

---

---

# Ruby Terminal App

TV Show Organiser

## Developers:

Holly King

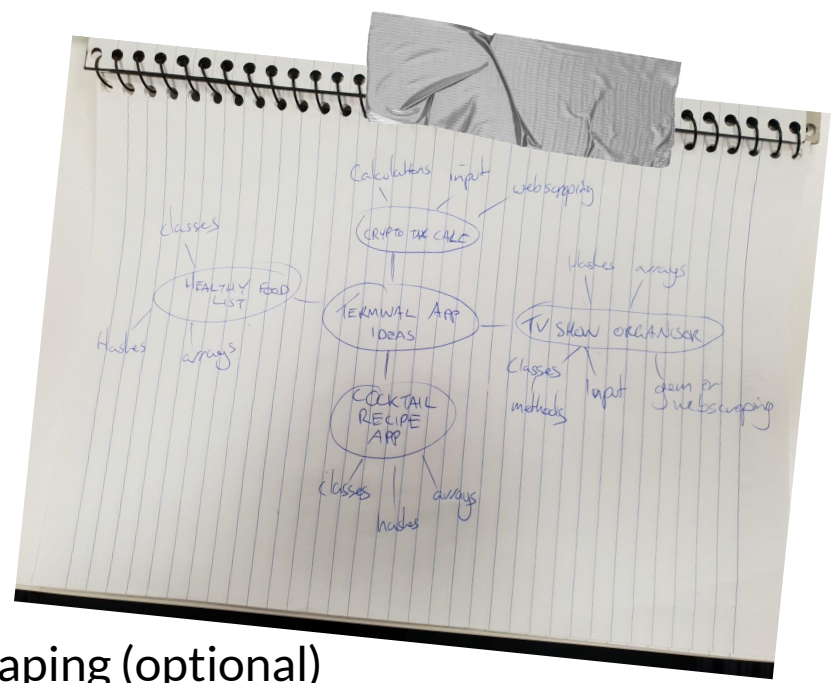
Kieran Lockyer

---

# Brainstorming

Initial ideas:

- **Crypto Currency Tax Calculator**
  - Web scraping was essential
- **TV Show Organiser**
  - Would require classes and web scraping (optional)
- **Cocktail Recipe Library**
  - Hardcoded Library with hashes and arrays
- **Health Food App**
  - Same as above



# Why We Chose TV Show Organiser

TV Show Organiser had the most pros going it's way:

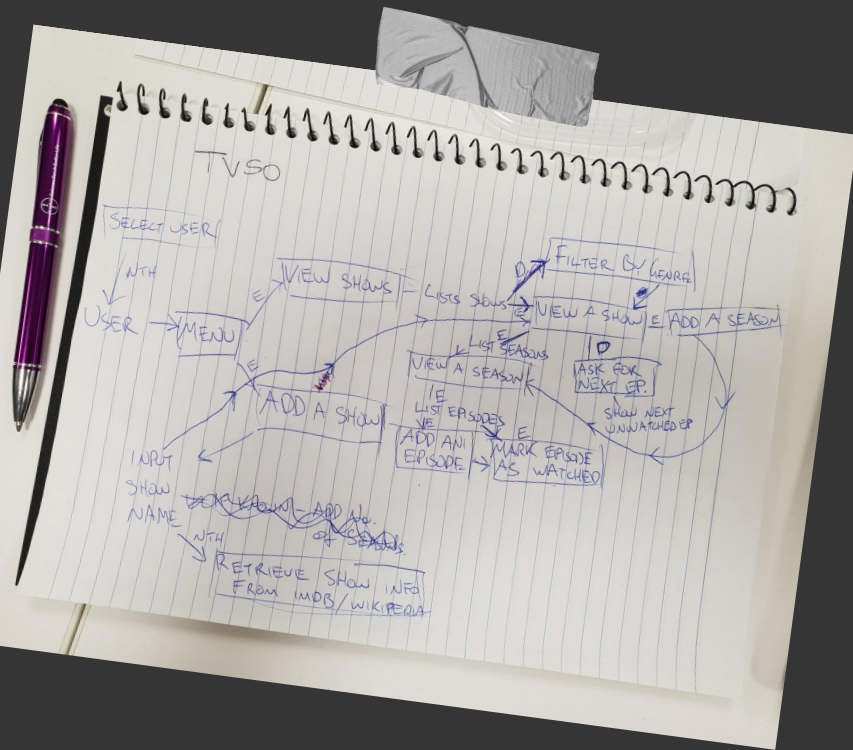
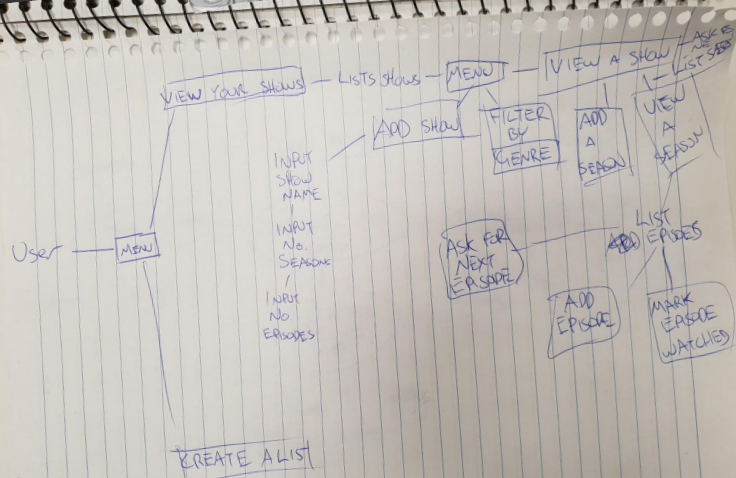
- Solving a real world problem
- Option to add web scraping, but not essential
- Used the most of our knowledge, without overcommitting.
- Fun!



