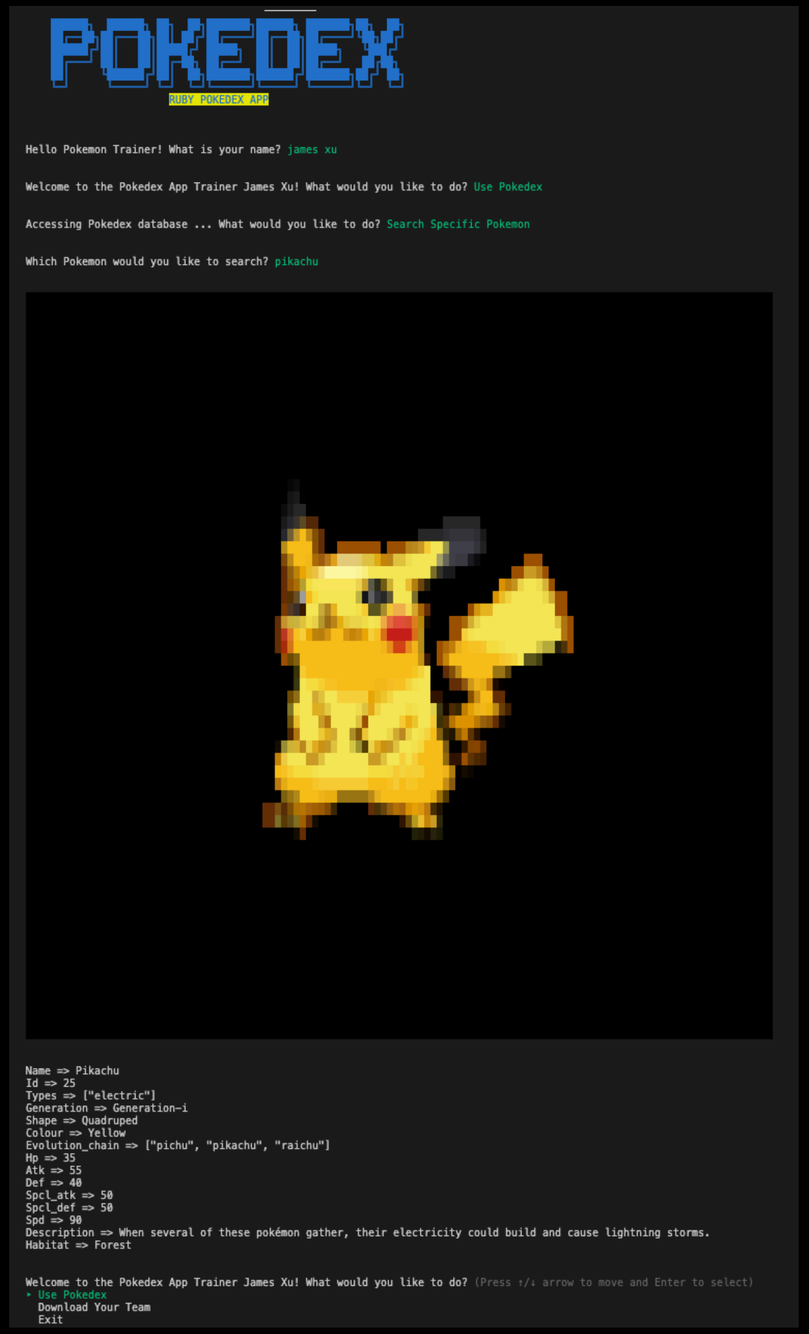
Pokedex App

- App Features; how is it used?
- App Logic & code
- Development process review
  - Challenges
  - Ethical issues
  - Favourite parts

App Features
What can it do?

