Lister App

R4 Link to Source Control

https://github.com/Yearnsmith/DerickYearnsmith_t1a3

R5 Software Development Plan

Purpose and Scope

Describe at a high level what app will do

Lister is a list manager. It allows users to create, edit, and save lists.

identify the problem it will solve and explain why you are developing it

1. Maintaining Lists

Makers and Creators use many kinds of lists:

- Kanban Planning
- Dependencies
- issues
- future projects
- new features
- lists of important lists!

Often ideas coem to us in the middle of an unrelated task. We have a choice to either let the idea go free or break our flow, load up a .txt, .doc, or pull out an old fashioned notepad and pen to write them down. For the creative type who uses a CLI, Lister enables them to quickly add to, remove from, or export a list from their CLI without changing directory, or opening a new terminal.

Lister also allows users to interact with lists in an interactive, distraction-free environment, with options to edit list items, and re-order them.

I am developing this app to help people like myself keep on task and not be distracted by new, exciting thoughts as they arise. It also will allow me to:

1. Work with file I/O
2. Feed inputs from the command line into the list
3. Perform interesting tasks with arrays, hashes, and other data-types
4. I can eventually port it to a Rails app, or other GUI interface for the Interactive Mode

identify the target audience

Lister's target audience is creators who work in a CLI, and don't enjoy being distracted or discarding cool ideas.

It's also aimed at people who like making lists.

explain how a member of the target audience will use it

There are two ways to use Lister:

CLI/Quick mode

`lister <flag> <list-title> [ <item>...]`

Flags

- -a, –**a**dd
- append an item to a list
- -c, –**c**reate
- create a new list
- -e, –**e**cho
- print a list to command line
- -r, –**r**emove
- remove an item from a list
- -x, –e**x**port
- export a list to `Lister` folder in `My Documents`

List names are entered two ways: 1. Single words for titles with 1 word *e.g.* `Ideas` 2. Strings for list titles with multiple words *e.g.* `"Killer Novel Ideas"`

The item to be added or removed

Interactive Mode

`lister [<file-name>]`

will load lister in Interactive Mode, along with the file in the argument.

If the file doesn't exit, Lister will create the file, will create the file and interact with it.

Running `lister` without arguments will load lister into a cleared screen and the user will be prompted to create or load a new list

Interactive mode clears the screen, and allows the user to interact with the list within a user-interface environment.

Stretch Goal *Use* `xterm` *to run* `smcup` *(or* `\e[?1049h`*) to act like less and open an altscreen.*

Operating within Interactive mode

Using Interactive Mode, Lister makes use of TTY-Prompt to give user an interactive list of options to choose between.

There is a new screen each time the user selects an item, which changes depending on the function.

R6 Develop a list of features that will be included in the application.

Interactive Mode

Interactive Mode is just that: the screen will clear, and users will be presented with a menu for interacting with Lister. This feature will make heavy use of the `TTY-Prompt` and `colorize` gems. It will use a control structure for selecting which feature to use.

The user will first be presented with a menu asking about the following options:

- Create List
- Edit List
- Delete List
- ¿Export List?

Create List

This is the main feature of the Lister. The app will prompt the user for a title, and instantiate a new `List` object. All list items will

Edit List

Add Item

User can add an item to the list, and the program will append the item to the list array. It does this by invoking an `.add-item` method in the `list` class.

Remove Item

Save List

Export List

Line Mode

In Line Mode, the user will execute and run the program without an interface appearing.

This will be part of a flow control loop at the beginning of the program interpreted code, checking which flags have been executed, and performing one of the following actions

```
lister -a <list-name> <list-item>
```
Lister will take `$3`'s value and `append` it to a hash with the `:name` of `$2`'s value. Lister will print a confirmation the item has been appended. The program achieves this by invoking an `.add-item` method in the `List` class.
Lister will invoke `.create_list` from `List` and pass in the shell variables.

```
lister -r <list-name> <list-item>
```
Will do the same, except it will `pop` `$3` from `$2` by invoking `.remove_item` from `List`. If the list, or list item doesn't exist, Lister will inform the user and close.
stretch goal: *Suggest a list based on similar names. But ew, regex*

```
lister -e <list-name>
```
Lister will `puts` the list into the terminal, by overiding the `.to_s` method in `List`.

```
lister -x <list-name>
```
Lister will invoke the `export_list` method in `List` class, and create a file in the ¿user's home's Documents directory.?

R7 Develop an outline of the user interaction and experience for the application.

Your outline must include:

- *how the user will find out how to interact with / use each feature*
- *how the user will interact with / use each feature*
- *how errors will be handled by the application and displayed to the user*

```
lister --add <list> <list-item>
```

4

R8 Develop a diagram which describes the control flow of your application.

- *show the workflow/logic and/or integration of the features in your application for each feature.*
- *utilise a recognised format or set of conventions for a control flow diagram, such as UML.*

R9 Develop an implementation plan

- *outlines how each feature will be implemented and a checklist of tasks for each feature*
- *prioritise the implementation of different features, or checklist items within a feature*
- *provide a deadline, duration or other time indicator for each feature or checklist/checklist-item*

Utilise a suitable project management platform to track this implementation plan

Trello Link

> Your checklists for each feature should have at least 5 items.

R10 Design help documentation which includes a set of instructions which accurately describe how to use and install the application.

You must include:
- *steps to install the application*
- *any dependencies required by the application to operate*
- *any system/hardware requirements*