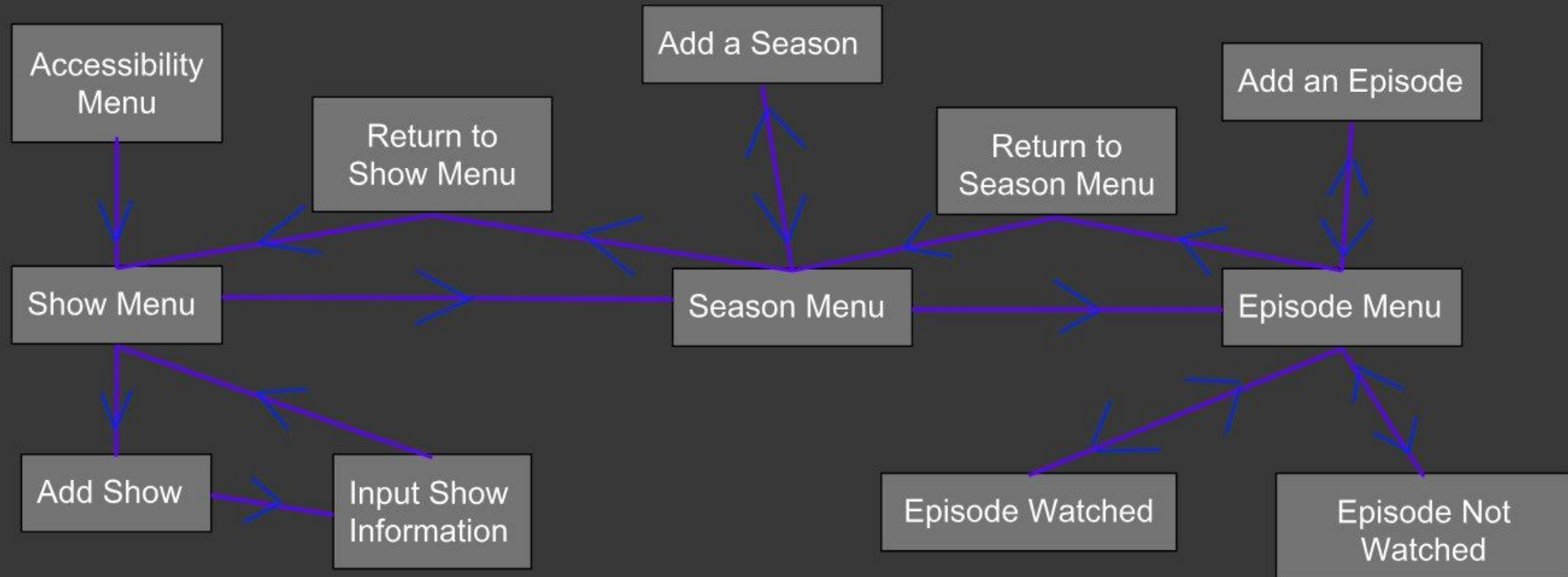
# 1. Development

- **Thought of Possible Use cases**
  - Multiple shows being tracked
  - Multiple users with different shows
- **Created User Workflow**
  - What is the most logical way to step through the menu
- **Broke Problems Down Into Features and Stories**
  - Which features are essential and which can we get away with adding later or not implementing at all.

