

Cài đặt:

```
pip install -r requirements.txt
```

Cấu trúc:

File-project

|_____ input: chứa các file input

|_____ output: chứa các file output được trả về

|_____ project: chương trình

|_____ resources

|_____ annotations.xlsx

|_____ templateHTML.html

|

|_____ src

|_____ myconstants.py

|_____ ... code

|

|

|_____ gen_flows_txt.py: (YC1) đọc UESCASES → FLOWS

|_____ gen_testcases_excel.py: (YC2) đọc FLOWS → TESTCASE

|_____ gen_template_html.py: (YC3) đọc FLOW → UI HTML

Giới thiệu:

annotation.xlsx

File định nghĩa các quy tắc regex, các annotation, html tương ứng

Gồm có các sheet:

- **Element:** để định nghĩa các đối tượng html
Trong đó:
 - o annotation: tên viết tắt của element (tên này được sử dụng trong file txt: usecases, flows)
 - o meaning: ý nghĩa/tên đầy đủ của element (tên này được sử dụng trong file excel: testcases)
 - o html-open: tag mở trong html
 - o html-close: tag đóng trong html
(Một số element có html chỉ gồm 1 tag → chỉ ghi vào html-open)
-

	A	B	C	D
1	annotation	meaning	html-open	html-close
2	#ao	alert box output		
3	#bi	button onclick input	<button type="button">	</button>
4	#fi	file input	<input type="file"> </br>	
5	#li	link onclick input		
	#si	selection input	<select> <option>A</option> <option>B</option> </select>	
6				
7	#ti	text input	<input type="text"> </br>	
8	#to	text output	<p>	</p>
9				

elements | states | flows | events | annotations | (+)

- **States:** gồm các annotation chỉ trạng thái vd: #empty, #invalid, ..
Dùng để nhận biết các trạng thái khi sinh testcase → expected output: vd: #empty,..

	A	B	C	D	E	F
1	annotation					
2	#empty					
3	#invalid					
4	#valid					
5						

elements | **states** | flows | events | annotations

- **Flows:** gồm các kiểu flow

Dùng để đọc file usecase: Khi đọc 1 dòng bắt đầu bằng những từ này → bắt đầu 1 flow

	A	B	C	D	E	F
1	flow					
2	basic flow					
3	alternative flow					
4	alternate flow					
5	exception flow					
6						
	elements	states	flows	events	annotations	

- **Events:** chứa biểu thức chính quy (regex) của các event (hành động trong flow)
Biểu thức chính quy này để phát hiện khi nào 1 event xuất hiện (giúp đọc file usecase)
Trong này em đang để 2 kiểu regex, tuy nhiên chương trình sẽ chỉ sử dụng regex đầu tiên (ở dòng số 2) để nhận diện

Nếu muốn cô có thể thêm/thay thế dòng 2 bằng regex khác

Ví dụ:

^: bắt đầu dòng

\d: các ký tự số

Example: ví dụ về 1 số chuỗi match với regex

	A	B	C	D	E	F
1	event	example				
2	^(?:\d+)*\.	1.	2.	3.	3'.	3'''.
3	^(?:\d+)[a-z]\.\s	1.	2.	2a.	2b.	3.
4						
	elements	states	flows	events	annotations	+

- **Annotations:** tương tự như events

Bảng này chứa biểu thức chính quy của các annotation trong 1 event

Chương trình sử dụng regex ở dòng đầu tiên để nhận diện các annotation

Ví dụ:

Nếu dùng kiểu 1, các annotation bắt buộc phải có dạng:

[+ # + **chữ hoặc số** + **dấu gạch ngang (-)** + **chữ hoặc số** +]

Nếu dùng kiểu 2, các annotation có thể chứa thêm dấu cách trước hoặc sau dấu ngoặc vuông

annotation	example				
\[\# (?: [a-z0-9]+) (?: (?: \-) (?: [a-z0-9]+)) * \]	[#ti]	[#a-b]	[#ab]	[#a123]	[#msje-0]
\[(?: \s) * \# (?: [a-z0-9]+) (?: (?: \-) (?: [a-z0-9]+)) * (?: \s) * \]	[#ti]	[#a-b]	[#ab]	[#a123]	[#msje-0]
elements	states	flows	events	annotations	+

templateHTML.html

file chứa khung html:

```
templateHTML.html X
project > resources > templateHTML.html > html > body > form
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Template HTML</title>
8  </head>
9  <body>
10     <form>
11
12     [#content]
13
14     </form>
15 </body>
16 </html>
```

