

# To-do List APP

Zumin Yu

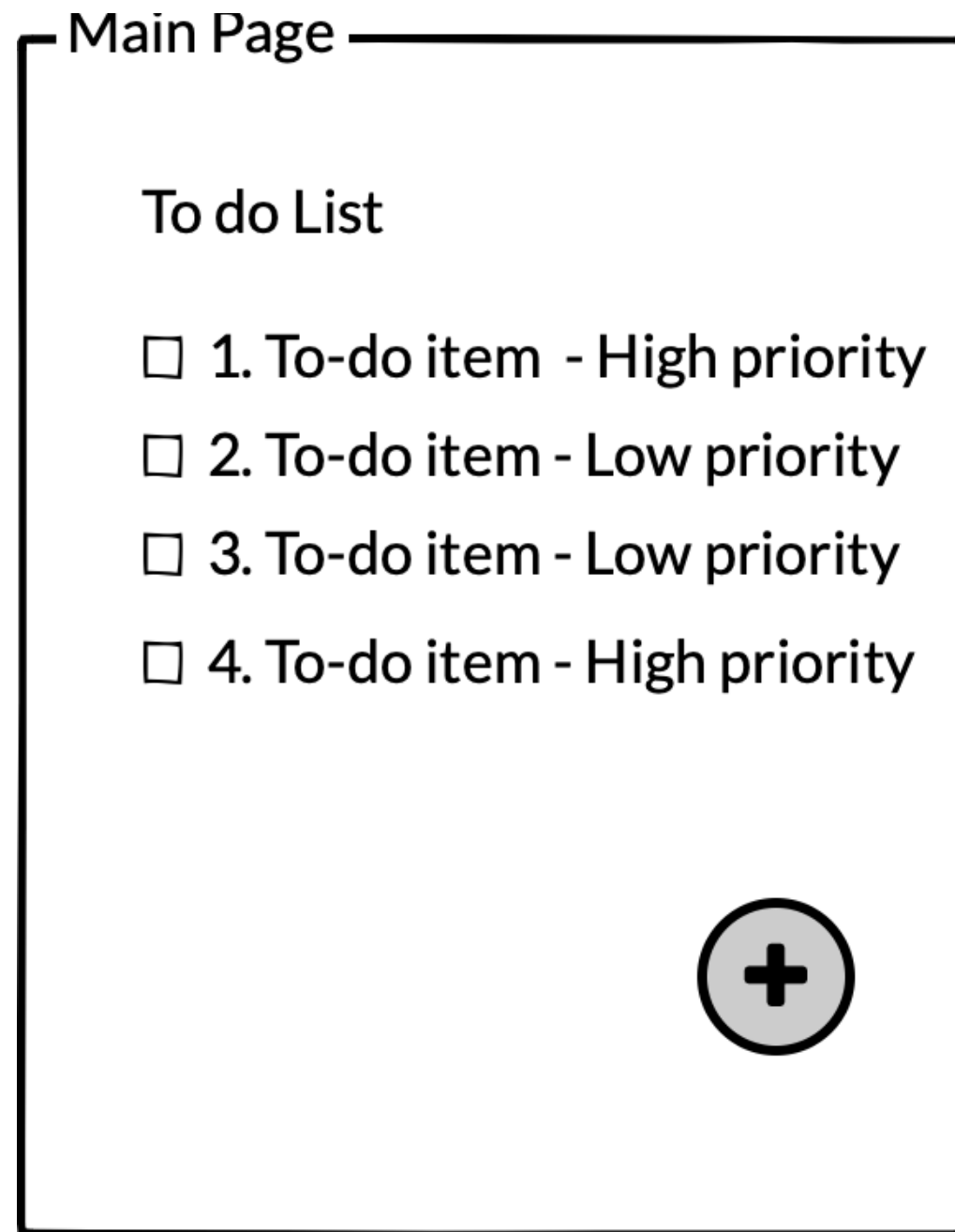


# About

## A TO-DO LIST

- Design for terminal application
- Main feature/function
  - Add function
  - delete function
  - Search function
  - Sort function
- Deployment plan





# Overall structure

TO DO LIST - SEPERATE INTO 2 PART  
CORE CLASS AND UX

- Main page  
Contains a current to-do lists.  
The options that lead to different page.
- Add page  
Be able to add to-do items to the list.
- Delete page  
Be able to delete selected to-do from the list.
- Search page  
Be able to search particular to-do item by type  
in the keyword

```

3  class App
4    attr_reader :to_do_list
5
6  def initialize
7    @to_do_list = []
8  end
9
10 def add(todo_name, high_priority)
11   # create the single to_do list
12   add_todo = TodoItem.new(todo_name, high_priority)
13   # push the new generated to_do to array of todo_list
14   @to_do_list << add_todo
15 end
16

```

```

2
3  class TodoItem
4    attr_accessor :todo_name, :high_priority
5
6  def initialize(todo_name, high_priority)
7    @todo_name = todo_name
8    @high_priority = high_priority
9  end
10

```

## CREATE AN CORE FUNCTION

Core function:

- class Todoitem
- class App

Application class has array of todoitem class.

The app class includes add, delete, search method.

# Overview of code

## - core function

# Overview of code

- main page

CREATE UX SECTION

Main page:  
focus on user input and provide options  
go to the other page.

```
5 class MainPage
6
7 def initialize
8   @app = App.new
9 end
10
11
12 def get_user_input
13   puts "Welcome to use To-do-list, let's get start!"
14
15   loop do
16     puts "-----"
17     puts "What would you like to do?"
18     puts "1. Add a to-do."
19     puts "2. Delete a to-do from the list."
20     puts "3. Search to-do by keyword."
21     puts "4. Display the to-do list by date."
22     puts "5. Exit"
23     choice = gets.chomp
24
25     case choice
26     when "1"
27       show_add_page
28     when "2"
29       display_todo
30       show_delete_page
31     when "3"
32       show_search_page
```

# Overview of code

## - add page

CREATE UX SECTION

Add page:

focus on user input and add to-do item.  
Also be able to go back to main page  
when finish adding to-do.

```
6  class AddPage
7
8  def initialize(app)
9    @app = app
10 end
11
12 def run
13   loop do
14     puts "Please enter todo name."
15     todo_name = gets.chomp
16     puts "please enter the priority of this todo. if it's high priorit
17     high_priority = gets.chomp.downcase
18     if high_priority == "h"
19       high_priority_value = true
20     elsif high_priority == "l"
21       high_priority_value = false
22     end
23
24     @app.add(todo_name, high_priority_value)
25
26     puts "Do you want to add more todo? (y/n) "
27     choice = gets.chomp.downcase
28     if choice == "n"
29       break
30     end
31   end
32 end
33 end
```



# Favourite

TDD

VARIABLE  
SCOPE



GOOGLE

RUBY GEMS

# challenges

# Let's Try the app