# Workflow Versions



# Final Workflow Version!

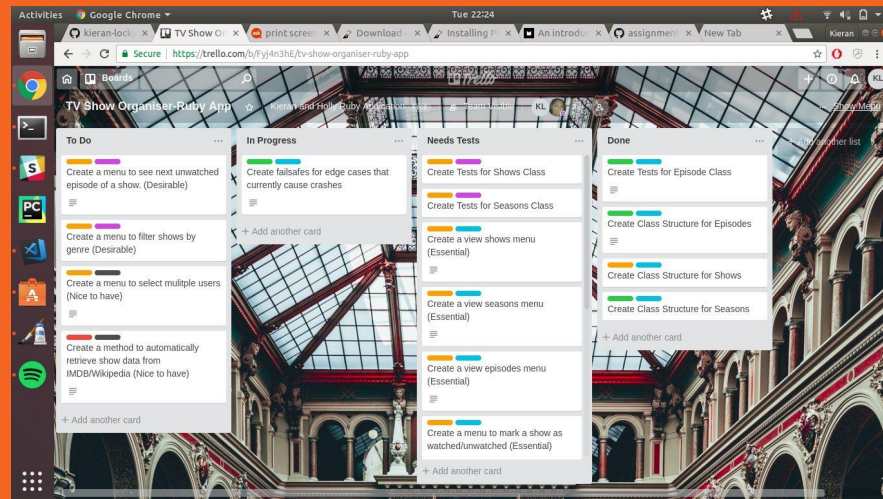


# Stories

As a user of the app I want to be able to:

- Add Shows to track
- Add different numbers of seasons to different shows
- Add different numbers of episodes to different seasons
- Not be confused when navigating the menus
- Be able to easily set show as watched
- Undo watching a show if I set it as watched by mistake
- Remove shows when I'm done watching them

# We used Trello to track our stories



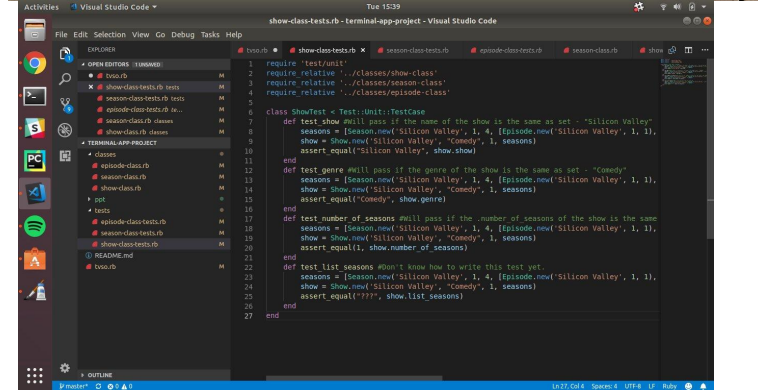
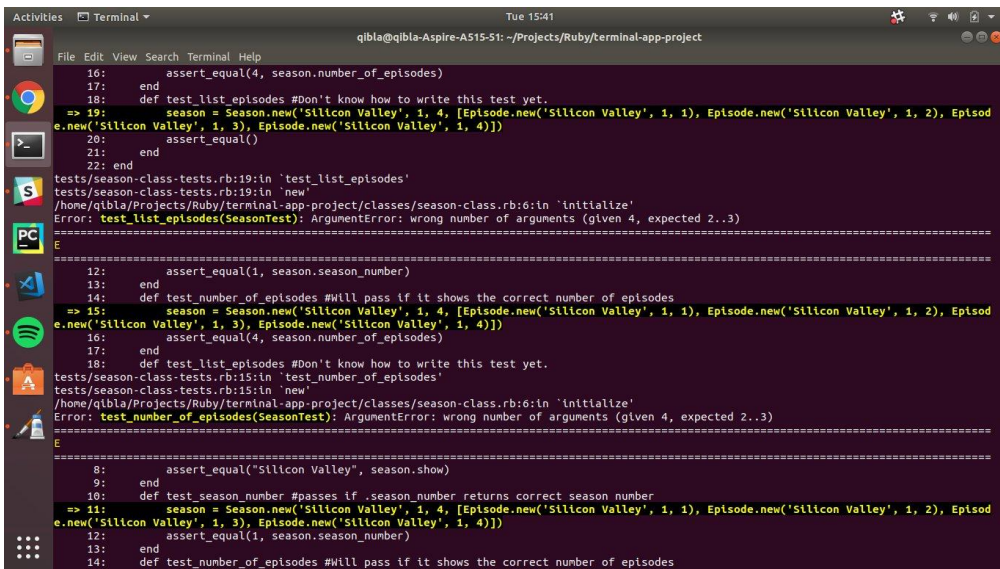
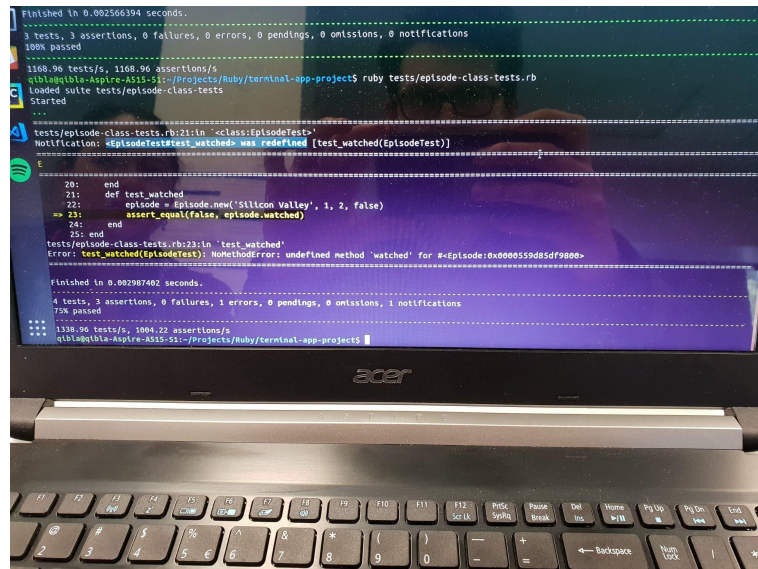
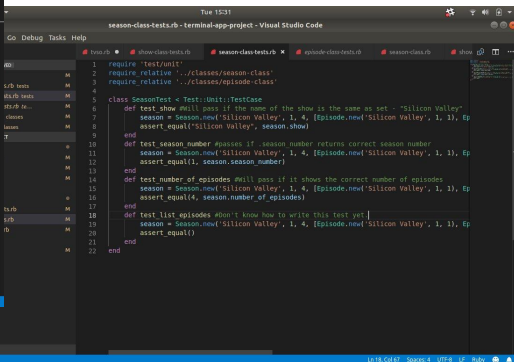
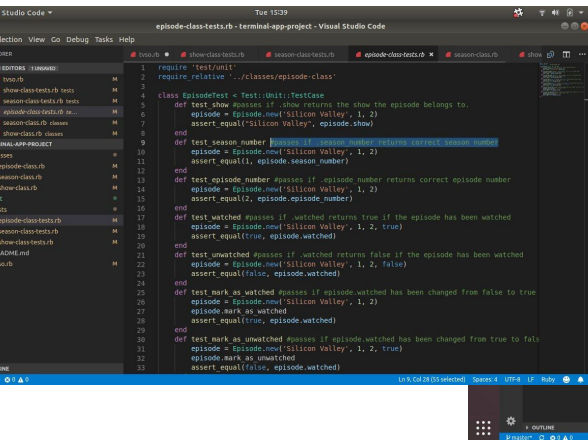


## 2. Starting to code

- **First step is to think about how to structure the data**
  - Episodes, Seasons and Shows as classes
  - Menu could be functions or class
  - Seasons and Episodes stored in arrays.
- **Next we wrote some kick ass tests!**
  - Testing is hard!!!
- **Then we proceeded to break everything and get headaches aka**
  - Palm's are sweaty, code spaghetti



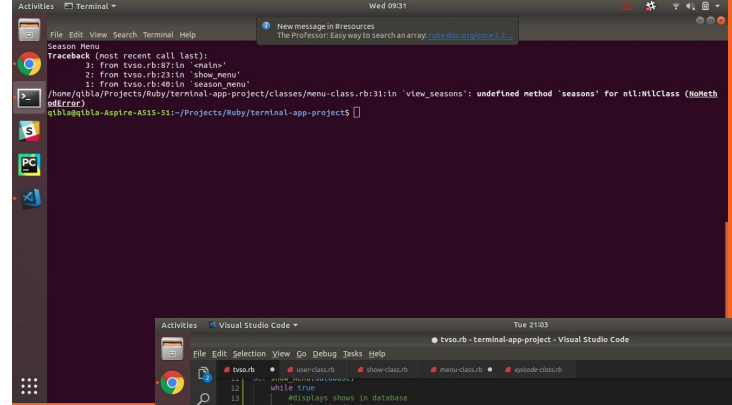
## Mixed success with tests



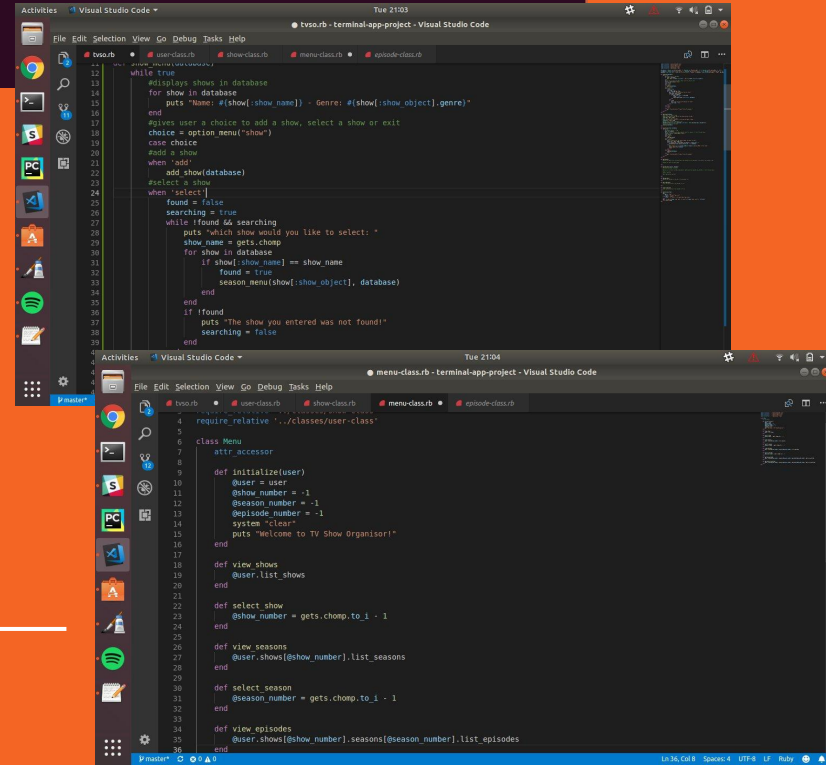
# Coding

## Lessons learned:

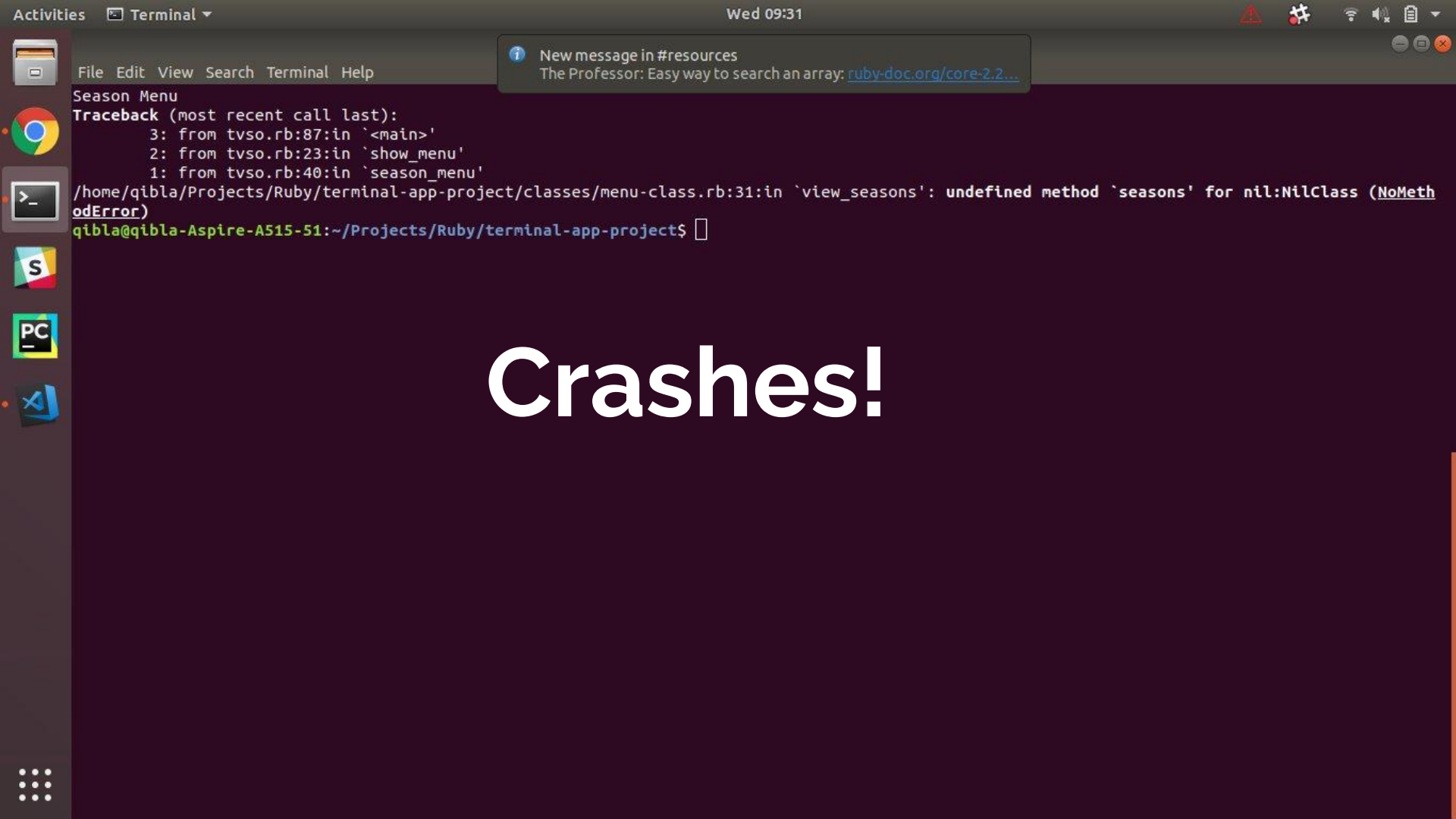
- Think before you type.
- Take breaks when you get frustrated.
- When your code gets messy, could a class do this better?
- Keep things simple and separate.
- Learn to read error messages.



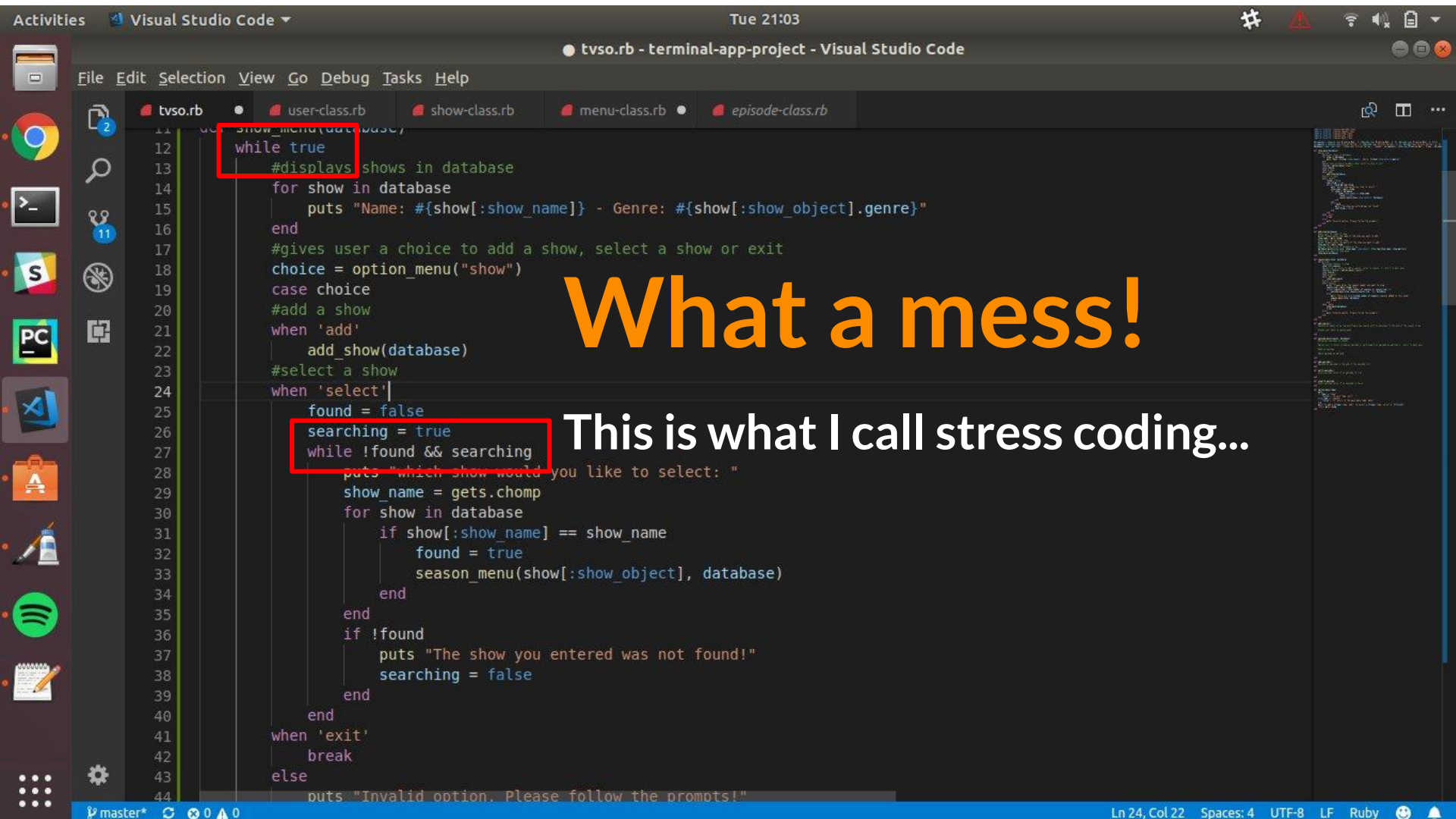
A terminal window showing a Ruby error message. The error message is: `view_seasons: undefined method 'seasons' for nil:NilClass (NoMethodError)`. The terminal also shows the command `git log` and the output of `git log`.



Two screenshots of the Visual Studio Code editor. The top screenshot shows a file named `show-class.rb` with Ruby code for a `Show` class. The bottom screenshot shows a file named `menu-class.rb` with Ruby code for a `Menu` class. Both files are part of a project named `terminal-app-project`.



# Crashes!



What a mess!

This is what I call stress coding...



```
4 require_relative '../classes/show-class'
5
6 class Menu
7   attr_accessor
8
9   def initialize(user)
10     @user = user
11     @show_number = -1
12     @season_number = -1
13     @episode_number = -1
14     system "clear"
15     puts "Welcome to TV Show Organisor!"
16   end
17
18   def view_shows
19     @user.list_shows
20   end
21
22   def select_show
23     @show_number = gets.chomp.to_i - 1
24   end
25
26   def view_seasons
27     @user.shows[@show_number].list_seasons
28   end
29
30   def select_season
31     @season_number = gets.chomp.to_i - 1
32   end
33
34   def view_episodes
35     @user.shows[@show_number].seasons[@season_number].list_episodes
36   end
37 end
```

