

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH



KIỂM TRA PHẦN MỀM (CO3015)

Project #2

Black-Box Testing

GVHD: GS.TS. Bùi Hoài Thắng

Lớp: L02

Nhóm: BK-daklak

SVTH:	Nguyễn Thành Long	1914003
	Nguyễn Thúc Quân	1914837
	Lê Ngọc Tiến	1915473
	Nguyễn Đình An	1912526
	Nguyễn Trần Hoàng	1913446

Tp. Hồ Chí Minh, Tháng 03/2022

Mục lục

Phần A: (Application: Hotel manager)	5
Phân công công việc:	6
Tổng quan các function và non-function:	6
Các non-functional requirements:	6
Search room, Book room:	7
1. Môi trường kiểm thử:	7
2. Các thuật toán được áp dụng:	7
3. Test scenario:	8
Login	10
1. Environment:	10
2. Các thuật toán kiểm thử Black-Box Testing được sử dụng trong bài	10
3. Test scenario:	10
Test scenario 1:	10
Test scenario 2:	11
Test scenario 3:	11
Test scenario 4: Chức năng Login	12
Sign up	12
1. Environment:	13
2. Các thuật toán kiểm thử Black-Box Testing được sử dụng trong bài:	13
3. Test scenario:	13
Test scenario 1:	13
Test scenario 2:	14
Test scenario 3:	14
Test scenario 4:	15
Test scenario 5:	16
Manager room, edit room.	16

1. Môi trường kiểm thử:	16
2. Các thuật toán được áp dụng:	16
3. Test scenario:	17
Add a new room.	19
1. Môi trường kiểm thử:	19
2. Các thuật toán được áp dụng:	19
3. Test scenario:	19
Phần B:	21
Tổng quan:	21
Sort:	21
1. Application: SORT của trang web https://visualgo.net/en :	21
2. Environment:	21
3. Các thuật toán kiểm thử Black-Box Testing được sử dụng trong bài:	21
4. Non-function:	22
Test scenario 1: Các giá trị nhập vào thỏa mãn x sao cho	22
Test scenario 2: Số các phần tử nhập vào thỏa mãn:	22
Note:	23
Function:	23
Test scenario: Chức năng Sorting để sắp xếp các số được nhập từ bàn phím.	23
Linked list:	25
1. Application: LINKED LIST của trang web https://visualgo.net/en :	25
2. Environment:	25
3. Các thuật toán kiểm thử Black-Box Testing được sử dụng trong bài:	25
4. Non-function:	26
Test scenario 1: Các giá trị nhập vào thỏa mãn x sao cho $1 \leq x \leq 50$	26
Test scenario 2: Số các phần tử nhập vào thỏa mãn:	26
Note:	26
5. Function:	27
Binary Heap:	30
1. Application: BINARY HEAP của trang web https://visualgo.net/en :	30
2. Environment:	30

3. Các thuật toán kiểm thử Black-Box Testing được sử dụng trong bài:	30
4. Non-function:	30
· Test scenario 1: Khởi tạo Binary Heap từ 1 đến 18 phần tử	30
Test scenario 2: Kiểm tra giới hạn giá trị của 1 phần tử khi khởi tạo Binary Heap	31
· Test scenario 3: Kiểm các giá trị giống nhau và giá trị đặc biệt khi khởi tạo Binary Heap	31
· Test scenario 4: Insert phần tử thứ 2 đến 18 trong Binary Heap	32
· Test scenario 5: Kiểm các giá trị giống nhau và giá trị đặc biệt khi thêm giá trị cho Binary Heap	32
· Test scenario 6: Cập nhật các phần tử từ vị trí 1 đến vị trí cuối trong Binary Heap	32
· Test scenario 7: Cập nhật các giá trị từ 1 đến 99 cho Binary Heap	33
· Test scenario 8: Kiểm các giá trị giống nhau và giá trị đặc biệt khi thêm giá trị cho Binary Heap	33
· Test scenario 9: Delete từ vị trí 1 đến vị trí cuối trong Binary Heap	33
· Test scenario 10: Delete các vị trí đặc biệt trong Binary Heap	34
5. Function:	34
Test scenario: Các chức năng Create, Insert, Update, Delete trong Binary Heap	34
Binary Search Tree:	42
1. ENVIRONMENT	42
2. CÁC THUẬT TOÁN KIỂM THỬ ĐƯỢC ÁP DỤNG	43
3. NON-FUNCTIONAL REQUIREMENTS	43
Test scenario 1	43
Test scenario 2:	43
4. FUNCTIONAL REQUIREMENTS	44
Test scenario: Chức năng Insert của cây BST	44
Hash Table:	47
1. Application: Hash Table của trang web https://visualgo.net/en :	47
2. Environment:	47
3. Các thuật toán kiểm thử Black-Box Testing được sử dụng trong bài:	47
4. Non-function:	47
Test scenario 1: Kiểm tra khả năng khởi tạo Hash Table	47

Test scenario 2: Kiểm tra khả năng nhập vào của chức năng Insert	48
Test scenario 3: Kiểm tra khả năng nhập vào của chức năng Search của Hash Table	48
Test scenario 4: Kiểm tra khả năng nhập vào của chức năng Remove của Hash Table	49
5. Function:	49
Test scenario: Chức năng Insert trong Binary Heap	49
Activity Graph:	50
Decision table:	50
Tài liệu tham khảo	51

A. Phần A: (Application: Hotel manager)

(link git 1.2)

I. Phân công công việc:

STT	Họ và Tên	MSSV	Công việc	Đóng góp
1	Nguyễn Thành Long	1914003	Book room, Responsive, Input field day check out, check in.	20%
2	Nguyễn Thúc Quân	1914837	Log-in, Log-out, Sign-in.	20%
3	Lê Ngọc Tiến	1915473	Add new room.	20%
4	Nguyễn Đình An	1912526	Book room, input filed children, adults.	20%
5	Nguyễn Trần Hoàng	1913446	Manager room, edit room.	20%

II. Tổng quan các function và non-function:

Trong báo cáo này nhóm sẽ tiến hành kiểm thử trang web Hotel Manager.

Các **functional requirement** mà nhóm sẽ tiến hành kiểm thử:

- Login, logout, sign up.
- Search room, Book room.
- Đối với admin:
 - + Manager room, edit room.
 - + Add a new room.
 - + Manager User account.

Các **non-functional requirement** mà nhóm sẽ tiến hành kiểm thử:

- Responsive.
- Giá trị ngày tháng của “check out” lớn hơn hoặc bằng “check in”.
- Input filed Adults > 0.
- Input field Children > 0.

III. Các non-functional requirements:

1. Responsive:

● Test scenario 1:

- Mục đích: Trang web có thể hiển thị tương thích với nhiều thiết bị có kích cỡ khác nhau.
- Quy trình thực hiện:
 - + Bước 1: Mở trang chính của giao diện trang web.
 - + Bước 2: Điều chỉnh lần lượt độ rộng của trang web.
- Các tình huống cần test:
 - + 320x568.
 - + 375x667.

+ 414x736.

+ 375x812.

+ 600x800.

+ 768x1024.

+ 1024x1366.

+ 1920x1080.

- Test scenario 2:

- Mục đích: kiểm tra ràng buộc ngày checkout phải lớn hơn hoặc bằng ngày checkin.
- Quy trình thực hiện:
 - + Bước 1: Nhấp vào ô check in tại màn hình chính, sau đó chọn ngày checkin.
 - + Bước 2: Nhấp vào ô check out tại màn hình chính, sau đó chọn ngày checkout.
- Các tình huống cần test: (Sử dụng thuật toán Boundary value analysis technique)
 - + Chọn ngày checkin lớn hơn ngày check out.
 - + Chọn ngày checkin nhỏ hơn ngày check out.
 - + Chọn ngày checkin bằng ngày check out.

2. Adults age

- Test scenario 1:

- Mục đích: Kiểm tra khi người dùng nhập vào tuổi adults không hợp lệ
- Quy trình thực hiện
 - + Bước 1: Mở trang chính của giao diện trang web.
 - + Bước 2: Bấm nút "Guest" xong nhập tuổi Adults vào
- Các tình huống cần test:
 - + Nhập tuổi "-12"
 - + Nhập tuổi "0"
 - + Nhập tuổi "5"
 - + Nhập tuổi "15"

IV. Search room, Book room:

- Bước 1: Sau khi khởi chạy trang web, ta vào trang theo đường link <http://localhost/hotel-management-php/index.php>
- Bước 2: Tại giao diện chính của trang (như hình dưới). Ta tiến hành nhập ngày tháng checkin, checkout, điền các field input rồi bấm chọn nút "Search" để tiến hành book phòng.

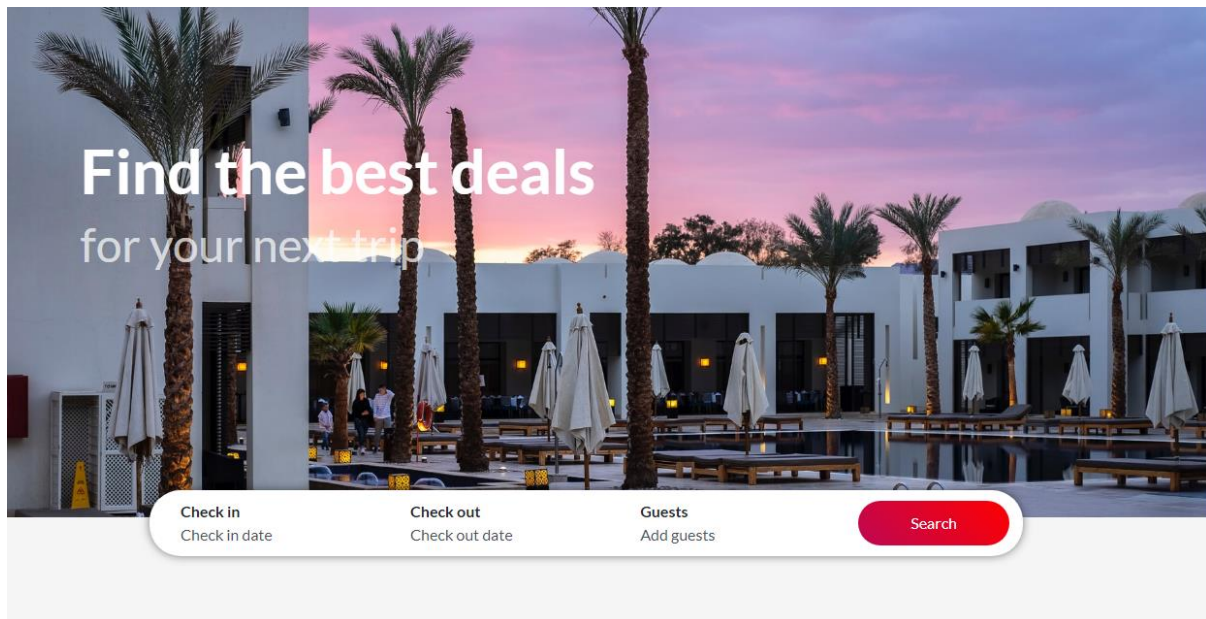
1. Môi trường kiểm thử:

Bài sẽ được kiểm thử trên hai môi trường trình duyệt đó là:

- Google Chrome.
- Firefox.

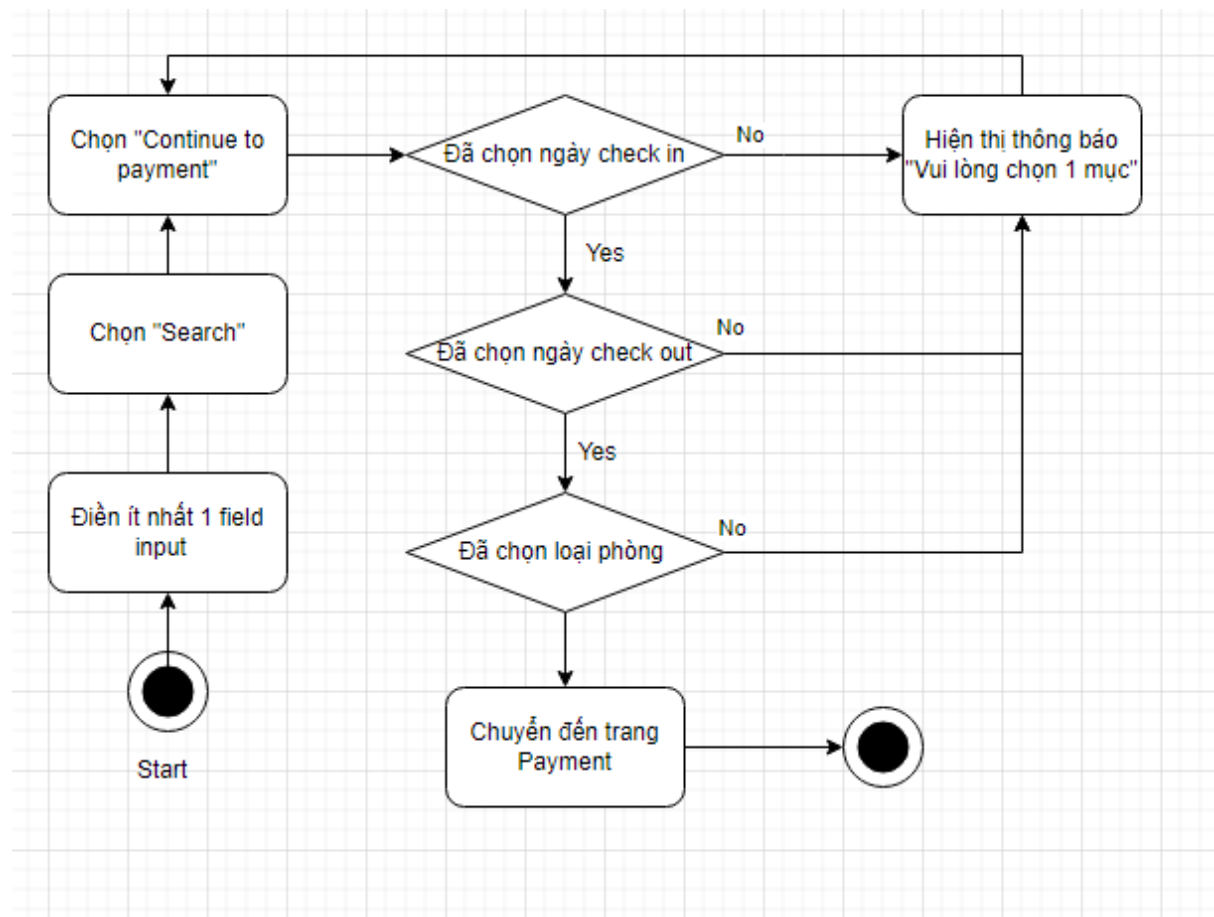
2. Các thuật toán được áp dụng:

- Boundary value analysis technique.
- Equivalence class partitioning technique.
- Decision table technique.
- Use-case testing.



