

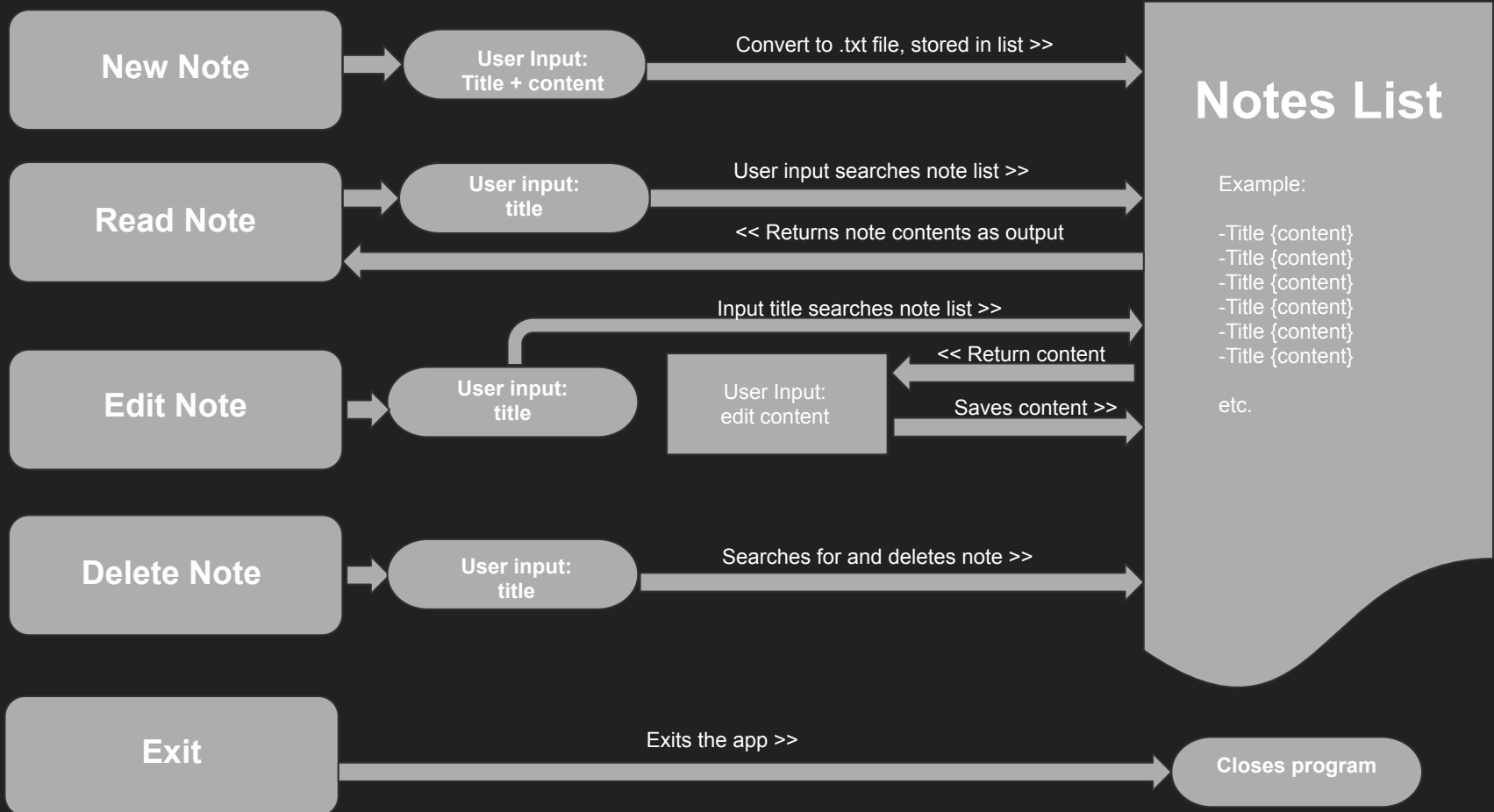
# Assignment 3 - Terminal App

App name: SuperNote

## DEVELOPMENT PLAN - extract from my README.md

- My application, SuperNotes, is a terminal application designed to do four things: take notes, read those notes, edit those notes, and delete those notes if necessary. SuperNote takes input from a user in the form of a note title and note content, then opens a text file and saves the user's input (the note heading and content) to the text file along with the date which the note was created. The user can also use SuperNote to look through a list of their created notes, or search for them by name, and either edit the note or delete the note. Basically, SuperNote is super basic. But it works, and it's super.
- It will solve all the client's note taking needs; even though there are hundreds of other note-taking apps on the market, you can rest assured that none of them take a note as SUPERbly as SuperNote!
- The target audience is anyone who wants to jump in on the fast-paced and always-expanding world of note-taking. From office workers to soccer-mums, if you need to take a note, I've got the app for you!
- I could explain how to use it here, under the Requirement 5 heading as asked, but I feel like that's a bit silly as Requirement 10 asks for the same thing, and I don't want to repeat myself. I got things to do.

