# T1A3 – TERMINAL APP

Richard Cho

## AGENDA

```
 ___ __  __ ____  
|_ _|  \/  / ___| 
 | || |\/| \___ \ 
 | || |  | |___) |
|___|_|  |_|____/ 
```

## STATEMENT OF PURPOSE

IMS stands for Inventory Management System and it’s a simple terminal app that allows users to manage and store their inventory.

The target audience for this app are mainly small businesses and individuals.

Users can see this as a free alternative to accounting software such as MYOB (~ $30/month).

## FEATURES - CRUD

### Create

Create new objects under the Inventory class. Objects will have the following attributes:

1. Name
2. Price
3. Quantity

### Save / Load

When an object is instantiated, their attributes will be saved as a CSV file and also a YAML file. The YAML files acts as a database and will be automatically loaded when we start the app.

### Update

Once objects have been created, we can update their attributes by reading the YAML database and altering the relevant values.

### Delete

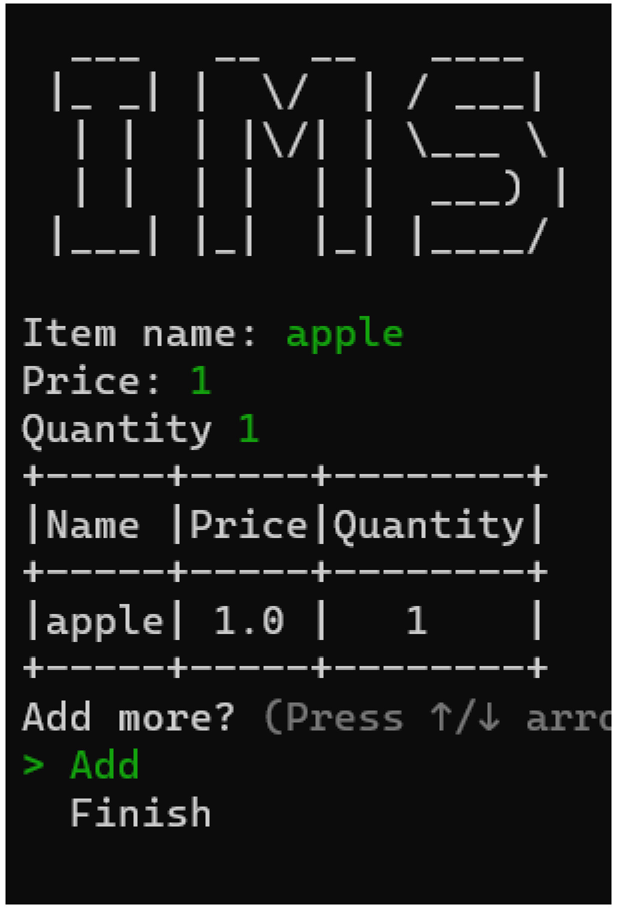
Read from the YAML file and remove an item from the list.