- Search specific Pokemon data
- Select from a list of Pokemon to view their data
- Display Pokemon team data from user input (in progress)
- Save data from user input as JSON file (in progress)





Searching for specific Pokemon by name

```
Welcome to the Pokedex App Trainer James Xu! What would you like to do? Use Pokedex

Accessing Pokedex database ... What would you like to do? Search Specific Pokemon

Which Pokemon would you like to search? pikachu
```

Displays relevant data about that Pokemon, pulls data from JSON files stored on PokeApi online

```
Name => Pikachu
Id => 25
Types => ["electric"]
Generation => Generation-i
Shape => Quadruped
Colour => Yellow
Evolution_chain => ["pichu", "pikachu", "raichu"]
Hp => 35
Atk => 55
Def => 40
Spcl_atk => 50
Spcl_def => 50
Spd => 90
Description => When several of these pokémon gather, their electricity could build and cause lightning storms.
Habitat => Forest


Welcome to the Pokedex App Trainer James Xu! What would you like to do? (Press ↑/↓ arrow to move and Enter to select)
‣ Use Pokedex
  Download Your Team
  Exit
```

```
Please select a pokemon to view more OR start typing a pokemon name to filter ... Charizard
```

```
Name => Charizard
Id => 6
Types => ["fire", "flying"]
Generation => Generation-i
Shape => Upright
Colour => Red
Evolution_chain => ["charmander", "charmeleon", "charizard"]
Hp => 78
Atk => 84
Def => 78
Spcl_atk => 109
Spcl_def => 85
Spd => 100
Description => Spits fire that is hot enough to melt boulders. known to cause forest fires unintentionally.
Habitat => Mountain
```

Filtering for specific Pokemon by name

```
Accessing Pokedex database ... What would you like to do? Browse Pokemon Database



Please select a pokemon to view more OR start typing a pokemon name to filter ... (Filter: "char")
▸ Charizard
  Charjabug
  Charmander
  Charmeleon
  Chimchar
```

Incorrect search, error message, return to home

```
Accessing Pokedex database ... What would you like to do? Search Specific Pokemon

Which Pokemon would you like to search? pifdsafdsaf

Not a valid search/Missing data from PokeApi

Welcome to the Pokedex App Trainer James Xu! What would you like to do? (Press ↑/↓ arrow to move and Enter to select)
▸ Use Pokedex
  Download Your Team
  Exit
```

In Progress

Soon to come

- Obtain user input
  - Save to file, parse through program to pull all Pokemon data at once

- Save pulled data as JSON on local machine
  - User may want to use pulled data in own project

App Logic
How does it work?

- Pokedex class
  - All pokemon api (application programming interface) calls linked to pokedex class
  - Using ruby gem poke-api-v2
  - Net/http - gem, allows http requests, connect with api

```ruby
class Pokedex
    attr_accessor :name

    def initialize(name)
        @name = name
    end
```

```ruby
class Pokedex
    attr_accessor :name

    def initialize(name)
        @name = name
    end

    def get_pokemon_general_info(pokemon_name)
        res = PokeApi.get(pokemon: pokemon_name)
        pokemon_type = []

        res.types.each do |item|
            pokemon_type << item.type.name.capitalize
        end
      You, 2 days ago • created pokedex class
        pokemon_info = {
            name: res.name,
            id: res.id,
            types: pokemon_type,
        }
        return pokemon_info.each do |key, value|
            puts key.to_s.capitalize + " => " + value.to_s.capitalize
        end
    end

    def get_pokemon_species_info(pokemon_name)
        res = PokeApi.get(pokemon_species: pokemon_name)
        evo_id = res.evolution_chain.get.id
        evo_res = PokeApi.get(evolution_chain: evo_id)
        evo_chain = []
        begin
            evo_chain << evo_res.chain.species.name
            evo_chain << evo_res.chain.evolves_to.first.species.name
            evo_chain << evo_res.chain.evolves_to.first.evolves_to.first.species.name
        rescue => e
        end


        pokemon_info = {
            generation: res.generation.name,
            shape: res.shape.name,
            colour: res.color.name,
            evolution_chain: evo_chain,
        }

        return pokemon_info.each do |key, value|
            puts key.to_s.capitalize + " => " + value.to_s.capitalize
        end
    end

    def get_pokemon_habitat_info(pokemon_name)
        res = PokeApi.get(pokemon_species: pokemon_name)

        pokemon_info = {
            habitat: res.habitat.name,
        }

        return pokemon_info.each do |key, value|
            puts key.to_s.capitalize + " => " + value.to_s.capitalize
        end
    end
```