## Options Menu:



## Gems used, and Methods created (to keep my code DRY)

supernote.rb X

README.md

supernote.rb

```
1  require 'date'
2  require 'pastel'
3  require 'tty-prompt'
4
5  prompt = TTY::Prompt.new(active_color: :yellow)
6  pastel = Pastel.new
7
8  def read(filename)
9    File.open(filename).each do |text|
10     puts text
11   end
12 end
13
14 def create(filename, content)
15   File.open(filename, "w") do |file|
16     file.write(content)
17   end
18 end
19
20 def note_list()
21   pastel = Pastel.new
22   puts pastel.yellow.underline("\nNote List:_____")
23   puts "\n"
24   puts Dir.glob("*.txt")
25   puts "\n"
26 end
27
28 def line()
29   pastel = Pastel.new
30   puts pastel.yellow.underline"_____""
31 end
32
```

## New Note method, with added Date function

```
66
67 case choice
68   when "New"
69     line()
70     puts pastel.yellow("\nGive your note a title:")
71     puts "\n"
72     filename = gets.chomp.to_s.gsub(" ", "_").downcase
73     filename = filename + ".txt"
74     if File.exists?(filename) == false
75       puts "\n"
76       line()
77       puts pastel.yellow("\nNow add something to " + "#{filename}:")
78       puts "\n"
79       date = Date.today
80       content = "Title: #{filename}" + "\nDate: #{date}" + "\n\n" + gets.chomp.to_s
81       create(filename, content)
82       puts pastel.bright_green("\nNote created!")
83       puts "\n"
84       choose2nd = %w(Back)
85       choice2nd = prompt.select("Go Back?", choose2nd).to_s
86       case choice2nd
87       when "Back"
88         puts "\n"
89         title()
90       else
91         puts pastel.red("Error!")
92       end
93     else
94       puts "\n"
95       puts pastel.red("Hey man. So, bad news, this note title already exists...")
96     end
97   end
98   when "Read"
```

# Ruby Gem: tty-prompt

Code:

```
title()

loop do

  puts pastel.yellow.underline("\nYour Options:_____")
  puts "\n"
  puts pastel.yellow("Read Note | New Note | Delete Note | Edit Note | Exit")
  puts "\n"

  choose = %w(Read New Delete Exit)
  choice = prompt.select("What would you like to do?", choose).to_s
```

In-app example (when creating a new note):

```
Your Options:_____
Read Note | New Note | Delete Note | Edit Note | Exit
What would you like to do? New
_____

Give your note a title:
my super new note
_____

Now add something to my_super_new_note.txt:
wassup wolf pack?

Note created!

Go Back? (Press ↑/↓ arrow to move and Enter to select)
• Back
```

## Challenges/Ethical Issues

The biggest challenge so far was not realising that there is a built-in Ruby function which lists all files, which can specify file type (`Dir.glob("*.txt")`). Before I figured this out, I wasted two full days trying to list all notes in an array, update that array each time I created a new note or delete an existing note (the easy parts), then save that array in a aesthetically pleasing way to its own text file - 'list.txt' - in order to 'save' it when i closed the program. Of course there was a long list of problems trying to do it this way, many of which came from deleting a note then trying to update (actually *overwrite*) the 'list.txt' file with the new array. Anyway, *issues, man. Issues*. `Dir.glob("*.txt")` saved SuperNotes.

My second biggest challenge will be to implement the ability to edit my notes in-app. I have *some* idea of how to do it but have not attempted it thus far as I know it will involve hours of trial and error and reading the documentation. Yay...

On the ethical side of things.... Well, it's a notes app with zero security and I make damn sure users know that in the README file. So unless someone is going to use it to store nuclear launch codes or the data they stole from your deranged aunt's vegan cat facebook group, I think I'm in the clear.

# My Favourite Part?

Getting it to work.

```
pepsi@LukeRazer: ~/my_ubuntu  x  +  v  -  □  x

      SUPER
NOTES
NotES!  notesss  NOTES  n0teS.
notes..  *notes!*  notes..

Welcome to SUPERNOTES

A SUPER app for taking HEAPS of NOTES and absolutely f*cking NOTHING else.

**finger guns**

Your options:-----
New Note
Read Previous Note
Exit Super Notes

What would you like to do? (Press ↑/↓ arrow to move and Enter to select)
▸ Read
  New
  Delete
  Exit
```



Preview time!