3. Test scenario:

- Chức năng: Tìm phòng và đặt phòng dựa trên ngày check in, check out, và loại phòng.
- Mục đích: Kiểm tra chức năng đặt phòng, tìm phòng.
- Quy trình thực hiện:
 - Bước 1: Nhập ít nhất một field trong màn hình chính.
 - Bước 2: Bấm chọn "Search".
 - Bước 3: Tiến hành điền các field input còn thiếu.
 - Bước 4: Bấm chọn "Continue to payment".
 - Bước 5: Chuyển hướng đến trang Payment.
- Activity graph:



- Decision Table:

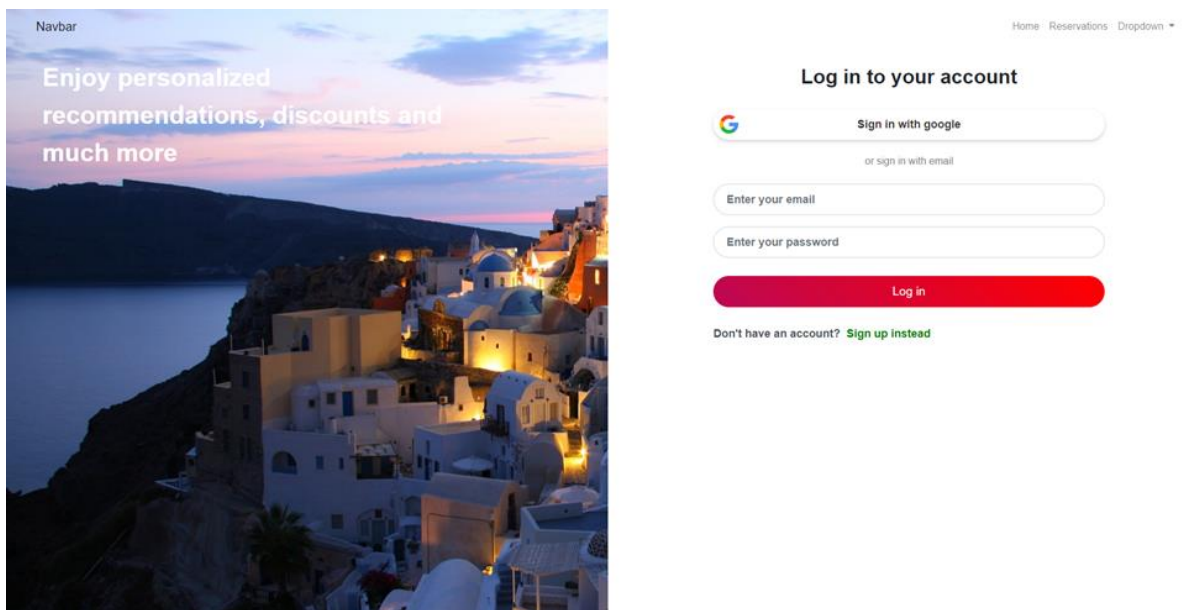
	Rule 1-4	Rule 4-6	Rule 7	Rule 8
<u>Conditions:</u>				
Đã chọn ngày check in	F	T	T	T
Đã chọn ngày check out	--	F	T	T
Đã chọn loại phòng	--	--	F	T
<u>Actions:</u>				
Chuyển trang Payment				X
Xuất ra thông báo	X	X	X	

- Các tình huống cần test:

- Không chọn ngày check in.
- Không một ngày check out.
- Không chọn loại phòng.
- Không điền field input nào trong trang chính.
- Chọn mỗi ngày check in.
- Chọn mỗi ngày check out.
- Chọn đầy đủ các field.
- Chọn hai field trong ba field, field còn lại để trống.

V. Login

Người dùng chọn vào mục Login và sẽ xuất hiện giao diện như bên dưới:



1. Environment:

Bài kiểm thử được thực hiện trên hai trình duyệt cụ thể là:

- Google chrome.
- Firefox.

2. Các thuật toán kiểm thử Black-Box Testing được sử dụng trong bài

- Boundary value analysis technique.
- Equivalence class partitioning technique.

3. Test scenario:

- *Test scenario 1:*

Giá trị username có độ dài từ 11 đến 30

Mục đích: Kiểm tra giá trị hợp lệ của trường Username trong form Login.

Quy trình thực hiện:

- Bước 1: Chọn mục login

- Bước 2: Nhập Username
- Bước 3: Ấn nút Login

Các tình huống cần test:

- Username có độ dài là 11
- Username có độ dài là 30
- Username có độ dài là 20
- Username có độ dài < 11
- Username có độ dài > 30

- *Test scenario 2:*

Giá trị Username định dạng là email

Mục đích: Kiểm tra định dạng của trường Username

Quy trình thực hiện:

- Bước 1: Chọn mục login
- Bước 2: Nhập Username
- Bước 3: Ấn nút Login

Các tình huống cần test:

- Username có dạng là email
- Username không có dạng là email

- *Test scenario 3:*

Password có độ dài từ 6 đến 10

Mục đích: Kiểm tra giá trị hợp lệ của trường Password

Quy trình thực hiện:

- Bước 1: Chọn mục login
- Bước 2: Nhập Password
- Bước 3: Nhấn nút Login

Các tình huống cần test

- Password có độ dài là 6
- Password có độ dài là 10
- Password có độ dài là 8
- Password có độ dài < 6
- Password có độ dài > 10

- *Test scenario 4:*

Chức năng Login

Mục đích: Kiểm tra chức năng Login

Quy trình thực hiện:

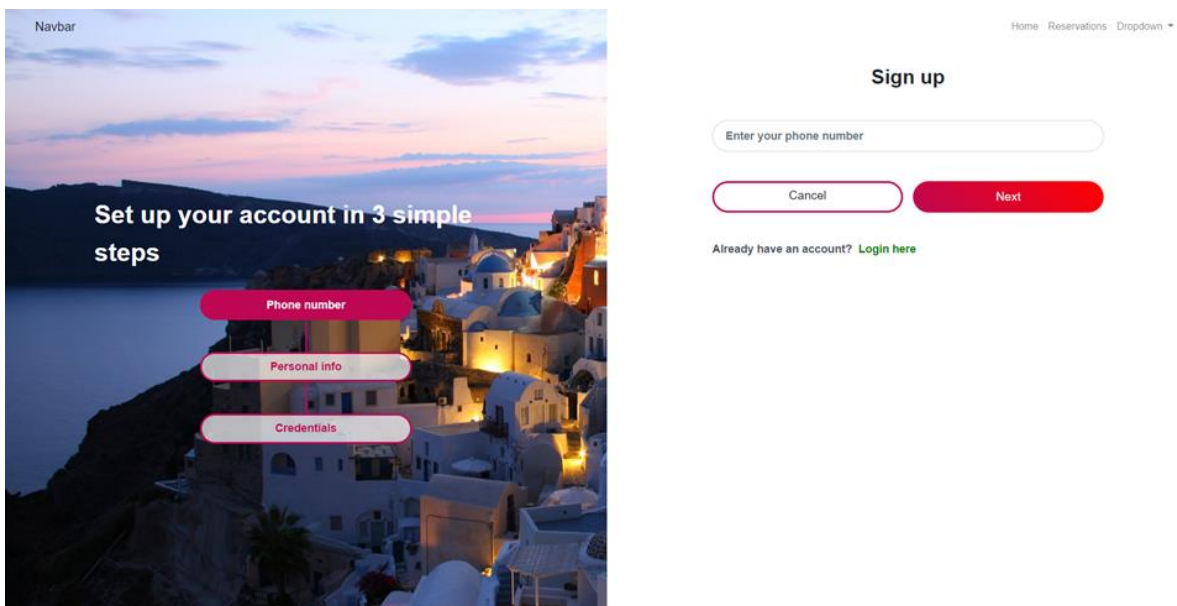
- Bước 1: Chọn vào mục Login
- Bước 2: Nhập các giá trị email, password
- Bước 3: Ấn nút Login

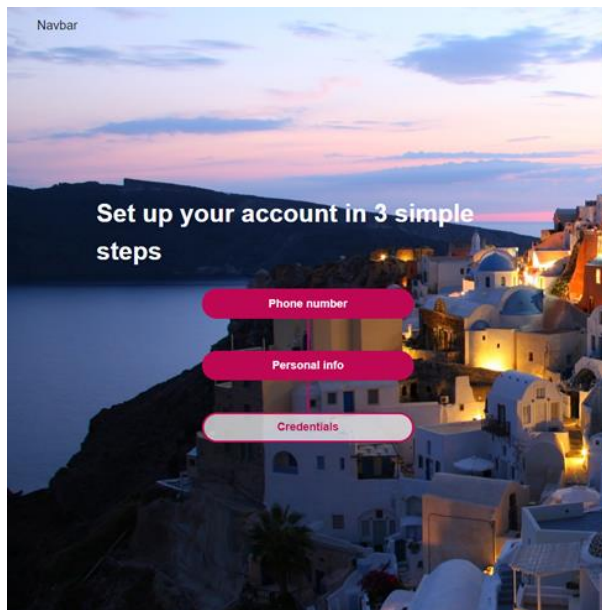
Đầu vào:

- Username và password đúng
- Username đúng nhưng password sai
- Username sai nhưng password đúng
- Cả username và password để sai

VI. Sign up

Người dùng chọn vào mục Sign up và sẽ xuất hiện giao diện như bên dưới:





Home Reservations Dropdown

Sign up

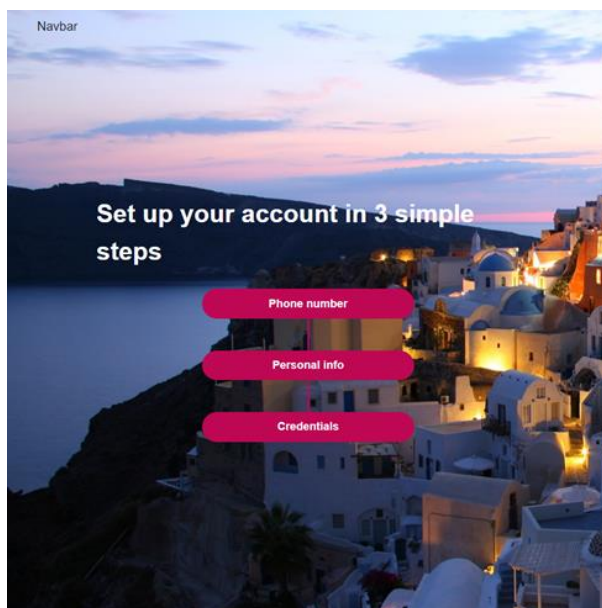
Enter your first name

Enter your last name

Enter your birth date

Back Next

Already have an account? [Login here](#)



Home Reservations Dropdown

Sign up

Enter your email

Enter your password

Back Signup

Already have an account? [Login here](#)

1. Environment:

Bài kiểm thử được thực hiện trên hai trình duyệt cụ thể là:

- Google chrome.
- Firefox.

2. Các thuật toán kiểm thử Black-Box Testing được sử dụng trong bài:

- Boundary value analysis technique.
- Equivalence class partitioning technique.