Create

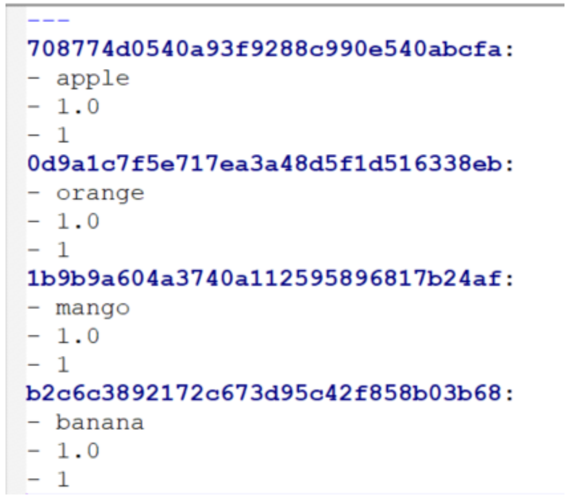
Save / Load

Update

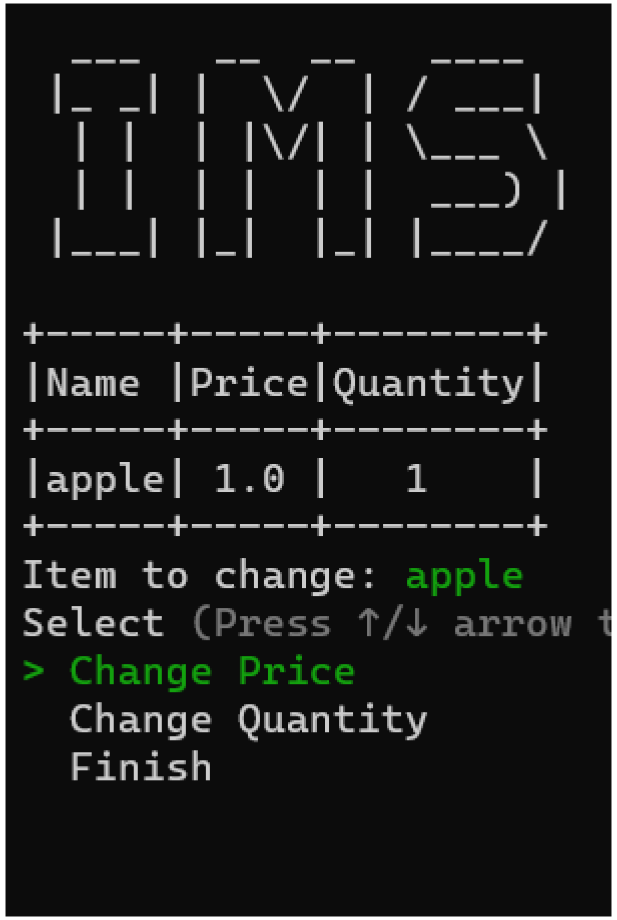
Delete



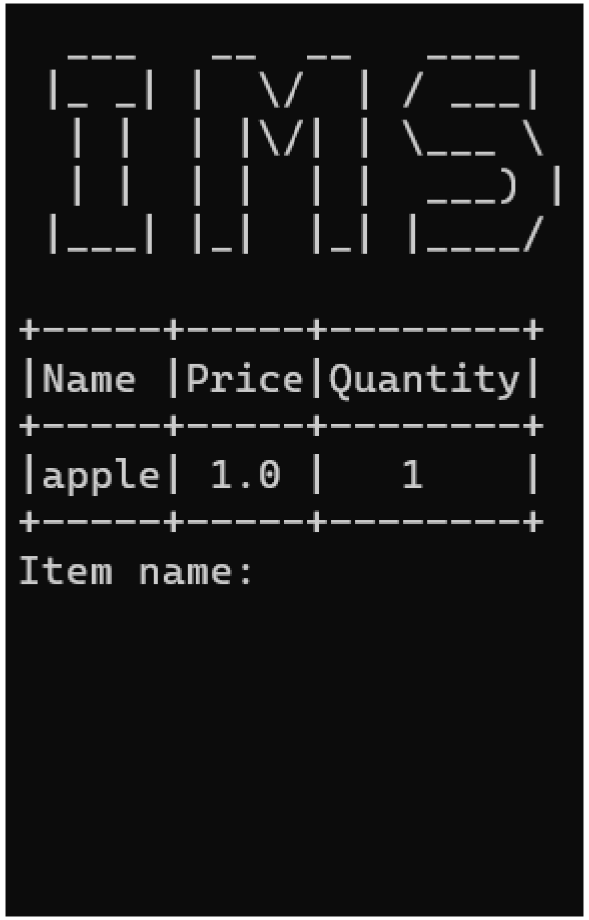
```
 ___ __  __ ____  
|_ _|  \/  / ___| 
 | || |\/| \___ \ 
 | || |  | |___) |
|___|_|  |_|____/ 

Item name: apple
Price: 1
Quantity 1
+-----+-----+--------+
|Name |Price|Quantity|
+-----+-----+--------+
|apple| 1.0 |   1    |
+-----+-----+--------+
Add more? (Press ↑/↓ arro
> Add
  Finish
```



```
---
708774d0540a93f9288c990e540abcfa:
- apple
- 1.0
- 1
0d9a1c7f5e717ea3a48d5f1d516338eb:
- orange
- 1.0
- 1
1b9b9a604a3740a112595896817b24af:
- mango
- 1.0
- 1
b2c6c3892172c673d95c42f858b03b68:
- banana
- 1.0
- 1
```