Khi thực hiện yêu cầu 3: reading the txt files to generate UI prototype in html files

File txt chứa flow → đọc các annotation → chuyển về html tương ứng

Đoạn html tương ứng của các annotation đó sẽ được chèn vào tag: **[#content]** ở file html trả về

Ví dụ:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Template HTML</title>
</head>
<body>
    <form>

        <label>user-name</label>
        <input type="text"><br></br>
        <label>password</label>
        <input type="text"><br></br>
        <button type="button">login</button>

    </form>
</body>
</html>
```

myconstants.py

lưu các hằng số cố định giúp chương trình đọc các file trong resources:

annotations.xlsx, templateHTML.html

Ví dụ: tên sheet, tên các cột của file annotations.xlsx

```
INPUT_FOLDER_PATH = "../input/"
OUTPUT_FOLDER_PATH = "../output/"

ANNOTATION_SHEET = "annotations"
ANNOTATION_COLUMN = "annotation"

ELEMENT_SHEET = "elements"
ELEMENT_COLUMN = "annotation"
ELEMENT_MEANING_COLUMN = "meaning"
ELEMENT_HTML_O_COLUMN = "html-open"
ELEMENT_HTML_C_COLUMN = "html-close"

STATE_SHEET = "states"
STATE_COLUMN = "annotation"
FLOW_SHEET = "flows"
FLOW_COLUMN = "flow"
EVENT_SHEET = "events"
EVENT_COLUMN = "event"

HTML_REPLACE_TAG = "[#content]"
```

gen_flows_txt.py

đọc file usecase (txt) và trả về file flows (txt)

Input:

Basic flow:

1. Người dùng nhập vào [#ti][#user-name] tên người dùng và mật khẩu [#ti][#password]
2. Người dùng yêu cầu hệ thống đăng nhập với tên người dùng và mật khẩu đã nhập [#bi][#login].
3. [#valid] [#pw] Hệ thống hiển thị trang thông báo cho người dùng biết đã đăng nhập thành công [#to]

Output:

```
1. #ti #user-name #ti #password -> 2. #bi #login -> 3. #valid #pw #to
```

gen_testcases_excel.py

đọc file flows (txt) trả về file testcases (excel)

Input:

```
flows - Notepad
File Edit View

1. #ti #user-name #ti #password -> 2. #bi #login -> 3. #valid #pw #to
1. #ti #user-name #ti #password -> 2. #bi #login -> 3'. #invalid #pw #to
1. #ti #user-name #ti #password -> 2. #bi #login -> 3''. #empty #pw #to
```

Output:

i19

	A	B	C	D	E	F	G	H
		STT	input element	input label	input data	expected output element	expected output label	expected output state
0		1	text input	user-name		text output	pw	valid
1			text input	password				
2			button onclick input	login				
3		2	text input	user-name		text output	pw	invalid
4			text input	password				
5			button onclick input	login				
6		3	text input	user-name		text output	pw	empty
7			text input	password				
8			button onclick input	login				

Sheet1

gen_template_html.py

đọc file flow (txt) trả về UI template html

Input:

File gồm 1 flow (như hình dưới)

Trong trường hợp file có nhiều flow, sẽ chỉ lấy flow đầu tiên để sinh template