3. Test scenario:

- *Test scenario 1:*

Phone number có độ dài là 10

Mục đích: Kiểm tra giá trị hợp lệ của trường Phone number trong Sign up

Quy trình thực hiện:

- Bước 1: Chọn mục Sign up
- Bước 2: Nhập phone number
- Bước 3: Ấn nút Next

Các tình huống cần test:

- Phone number có 10 số
- Phone number khác 10 số
- Phone number chứa ký tự đặc biệt

● *Test scenario 2:*

Birthday không được là ngày sau ngày hiện tại

Mục đích: Kiểm tra giá trị hợp lệ của trường Birthday

Quy trình thực hiện:

- Bước 1: Chọn mục Sign up
- Bước 2: Nhập phone number
- Bước 3: Ấn nút Next
- Bước 4: Nhập first name và last name, birthday
- Bước 5: Nhấn next

Các tình huống cần test:

- Birthday bé hơn hoặc bằng ngày hiện tại
- Birthday là ngày ở tương lai

● *Test scenario 3:*

Giá trị Username định dạng là email

Mục đích: Kiểm tra định dạng của trường Username

Quy trình thực hiện:

- Bước 1: Chọn mục Sign up

- Bước 2: Nhập phone number
- Bước 3: Ấn nút Next
- Bước 4: Nhập first name và last name, birthday
- Bước 5: Nhấn next
- Bước 6: Nhập username, password
- Bước 7: Nhấn sign up

Các tình huống cần test:

- Username có dạng là email
- Username không có dạng là email

- *Test scenario 4:*

Giá trị Username có độ dài từ 11 tới 30

Mục đích: Kiểm tra định dạng của trường Username

Quy trình thực hiện:

- Bước 1: Chọn mục Sign up
- Bước 2: Nhập phone number
- Bước 3: Ấn nút Next
- Bước 4: Nhập first name và last name, birthday
- Bước 5: Nhấn next
- Bước 6: Nhập username, password
- Bước 7: Nhấn sign up

Các tình huống cần test:

- Username có độ dài là 11
- Username có độ dài là 30
- Username có độ dài là 20
- Username có độ dài < 11
- Username có độ dài > 30

- *Test scenario 5:*

Password có độ dài từ 6 đến 10

Mục đích: Kiểm tra giá trị hợp lệ của trường Password

Quy trình thực hiện:

- Bước 1: Chọn mục Sign up
- Bước 2: Nhập phone number
- Bước 3: Ấn nút Next
- Bước 4: Nhập first name và last name, birthday
- Bước 5: Nhấn next
- Bước 6: Nhập username, password
- Bước 7: Nhấn sign up

Các tình huống cần test

- Password có độ dài là 6
- Password có độ dài là 10
- Password có độ dài là 8
- Password có độ dài < 6
- Password có độ dài > 10

VII. Manager room, edit room.

- Bước 1: Sau khi khởi chạy trang web, ta vào trang theo đường link <http://localhost/hotel-management-php/admin/rooms.php>
- Bước 2: Tại giao diện chính của trang (như hình dưới). Ta tiến hành xóa hoặc edit các room khi bấm vào các action ở cột cuối cùng

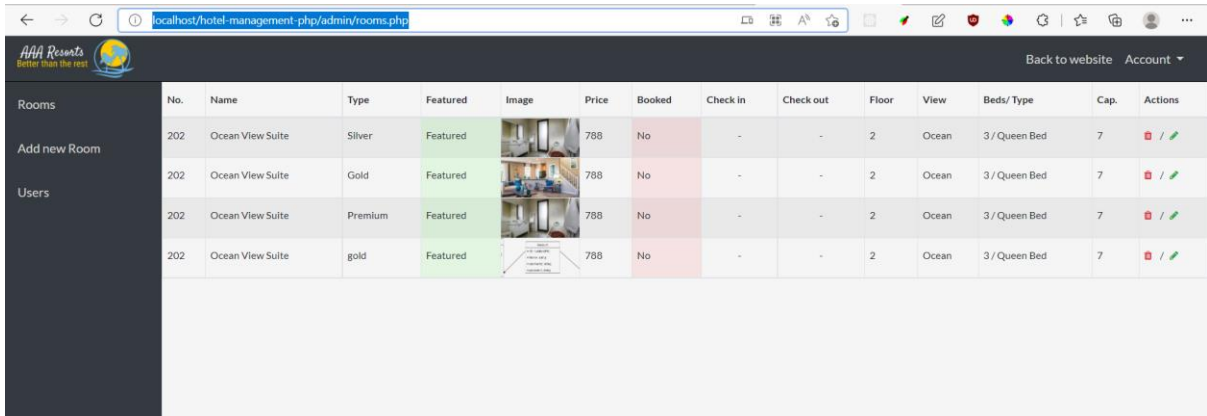
1. Môi trường kiểm thử:

Bài sẽ được kiểm thử trên hai môi trường trình duyệt đó là:













- Google Chrome
- Firefox

2. Các thuật toán được áp dụng:

- Boundary value analysis technique.
- Equivalence class partitioning technique.
- Decision table technique.
- Use-case testing.

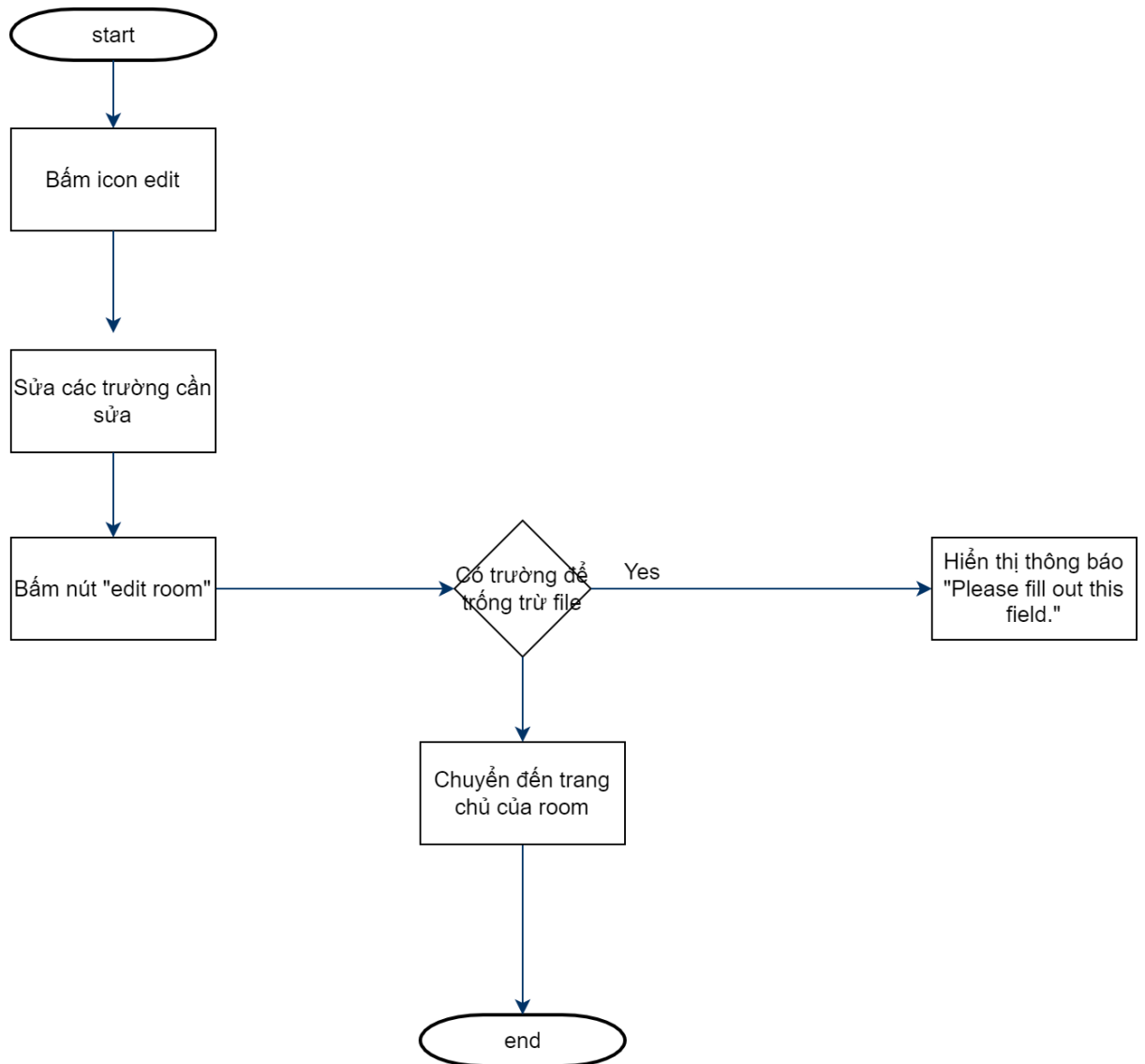


The screenshot shows a web browser window with the URL `localhost/hotel-management-php/admin/rooms.php`. The page displays a table of rooms with the following columns: No., Name, Type, Featured, Image, Price, Booked, Check in, Check out, Floor, View, Beds/ Type, Cap., and Actions. The table contains four rows of data, all for 'Ocean View Suite' rooms on floor 2, priced at 788. The 'Booked' status is 'No' for all, and 'Check in' and 'Check out' are both '-'. The 'Actions' column contains edit and delete icons for each row.

No.	Name	Type	Featured	Image	Price	Booked	Check in	Check out	Floor	View	Beds/ Type	Cap.	Actions
202	Ocean View Suite	Silver	Featured		788	No	-	-	2	Ocean	3 / Queen Bed	7	 
202	Ocean View Suite	Gold	Featured		788	No	-	-	2	Ocean	3 / Queen Bed	7	 
202	Ocean View Suite	Premium	Featured		788	No	-	-	2	Ocean	3 / Queen Bed	7	 
202	Ocean View Suite	gold	Featured		788	No	-	-	2	Ocean	3 / Queen Bed	7	 

3. Test scenario:

- Chức năng: Chỉnh sửa lại phòng cho phù hợp khi có nhu cầu
- Mục đích: Kiểm tra chức năng chỉnh sửa phòng
- Quy trình thực hiện
 - Bước 1: Bấm nút icon edit ở cột action ngoài cùng bên phải
 - Bước 2: Chuyển hướng đến trang edit room
 - Bước 3: Chỉnh sửa các field cần sửa
 - Bước 4: Bấm nút "Edit room" để hoàn tất việc chỉnh sửa
 - Bước 5: Chuyển hướng đến trang room
- Activity graph



- Decision Table

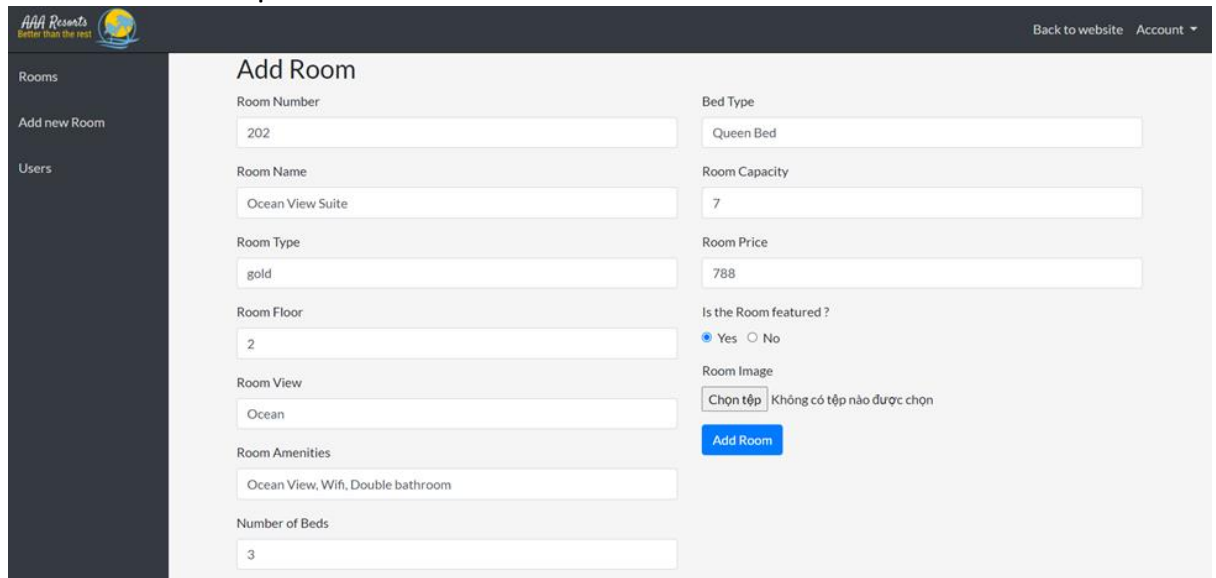
	Rule 1-4	Rule 4-6	Rule 7	Rule 8
<u>Conditions:</u>				
Không để trống	F	T	T	T
Chọn file	--	F	T	T
Điền chữ vào trường number	--	--	F	T
<u>Actions:</u>				
Chuyển trang Room		X	X	
Xuất ra thông báo	X			X

- Các tình huống cần test

- Để trống trường "Room Number"
- Để trống trường "Room Name"
- Để trống trường "Room Type"
- Để trống trường "Room Floor"
- Để trống trường "Room View"
- Để trống trường "Room Amenities"
- Để trống trường "Number of Beds"
- Để trống trường "Bed Type"
- Để trống trường "Room Capacity"
- Để trống trường "Room Room feature"
- Để trống trường "Room Image"
- Không để trống trường nào

VIII. Add a new room.

Để truy cập vào chức năng Add new Room, người dùng phải đăng nhập bằng tài khoản admin và chọn vào tab Add new Room như hình dưới:



1. Môi trường kiểm thử:

Bài sẽ được kiểm thử trên hai môi trường trình duyệt đó là:

- Google Chrome
- Firefox

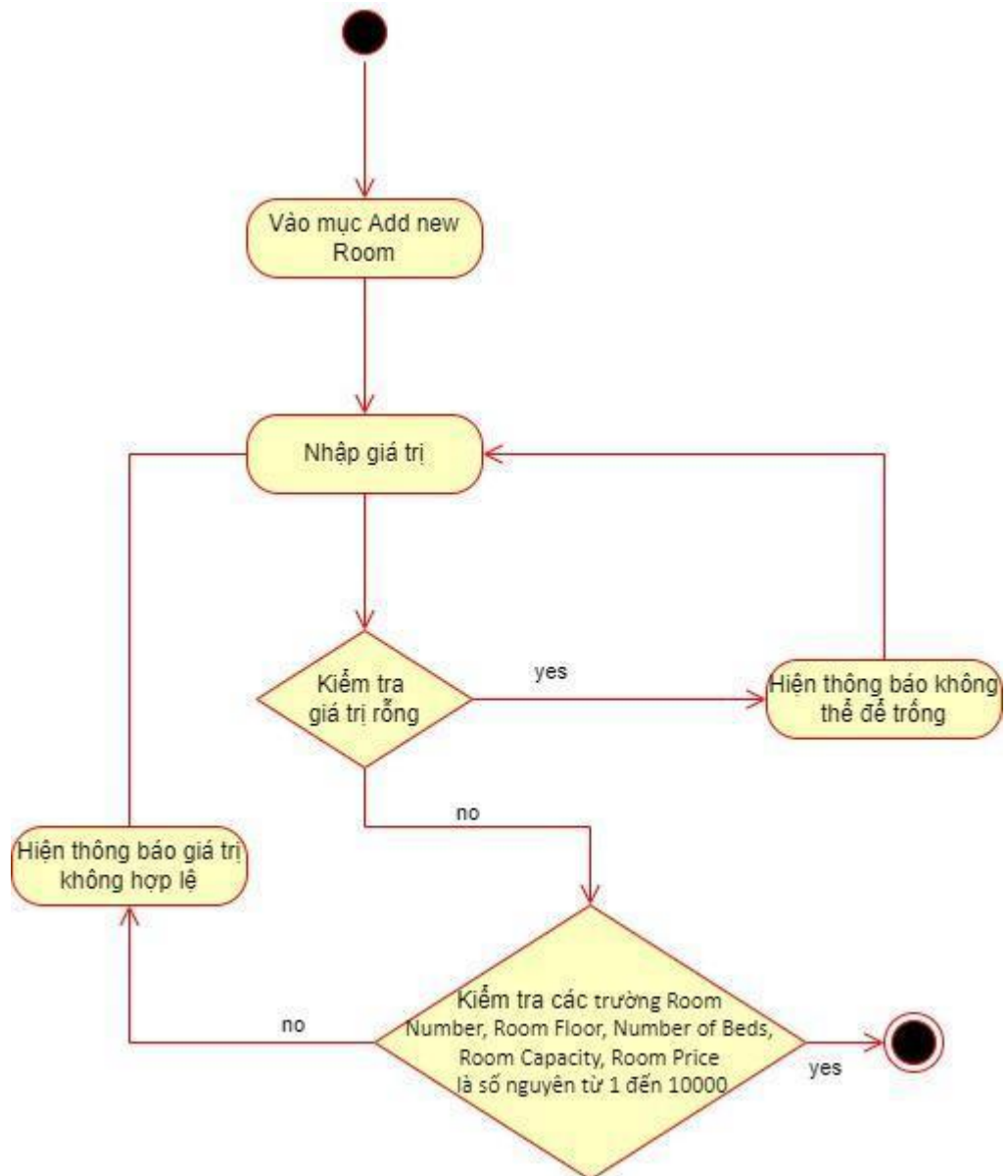
2. Các thuật toán được áp dụng:

- Boundary value analysis technique.
- Equivalence class partitioning technique.
- Decision table technique.
- Use-case testing.

3. Test scenario:

- Chức năng: Add new Room
- Mục đích: Kiểm tra chức năng Add new Room
- Quy trình thực hiện
 - Bước 1: Đăng nhập bằng tài khoản admin và vào mục Add new Room
 - Bước 2: Nhập các giá trị
 - Bước 3: Ấn nút Add Room

- Activity graph



- Decision Table

	Rule 1-2	Rule 3	Rule 4
<u>Conditions:</u>			
Đầu vào rỗng	T	F	T
Đầu vào các trường Room Number, Room Floor, Number of Beds, Room Capacity, Room Price là số nguyên từ 1 đến 10000	--	T	F
<u>Actions:</u>			
Nhập thành công		X	
Thông báo lỗi	X		X

- Các tình huống cần test
 - Không nhập đầu vào.
 - Đầu vào các trường Room Number, Room Floor, Number of Beds, Room Capacity, Room Price là số nguyên từ 1 đến 10000.
 - Đầu vào các trường Room Number, Room Floor, Number of Beds, Room Capacity, Room Price không phải số nguyên từ 1 đến 10000.

B. Phần B:

I. Tổng quan:

II. Sort:

1. Application: SORT của trang web <https://visualgo.net/en>:

Bước 1: Vào trang <https://visualgo.net/en>.

Bước 2: Chọn Sorting trong tab đầu tiên (như hình bên dưới).

2. Environment:

Bài kiểm thử được thực hiện trên hai trình duyệt cụ thể là:

- Google chrome.
- Firefox.

3. Các thuật toán kiểm thử Black-Box Testing được sử dụng trong bài:

- Boundary value analysis technique.
- Equivalence class partitioning technique.

- Decision table technique.
- Use-case testing.

4. Non-function:

- *Test scenario 1: Các giá trị nhập vào thỏa mãn x sao cho*

Mục đích: Kiểm tra vùng giá trị có thể nhập vào để thực hiện Sorting.

Quy trình thực hiện:

Bước 1: Chọn Create.

Bước 2: Nhập số.

Bước 3: Chọn Go.

Các tình huống cần test: (được suy ra từ hai thuật toán **Equivalence class partitioning technique** và **Boundary value analysis technique**).

(được viết gọn lần lượt là E và B)

- Nhập số trong (1, 50). (E và B)
- Nhập số 1. (B)
- Nhập số 50. (B)
- Nhập số < 1 . (B và E)
- Nhập số > 50 . (B và E)

(Trong file excel Sheet/SORT các Test case NF-001 đến NF-005.)

- *Test scenario 2: Số các phần tử nhập vào thỏa mãn:*

Mục đích: Kiểm tra số phần tử có thể nhập vào để thực hiện Sorting.

Quy trình thực hiện:

Bước 1: Chọn Create.

Bước 2: Nhập số.

Bước 3: Chọn Go.

Các tình huống cần test: (được suy ra từ hai thuật toán **Equivalence class partitioning technique** và **Boundary value analysis technique**).

(được viết gọn lần lượt là E và B)

- Nhập số phần tử trong khoảng (1, 20) cụ thể là 10. (B và E)
- Nhập một phần tử. (B)
- Nhập 20 phần tử. (B)
- Không nhập phần tử nào. (B và E)
- Nhập lớn hơn 20 phần tử. (B và E)
- Nhập 19 phần tử. (B)

([Trong file excel Sheet/SORT các Test case NF-006 đến NF-011.](#))

● *Note:*

- Khi nhập các phần tử phải được ngăn cách bởi dấu “,”.
- Có thể nhập các số thập phân (có thể là 1.3, 2.4,...).

5. Function:

- *Test scenario: Chức năng Sorting để sắp xếp các số được nhập từ bàn phím.*

Mục đích: Kiểm tra chức năng Sorting của trang web.

Quy trình thực hiện:

Bước 1: Chọn Create.

Bước 2: Nhập số.

Bước 3: Chọn Go.

Bước 4: Chọn Sort (ở tab dưới Create).

Bước 5: Chọn Go.

Activity Graph:

Decision table:

	Rule 1-8	Rule 9-12	Rule 13-14	Rule 15	Rule 16
<u>Conditions:</u>					

Đầu vào rỗng.	T	F	F	F	F
Đầu vào lớn hơn 20 phần tử.	—	T	F	F	F
Đầu vào có giá trị trong đoạn [1, 50].	—	—	F	T	T
Đầu vào chỉ là các số.	—	—	—	T	F
<u>Actions:</u>					
Thực hiện Sort.				X	
Xuất ra Message.	X	X	X		X

Các tình huống cần test:

· **Dựa vào *Decision table*:**

- Không nhập đầu vào.
- Nhập đầu vào có số phần tử lớn hơn 20.
- Nhập vào số phần tử trong đoạn [1, 20] và trong các số nhập vào có chứa một số không nằm trong đoạn [1, 50].
- Nhập vào số phần tử trong đoạn [1, 20] và các số nhập vào nằm trong đoạn [1, 50] nhưng có chứa vài kí tự đặc biệt khác số.
- Nhập vào số phần tử trong đoạn [1, 20] và các số nhập vào nằm trong đoạn [1, 50].

· **Dựa vào *Use-case* , *Equivalence class partitioning technique* và *Boundary value analysis technique*.**

- Không nhập đầu vào.
- Nhập vượt quá 20 phần tử.
- Nhập trong đoạn [1, 20] phần tử.
- Nhập các số trong đoạn [1, 50].
- Nhập các số nhỏ hơn 1 (có thể gần 1).
- Nhập các số lớn hơn 50 (có thể gần 50).
- Nhập các số trong đoạn [1, 50].
- Nhập không phải là các số (có thể là ",", các kí tự đặc biệt,...).

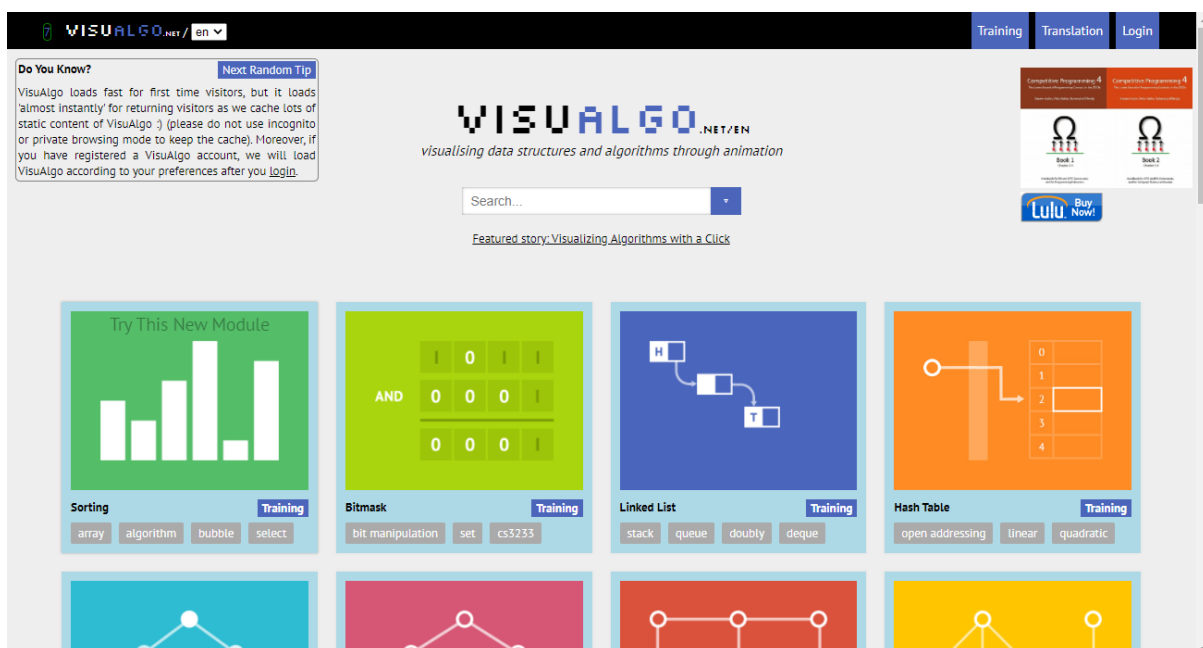
(Trong file excel Sheet/SORT các Test case F-001 đến F-008.)

III. Linked list:

1. Application: LINKED LIST của trang web <https://visualgo.net/en>:

Bước 1: Vào trang <https://visualgo.net/en>.

Bước 2: Chọn Sorting trong tab thứ 3 (như hình bên dưới).



2. Environment:

Bài kiểm thử được thực hiện trên hai trình duyệt cụ thể là:

- Google chrome.
- Firefox.

3. Các thuật toán kiểm thử Black-Box Testing được sử dụng trong bài:

- Boundary value analysis technique.
- Equivalence class partitioning technique.
- Decision table technique.
- Use-case testing.

4. Non-function:

- *Test scenario 1: Các giá trị nhập vào thỏa mãn x sao cho $1 \leq x \leq 50$*

Mục đích: Kiểm tra vùng giá trị có thể nhập vào để thực hiện các tác vụ khác.

Các tình huống cần test:

- Nhập số trong (1, 50).
- Nhập số 1.
- Nhập số 50.
- Nhập số < 1 .
- Nhập số > 50 .

- *Test scenario 2: Số các phần tử nhập vào thỏa mãn:*
 $1 \leq x \leq 20$.

Mục đích: Kiểm tra số phần tử có thể nhập vào để thực hiện các tác vụ khác.

Quy trình thực hiện:

Bước 1: Chọn Create.

Bước 2: Nhập mảng.

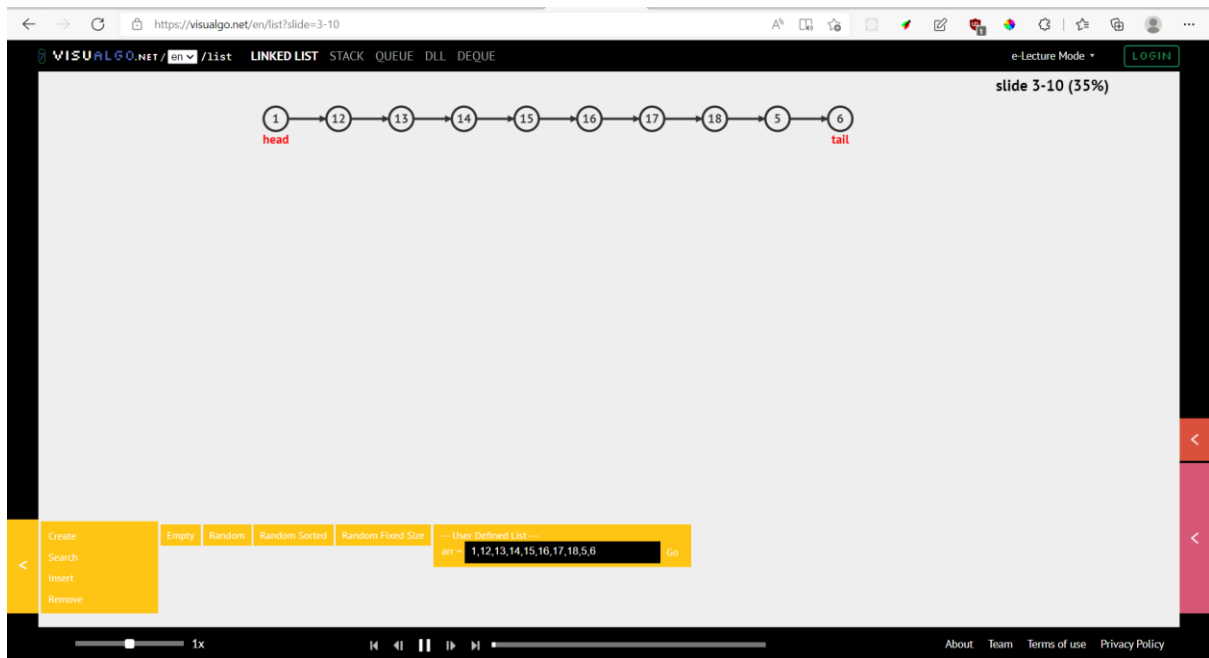
Bước 3: Chọn Go.