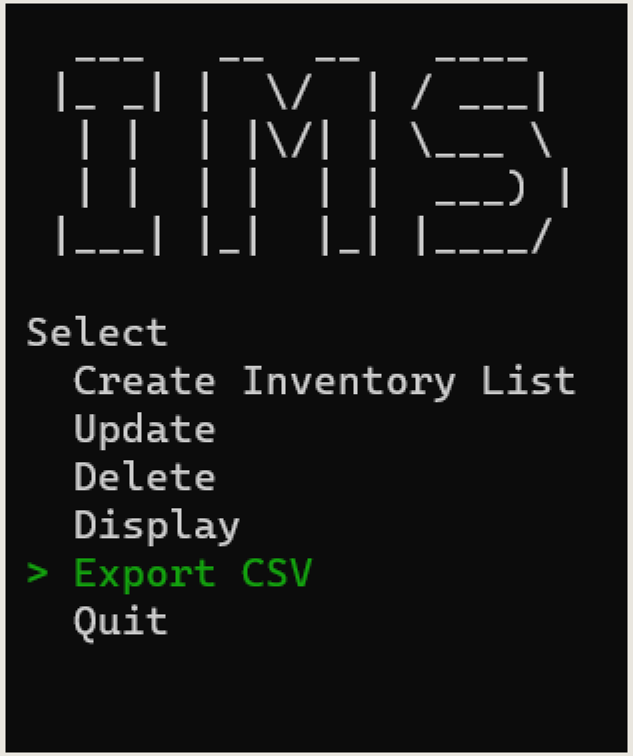
```
 ___ __  __ ____  
|_ _|  \/  / ___| 
 | || |\/| \___ \ 
 | || |  | |___) |
|___|_|  |_|____/ 

+-----+-----+--------+
|Name |Price|Quantity|
+-----+-----+--------+
|apple| 1.0 |   1    |
+-----+-----+--------+
Item to change: apple
Select (Press ↑/↓ arrow t
> Change Price
  Change Quantity
  Finish
```



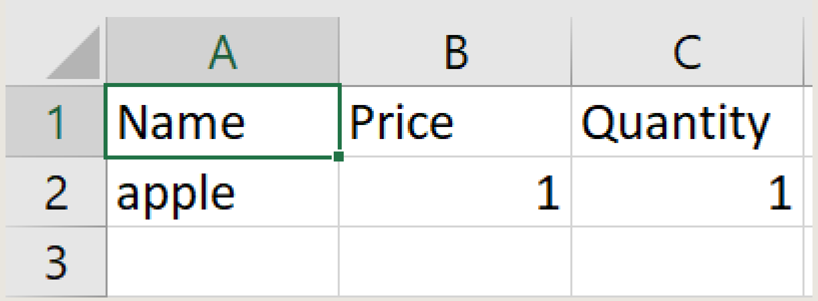
```
 ___ __  __ ____  
|_ _|  \/  / ___| 
 | || |\/| \___ \ 
 | || |  | |___) |
|___|_|  |_|____/ 

+-----+-----+--------+
|Name |Price|Quantity|
+-----+-----+--------+
|apple| 1.0 |   1    |
+-----+-----+--------+
Item name:
```

## FEATURES – EXPORT TO CSV

Exporting inventory data to a CSV file which can then be used on other programs such as Microsoft Excel.



```
 ___ __  __ ____
|_ _|  \/  / ___|
 | || |\/| \___ \
 | || |  | |___) |
|___|_|  |_|____/

Select
  Create Inventory List
  Update
  Delete
  Display
> Export CSV
  Quit
```

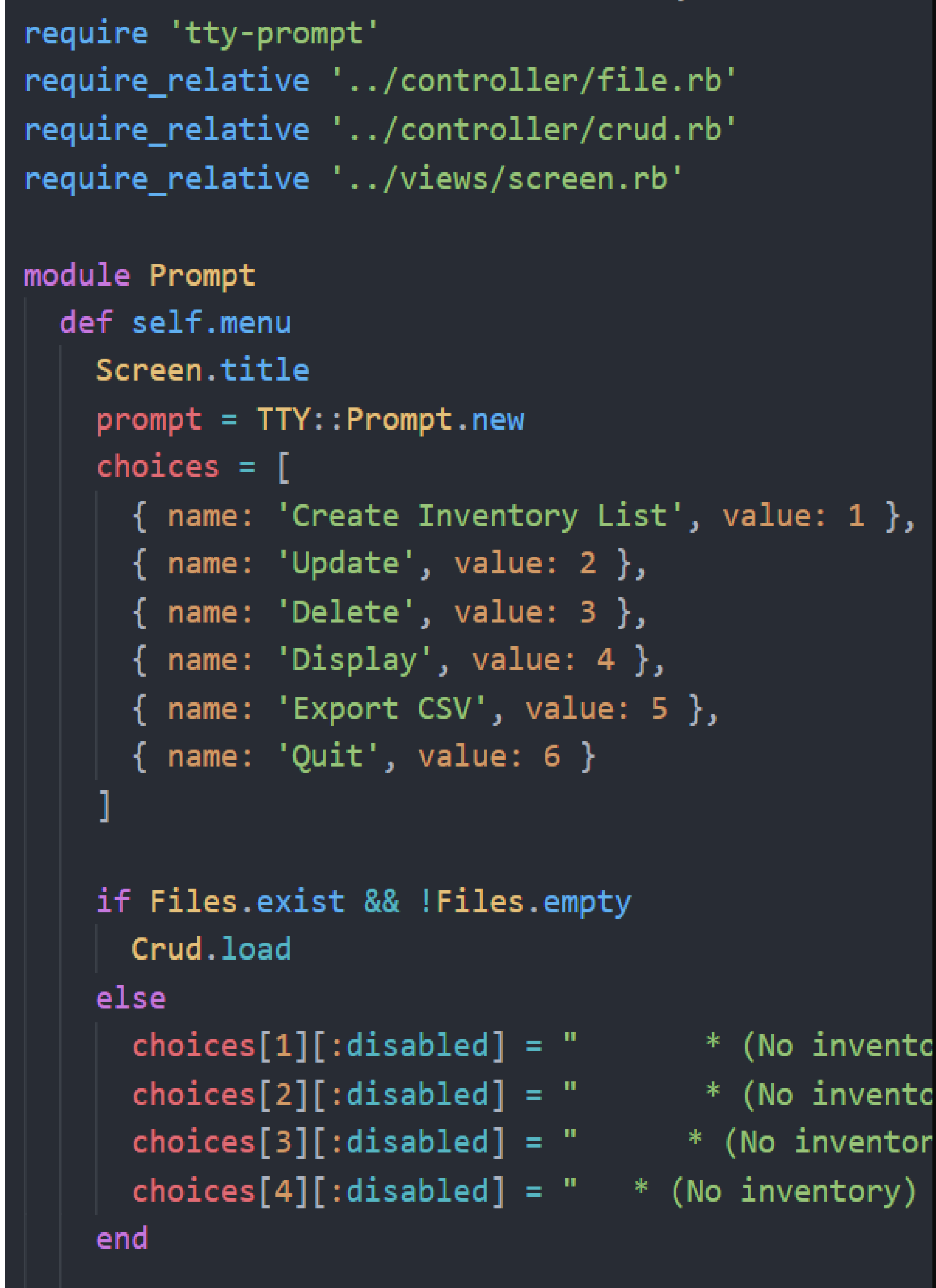


|   | A | B | C |
|---|---|---|---|
| 1 | Name | Price | Quantity |
| 2 | apple | 1 | 1 |
| 3 |  |  |  |

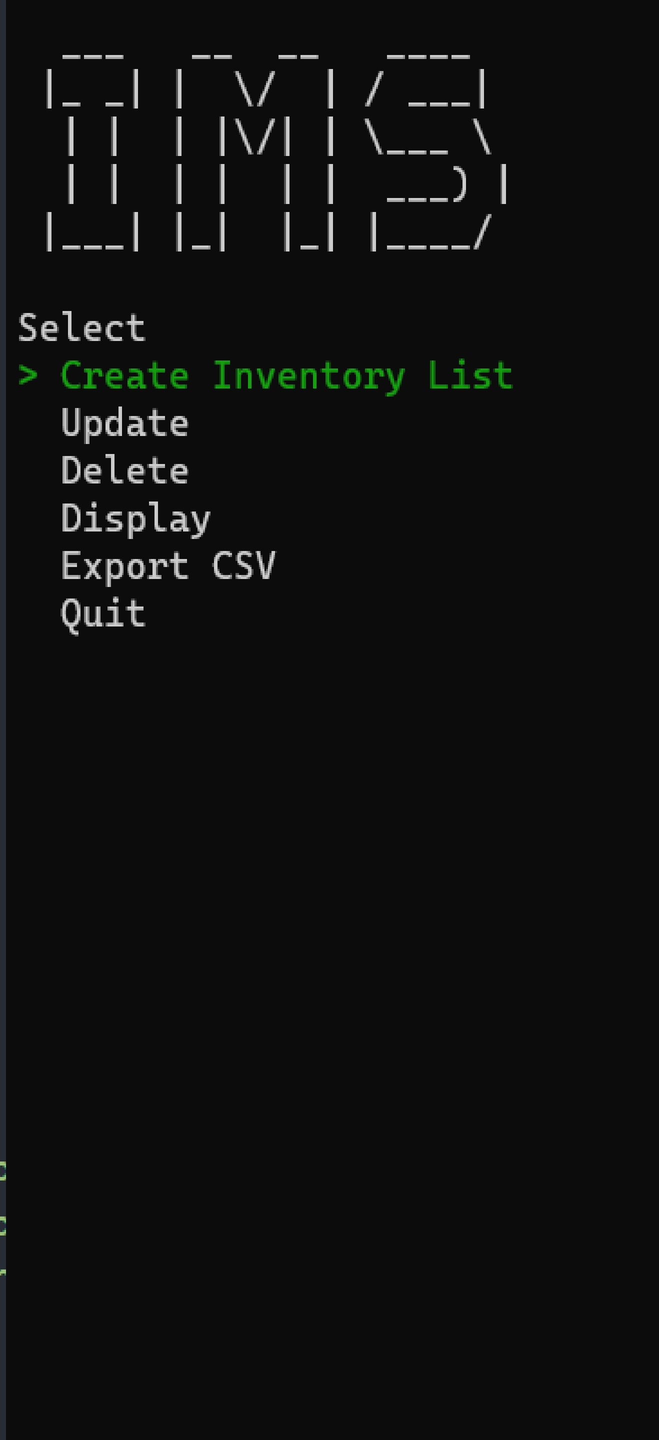
This is the main menu. Users can select from several different options once they have inputted some items into the app.

The selection menu allows users to cycle through the options and doesn't require the users to input any commands via the terminal.

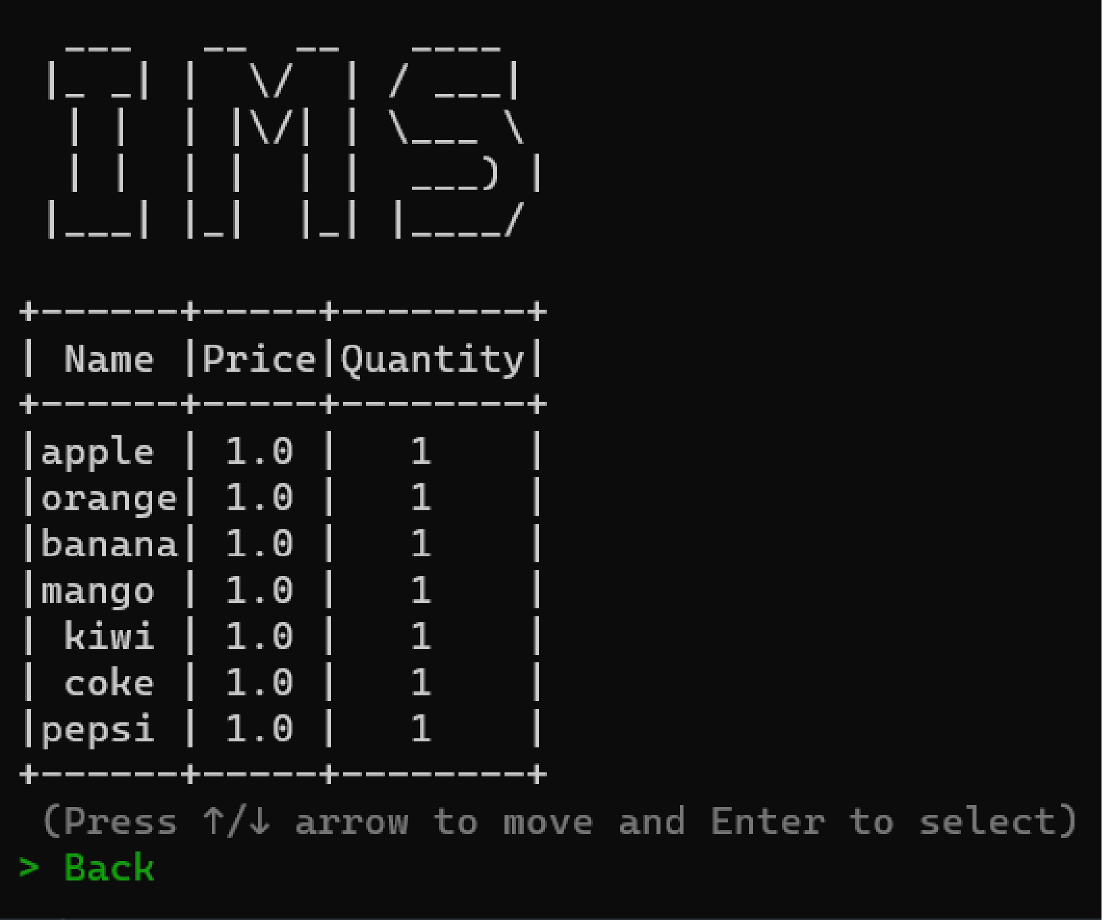
This makes it more intuitive and easier to use.



