

# **Test Plan**

## **Rent My Ride**

presented by

### **Group : CPCJAI**

Theerapat Boonrod	6630149621	Manual Test
Theerach Sae-Ngow	6630154721	Manual Test
Phaolap Kulteera	6630199021	Playwright
Chawanrat Thoop-On	6632043021	Manual Test
Tinnapat Sittisawan	6632088321	Manual Test
Pongpitchaya Jaidee	6632141921	Manual Test
Peeradon Thienpongkasem	6632154021	Playwright/Jest
Lattachot Cumpeerosakul	6632195821	Manual Test
Weeraphat Lerdcharoenvorakul	6632209021	Manual Test
Aekkorn Panbangna	6632263821	Manual Test
Phakin Wattanayakorn	6430302021	Jest

2110423 Software Engineering

22 October 2025

Department of Computer Engineering Faculty of Engineering

## Black-box testing using PlayWright Framework

### **EPIC 2 : Authenticated System**

#### **User Story 2-1 (Sign-up)**

- Input Equivalence Class

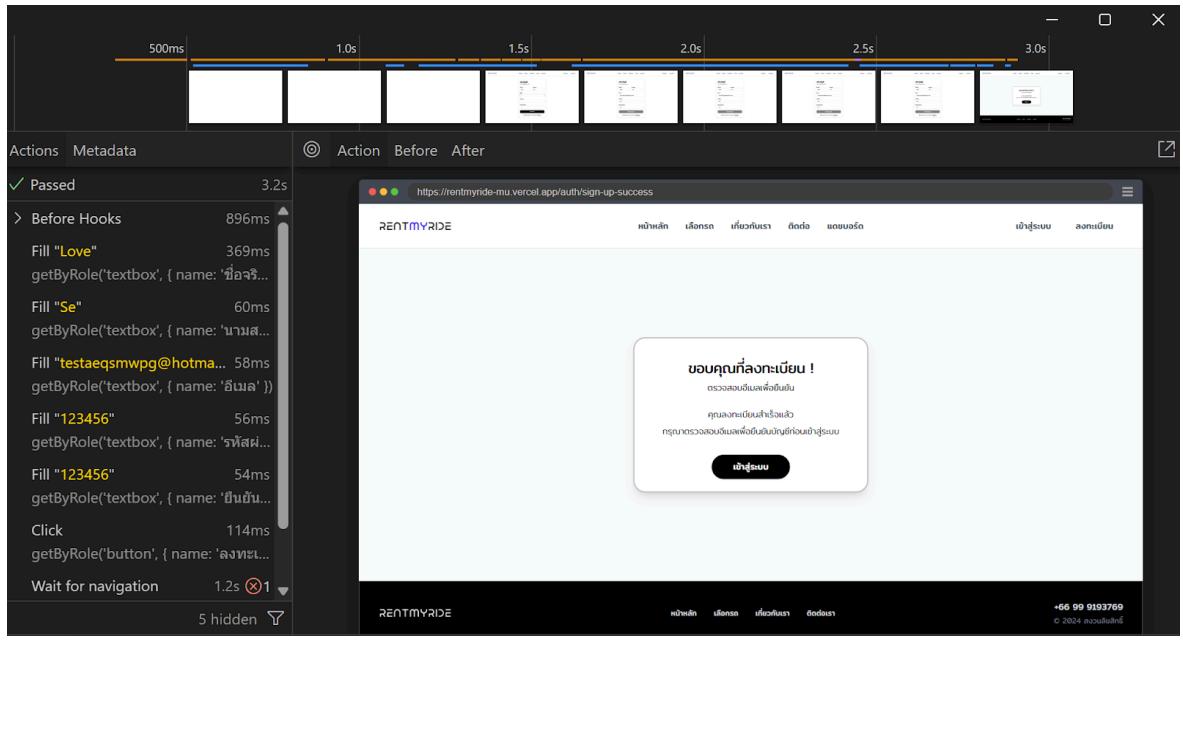
Equivalent Class ID	Input	Validation
1	Name, Surname, Email, Password, ConfirmPassword is correct format	valid
2	Name, Surname, Email, Password, ConfirmPassword is empty	invalid
3	Email without @	invalid
4	Email without . (dot)	invalid
5	Email with special characters that violate the rules ( ) , : ; < > [ ] space \ " /	invalid
6	Email with . more than one	invalid
7	Email with @ more than one	invalid
8	Duplicate email in the system	invalid
9	Password length less than 6	invalid
10	Confirm password does not match password	invalid

- Testcase

Pre condition : Email is not yet in the system.

