

Reminder Terminal Application

Presented by Khaing Tun

Planning for Ruby CLI

- ▶ Ideas taken from Pretty simple design
- ▶ Keep the simple algorithm to start the app
- ▶ Designing the workflow was pretty struggle
- ▶ Use the TTY-prompt for the menu

Obstacles I faced..

- ▶ Wanted to make CLI game but tried to keep simple as starter in limited time
- ▶ To keep things on track and make simple reminder app
- ▶ Finding gems that suited my app

Design and Organizing project

The image shows a Trello board titled "Ruby Terminal App". The board is organized into four main columns: Backlog, Design, Progress and problems, and Doing.

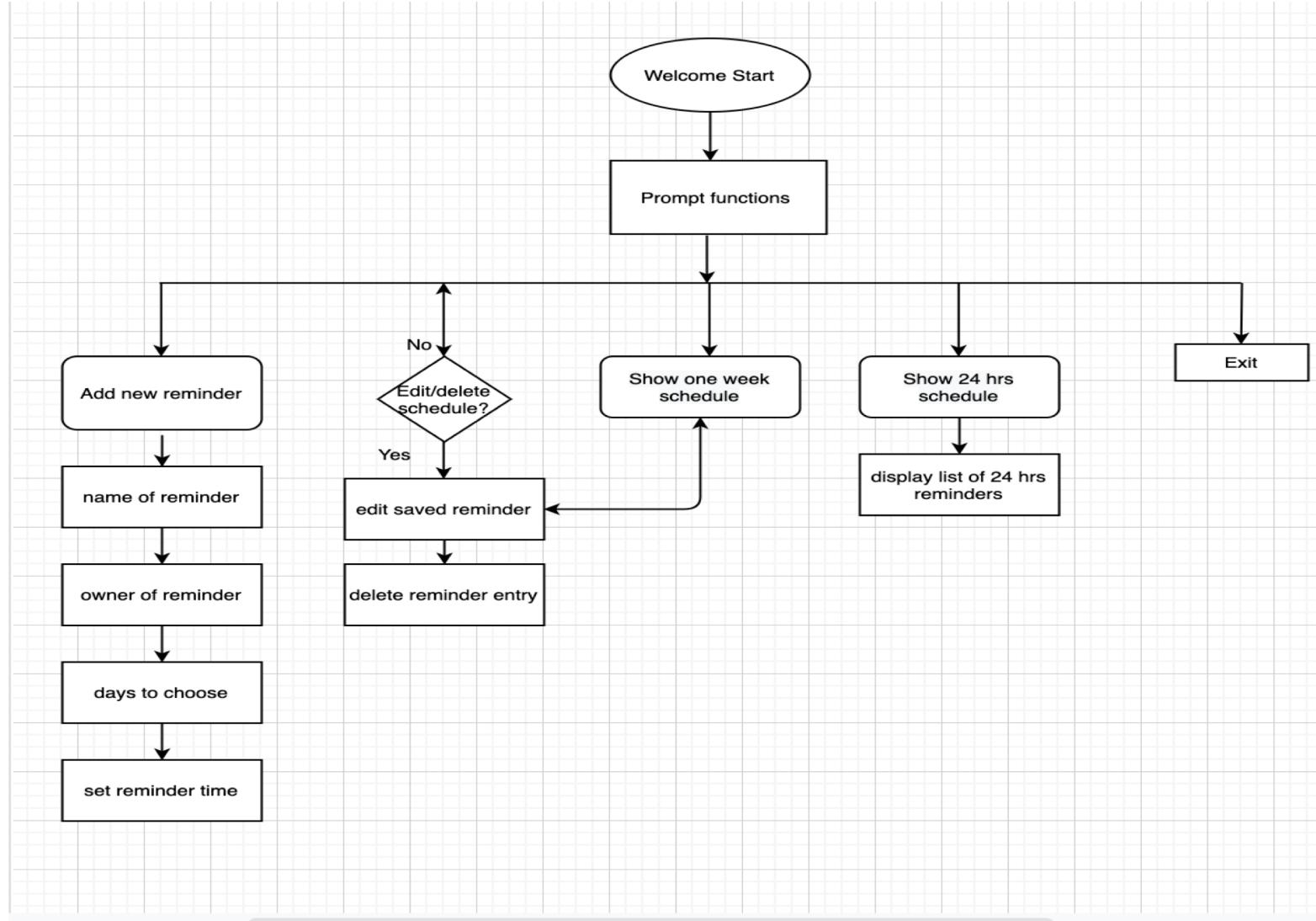
- Backlog:** Contains cards for "Ideas and planning", "provide attribution to reference sources", "originally planned on doing a adventure game but ran into some roadblocks when first playing around with it that made me realise i was biting off more than i could chew.", "still haven't added functions. some simpler methods and added in tty-prompt to help the user through the menus.", and "write documentation for application".
- Design:** Contains cards for "Ruby gems", "testing gems and compatibility", "work flow and pseudo code", and a "+ Add another card" button.
- Progress and problems:** Contains cards for "To Do", "coding and debugging", "rspec and tdd", "gem testing colorize and welcome text", "privacy issues", "don't forget to commit git !!", "write readme in markdown", and a "+ Add another card" button.
- Doing:** Contains cards for "ruby coding", "rspec and tdd", "unique variable name as the coding get pretty messy", "spec and gem", "tty prompt usage", "getting error when press enter", and a "+ Add another card" button.