Các tình huống cần test:

- Nhập số phần tử trong khoảng (1, 20)
- Nhập một phần tử.
- Nhập 20 phần tử.
- Không nhập phần tử nào.
- Nhập lớn hơn 20 phần tử

- *Note:*

- Khi nhập các phần tử phải được ngăn cách bởi dấu “,”.
- Có thể nhập các số thập phân (có thể là 1.3, 2.4,...).



5. Function:

5.1: Create

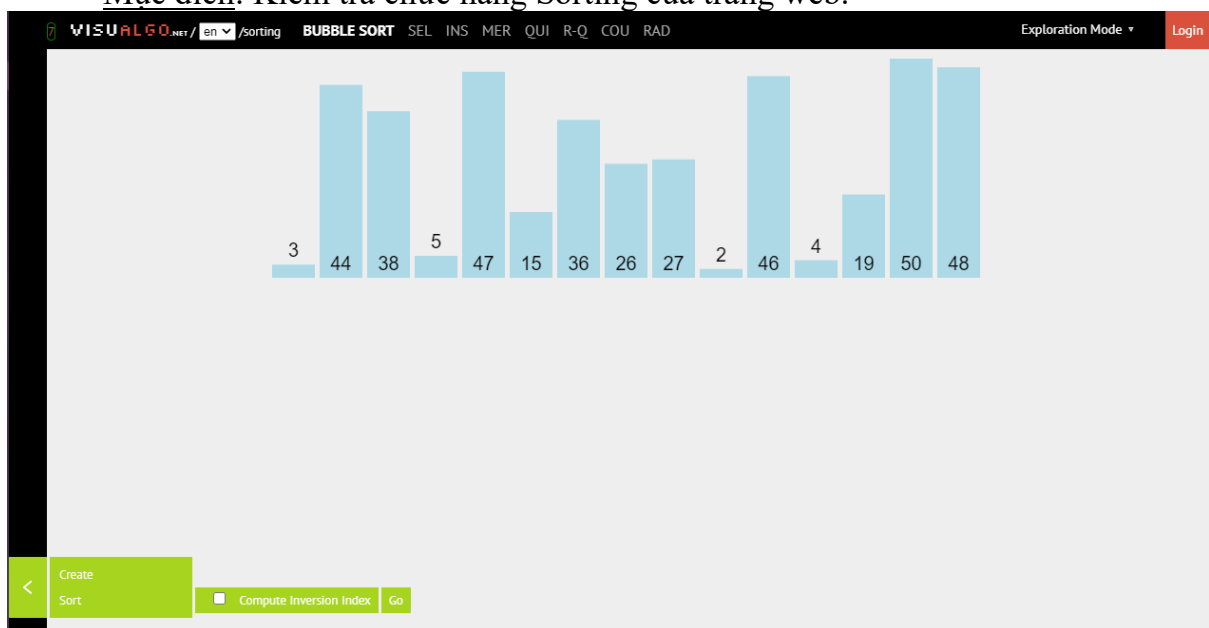
5.2: Remove

5.3: Insert

5.4: Search

• **Test scenario:** Chức năng Search để tìm kiếm các số được nhập từ bàn phím.

Mục đích: Kiểm tra chức năng Sorting của trang web.



Quy trình thực hiện:

Bước 1: Chọn Create.

Bước 2: Nhập số.

Bước 3: Chọn Go.

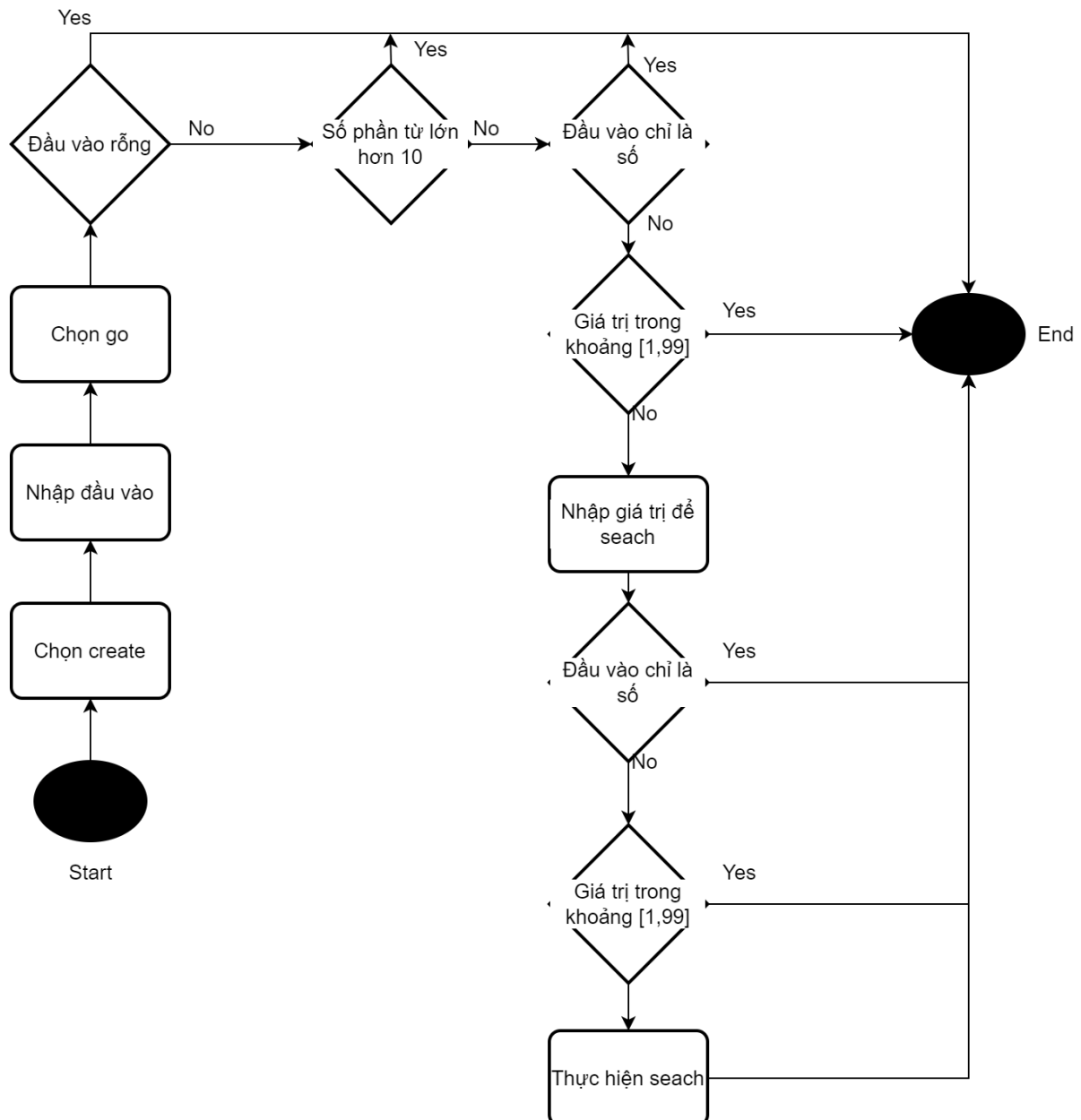
Bước 4: Chọn Search(ở tab dưới Create).

Bước 5: Chọn Go.

Đầu vào:

- Số trong đoạn $1 \leq x \leq 50$.
- Số nhỏ hơn 1.
- Số lớn hơn 50.
- Không nhập gì cả.
- Nhập vào nhưng không phải số (có thể là các kí tự nào đó).
- Nhập vào số phần tử trong đoạn $1 \leq x \leq 20$.
- Nhập vào số phần tử lớn hơn 20.
- Nhập vào các số thập phân.

Activity Graph:



Decision table:

	Rule 1-7	Rule 8-12	Rule 13-14	Rule 15	Rule 16
Conditions					
Đầu vào rỗng	T	F	F	F	F
Đầu vào lớn hơn 10 phần tử	-	T	F	F	F
Đầu vào có giá trị trong đoạn [1,99]	-	-	F	T	T
Đầu vào chỉ là số	-	-	T	T	F
Actions					
Thực hiện search				X	
Xuất ra message	X	X	X		X

Các tình huống cần test:

- Dựa vào Decision table:
 - Không nhập đầu vào.
 - Nhập đầu vào có số phần tử lớn hơn 20.
 - Nhập vào số phần tử trong đoạn [1, 20] và trong các số nhập vào có chứa một số không nằm trong đoạn [1, 99].
 - Nhập vào số phần tử trong đoạn [1, 20] và các số nhập vào nằm trong đoạn [1, 99] nhưng có chứa vài kí tự đặc biệt khác số.
 - Nhập vào số phần tử trong đoạn [1, 20] và các số nhập vào nằm trong đoạn [1, 99].
- Dựa vào Use-case , Equivalence class partitioning technique và Boundary value analysis technique.

- Không nhập đầu vào.
- Nhập vượt quá 20 phần tử.
- Nhập trong đoạn [1, 20] phần tử.
- Nhập các số trong đoạn [1, 99].
- Nhập các số nhỏ hơn 1 (có thể gần 1).
- Nhập các số lớn hơn 99 (có thể gần 99).
- Nhập các số trong đoạn [1, 99].
- Nhập không phải là các số (có thể là “,” , các kí tự đặc biệt,..).

IV. Binary Heap:

1. Application: BINARY HEAP của trang web <https://visualgo.net/en>:

Bước 1: Vào trang <https://visualgo.net/en>.

Bước 2: Để truy cập vào chức năng Hash Table, người dùng chọn mục được mô tả như hình bên dưới:



2. Environment:

Bài kiểm thử được thực hiện trên hai trình duyệt cụ thể là:

- Google chrome.
- Firefox.

3. Các thuật toán kiểm thử Black-Box Testing được sử dụng trong bài:

- Boundary value analysis technique.
- Equivalence class partitioning technique.
- Decision table technique.
- Use-case testing.

4. Non-function:

- *Test scenario 1: Khởi tạo Binary Heap từ 1 đến 18 phần tử*

Mục đích: Kiểm tra số lượng phần tử có thể khởi tạo cho Binary Heap

Quy trình thực hiện:

- Bước 1: Chọn Create(A) - $O(N \log N)$ hoặc Create(A) - $O(N)$
- Bước 2: Nhập số
- Bước 3: Nhấn Go

Các tình huống cần test:

- Nhập số phần tử trong khoảng (1, 18)
- Nhập một phần tử.
- Nhập 18 phần tử.
- Không nhập phần tử nào.
- Nhập lớn hơn 18 phần tử
- *Test scenario 2: Kiểm tra giới hạn giá trị của 1 phần tử khi khởi tạo Binary Heap*

Mục đích: Kiểm tra giới hạn giá trị của 1 phần tử khi khởi tạo Binary Heap

Quy trình thực hiện:

- Bước 1: Chọn Create(A) - $O(N \log N)$ hoặc Create(A) - $O(N)$
- Bước 2: Nhập số
- Bước 3: Nhấn Go

Các tình huống cần test:

- Nhập số phần tử trong khoảng (-100, 100)
- Nhập số phần tử ngoài khoảng (-100, 100)
- *Test scenario 3: Kiểm các giá trị giống nhau và giá trị đặc biệt khi khởi tạo Binary Heap*

Mục đích: Kiểm tra các giá trị hợp lệ khi khởi tạo Binary Heap

Quy trình thực hiện:

- Bước 1: Chọn Create(A) - $O(N \log N)$ hoặc Create(A) - $O(N)$
- Bước 2: Nhập số
- Bước 3: Nhấn Go

Các tình huống cần test:

- Nhập phần tử có ký tự đặc biệt

- Nhập phần tử là là số không nguyên
 - Nhập các phần tử có giá trị trùng nhau
- *Test scenario 4: Insert phần tử thứ 2 đến 18 trong Binary Heap*

Mục đích: Kiểm tra số lượng phần tử có thể thêm cho Binary Heap

Quy trình thực hiện:

- Bước 1: Chọn Insert(v)
- Bước 2: Nhập số
- Bước 3: Nhấn Go

Các tình huống cần test:

- Thêm phần tử thứ 2
 - Thêm phần tử thứ 3 đến 17
 - Thêm phần tử thứ 18
 - Thêm phần tử thứ 19
- *Test scenario 5: Kiểm các giá trị giống nhau và giá trị đặc biệt khi thêm giá trị cho Binary Heap*

Mục đích: Kiểm tra các giá trị hợp lệ khi thêm giá trị cho Binary Heap

Quy trình thực hiện:

- Bước 1: Chọn Insert(v)
- Bước 2: Nhập số
- Bước 3: Nhấn Go

Các tình huống cần test:

- Thêm phần tử có ký tự đặc biệt
 - Thêm phần tử là là số không nguyên
 - Thêm phần tử có giá trị đã có trong Binary Heap
- *Test scenario 6: Cập nhật các phần tử từ vị trí 1 đến vị trí cuối trong Binary Heap*

Mục đích: Kiểm tra các vị trí có thể cập nhật cho Binary Heap

Quy trình thực hiện:

- Bước 1: UpdateKey(i, newv)
- Bước 2: Nhập số

- Bước 3: Nhấn Go

Các tình huống cần test:

- Cập nhật phần tử ở vị trí 1
- Cập nhật phần tử ở vị trí 2 đến vị trí kế cuối
- Cập nhật phần tử ở vị trí cuối cùng
- Cập nhật phần tử ở vị trí không tồn tại
- *Test scenario 7: Cập nhật các giá trị từ 1 đến 99 cho Binary Heap*

Mục đích: Kiểm tra các giá trị có thể cập nhật cho Binary Heap

Quy trình thực hiện:

- Bước 1: UpdateKey(i, newv)
- Bước 2: Nhập số
- Bước 3: Nhấn Go

Các tình huống cần test:

- Cập nhật phần tử có giá trị 0
- Cập nhật phần tử có giá trị từ 1 đến 99
- Cập nhật phần tử có giá trị lớn hơn 99
- Cập nhật phần tử có giá trị rỗng
- *Test scenario 8: Kiểm các giá trị giống nhau và giá trị đặc biệt khi thêm giá trị cho Binary Heap*

Mục đích: Kiểm tra các giá trị hợp lệ khi cập nhật cho Binary Heap

Quy trình thực hiện:

- Bước 1: UpdateKey(i, newv)
- Bước 2: Nhập số
- Bước 3: Nhấn Go

Các tình huống cần test:

- Cập nhật phần tử có giá trị không phải số nguyên
- Cập nhật phần tử có giá trị đã tồn tại trong Binary Heap
- *Test scenario 9: Delete từ vị trí 1 đến vị trí cuối trong Binary Heap*

Mục đích: Kiểm tra các vị trí có thể xóa trong Binary Heap

Quy trình thực hiện:

- Bước 1: Delete(i)
- Bước 2: Nhập số
- Bước 3: Nhấn Go

Các tình huống cần test:

- Xóa phần tử ở vị trí 1
- Xóa phần tử ở vị trí 2 đến vị trí kế cuối
- Xóa phần tử ở vị trí cuối cùng
- Xóa phần tử ở vị trí nằm ngoài kích thước Binary Heap
- *Test scenario 10: Delete các vị trí đặc biệt trong Binary Heap*

Mục đích: Kiểm tra các vị trí hợp lệ có thể xóa trong Binary Heap

Quy trình thực hiện:

- Bước 1: Delete(i)
- Bước 2: Nhập số
- Bước 3: Nhấn Go

Các tình huống cần test:

- Xóa phần tử ở vị trí rỗng
- Xóa phần tử ở vị trí không phải số nguyên

5. Function:

- *Test scenario: Các chức năng Create, Insert, Update, Delete trong Binary Heap*

Mục đích: Kiểm tra các chức năng Create, Insert, Update, Delete trong Binary Heap

a) Create

Quy trình thực hiện:

Bước 1: Chọn Create.

Bước 2: Nhập số.

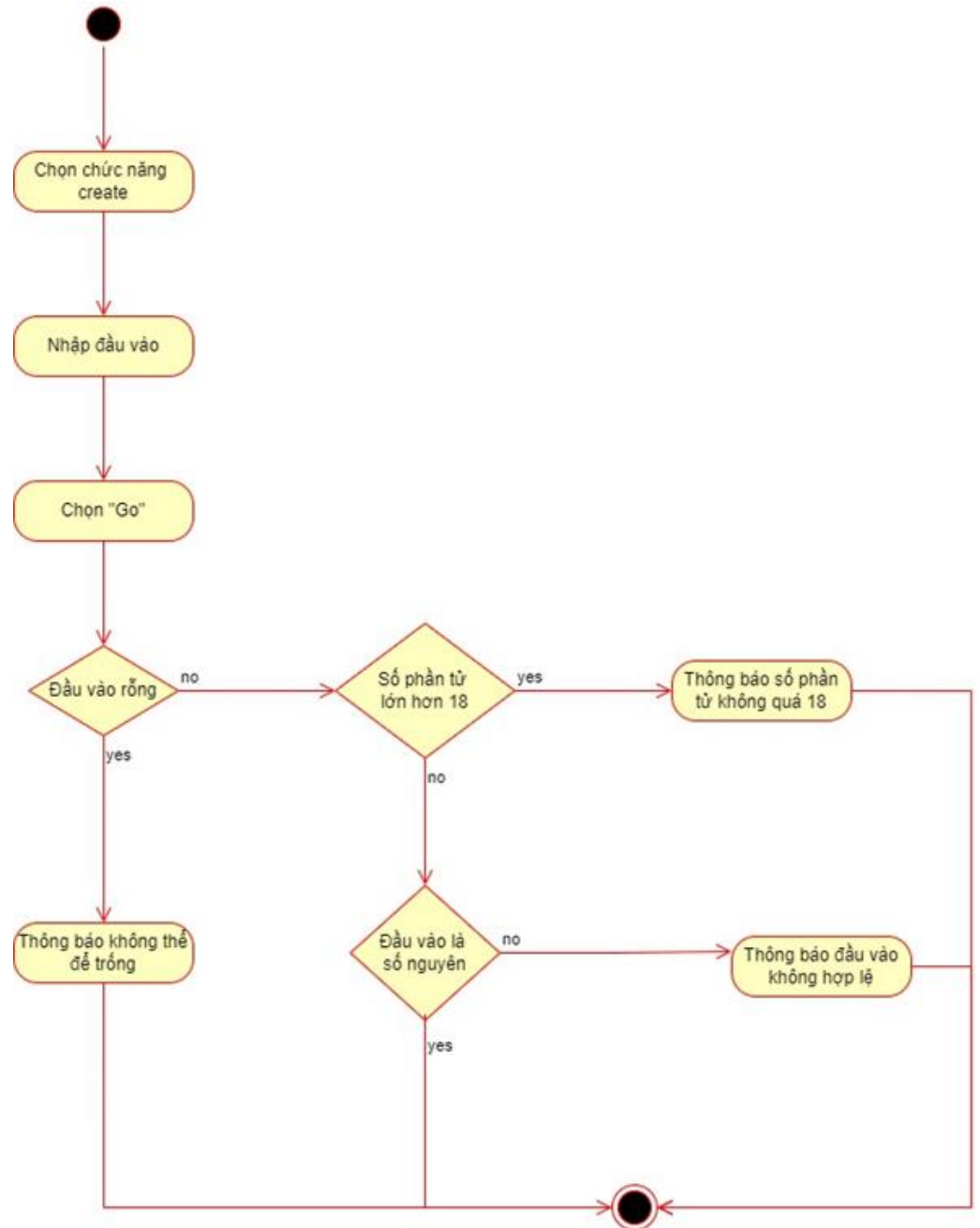
Bước 3: Chọn Go.

Đầu vào:

- Số trong đoạn [-100 .. 100].
- Số ngoài đoạn [-100 .. 100].
- Không nhập gì cả.
- Nhập vào các ký tự đặc biệt.
- Nhập vào số phần tử trong đoạn $1 \leq x \leq 18$.
- Nhập vào số phần tử lớn hơn 18.
- Nhập vào các số không nguyên.

- Nhập vào các số trùng nhau.

Activity Graph:



Decision table:

	Rule 1-4	Rule 5-6	Rule 7	Rule 8
<u>Condition:</u>				
Đầu vào rỗng	T	F	F	F
Đầu vào lớn hơn 18 phần tử	-	T	F	F

Đầu vào là các số nguyên	-	-	T	F
Action:				
Thực hiện create			X	
Thông báo lỗi	X	X		X

Các tình huống cần test dựa vào Decision table:

- Không nhập đầu vào.
- Nhập đầu vào có số phần tử lớn hơn 18.
- Nhập vào số phần tử trong đoạn [1, 18] và trong các số nhập vào có chứa ít nhất 1 phần tử không phải số nguyên.
- Nhập vào số phần tử trong đoạn [1, 18] và các số nhập vào đều là số nguyên.

b) Insert

Quy trình thực hiện:

Bước 1: Chọn Insert.

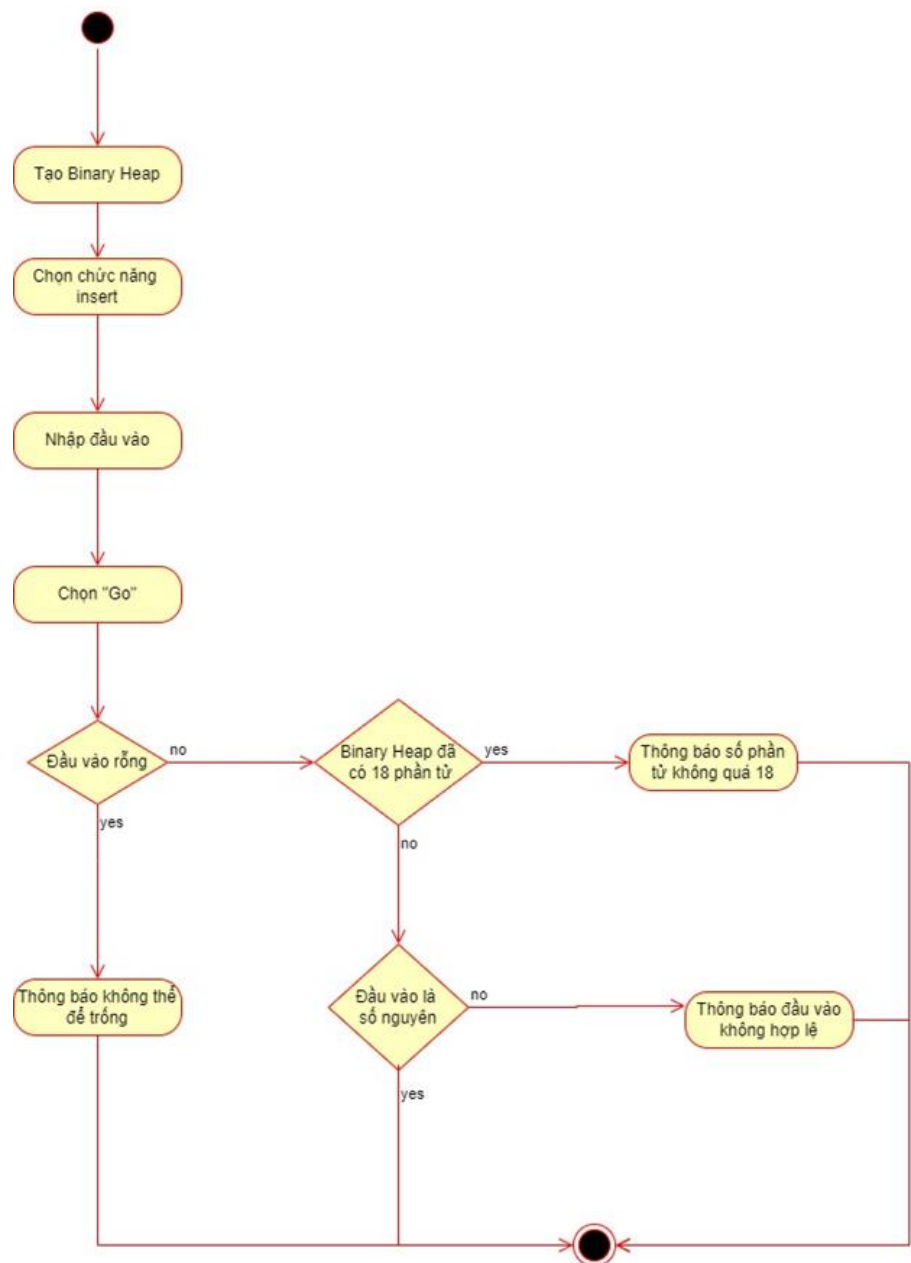
Bước 2: Nhập số.

Bước 3: Chọn Go.

Đầu vào:

- Số trong đoạn [-100 .. 100].
- Số ngoài đoạn [-100 .. 100].
- Không nhập gì cả.
- Nhập vào các ký tự đặc biệt.
- Chèn phần tử có vị trí trong đoạn $1 \leq x \leq 18$.
- Chèn phần tử có vị trí lớn hơn 18.
- Nhập vào các số không nguyên.
- Nhập vào số đã tồn tại.

Activity Graph:



Decision table:

	Rule 1-4	Rule 5-6	Rule 7	Rule 8
<u>Condition:</u>				
Đầu vào rỗng	T	F	F	F
Binary Heap đã có 18 phần tử	-	T	F	F
Đầu vào là số nguyên	-	-	T	F

Action:				
Thực hiện insert			X	
Thông báo lỗi	X	X		X

Các tình huống cần test dựa vào Decision table:

- Không nhập đầu vào.
- Insert vào Binary Heap có 18 phần tử.
- Insert vào Binary Heap có [1 .. 17] phần tử và đầu vào là số nguyên.
- Insert vào Binary Heap có [1 .. 17] phần tử và đầu vào không phải số nguyên.

c) Update

Quy trình thực hiện:

Bước 1: Chọn UpdateKey.