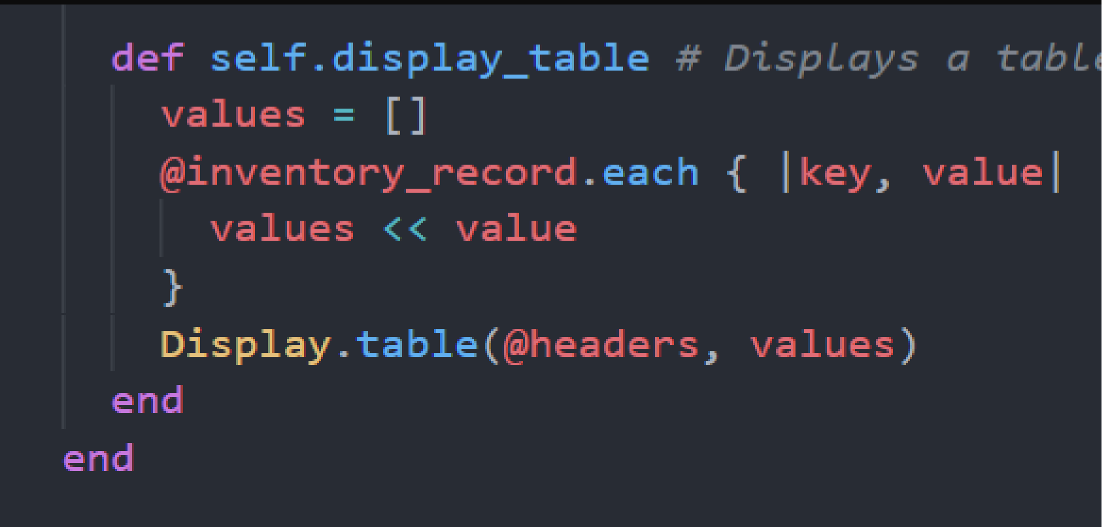
```ruby
require 'tty-prompt'
require_relative '../controller/file.rb'
require_relative '../controller/crud.rb'
require_relative '../views/screen.rb'

module Prompt
  def self.menu
    Screen.title
    prompt = TTY::Prompt.new
    choices = [
      { name: 'Create Inventory List', value: 1 },
      { name: 'Update', value: 2 },
      { name: 'Delete', value: 3 },
      { name: 'Display', value: 4 },
      { name: 'Export CSV', value: 5 },
      { name: 'Quit', value: 6 }
    ]

    if Files.exist && !Files.empty
      Crud.load
    else
      choices[1][:disabled] = "       * (No invento
      choices[2][:disabled] = "       * (No invento
      choices[3][:disabled] = "      * (No inventor
      choices[4][:disabled] = "   * (No inventory)
    end
```



```
 ___   __  __   ____
|_ _| |  \/  | / ___|
 | |  | |\/| | \___ \
 | |  | |  | |  ___) |
|___| |_|  |_| |____/

Select
> Create Inventory List
  Update
  Delete
  Display
  Export CSV
  Quit
```



```
  ___   __  __    ____
 |_ _| |  \/  | / ___|
  | |  | |\/| | \___ \
  | |  | |  | |  ___) |
 |___| |_|  |_| |____/

+------+-----+--------+
| Name |Price|Quantity|
+------+-----+--------+
|apple | 1.0 |   1    |
|orange| 1.0 |   1    |
|banana| 1.0 |   1    |
|mango | 1.0 |   1    |
| kiwi | 1.0 |   1    |
| coke | 1.0 |   1    |
|pepsi | 1.0 |   1    |
+------+-----+--------+
 (Press ↑/↓ arrow to move and Enter to select)
> Back
```

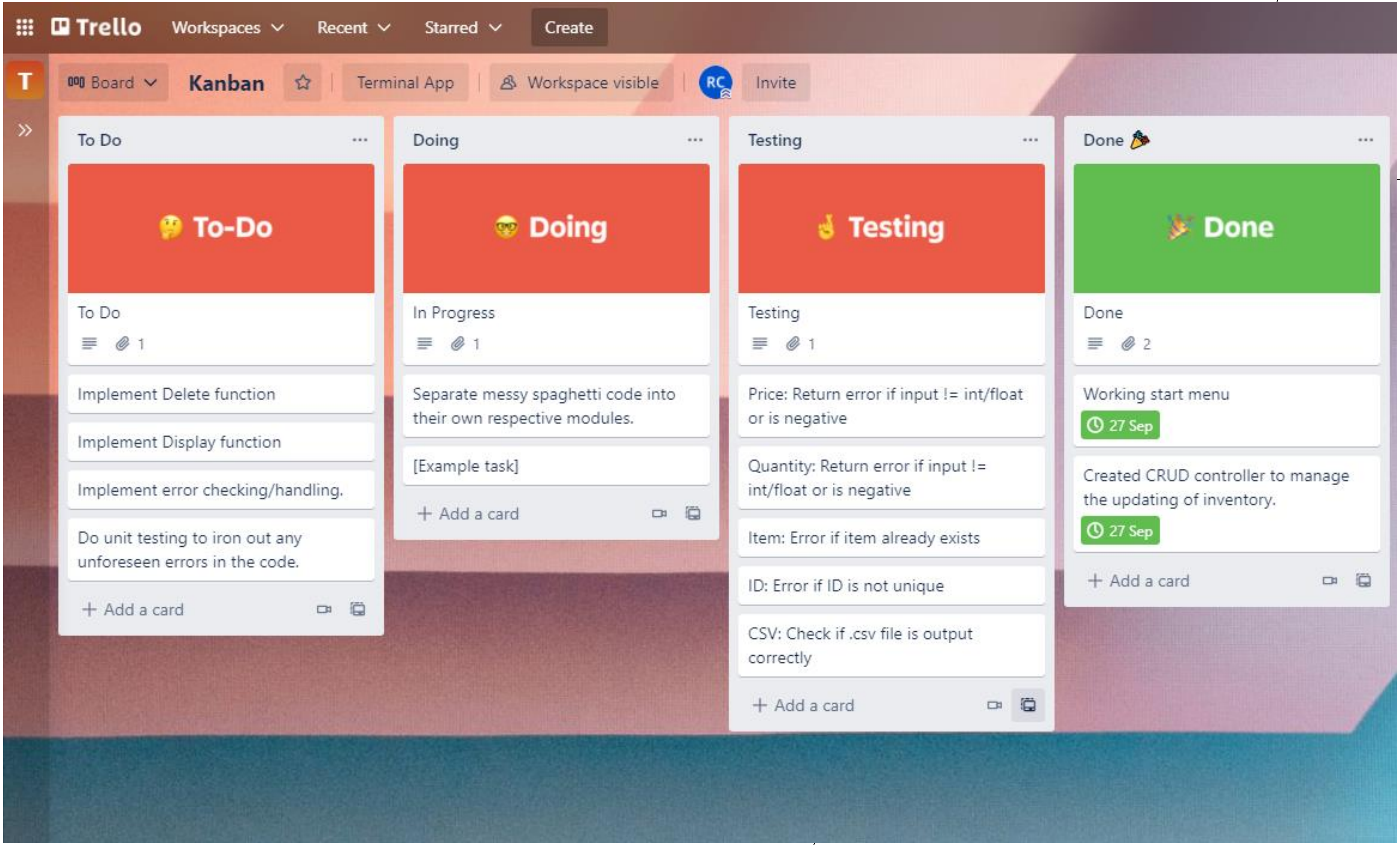


```
  def self.display_table # Displays a tabl
    values = []
    @inventory_record.each { |key, value|
      values << value
    }
    Display.table(@headers, values)
  end
end
```

The app can display a list of all items stored in the inventory to the user as an ascii table generated by tty-table.

## RUBY GEMS

| | |
|---|---|
| TTY-Font | App logo |
| TTY-Prompt | Menu selection and prompts |
| TTY-Table | Displays the list of items as an ASCII table |
| CSV | Reading and writing .CSV files |
| YAML | Reading and writing .YAML files |
| SecureRandom | Generate unique ID for inventory items |


Trello Workspaces Recent Starred Create

Board Kanban Terminal App Workspace visible RC Invite

**To Do**

To-Do

To Do

1

Implement Delete function

Implement Display function

Implement error checking/handling.

Do unit testing to iron out any unforeseen errors in the code.

+ Add a card

**Doing**

Doing

In Progress

1

Separate messy spaghetti code into their own respective modules.

[Example task]

+ Add a card

**Testing**

Testing

Testing

1

Price: Return error if input != int/float or is negative

Quantity: Return error if input != int/float or is negative

Item: Error if item already exists

ID: Error if ID is not unique

CSV: Check if .csv file is output correctly

+ Add a card

**Done**

Done

Done

2

Working start menu

27 Sep

Created CRUD controller to manage the updating of inventory.