Connect with api, specify what data to pull, push into array, set k-v pairs, format data and print to screen

```ruby
    def get_pokemon_species_info(pokemon_name)
        res = PokeApi.get(pokemon_species: pokemon_name)
        evo_id = res.evolution_chain.get.id
        evo_res = PokeApi.get(evolution_chain: evo_id)
        evo_chain = []
        begin
            evo_chain << evo_res.chain.species.name
            evo_chain << evo_res.chain.evolves_to.first.species.name
            evo_chain << evo_res.chain.evolves_to.first.evolves_to.first.species.name
        rescue => e
        end


        pokemon_info = {
            generation: res.generation.name,
            shape: res.shape.name,
            colour: res.color.name,
            evolution_chain: evo_chain,
        }

        return pokemon_info.each do |key, value|
            puts key.to_s.capitalize + " => " + value.to_s.capitalize
        end
    end
```

Converts image into pixels on terminal, parameters set using image.scale, image.rows (gems: paint, rmagick)

```ruby
    def ascii_image(pokemon_name)
        puts ("\n") * 2
        res = PokeApi.get(pokemon: pokemon_name)
        res = res.sprites.front_default
        uri = URI.parse(res)
        response = Net::HTTP.get_response(uri)
        image = Magick::ImageList.new
        image.from_blob uri.read
        image = image.scale(120/ image.columns.to_f)
        image = image.scale(image.columns, image.rows / 2)
        cur_row = 0
        image.each_pixel do |pixel, col, row|
        color = pixel.to_color(Magick::AllCompliance, false, 8)
        if cur_row != row
            puts
            cur_row = row
        end
        print Paint[' ', '', color]
        end
        puts
        puts ("\n") * 2
    end

end
```

App Logic
How does it work?

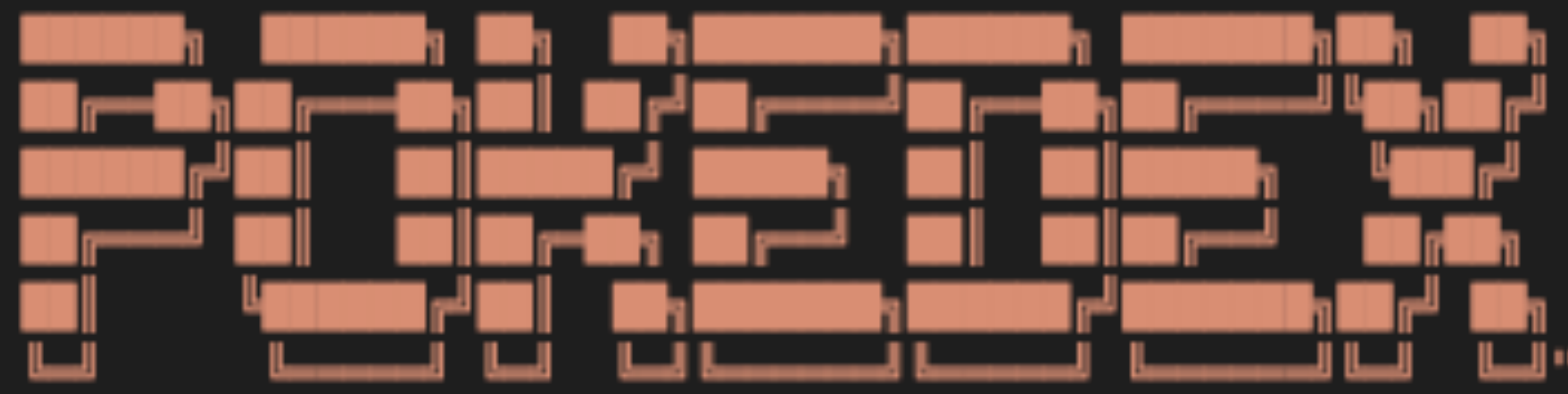
- Main.rb
  - Utilise methods to pull data when called upon in the menu
  - Action done, cycles back to home screen to reselect options
  - Nested hashes of k-v pairs to select menu option (gem: tty-prompt)

```ruby
#Generates the Pokedex app logo
self.logo

#Initiates tty-prompt
prompt = TTY::Prompt.new

#Obtain and save user_name, re-asks user for input if blank
begin
    puts ("\n") * 3
    user_name = prompt.ask("Hello Pokemon Trainer! What is your name?")
    raise StandardError if user_name.nil? || user_name == ""
rescue
    puts "Please enter a name."
    retry
else
    user_name = string_capitalize(user_name)
end
```

```ruby
#Saves menu choices in a variable as a hash of key value pairs
#Select from prompted choices, allows cycling of menu once selection reaches bottom of options and oves user to specific interface as per selection

while true
    choices = {"Use Pokedex": 1, "Download Your Team": 2, "Exit": 3}
    puts ("\n") * 2
    choice = prompt.select("Welcome to the Pokedex App Trainer #{user_name}! What would you like to do?", choices, cycle: true)
    case choice
    when 1
        puts ("\n") * 2
        choices = {"Search Specific Pokemon": 1, "Browse Pokemon Database": 2, "Go Back": 3}
        choice = prompt.select("Accessing Pokedex database ... What would you like to do?", choices, cycle: true)
        puts ("\n") * 2
        case choice      You, 2 days ago • fixed single pokemon lookup api call on main.rb
            when 1
                begin
                    choice = prompt.ask("Which Pokemon would you like to search?")
                    choice = choice.downcase
                    search_result = Pokedex.new(choice)
                    search_result.paint(choice)
                    search_result.get_pokemon_general_info(choice)
                    search_result.get_pokemon_species_info(choice)
                    search_result.get_pokemon_stats(choice)
                    search_result.get_pokemon_desc_info(choice)
                    search_result.get_pokemon_habitat_info(choice)
                rescue => e
                    puts "Not a valid search/Missing data from PokeApi".colorize(:red)
                end
            when 2
                begin
                puts ("\n") * 2
                all_pokemon_names = self.get_all_pokemon_names
                choice = prompt.select("Please select a pokemon to view more OR start typing a pokemon name to filter ...", all_pokemon_names, filter: true)
                choice = choice.downcase
                search_result = Pokedex.new(choice)
                search_result.paint(choice)
                search_result.get_pokemon_general_info(choice)
                search_result.get_pokemon_species_info(choice)
                search_result.get_pokemon_stats(choice)
                search_result.get_pokemon_desc_info(choice)
                search_result.get_pokemon_habitat_info(choice)
                #External PokeApi has some missing habitat values set as NIL, crashes app if not rescued
                rescue => e
                    puts "Habitat => N/A".colorize(:red)
                end
                puts ("\n") * 2
            when 3
        end
    when 2
        puts ("\n") * 2
        puts "in progress"
        #parse json file of pokemon team to pull data for whole team
    when 3
        puts ("\n") * 2
        puts "Goodbye Trainer #{user_name}!"
        puts ("\n") * 2
        exit
    end
end
```

- While true keeps menu system in loop; cycles back to start
- Nested option using tty-prompt (select 1, 2 or 3)
- Calls upon methods to pull data from api
- .colorize to emphasise errors (gem: colorise)
- Methods separate api calls depending on data type
- \n puts to space command inputs (readability purposes)

```ruby
#method to print applicatin logo colourised
def self.logo
    puts "
```



```ruby
".colorize(:blue)

    puts "                                " + "RUBY POKEDEX APP".colorize(:color => :blue, :background => :yellow)
end
```

Method used to call app title screen

```ruby
def get_pokemon_species_info(pokemon_name)
    pokemon_name = pokemon_name.split("-")[0]
```

Line remove all special suffixes - leads to same data point in api
(remove duplicates, reduce nil data calls)

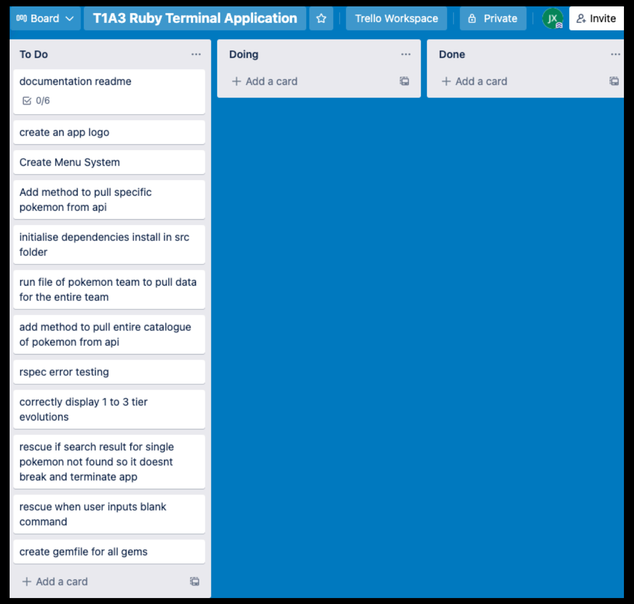
```ruby
puts "Your team is #{pokemon_name_file}"
puts "What would you like to name your file?"
file_directory_name = gets.chomp
file_name = "./saved_teams/#{file_directory_name}.txt"
File.write(file_name, pokemon_name_file)
pokemon_name_file = []
end
```

Saves pokemon name into an arr txt

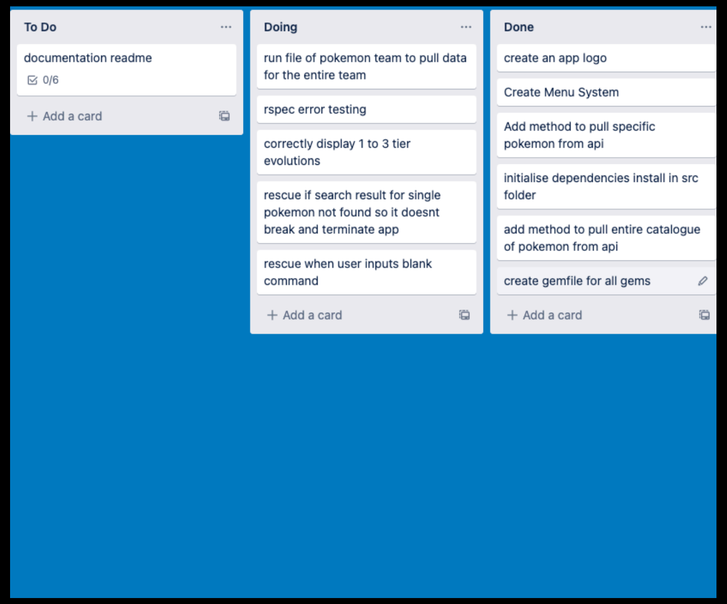
```ruby
new_hash.merge!(search_result.get_pokemon_general_info(item))
new_hash.merge!(search_result.get_pokemon_species_info(item))
new_hash.merge!(search_result.get_pokemon_stats(item))
new_hash.merge!(search_result.get_pokemon_desc_info(item))
new_hash.merge!(search_result.get_pokemon_habitat_info(item))
JSON.pretty_generate(new_hash)
puts ("\n") * 2
export_yn = prompt.yes?("Export to file?")
if export_yn == true
file_name = "./saved_teams/#{item.to_s}_export.txt"
File.write(file_name, new_hash)
end
```