# Breakthrough!

episode-class.rb

tvso.rb

show-class.rb

season-class.rb

menu-class.rb

user-class.rb




```
93     ESpeak::Speech.new("Episode Menu").speak
94   end
95   main.view_episodes #lists the episodes in season
96   print "Press 1 to mark an episode as watched, 2 to mark an episode as unwatched, 3 to add an episode or 4 to return to 'Season Menu'"
97   if audio
98     ESpeak::Speech.new("Press 1 to mark an episode as watched, 2 to mark an episode as unwatched, 3 to add an episode or 4 to return to 'Season Menu'").speak
99   end
100  choice = gets.chomp
101  case choice
102  when "1"
103    print "Please enter the episode number: "
104    if audio
105      ESpeak::Speech.new("Please enter the episode number: ").speak
106    end
107    if main.select_episode # sets the menu attribute for episode number to the corresponding episode number in the episode object
108      main.episode_watched #sets the episode as watched
109    end
110  when "2"
111    print "Please enter the episode number: "
112    if audio
113      ESpeak::Speech.new("Please enter the episode number: ").speak
114    end
115    if main.select_episode # sets the menu attribute for episode number to the corresponding episode number in the episode object
116      main.episode_not_watched #sets the episode as unwatched
117    end
118  when "3"
119    main.add_episode #adds an episode object to the season object
120  when "4"
121    episode_menu = false # sets condition to loop episode menu to false, breaking the loop so you don't return to the loop
122    season_menu(main, audio) #returns user to season menu to make further changes to the season
123  else
124    puts "Invalid Selection. Please try again!"
125  end
```

# Comments Galore!






← → ↻ [GitHub, Inc. \[US\]](#) | <https://github.com/kieran-lockyer/terminal-app-project/commits/master> ☆

Branch: master ▾

Commits on Sep 5, 2018

- Fixed bugs and added audio guide for accessibility**  
kieran-lockyer committed 6 hours ago  ddc**b100** <>
- Last bug fixed... I think...**  
kieran-lockyer committed 9 hours ago  f099bf**3** <>
- Mostly working menu with 1 bug, saves data to file and retrieves on r...** ...  
kieran-lockyer committed 9 hours ago  421214**9** <>

Commits on Sep 4, 2018

- Created menu class and basic menu workflow to intereact with object l...** ...  
kieran-lockyer committed 22 hours ago  010223**2** <>
- Added some screenshots to the ppt(presentation) folder and have flesh...** ...  
kieran-lockyer committed a day ago  7da903**c** <>
- Added files required to run, store classes, and test classes.**  
kieran-lockyer committed a day ago  14e321**8** <>
- Update README.md**  
kieran-lockyer committed a day ago Verified  ce2fae**e** <>
- Initial commit**  
kieran-lockyer committed a day ago Verified  3c87e0**0** <>

# GIT



## 3. Finishing Touches

### → Accessibility

- We added a text to speech option for visually impaired
- Advantage of OOP, very little code required to implement

### → Documentation

- We mostly stayed true to the original outline, with only minor changes to the user workflow.
- We did use a gem to add TTS so that needed to be added to pre-requisites



# README

## TV Show Organiser

### Problem Definition

Have you ever felt the pain of coming back to a show from a long hiatus (maybe due to being a good student and swapping Game of Thrones for your History text book so you can ace your final exam) and not remembering where you left off? Have you spent 20 minutes watching an episode, only to realise you've already seen it? How did that make you feel? If you're anything like me, you might have given up on watching that series all together rather than dig through episode by episode to try and find the right place to pick up again.

### Solution

TVSO is a simple app that runs in the terminal that allows you to easily keep track of all your favourite shows from one easy to use location. Simply load up TVSO, add the details of a show you want to track, and mark off each episode as you watch it.

Never waste your time unintentionally rewatching shows again.

### Ethical Considerations

We are not responsible for any shows obtained by illegal methods such as torrents or bootleg dvd's. If you enjoy the work, support the creators by paying for it.

### Features:

- Add TV shows you are currently watching
- Add seasons to your shows
- Add episodes to your seasons
- View list of shows you have added
- View list of seasons and episodes for each added TV Show
- Mark a show as watched or not yet watched

### Prerequisites:

Espeak Ruby Gem This gem is required for TTS Accessibility functionality and must be required in tvsorb and menu-class.rb  
<https://rubygems.org/gems/espeak-ruby/versions/1.0.3>

### Launching the App:

Open up terminal and navigate to the directory that contains the terminal app files and type `ruby tvso.rb`

### Using the App:

The app was designed to take as little input as possible to navigate the menu. Each menu can be navigated by typing a number corresponding with the onscreen prompts.

Shut Up and Show Us the Code!

—



## 4. Closing

- **Coding is fun!**
  - ...when everything is working the way you want it to...
- **Learn to enjoy the struggle.**