

T1A3 Presentation

Kaiqi Liao

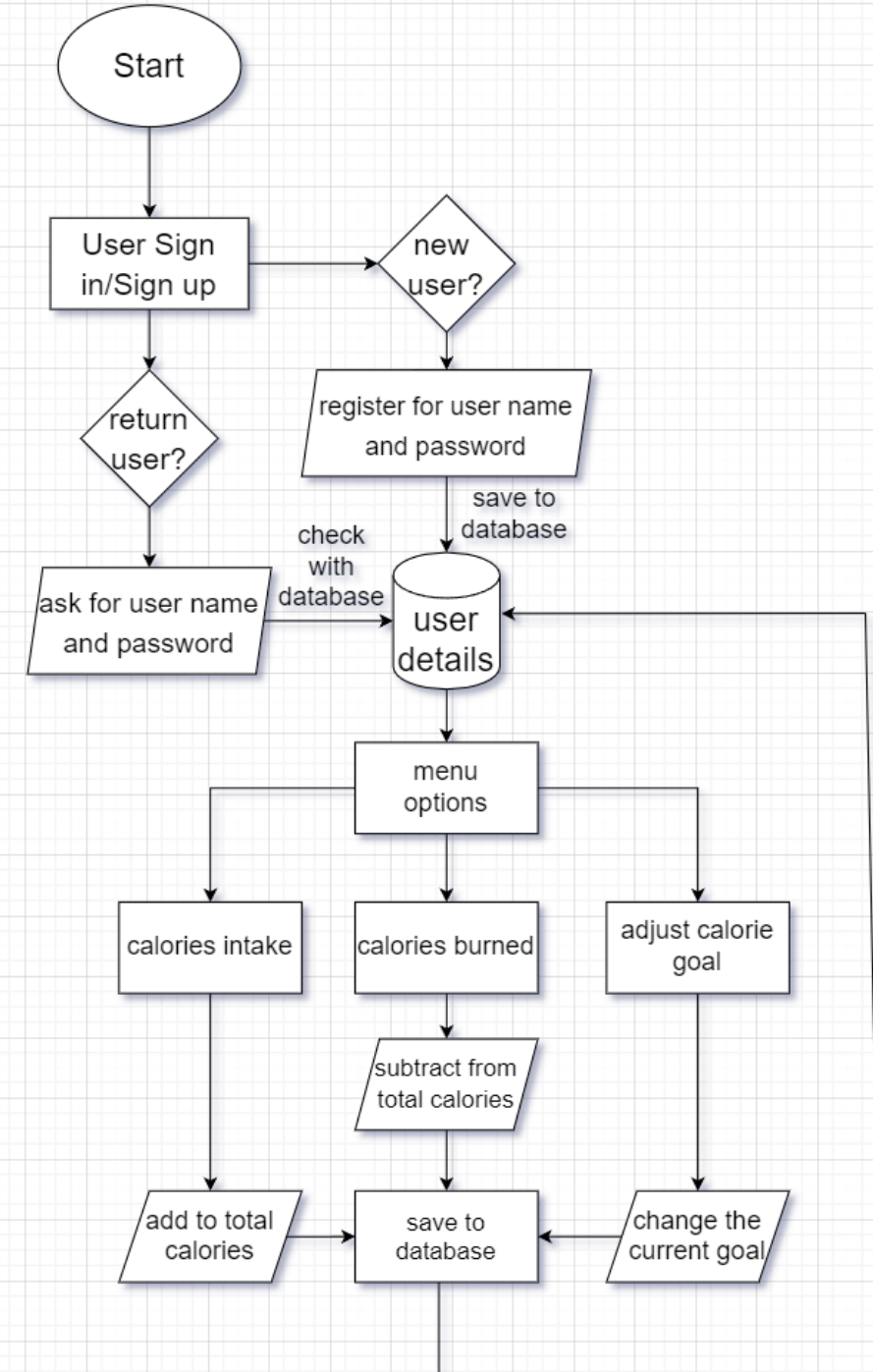
Daily Calories Tracker

- Wanted to record my calorie when losing weight
- Also got inspiration from Apple Watch
- Current apps are too “fancy” and asking for subscription



Flow Chart

- Use a JSON file to store user details
- User sign in or sign-up feature
- View current daily calories goal
- Update daily calories progression