Runs search depending on pokemon in saved arr.txt, saves pokemon as JSON file

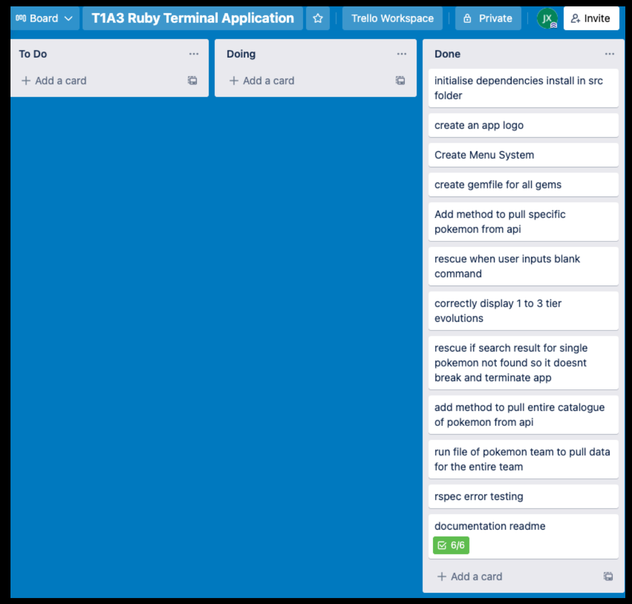
Start



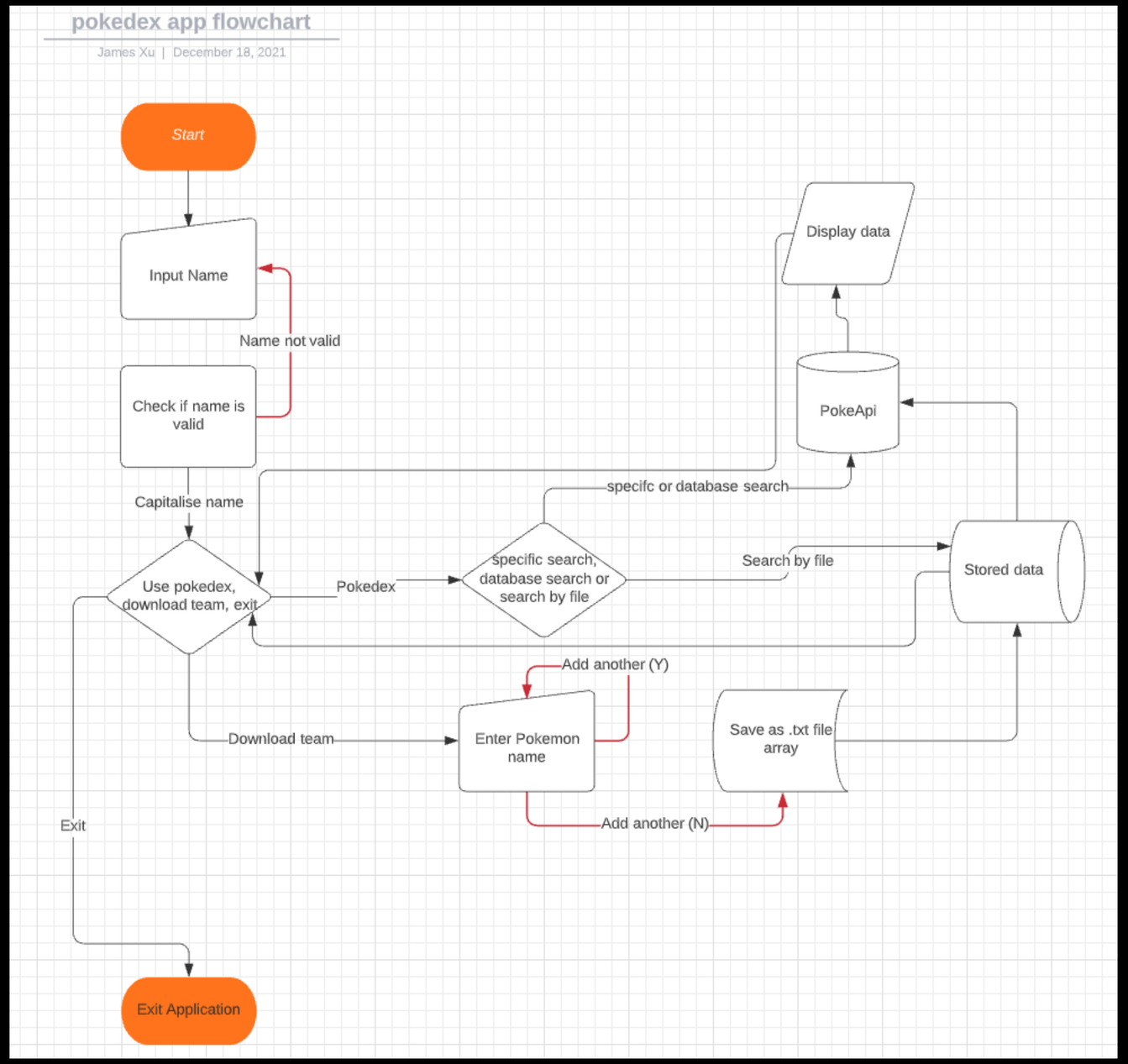
Doing



Done



Flowchart


pokedex app flowchart
James Xu | December 18, 2021
Start
Input Name
Name not valid
Check if name is valid
Capitalise name
Use pokedex, download team, exit
Pokedex
specific search, database search or search by file
specifc or database search
Display data
PokeApi
Search by file
Stored data
Add another (Y)
Download team
Enter Pokemon name
Save as .txt file array
Add another (N)
Exit
Exit Application

Development Process Review

Challenges, favourite parts, ethics?

- Challenge
  - Utilising APIs
  - Underestimating features
  - Keeping code DRY
  - Debugging

Development Process Review

Challenges, favourite parts, ethics?

- Favourite Parts
  - Images (rmagick, paint)
  - A (somewhat) operating application!
  - Solving issues (self-troubleshooting)

Development Process Review

Challenges, favourite parts, ethics?

- Ethics
  - Ensuring API is open-source
  - Not profiting off copyrighted sources (pokemon)