27 Sep

+ Add a card

## CHALLENGES

### Understanding OOP Concepts

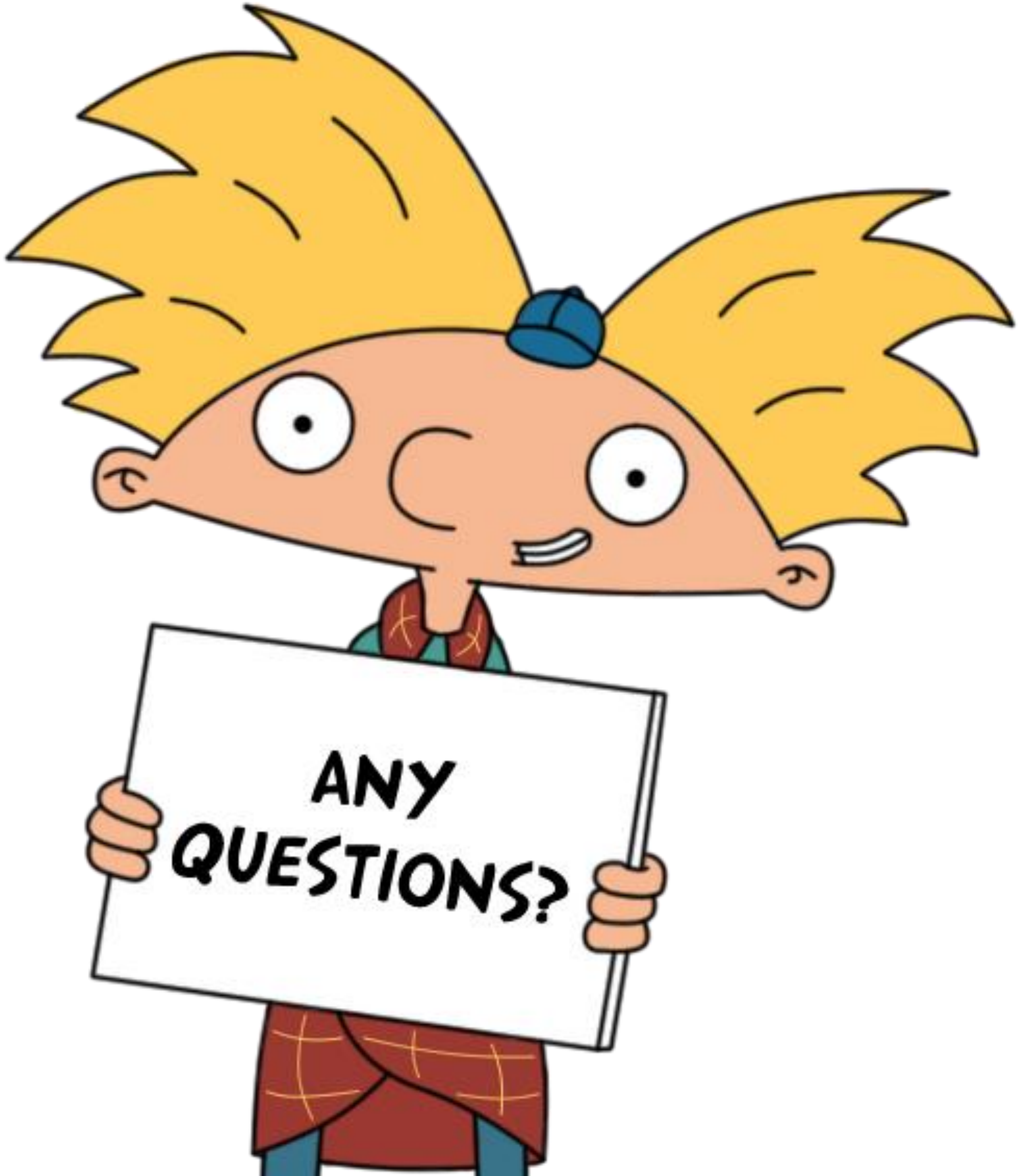
- Self
- Classes
- Modules
- Trying to stay away from spaghetti code and separate them into separate files,

### Implementing features

- More features = more complexity = more time

### Testing

- Fixing errors
- Error handling – Thinking of all the different ways a piece of code can break.
- Making sure the app behaves the way it’s meant to.


ANY
QUESTIONS?