A magnifying glass icon is positioned over the "Design" column. The background features a green abstract pattern and a pink notepad with the text "TODAY I AM GRATEFUL".

Feature of application

- ▶ Welcome Prompt menu
- ▶ Add new reminder
- ▶ Edit/delete schedule
- ▶ Show 1 week schedule
- ▶ Show 24 hrs reminder schedule
- ▶ Exit

Flow diagram of App



Welcome page of CLI

```
bash reminder-app.sh --help
```

Code snippet classes and methods used

```
# This file is the list of classes and methods used to run the app
require "tty-prompt"
require "tod"
require "io/console"
require "date"
require "time"
require "colorize"
require_relative ("reminder.rb")
require_relative ("db.rb")
require_relative ("text.rb")
class App
  attr_accessor :reminders

  def initialize
    @prompt = TTY::Prompt.new(symbols: { marker: "->" })
    @reminders = Db.read_from_file
  end

  def run
    loop do
      clear
      main_menu
    end
  end

  def main_menu
    titlebar
    choices = [
      { name: "Add new reminders", value: -> { addReminderWeekly } },
      { name: "View, edit or delete existing reminders", value: -> { remindersMenu } },
      { name: "Show 1 week schedule", value: -> { reminderWeek } },
      { name: "Show next 24 hour schedule", value: -> { reminderDay } },
      { name: "Exit", value: -> [
        clear,
        titlebar
      ] }
    ]
    choices.each do |choice|
      choice[:value].call
    end
  end

  def addReminderWeekly
    puts "Enter reminder details"
    puts "Title: "
    title = gets.chomp
    puts "Description: "
    description = gets.chomp
    puts "Due Date (YYYY-MM-DD): "
    due_date = gets.chomp
    puts "Priority (1-5): "
    priority = gets.chomp
    reminder = Reminder.new(title, description, due_date, priority)
    Db.add_reminder(reminder)
    puts "Reminder added successfully!"
  end

  def remindersMenu
    puts "Reminders menu"
    puts "1. Add new reminders"
    puts "2. View, edit or delete existing reminders"
    puts "3. Show 1 week schedule"
    puts "4. Show next 24 hour schedule"
    puts "5. Exit"
    choice = gets.chomp
    case choice
    when "1"
      addReminderWeekly
    when "2"
      editReminder
    when "3"
      showWeekSchedule
    when "4"
      showDaySchedule
    when "5"
      exit
    else
      puts "Invalid choice"
    end
  end

  def editReminder
    puts "Edit reminder"
    puts "1. Add new reminders"
    puts "2. View, edit or delete existing reminders"
    puts "3. Show 1 week schedule"
    puts "4. Show next 24 hour schedule"
    puts "5. Exit"
    choice = gets.chomp
    case choice
    when "1"
      addReminderWeekly
    when "2"
      editReminder
    when "3"
      showWeekSchedule
    when "4"
      showDaySchedule
    when "5"
      exit
    else
      puts "Invalid choice"
    end
  end

  def showWeekSchedule
    puts "Show 1 week schedule"
    puts "1. Add new reminders"
    puts "2. View, edit or delete existing reminders"
    puts "3. Show 1 week schedule"
    puts "4. Show next 24 hour schedule"
    puts "5. Exit"
    choice = gets.chomp
    case choice
    when "1"
      addReminderWeekly
    when "2"
      editReminder
    when "3"
      showWeekSchedule
    when "4"
      showDaySchedule
    when "5"
      exit
    else
      puts "Invalid choice"
    end
  end

  def showDaySchedule
    puts "Show next 24 hour schedule"
    puts "1. Add new reminders"
    puts "2. View, edit or delete existing reminders"
    puts "3. Show 1 week schedule"
    puts "4. Show next 24 hour schedule"
    puts "5. Exit"
    choice = gets.chomp
    case choice
    when "1"
      addReminderWeekly
    when "2"
      editReminder
    when "3"
      showWeekSchedule
    when "4"
      showDaySchedule
    when "5"
      exit
    else
      puts "Invalid choice"
    end
  end

  def clear
    system("clear")
  end

  def titlebar
    puts "-----"
    puts "Reminder Application"
    puts "-----"
  end

  def exit
    puts "Exiting application..."
    exit
  end
end
```

