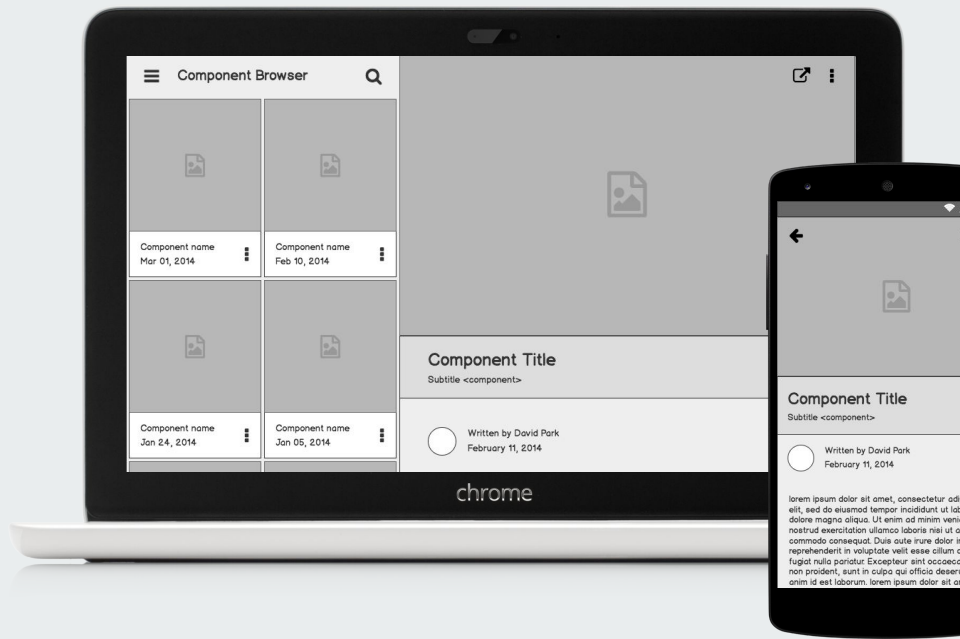




Dylan Esteban

Terminal App | Coder Academy 2021

MVP

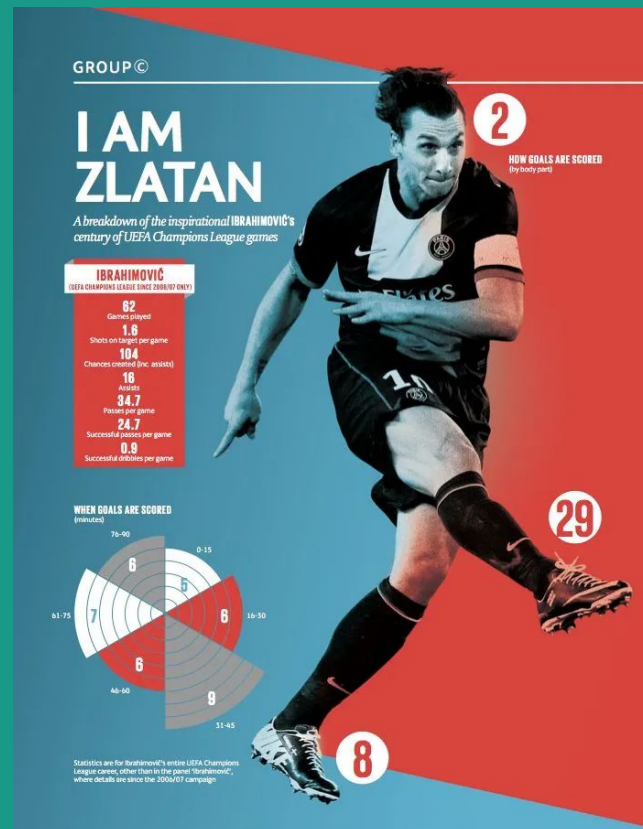


App Idea | Sports Stats App