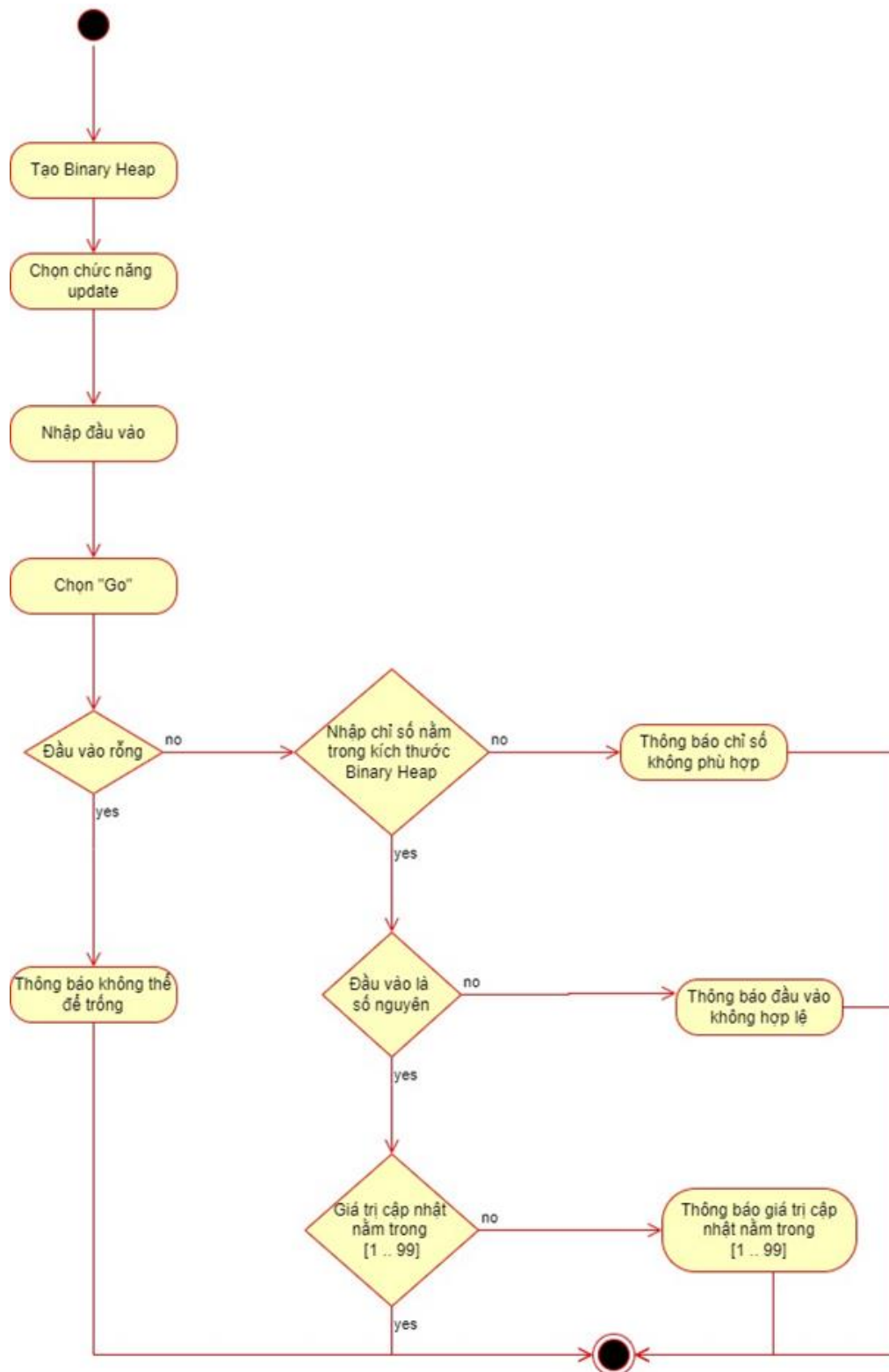
Bước 2: Nhập số.

Bước 3: Chọn Go.

Đầu vào:

- Số trong đoạn [1 .. 99].
- Số ngoài đoạn [1 .. 99].
- Không nhập gì cả.
- Cập nhật phần tử từ vị trí 1 đến vị trí cuối.
- Cập nhật phần tử có vị trí lớn hơn vị trí cuối.
- Nhập vào các số không nguyên.
- Nhập vào số đã tồn tại.

Activity Graph:



Decision table:

	Rule 1-8	Rule 9-12	Rule 13-14	Rule 15	Rule 16

<u>Condition:</u>					
Đầu vào rỗng	T	F	F	F	F
Đầu vào chỉ số nằm trong kích thước Binary Heap	-	F	T	T	T
Đầu vào là số nguyên	-	-	F	T	T
Đầu vào giá trị cập nhật nằm trong [1 .. 99]	-	-	-	T	F
<u>Action:</u>					
Thực hiện update				X	
Thông báo lỗi	X	X	X		X

Các tình huống cần test dựa vào Decision table:

- Không nhập đầu vào.
- Nhập chỉ số đầu vào lớn hơn kích thước Binary Heap.
- Nhập chỉ số đầu vào không lớn hơn kích thước Binary Heap và đầu vào không phải số nguyên.
- Nhập chỉ số đầu vào không lớn hơn kích thước Binary Heap và đầu vào là số nguyên nằm trong khoảng [1 .. 99].
- Nhập chỉ số đầu vào không lớn hơn kích thước Binary Heap và đầu vào là số nguyên nằm ngoài khoảng [1 .. 99].

d)Delete

Quy trình thực hiện:

Bước 1: Chọn Delete.

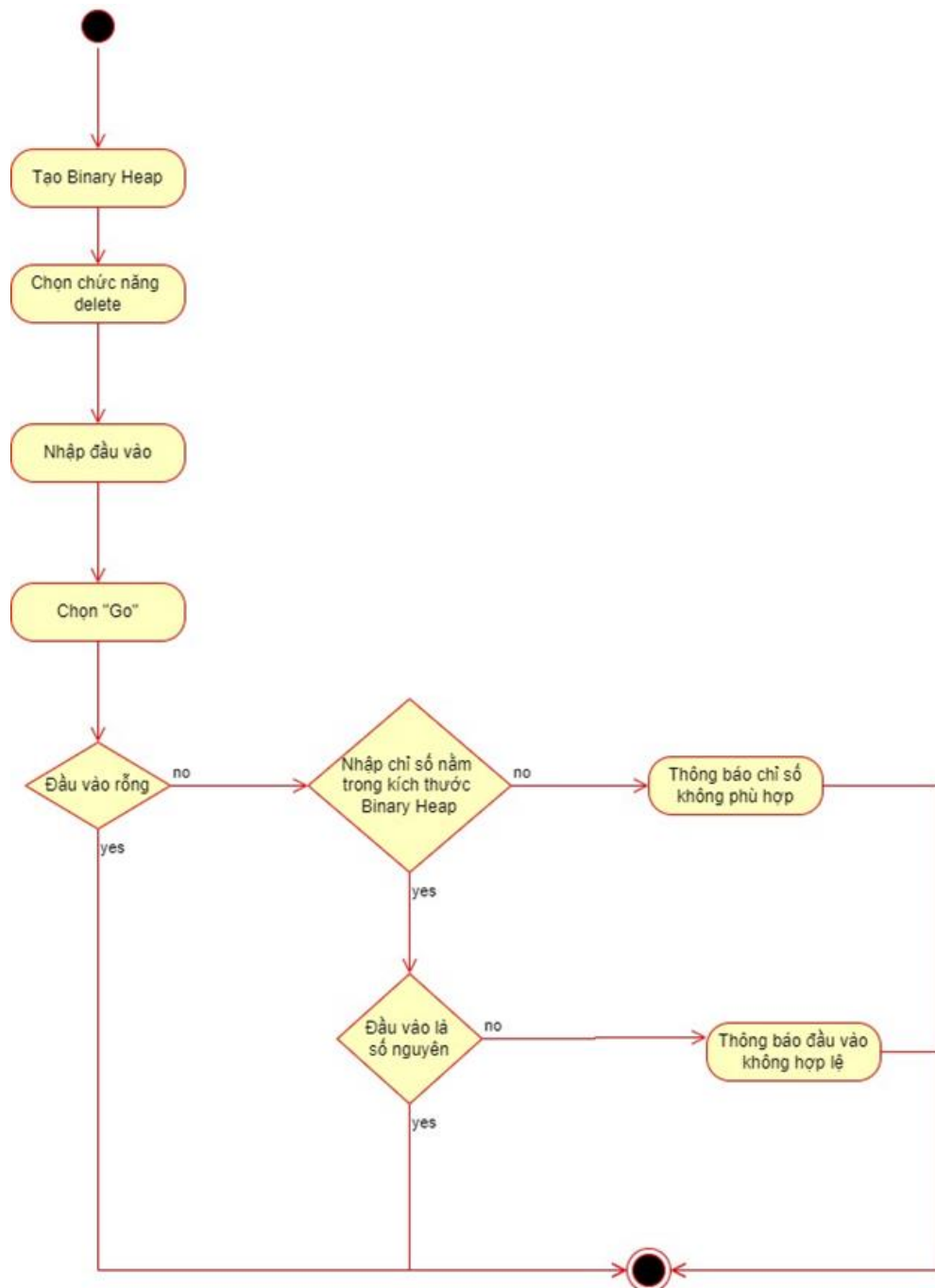
Bước 2: Nhập số.

Bước 3: Chọn Go.

Đầu vào:

- Không nhập gì cả.
- Xóa phần tử từ vị trí 1 đến vị trí cuối.
- Xóa phần tử có vị trí lớn hơn vị trí cuối.
- Nhập vào các số không nguyên.

Activity Graph:



Decision table:

	Rule 1-4	Rule 5-6	Rule 7	Rule 8
<u>Condition:</u>				
Đầu vào rỗng	T	F	F	F
Đầu vào chỉ số nằm trong kích thước Binary Heap	-	F	T	T

Đầu vào là số nguyên	-	-	T	F
Action:				
Thực hiện delete			X	
Thông báo lỗi	X	X		X

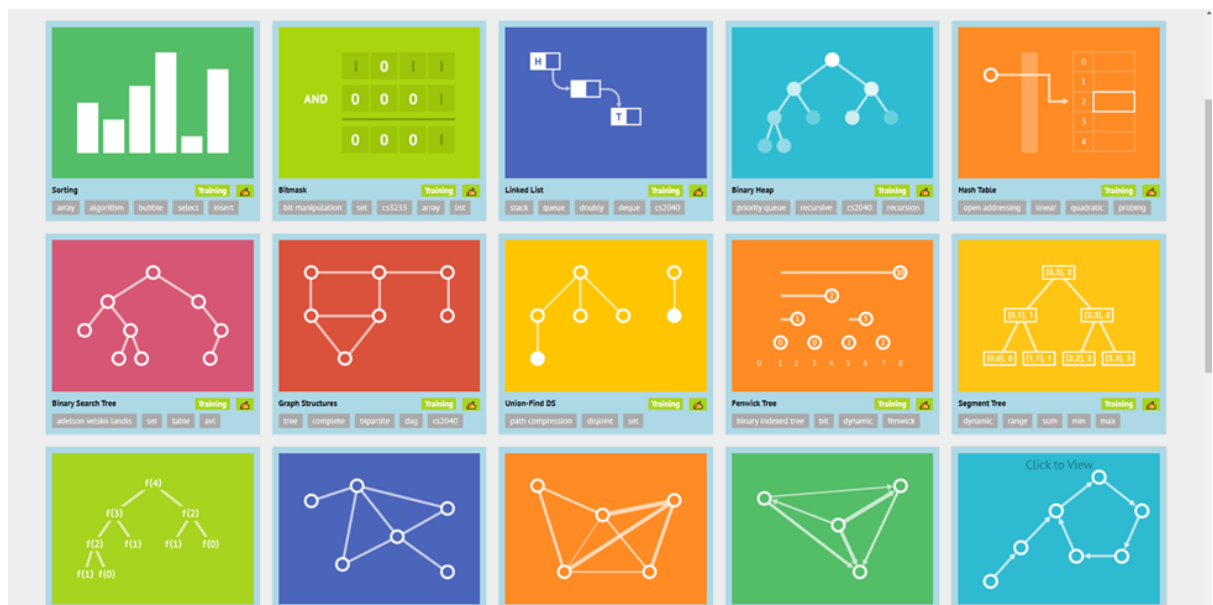
Các tình huống cần test dựa vào Decision table:

- Không nhập đầu vào.
- Nhập chỉ số đầu vào lớn hơn kích thước Binary Heap.
- Nhập chỉ số đầu vào không lớn hơn kích thước Binary Heap và đầu vào là số nguyên.
- Nhập chỉ số đầu vào không lớn hơn kích thước Binary Heap và đầu vào không phải số nguyên.

V. Binary Search Tree:

Bước 1: Vào trang <https://visualgo.net/en>.

Bước 2: Chọn Binary Search Tree (như hình bên dưới).



1. ENVIRONMENT

Bài kiểm thử được thực hiện trên hai trình duyệt cụ thể là:

- Google chrome.

- Microsoft Edge.

2. CÁC THUẬT TOÁN KIỂM THỬ ĐƯỢC ÁP DỤNG

- Boundary value analysis technique.
- Equivalence class partitioning technique.
- Use-case testing.
- Decision table technique.

3. NON-FUNCTIONAL REQUIREMENTS

• *Test scenario 1*

Các giá trị nhập vào thỏa mãn x sao cho $1 \leq x \leq 100$

Mục đích: Kiểm tra vùng giá trị có thể nhập vào để thực hiện Binary Search Tree.