Main menu code snippet

```
#Reminder menu for the app
def reminder_weekly_input
    reminder_name = @prompt.ask("What is the name of the reminder?", required: :true)
    reminder_owner = @prompt.ask("Who is this reminder for?", required: :true)
    choices = %w(Sunday Monday Tuesday Wednesday Thursday Friday Saturday)
    reminder_days_taken = @prompt.multi_select("Which days of the week do you take it?", choices, per_page: 7, help: "\n(Press ↑/↓ arrow keys to navigate, Space to select and Enter to continue)", show_help: :always, min: 1)
    reminder_times_taken = time_input
    return reminder_name, reminder_owner, reminder_days_taken, reminder_times_taken
end

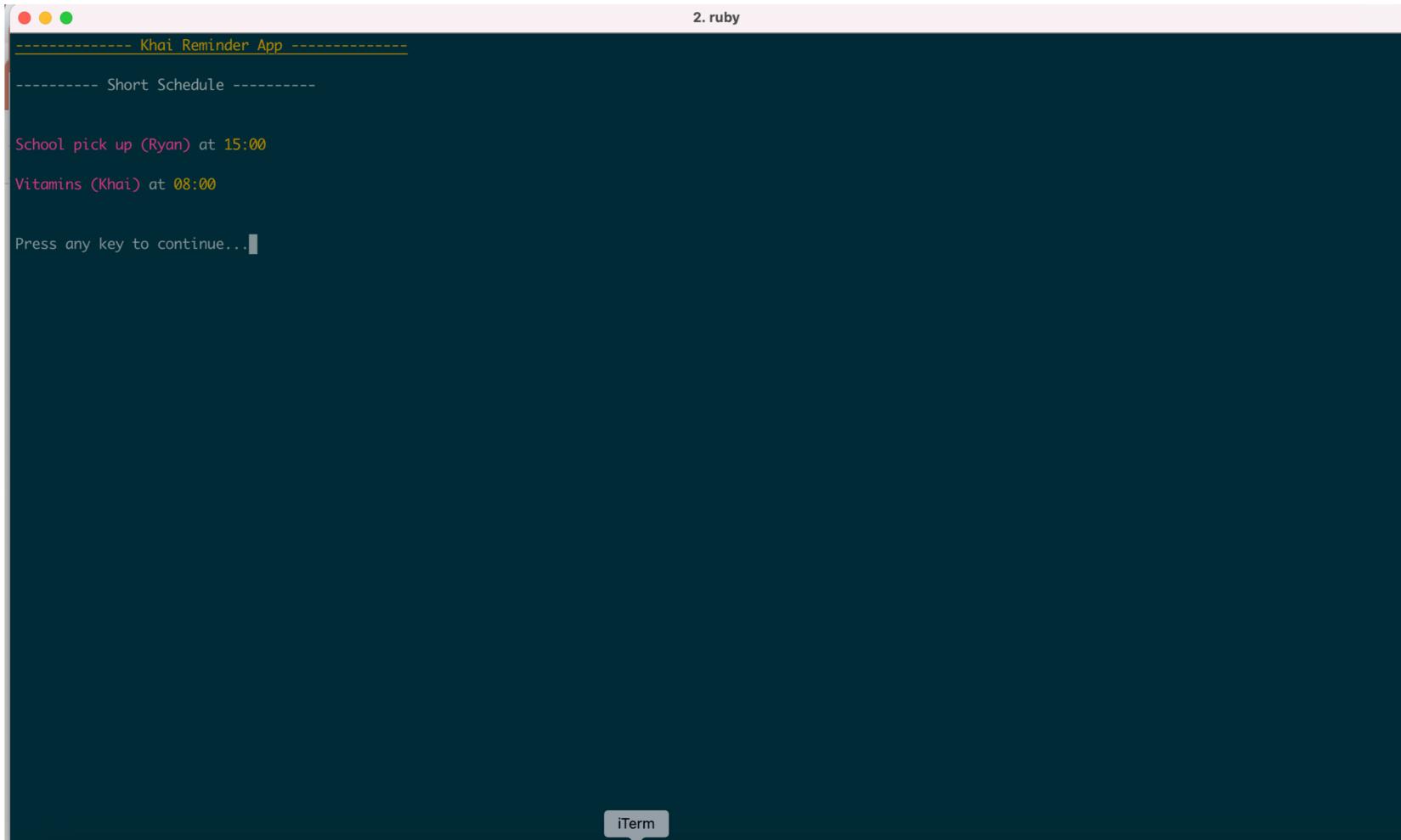
def add_reminder_weekly(argv = false)
    clear
    titlebar
    puts "Add reminder taken on weekly schedule\n\n".colorize(:light_cyan)
    @reminders << ReminderWeekly.new(*reminder_weekly_input)
    Db.write_to_file(reminders)
    clear
    titlebar
    puts "Reminder added!\n\n"
    reminders.last.display_reminder
    continue
end

def edit_reminder_weekly(index)
    clear
    titlebar
    puts "Edit reminder\n".colorize(:light_cyan)
    puts "You are editing " + "#{reminders[index].name}".colorize(:magenta) + "\n\n"