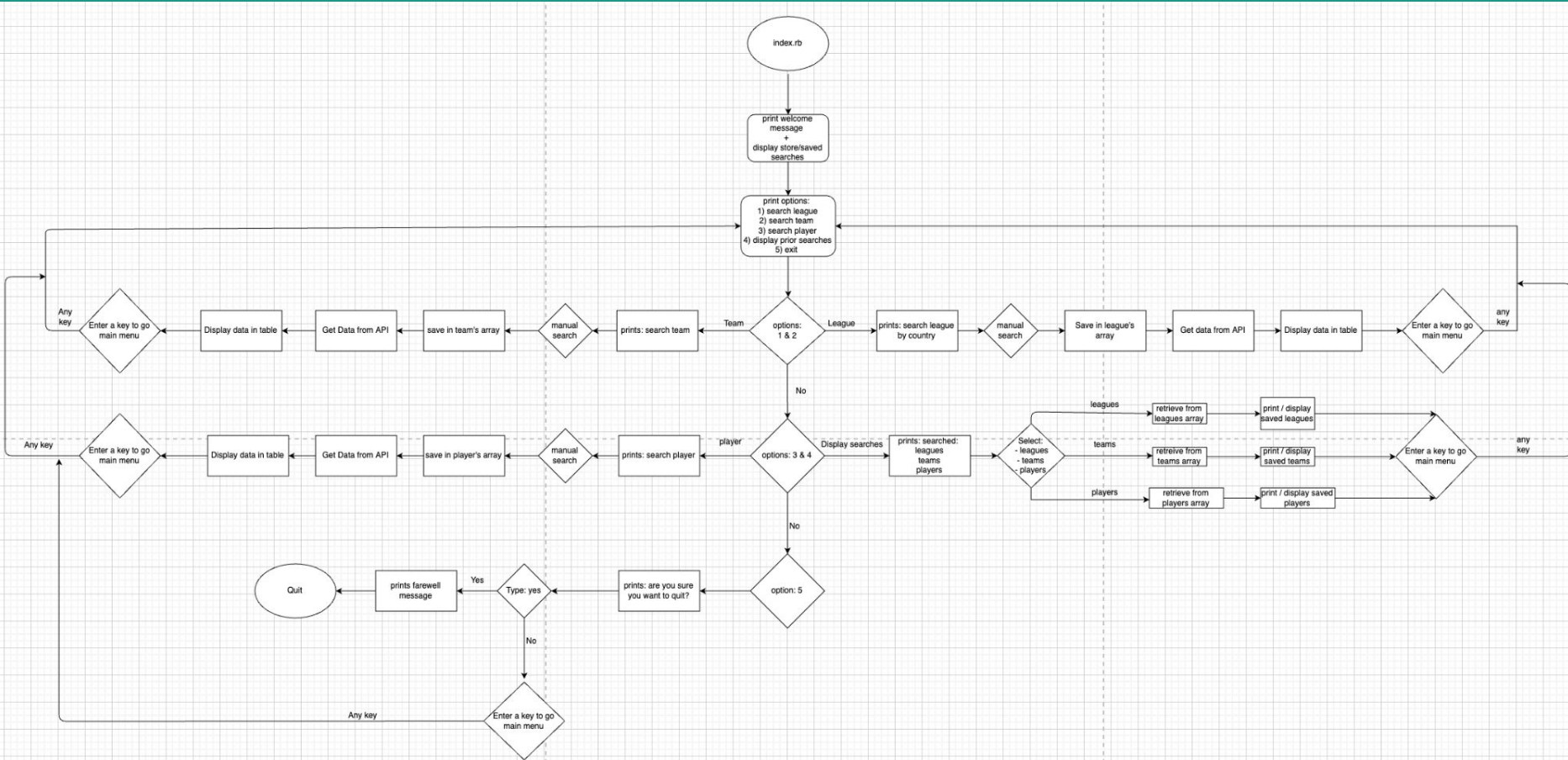
1. Design Process
2. Review of some features and walk-through
3. Review of build: challenges, favourite features, etc

Purpose:

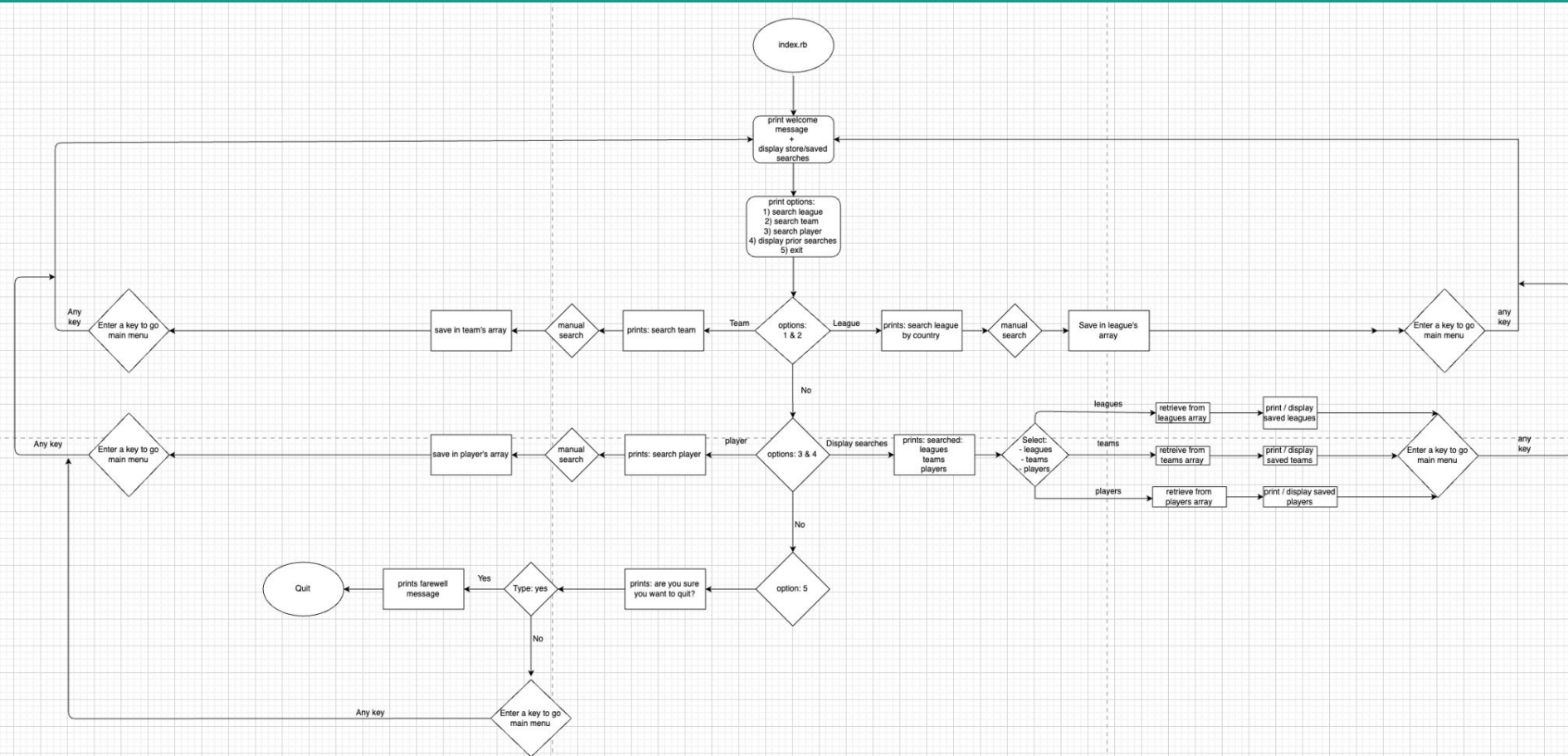
- **Problem:** statistics are important and helpful
- **Need:** a means to easily access them
- **Action:** write a Ruby program



Design process | Original Flowchart



Design process | Current Flowchart



Features | Interactive Menu

- Empty arrays to store user selections
- Short-circuit logic for interactive menu ensuring efficient code + DRY
- Menu storing prior input

TERMINAL

```
zsh + v [ ] [ ] [ ] [ ] [ ] [ ]  
  
Hi! Welcome to Footy Stats Look-Up!  
You have no prior searches. Let's change that!  
Do you have a favourite league, team or player?  
Hit display to view your prior searches! (Press  
↑/↓ arrow to move and Enter to select)  
▶ League  
  Team  
  Player  
  Display  
  Exit
```

```
# empty arrays for user in-put  
fav_league = []  
fav_team = []  
fav_player = []  
  
#menu loop + gems: colorize & tty-prompt  
while true  
  puts "Hi! Welcome to Footy Stats Look-Up!".red.italic  
  
  if ( fav_league.empty? && fav_team.empty? && fav_player.empty?)  
    puts "You have no prior searches. Let's change that!"  
  else  
    puts "You currently have #{out_put(fav_league.length, 'league')}, #{out_put(fav_team.length, 'team')} & #{out_put(fav_player.length, 'player')}"  
  end  
  
  #menu user-input + Gem  
  menu_input = prompt.select("Do you have a favourite league, team or player? Hit display to view your prior searches!", %w(League Team Player Display Exit))
```

Features | Save feature

Hi! Welcome to Footy Stats Look-Up!