Quy trình thực hiện:

- Bước 1: Nhấn vào Insert
- Bước 2: Nhập số
- Bước 3: Chọn Go

Các tình huống cần test:

- Nhập số trong (1, 100).
- Nhập số 1.
- Nhập số 100.
- Nhập số < 1 .
- Nhập số > 100
- Nhập số 50
- Nhập số 2
- Nhập số 99

• *Test scenario 2:*

Giá trị nhập không được trùng với giá trị đã có trong cây nhị phân.

Mục đích: Kiểm tra giá trị thêm vào

Quy trình thực hiện:

- Bước 1: Nhấn vào Insert

- Bước 2: Nhập số
- Bước 3: Chọn Go

Các tình huống cần test:

- Số đã có trong cây nhị phân
- Số chưa có trong cây nhị phân

4. FUNCTIONAL REQUIREMENTS

- *Test scenario: Chức năng Insert của cây BST*

Mục đích: Kiểm tra các chức năng Insert trên cây BST của VisualGo.

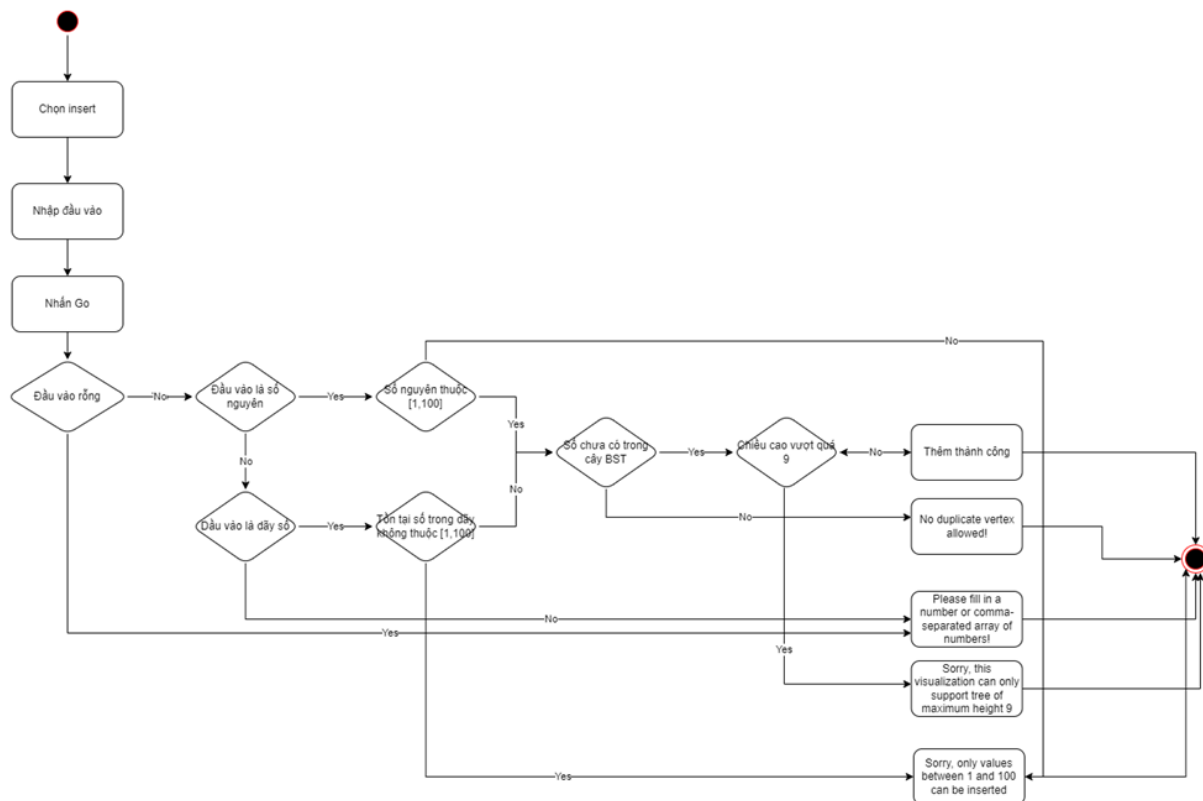
Quy trình thực hiện (Insert):

- Bước 1: Chọn Insert
- Bước 2: Nhập số.
- Bước 3: Chọn Go

Các tình huống cần test:

- Số x sao cho $1 \leq x \leq 100$
- Số nhỏ hơn 1.
- Số lớn hơn 100.
- Nhập dãy số trong khoảng $[1,100]$
- Không nhập gì cả.
- Nhập vào nhưng không phải số (có thể là các kí tự nào đó).

Activity Graph:



Hình 1. Sơ đồ hoạt động Insert

Decision table:

	Rule 1-64	Rule 65 - 68	Rule 69 - 72	Rule 75- 82	Rule 83- 98	Rule 99- 114	Rule 115- 122	Rule 123- 126	Rule 127	Rule 128
Condition:										
Đầu vào rỗng	T	F	F	F	F	F	F	F	F	F
Đầu vào là số nguyên	-	T	T	T	T	F	F	F	F	F
Đầu vào là dãy số	-	-	-	-	-	F	T	T	T	T

Tồn tại số trong dãy không thuộc [1,100]	-	-	-	-	-	-	T	F	F	F
Số chưa có trong cây BST	-	T	T	T	F	-	-	F	T	T
Số nguyên thuộc [1,100]	-	T	T	F	-	-	-	-	-	-
Chiều cao vượt quá 9	-	T	F	-	-	-	-	-	F	T
Action:										
Thực hiện insert			X						X	
Hiển thị lỗi	X	X		X	X	X	X	X		X

Các tình huống cần test dựa vào Decision table:

- Đầu vào rỗng
- Đầu vào là số nguyên thuộc [1,100] , số chưa có trong BST và chiều cao của cây không vượt quá 9
- Đầu vào là số nguyên không thuộc [1,100] , số chưa có trong BST
- Đầu vào là số nguyên đã có có trong BST
- Đầu vào là ký tự đặc biệt
- Đầu vào là dãy số có số không thuộc [1,100]
- Đầu vào là dãy số thuộc [1,100] đã có trong cây BST

- Đầu vào là dãy số thuộc $[1,100]$ chưa có trong cây BST, chiều cao cây không quá 9
- Đầu vào là dãy số thuộc $[1,100]$ chưa có trong cây BST, chiều cao cây quá 9

VI. Hash Table:

1. Application: Hash Table của trang web <https://visualgo.net/en>:

Bước 1: Vào trang <https://visualgo.net/en>.

Bước 2: Để truy cập vào chức năng Hash Table, người dùng chọn mục được mô tả như hình bên dưới:



2. Environment:

Bài kiểm thử được thực hiện trên hai trình duyệt cụ thể là:

- Google chrome.
- Firefox.

3. Các thuật toán kiểm thử Black-Box Testing được sử dụng trong bài:

- Boundary value analysis technique.
- Equivalence class partitioning technique.

4. Non-function:

- *Test scenario 1: Kiểm tra khả năng khởi tạo Hash Table*

Mục đích: Kiểm tra số lượng phần tử có thể khởi tạo cho Hash Table

Quy trình thực hiện:

- Bước 1: Chọn Create
- Bước 2: Nhập số
- Bước 3: Nhấn Go

Các tình huống cần test:

- Nhập số phần tử trong khoảng (5, 19)
- Nhập kích thước là 5
- Nhập kích thước là 19
- Không nhập gì hết
- Nhập lớn hơn 19
- Nhập bé hơn 5

● *Test scenario 2: Kiểm tra khả năng nhập vào của chức năng Insert*

Mục đích: Kiểm tra khả năng thêm giá trị vào hash table của trang web.

Tiền điều kiện: Hash table đã được khởi tạo.

Quy trình thực hiện:

- Bước 1: Chọn Insert
- Bước 2: Nhập số
- Bước 3: Nhấn Go

Các tình huống cần test:

- Nhập số trong khoảng [0, 99]
- Số bé hơn 0
- Số lớn hơn 99
- Không nhập gì cả
- Chuỗi bắt đầu bởi số
- Chuỗi không bắt đầu bởi số

● *Test scenario 3: Kiểm tra khả năng nhập vào của chức năng Search của Hash Table*

Mục đích: Kiểm tra khả năng Search của Hash Table

Tiền điều kiện: Hash table đã được khởi tạo.

Quy trình thực hiện:

- Bước 1: Chọn Search

- Bước 2: Nhập số
- Bước 3: Nhấn Go

Các tình huống cần test:

- Nhập số trong đoạn [0, 99]
- Số bé hơn 0
- Số lớn hơn 99
- Không nhập gì cả
- Chuỗi bắt đầu bởi số
- Chuỗi không bắt đầu bởi số

- *Test scenario 4: Kiểm tra khả năng nhập vào của chức năng Remove của Hash Table*

Mục đích: Kiểm tra khả năng Remove của Hash Table

Tiền điều kiện: Hash table đã được khởi tạo.

Quy trình thực hiện:

- Bước 1: Chọn Remove
- Bước 2: Nhập số
- Bước 3: Nhấn Go

Các tình huống cần test:

- Nhập số trong đoạn [0, 99]
- Số bé hơn 0
- Số lớn hơn 99
- Không nhập gì cả
- Chuỗi bắt đầu bởi số
- Chuỗi không bắt đầu bởi số

5. Function:

- *Test scenario: Chức năng Insert trong Binary Heap*

Mục đích: Kiểm tra các chức năng Insert trong Binary Heap

Quy trình thực hiện:

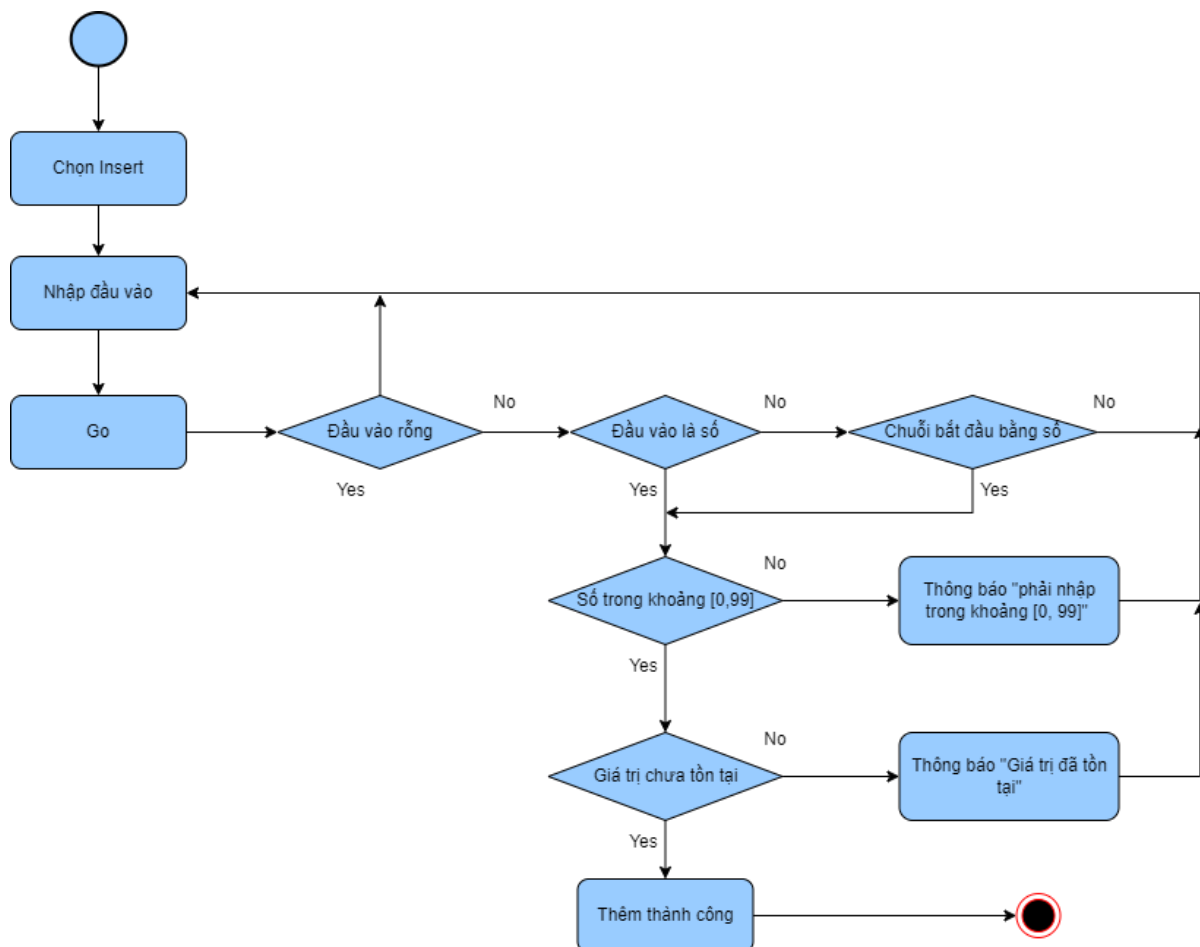
- Bước 1: Chọn Insert.
- Bước 2: Nhập số.
- Bước 3: Chọn Go.

Đầu vào:

- Nhập số trong khoảng [0, 99]

- Số bé hơn 0
- Số lớn hơn 99
- Không nhập gì cả
- Chuỗi bắt đầu bởi số
- Chuỗi không bắt đầu bởi số

• *Activity Graph:*



• *Decision table:*

	Rule 1-8	Rule 9-12	Rule 13- 14	Rule 15	Rule 16
Condition:					
Đầu vào rỗng	T	F	F	F	F
Đầu vào là số hoặc là chuỗi bắt đầu bằng số	-	F	T	T	T
Đầu vào là số trong khoảng [0,99]	-	-	F	T	T
Đầu vào là số chưa tồn tại	-	-	-	F	T
Action:					
Thực hiện Insert					X
Thông báo lỗi	X	X	X	X	