**Black Box Testing**

* Scenario: Registering an account
* Input : Telephone

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case | Valid Input | Invalid Input | Expected Output | Actual Output |
| 1 | 2 alphabets (front & back) & 7 digits | 2 alphabets & 6 digits | Error | Error |
| 2 | 2 alphabets (front & back) & 7 digits | 2 alphabets & 8 digits | Error | Error |
| 3 | 2 alphabets (front & back) & 7 digits | 8 digits & 1 alphabets | Error | Error |
| 4 | 2 alphabets (front & back) & 7 digits | Empty | Error | Error |

* Scenario: Register an account
* Input : NRIC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case | Valid Input | Invalid Input | Expected Output | Actual Output |
| 1 | 8 digits | 7 digits | Error | Error |
| 2 | 8 digits | 9 digits | Error | Error |
| 3 | 8 digits | 7 digits & 1 non-digit-character | Error | Error |
| 4 | 8 digits | Empty | Error | Error |

**White Box Testing**

**Booking Appointment**

public static function build($type) {

$appointment = "\App\\" . $type . "Appointment";

if (class\_exists($appointment)) {

return new $appointment();

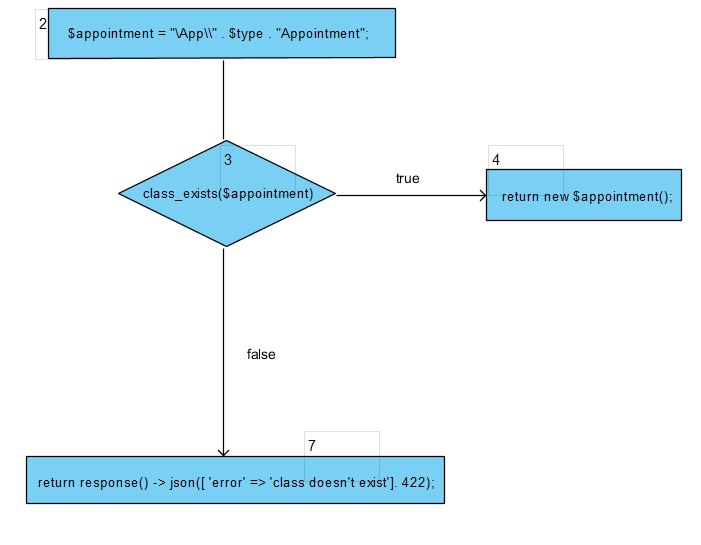
}

else {

return response()->json(['error' => 'class doesn’t exist’], 422);

}

}

****

* Basis path:

i.) 2 -> 3 -> 4

ii.) 2 -> 3 -> 7

* Test cases:

i.) $type will set to ‘’CHAS clinic/dental/hospital/polyclinic’’ when user select it, class will be created based on the type and we will create an appointment for these classes.

ii.) $type will set to ‘’CHAS clinic/dental/hospital/polyclinic’’ when user select it, however if user intercept and modify the data (e.g from “CHAS clinic” to “Private Clinic”), it will return ‘’class doesn’t exist’’ since there is no such class.

**Authenticate User**

if(!currentUser) { // if user not logged in

if (requiresAdmin || requiresAuth){ // the action requires Admin/Auth

next('/login'); // direct to login

} else {

next(); // proceed with the current request

}

} else { // if user is logged in

if (requiresAdmin){ // if the action requires Admin

if (store.getters.isAdmin) // if current account is Admin

next(); // procced with the current request

else

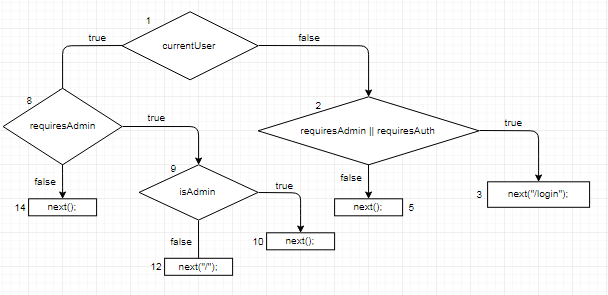
next('/') // direct to homepage

} else {

next();

}

}



* Basis path:

i.) 1 -> 2 -> 3

ii.) 1 -> 2 -> 5

iii.) 1 -> 8 -> 9 -> 10

iv.) 1 -> 8 -> 9 -> 12

v.) 1 -> 8 -> 14

* Test cases:

i.) If user is not logged in and action requires admin/auth, it will direct user to login.

ii.) If user is not logged in and the action do not requires admin/auth, the program will proceed with the current request.

iii.) If user is logged in and user is an admin and action requires admin, program will proceed with the current request.

iv.) If user is logged in and user is an admin, however action do not require admin, program will direct user to homepage.

v.) If user is logged in, and action do not requires admin then program will proceed with current request.