Read user details from JSON file

- Each user details stored as a hash inside the array
- Create the array inside ruby

```
# get all saved user details
userdata = JSON.load_file('userdata.json', symbolize_names: true)
```

```
[{:name=>"Jim", :password=>"1234", :goal=>1680, :progress=>880}, {:name=>"Bob", :password=>"2345", :goal=>1400, :progress=>0}, {:name=>"John", :password=>4567, :goal=>1600, :progress=>0}, {:name=>"Bill", :password=>"3344", :goal=>"1300", :progress=>400}]
```

```
{..} userdata.json > ...
You, 1 second ago | 1 author (You)

1  [
2    {
3      "name": "Jim",
4      "password": "1234",
5      "goal": 1680,
6      "progress": 880
7    },
8    {
9      "name": "Bob",
10     "password": "2345",
11     "goal": 1400,
12     "progress": 0
13   },
14   {
15     "name": "John",
16     "password": 4567,
17     "goal": 1600,
18     "progress": 0
19   }
20 ]
```

Check if user exists

- Pull all user name into an array
- Run a loop to see if the terminal input matches an existing user
- Load user details into current_user variable

```
all_user_names = []
userdata.each do |i|
  all_user_names << i[:name]
end
```

```
# set a hash to store details of user using the app
current_user = {}

answer = prompt.ask("What is your name?")

if all_user_names.include? answer
  # set current_user to this user
  userdata.each do |i|
    if i[:name] == answer
      current_user = i
      # p current_user
    end
  end
end
```

```
What is your name? Jim
{:name=>"Jim", :password=>"1234", :goal=>1680, :progress=>880}
Hi Jim, please enter your password
```


Password check

```
# password check
begin
  if current_user[:password] == prompt.ask("Hi #{answer}, please enter your password")
    puts "Welcome back #{answer}!"
    puts "Your current progress is #{current_user[:progress]}/#{current_user[:goal]} calories"
  else
    raise "Wrong password! Please try again"
  end
rescue => e
  puts e
  retry
end
```

```
Hi Jim, please enter your password 2222
Wrong password! Please try again
```

```
Welcome back Jim!
Your current progress is 880/1680 calories
```

New User Sign Up

```
newname = prompt.ask("New user? register a new user name:")
newpassword = prompt.ask("Please enter a password:")
newgoal = prompt.ask("What is your daily goal?")
puts "Welcome #{newname}!"
current_user = { name: newname, password: newpassword, goal: newgoal, progress: 0 }
userdata << current_user
File.write('userdata.json', JSON.pretty_generate(userdata))
```

```
What is your name? Kim
New user? register a new user name: Kim
Please enter a password: 3344
What is your daily goal? 1300
Welcome Kim!
```

```
{
  "name": "Kim",
  "password": "3344",
  "goal": "1300",
  "progress": 0
}
```

Main menu

```
def menu_choice(user)
  prompt = TTY::Prompt.new
  loop do
    choices = ["Add food intake", "Add workouts", "Adjust goals", "Exit"]
    selected = prompt.select("Please select from following options", choices, cycle: true)
    case selected
    when choices[0]
      food_intake(user)
    when choices[1]
      workout(user)
    when choices[2]
      adjust_goal(user)
    when choices[3]
      exit_app
      break
    end
  end
end
```

```
Welcome back Jim!
Your current progress is 880/1680 calories
Please select from following options (Press ↑/↓ arrow to move and Enter to select)
▸ Add food intake
  Add workouts
  Adjust goals
  Exit
```


Menu options

Use loops to keep running through options

User can go back to previous menu

```
def food_intake(user)
  prompt = TTY::Prompt.new
  loop do
    answer = prompt.ask("How many calories in your food?", convert: :int)
    user[:progress] += answer
    puts "Intake added!"
    puts "Today you have #{user[:progress]}/#{user[:goal]} calories so far"
    answer2 = prompt.yes?("Do you want to add more intake?")
    break unless answer2
  end
end
```

```
How many calories in your food? 200
Intake added!
Today you have 1080/1680 calories so far
Do you want to add more intake? (Y/n) █
```

Adjust the calorie goal

```
def adjust_goal(user)
  prompt = TTY::Prompt.new
  puts "Your current goal is #{user[:goal]} calories"
  answer = prompt.ask("Please enter your new daily goal", convert: :int)
  user[:goal] = answer
  puts "Daily goal adjusted, your new goal is #{answer} calories"
  answer2 = prompt.yes?("Do you want more actions?")
  exit_app unless answer2
end
```

```
Your current goal is 1680 calories
Please enter your new daily goal 1580
Daily goal adjusted, your new goal is 1580 calories
Do you want more actions? (Y/n) █
```

Save changes and exit

- Write changes back to the JSON file

```
# write user data back to json file  
File.write('userdata.json', JSON.pretty_generate(userdata))
```

- Display exit message

```
Please select from following options Exit  
Thanks for using Daily Calories Tracker! See you later!
```

Works to do

- Obviously DRY my code a lot
- Use gem like rainbow to better visualize the interface
- Improve the loop logic
- Test and debug
- Add [ARGV] for help option