Test Case ID	TC 1-1
Input	ชื่อจริง : Love นามสกุล : SE อีเมล : <a href="mailto:ILoveSe@gmail.com">ILoveSe@gmail.com</a> รหัสผ่าน : 123456 ยืนยันรหัสผ่าน : 123456
Expected Output	- go to sign-up-success page
Actual Output	- go to sign-up-success page
Equivalence Classes Covered	1

Playwright Result



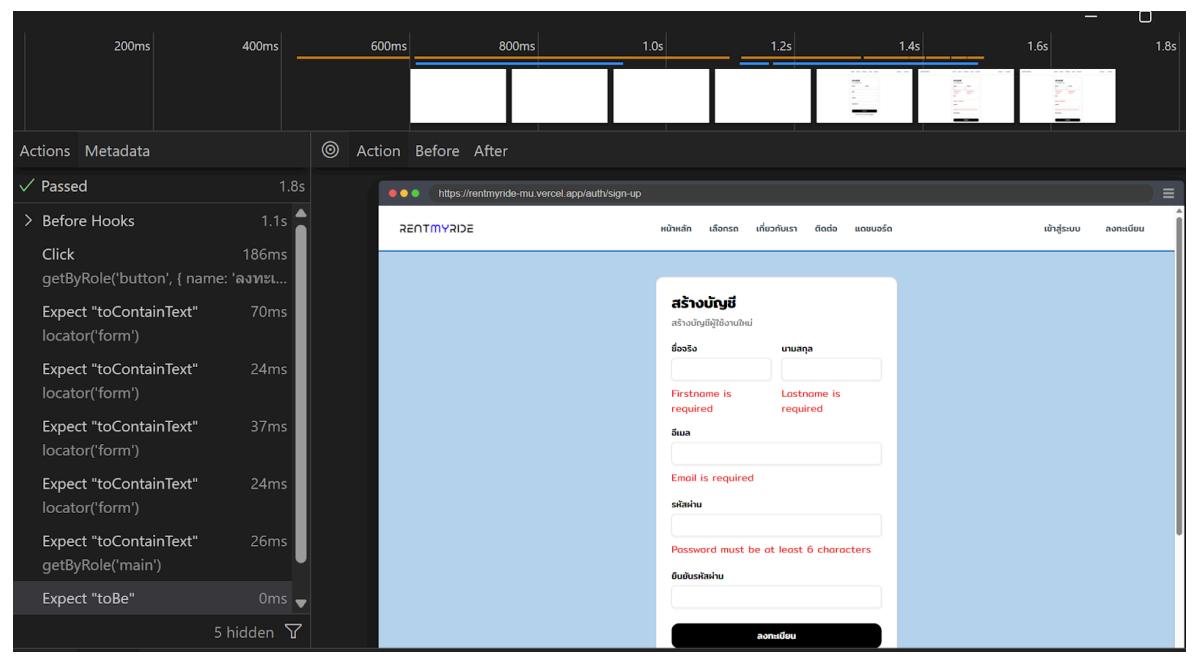
The screenshot shows the Playwright UI test results interface. At the top, there is a timeline from 0ms to 3.0s with screenshots at each second. Below the timeline is a table of actions:

Actions	Metadata	Action	Before	After
✓ Passed	3.2s			
> Before Hooks	896ms			
Fill "Love"	369ms	getByRole('textbox', { name: 'ชื่อจริ...' })		
Fill "Se"	60ms	getByRole('textbox', { name: 'นามส...' })		
Fill "testaeqsmwpg@hotmail.com"	58ms	getByRole('textbox', { name: 'อีเมล' })		
Fill "123456"	56ms	getByRole('textbox', { name: 'รหัสผ...' })		
Fill "123456"	54ms	getByRole('textbox', { name: 'ยืนยัน...' })		
Click	114ms	getByRole('button', { name: 'ตกลง...' })		
Wait for navigation	1.2s <span style="color: red;">⊗1</span>			
5 hidden				

The browser screenshot shows a success message: "ขอบคุณที่ลงทะเบียน!" (Thank you for registering!) with a "เข้าสู่ระบบ" (Log in) button. The URL in the address bar is https://rentmynde-mu.vercel.app/auth/sign-up-success.