```
File Edit View

1. #ti #user-name #ti #password -> 2. #bi #login -> 3. #valid #pw #to
```

Output:

← → ↻ ⓘ File | D:/file-project/output/123.html

user-name

password

Hướng dẫn sử dụng:

Quy tắc (khi đọc usecase):

- Các flow có dạng: "Basic flow...", "Alternative flow..",...
- Các event có dạng: "1. ", "12. "
- Các annotation có dạng: [#string] (*string chỉ gồm chữ, số và có thể có dấu gạch nối -*)
- Các annotation được phân loại thành:
 - + element: đối tượng html
 - + state: các trạng thái (của output)
 - + label: trong trường hợp không thuộc 2 nhóm trên
- Thứ tự cho 1 cụm annotation trong 1 event:
State + element + label

Ví dụ: (ELEMENT + LABEL)

[#text-output][#DayLaText][#button] → [#text-output][#DayLaText] , [#button]

1 nút có tên là Close sẽ phải viết thành:

[#button][#Close] → <button>Close</button>

Thay vì:

[#Close][#button] → <label>Close</label> <button></button>

Các bước thực hiện yêu cầu 1:

Chuyển từ file usecase → file flows:

B1: Cho file usecase cần đọc vào folder: **input**

B2: Ở command line:

- Đi đến folder **project** bằng lệnh cd
- Gõ câu lệnh:
py gen_flows_txt.py
+ tên file usecase input (đã được đặt trong folder input) + tên file flows output

```
PS D:\file-project> cd project
PS D:\file-project\project> py gen_flows_txt.py usecases.txt flows.txt
```

B3: sau đó, ở folder **output** sẽ xuất hiện file flows.txt

```
▼ output
≡ flows.txt
```

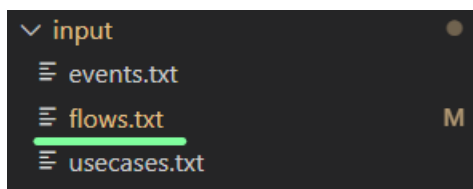
```
flows - Notepad
File Edit View

1. #ti #user-name #ti #password -> 2. #bi #login -> 3. #valid #pw #to
1. #ti #user-name #ti #password -> 2. #bi #login -> 3'. #invalid #pw #to
1. #ti #user-name #ti #password -> 2. #bi #login -> 3''. #empty #pw #to
```

Các bước thực hiện yêu cầu 2:

Tương tự như khi thực hiện yêu cầu 1

B1: Cho file flows cần đọc vào folder: **input**



B2: Ở command line, gõ lệnh:

py gen_testcases_excel.py

+ tên file usecase input (đã được đặt trong folder input) + tên file flows output

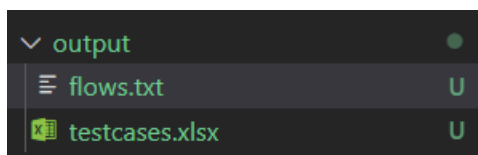
```
PS D:\file-project> cd project
PS D:\file-project\project> py gen_flows_txt.py usecases.txt flows.txt
PS D:\file-project\project> py gen_testcases_excel.py flows.txt testcases.xlsx
PS D:\file-project\project>
```

Ví dụ:

file input: flows.txt

File output: testcases.xlsx

B3: Ở folder output sẽ xuất hiện:



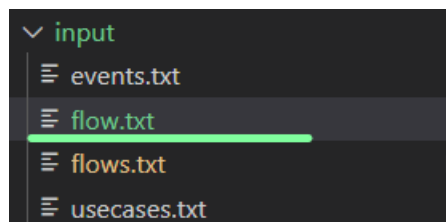
	B	C	D	E	F	G	H	I
	STT	put element	input label	input data	ed output	eted output	ted output	state
	1	text input	user-name		text output	pw	valid	
		text input	password					
		button on	login					
	2	text input	user-name		text output	pw	invalid	
		text input	password					
		button on	login					
	3	text input	user-name		text output	pw	empty	
		text input	password					
		button on	login					

(chú ý: cần kéo giãn các cột thủ công)

Các bước thực hiện yêu cầu 3:

Tương tự như khi thực hiện yêu cầu 1

B1: Cho file flow cần đọc vào folder: **input**



B2: Ở command line, gõ lệnh:

py gen_template_html.py

+ tên file usecase input (đã được đặt trong folder input) + tên file flows output

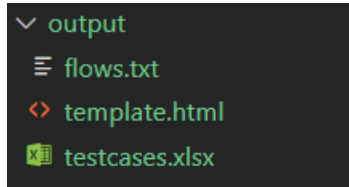
```
PS D:\file-project> cd project
PS D:\file-project\project> py gen_template_html.py flow.txt template.html
PS D:\file-project\project>
```

Ví dụ:

file input: flow.txt

File output: template.html

B3: Ở folder output sẽ xuất hiện: template.html



Mở file bằng chrome:

