**Trump v Kanye Quiz CLI**

**Table of contents**

[1. Project Links](https://github.com/Finbob12/ToDoList-CLI#Project-Links)  
[2. What does ToDoList CLI do?](https://github.com/Finbob12/ToDoList-CLI#What-does-ToDoList-CLI-do?)  
[3. What problem does this app solve? and why am I developing it?](https://github.com/Finbob12/ToDoList-CLI#What-problem-does-this-app-solve?-Why-am-I-developing-it?)  
[4. Target audience and how they will use it](https://github.com/Finbob12/ToDoList-CLI#Target-audience-and-how-they-will-use-it)  
[5. ToDoList CLI Features](https://github.com/Finbob12/ToDoList-CLI#ToDoList-CLI-Features)  
[6. An outline of user interaction](https://github.com/Finbob12/ToDoList-CLI#An-outline-of-user-interaction)  
[7. Control flow diagram for the app](https://github.com/Finbob12/ToDoList-CLI#Control-flow-diagram-for-the-app)  
[8. Implementation plan - Trello](https://github.com/Finbob12/ToDoList-CLI#Implementation-plan---Trello)  
[9. System requirements](https://github.com/Finbob12/ToDoList-CLI#System-requirements)  
[10. How to install and run ToDoList CLI](https://github.com/Finbob12/ToDoList-CLI#How-to-install-and-run-ToDoList-CLI)

**Project Links**

[Github: ToDoList CLI](https://github.com/Finbob12/ToDoList-CLI)

Variables

Loops

Input and Output

**Develop** a statement of **purpose** and **scope** for your application. It must include:  
- **describe** at a high level what the application will do  
- **identify** the problem it will solve and **explain** why you are developing it  
- **identify** the target audience  
- **explain** how a member of the target audience will use it

**Purpose and scope of the application**

**What does Trump v Kayne Quiz CLI do?**

The Trump v Kanye Quiz is a command line quiz that provides the user with 15 multiple choice questions. Users are asked to enter their name, and the program refers to them as this name from then on. If they do not enter a value, they are prompted to re-enter their name. The quiz then starts, with 15 quotes from Twitter and users have to identify the author – Trump or Kanye. If they get the correct answer, they receive a point are told their cumulative score. If they get it wrong, they are told their answer is incorrect.

At the end of the quiz, the user is given their score out of the total answers. They are then given the opportunity to double or nothing – to gamble all their points on the last question. They are then told their new score based on their answer to the final question.

**What problem does this app solve?**

This app doesn’t necessarily solve a problem - possibly boredom! It does improve user’s knowledge of the Donald Trump’s and Kanye West’s social media outputs; perhaps it’s a commentary on the state of politics and entertainment…

**Target audience**

This is a fun tongue-in-cheek app to entertain users and allow them to test their knowledge about popular culture figures. It is designed to be played by users with minimal to no knowledge of the command line, as they only have to answer simple prompts. Users who have never heard of Trump of Kanye can still play, as there are only two options to pick from so they still have a chance of being right.

Inputs are entered as ‘Name’, typing ‘Trump’ or ‘Kanye’ for each question (case insensitive) and then selecting Yes or No at end when deciding whether or not to gamble.

**Develop** a list of features that will be included in the application. It must include:  
- at least THREE **features**  
- **describe** each feature  
  
**Note:** **Ensure** that your features above allow you to demonstrate your understanding of the following language elements and concepts:  
- use of variables and the concept of variable scope  
- loops and conditional control structures  
- error handling  
  
**Consult with your educator** to check your features are sufficient .

**Trump v Kanye Features**

Features of the app include \*\*

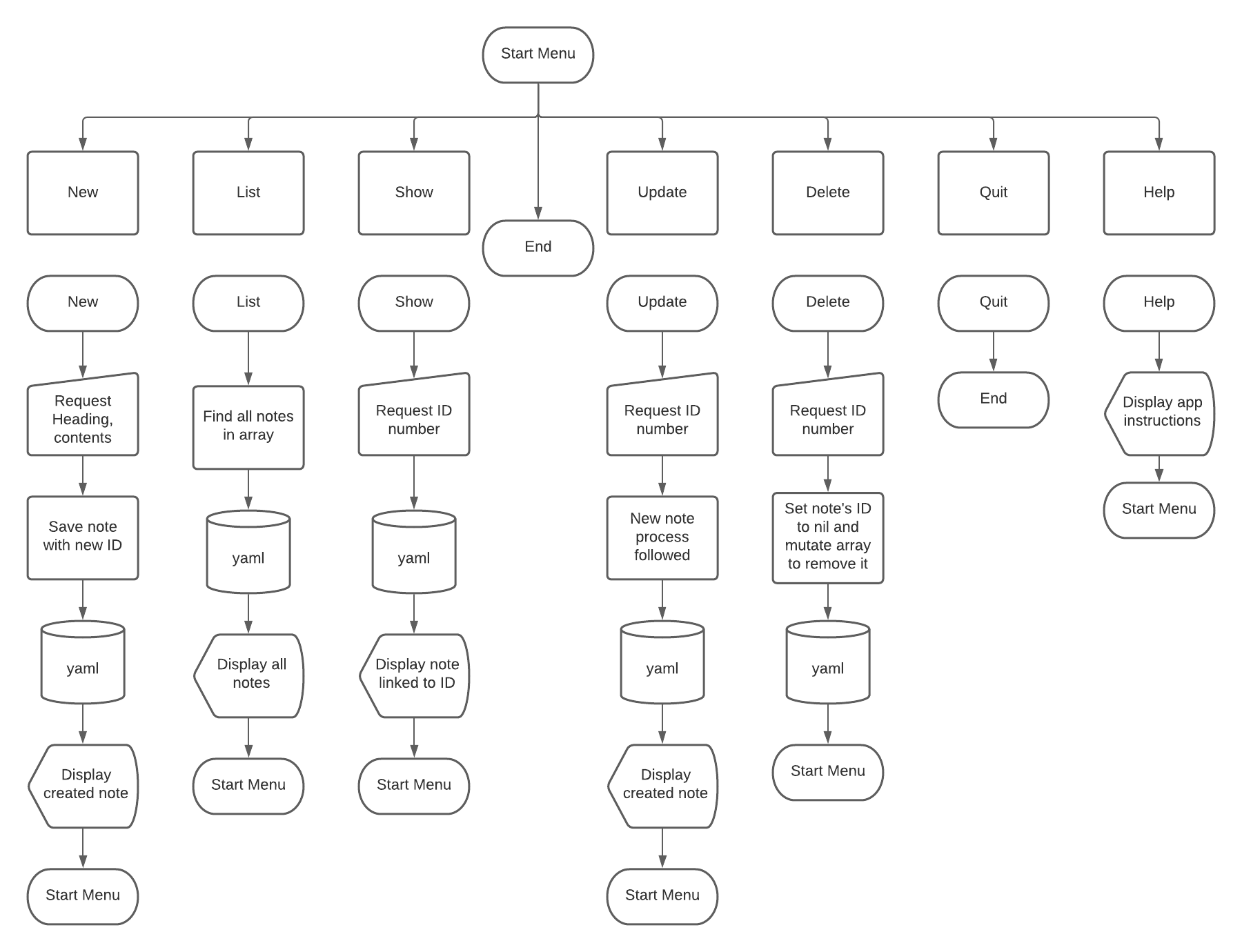
The first step of the app asks the user to enter their name, in order to engage them from the beginning and to personalise the user experience.

* The interactive menu. A self explanatory menu in your terminal guiding your navigation with simple text inputs from the user.
* New - The most important feature of the app, the ability to create your own notes and have the app save them.
* List - Easy accessibility to read notes you have previously saved.
* Show - View a specific note that has been previously written for closer inspection.
* Update - Ability to update or edit notes that have been previously created.
* Delete - Specify which note you would like to delete and it will be removed from the app.

**An outline of user interaction**

When the user first launches the application they will be greeted with a very simple, clear and consise menu that should be self-explantory. If the user is still after further clarification on what they should do, they can select the help option which will provide them with a detailed beakdown of the menu options available. Each feature can be easily used by selecting it from the main menu. Show, Update and Delete also require an ID number to be inputted for their functionality to be accessed. If these ID's are forgotten, the user can simply use the List feature to view them again. At any time if the user incorrectly inputs something, or attempts to input nothing for a note, the system will give them a message telling them what the exact issue was and how they can resolve it.

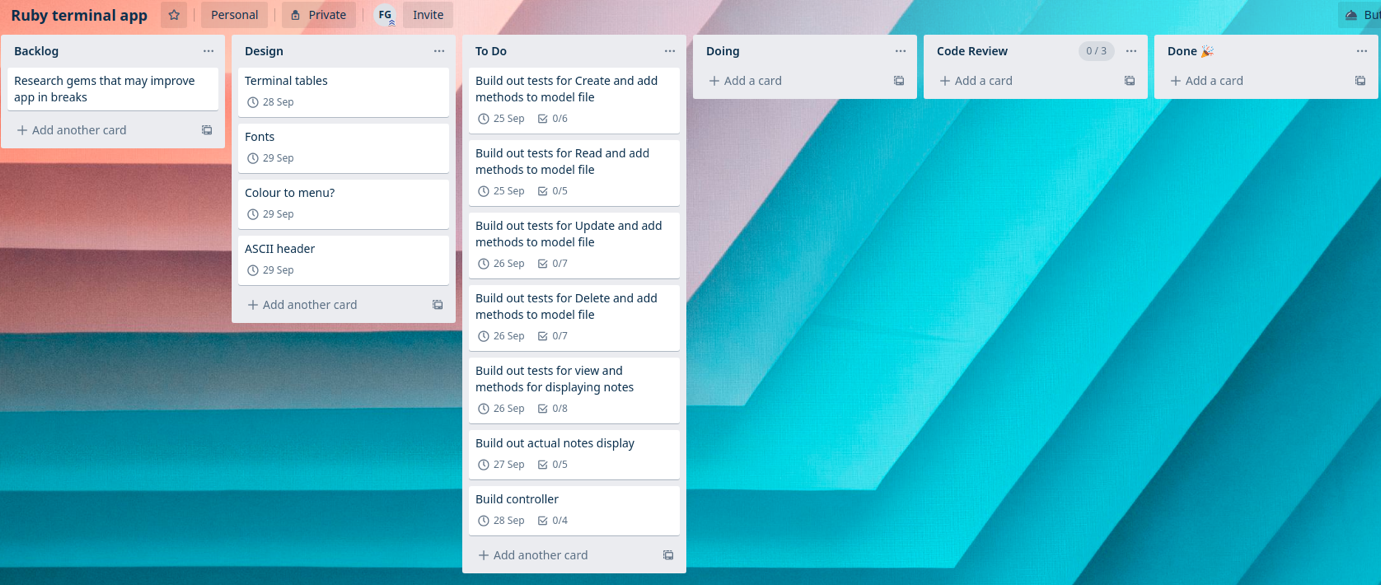
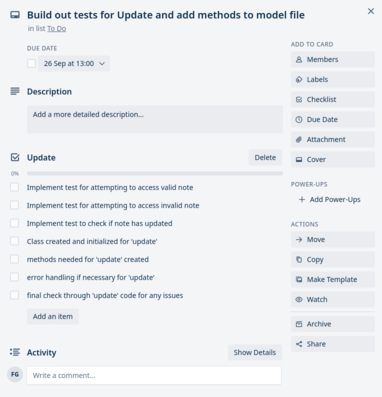
**Control flow diagram for the app**

[](https://github.com/Finbob12/ToDoList-CLI/blob/master/readme_images/flowchart.png)

**Implementation plan - Trello**

Link to the live Trello board is [here](https://trello.com/b/XyIxIzrm/ruby-terminal-app)

The below two images are from my initial Trello plan. I've set myself acceptable timeframes to complete each task, allowing for extra days before the project is due for any complications or extra features I may want to add. This initial plan changed a few times as I edited my board to better fit the exact scope of the project. An MVP was the main goal of the process, while also keeping in mind design elements that I wanted to add to make the project more appealing. Further screenshots of my Trello board as it was updated day to day are located [here](https://github.com/Finbob12/ToDoList-CLI/blob/master/readme_images/trello)

[](https://github.com/Finbob12/ToDoList-CLI/blob/master/readme_images/trello/24-sep-trello.png) [](https://github.com/Finbob12/ToDoList-CLI/blob/master/readme_images/trello/24-sep-checklist.png)

**System requirements**

The only system requirement is a computer that has bash script capability.

**Installation of the application**

* Follow these instructions to [download and install](https://www.ruby-lang.org/en/documentation/installation/) Ruby on your computer.
* Clone the Github repository or download the Zip file and extract it onto your local machine.
* Open your terminal and run the following command after navigating to the projects main folder.

$ gem install bundler

* After bundle is installed, run the next command below.

$ bundle install

* Then to launch the application type the following.

$ ./run\_app.sh

**Dependencies for the application**