    @reminders[index] = ReminderWeekly.new(*reminder_weekly_input)
    Db.write_to_file(reminders)
    clear
    titlebar
    puts "Reminder updated!\n\n"
    reminders[index].display_reminder
    continue
end
```

List of reminders for the week

```
2. ruby
Khai
Times taken:
08:00
-----
----- Saturday -----
Reminder name:
Vitamins
Owner:
Khai
Times taken:
08:00
-----
----- Sunday -----
Reminder name:
Walking
Owner:
Khai
Times taken:
06:00
Reminder name:
Vitamins
Owner:
Khai
Times taken:
08:00
-----
Press any key to continue...■
```

List of reminders for a day



The image shows a terminal window titled "2. ruby". Inside the terminal, there is a program titled "Khai Reminder App" which displays a "Short Schedule" with two entries: "School pick up (Ryan) at 15:00" and "Vitamins (Khai) at 08:00". The terminal prompt "Press any key to continue..." is visible at the bottom.

```
2. ruby
-----
Khai Reminder App -----
-----
Short Schedule -----
School pick up (Ryan) at 15:00
Vitamins (Khai) at 08:00
Press any key to continue... █
```

iTerm

Conclusions/Review

- ▶ Further planning and testing would have helped quite a bit but it is something I need to get better with.
- ▶ Polishing the overall look of the app.

Questions ?

Thank you.