You currently have 1 searched league, 0 searched teams & 3 searched players
Do you have a favourite league, team or player? Hit display to view your prior searches!

League
Team
Player
► Display
Exit

Method menu

```
# menu for interface method
def out_put(length, keyword)
  return "#{length} searched #{keyword}#{'s' if length != 1}"
end
```

Saving & displaying prior searches in menu

- Saves searches to empty array for later usage
- Displays number of searches from each category
- Allows user to manually keep track of searches
- Method pluralises results if length isn't equal to 1
- **1 league** but **0 teams** & **3 players** searched
- Not super important but aids in readability of app & language conventions

Features: Display prior searches

- Saved in an empty array and called upon when user wishes to find prior searches, be they leagues, individuals teams, or specific players
- Tty-prompt.select provides simple interface for navigating selections

Action needed:

- solve display arrays bug
- tty-table to store / present data for stats side by side comparison

```
You currently have 0 searched leagues, 0 searched teams & 3 searched players
Do you have a favourite league, team or player? Hit display to view your prior searches! Display
Searched Players
messi
ronaldo
sergio ramos
Press any key to continue
█
```

```
when 'Display'
  display = prompt.select("Searched", %w(Leagues Teams Players))
case display
when 'Leagues'
  puts fav_league
when 'Teams'
  puts fav_team
when 'Players'
  puts fav_player
end
```

Features | Exit feature

```
when 'Exit'  
  puts "Are you sure you want to quit? Type 'YES' to exit or type any key to return to main-menu"  
  quit_choice = gets.chomp  
  break if quit_choice.downcase == 'yes'  
else  
  # puts "Error: Invalid Input. Please try again."  
end  
puts "Press any key to continue"  
# gets  
system "clear"  
end  
puts "Goodbye!"
```

Exit

```
Are you sure you want to quit? Type 'YES' to  
exit or type any key to return to main-menu  
yes  
Goodbye!
```

Exiting Menu

- Case statement for options: when exit is used
- Break ends loop + transfers control out of the block of code, out of the iterator that had invoked the block, and moves to first expression following iterator
- System clear, ends with goodbye
- Code needs to be re-written; functioning, but bit messy

Gems |



- tty-prompt
- Colorize
- API-Football,
- tty-table

```
# frozen_string_literal: true

source "https://rubygems.org"

git_source(:github) {|repo_name| "https://github.com/#{repo_name}" }

# gem "rails"

gem "colorize", "~> 0.8.1"

gem "tty-prompt", "~> 0.23.1"

gem "api_football_v3", "~> 0.0.2"

gem "tty-table", "~> 0.12.0"

@client = ApiFootballV3::Client.new do |config|
  config.api_key = "3cb2644b3d9b2bf1350557f6c74e1a21"
  config.base_url = "https://v3.football.api-sports.io/"
end
```

```
# need to refactor this & use methods
case menu_input
when 'League'
  puts "Search league by country:"
  country = gets.chomp
  fav_league.push(country)
  # fav_league << country (?)
when 'Team' # consider drop down of teams instead
  puts "Search EPL team:" #restrict it to EPL only
  team = gets.strip
  fav_team.push(team)
  data = @client.teams(name: team.capitalize, league: 39, season: 2020)
  team_data = data["response"].first["team"]
  pp team_data
  team_statistics = @client.team_statistics(team: team_data["id"], league: 39, season: 2020)
  pp team_statistics

  team_statistics["response"]["goals"].each do |label, stats|
    puts "#{label}: #{stats}"
  end
end
```

- Colorize to add some colour and vibrancy to the user-experience
- TTY-Prompt powerful: helpful prompt types for gathering input
- API-Football Gem: Ruby Interface to access API-Football v3: issues with calling API

To hook-up:

- Tty-table to display data

Hi! Welcome to Footy Stats Look-Up!

You have no prior searches. Let's change that!

Do you have a favourite league, team or player? Hit display to view your prior searches!

```
League
Team
▶ Player
Display
Exit
```

Build Review |

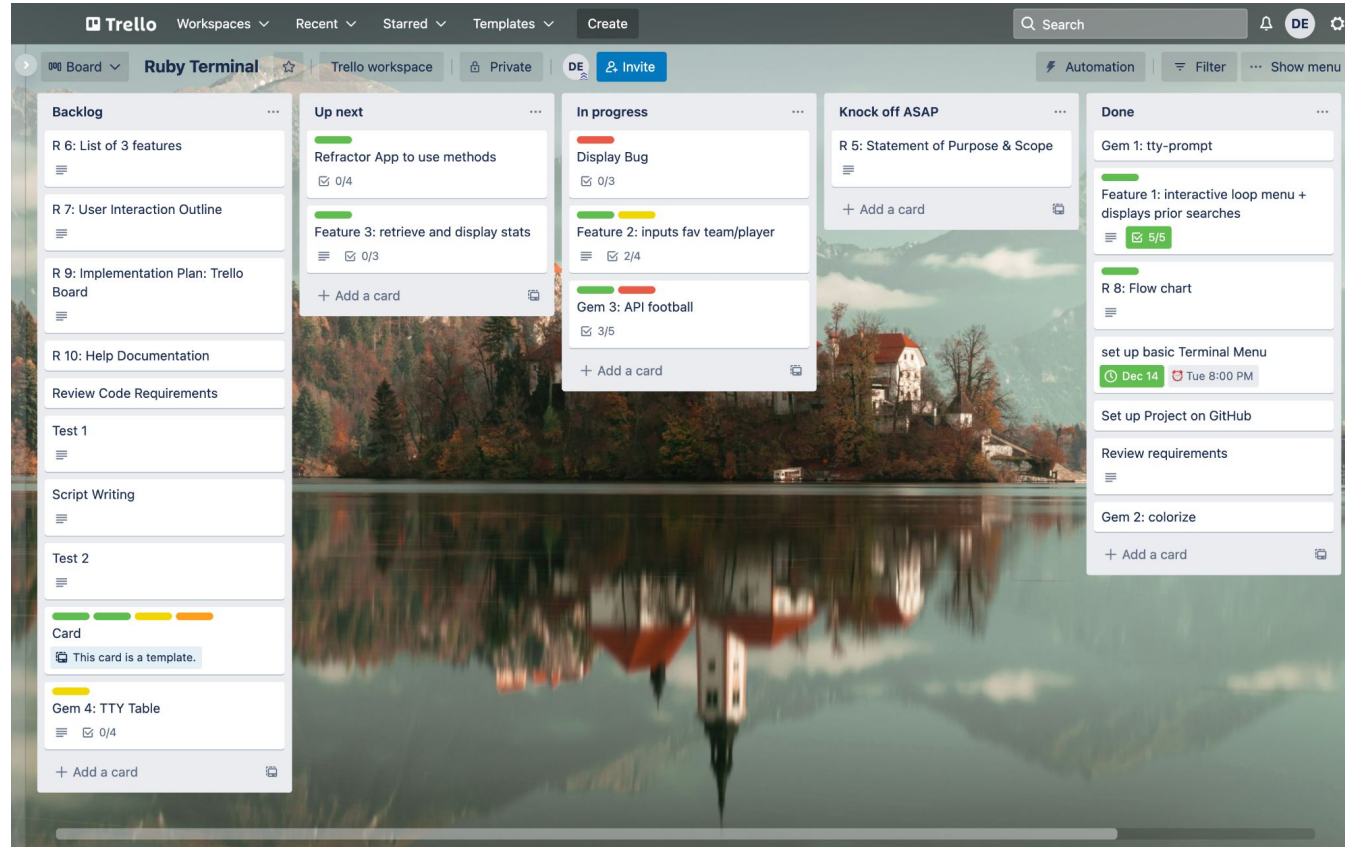
- Process: trello
 - Challenges
 - Favourite parts
-



Build Review: process

Process:

- Brainstorm
- Select idea
- Trello up n runnin'
- Flow-chart
- Some basic pseudocode
- Cry a little
- Code
- Revise learning material
- Code some more
- Cry again
- OMG it works.. I think..



Build Review: challenges & favourite parts

Challenges:

- Not focusing on the donut division: donuts and sprinkles = getting lost in the sauce
- Scaling back: biting off more than one can chew
- Indecisiveness and a lot of edited out code
- Maintaining a growth-mindset
 - We're all aboard a steady ship of navigators (educators) and rowers (students); just keep rowing
 - Trusting the process and comparing oneself to where you were yesterday

- Menu interface: accessible, easy to use
- Save feature which then displays prior searches
- Gems super fun
- Actually **building** something!



Thanks for coming!