Test Case ID	TC 1-2
Input	<p>ชื่อจริง : &lt;empty&gt;          นามสกุล : &lt;empty&gt;          อีเมล : &lt;empty&gt;          รหัสผ่าน : &lt;empty&gt;          ยืนยันรหัสผ่าน : &lt;empty&gt;</p>
Expected Output	<ul style="list-style-type: none"> <li>- Still on the signup page</li> <li>- Show text error under the field (when click “ลงทะเบียน” button)</li> </ul>
Actual Output	<ul style="list-style-type: none"> <li>- Still on the signup page</li> <li>- Show text error under the field (when click “ลงทะเบียน” button)</li> </ul>
Equivalence Classes Covered	2

### Playwright Result



The screenshot displays the Playwright test results for a signup page. At the top, a timeline shows the execution flow from 200ms to 1.8s. Below the timeline, a table lists the actions taken and their execution times:

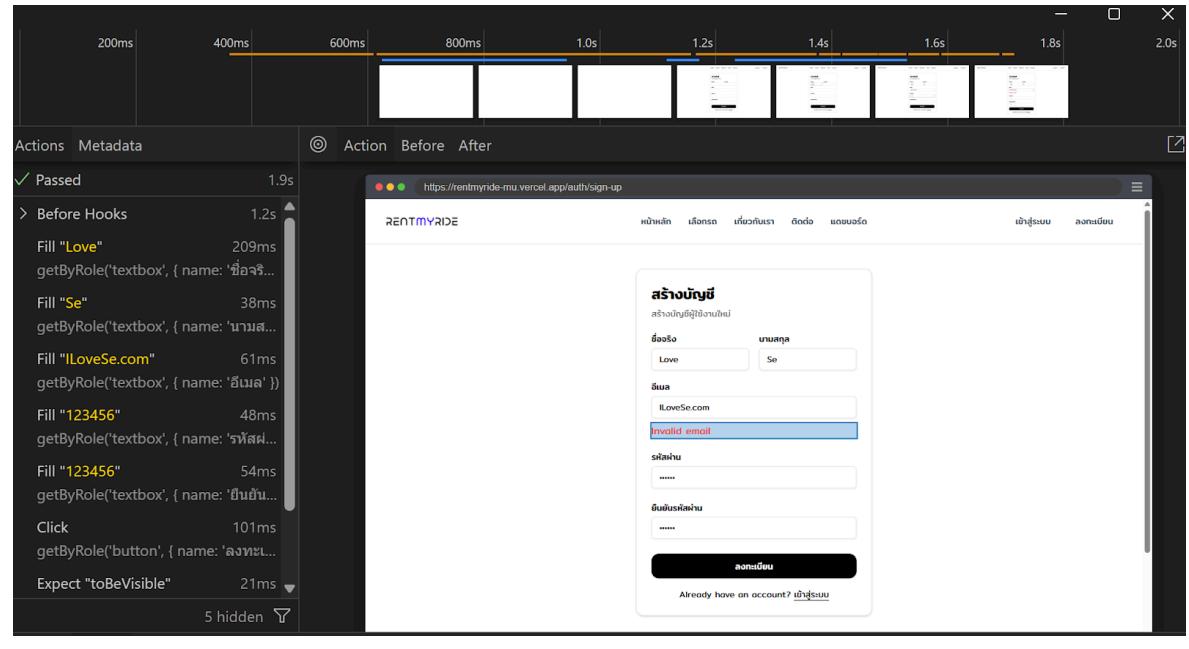
Actions	Metadata	Time
Passed		1.8s
> Before Hooks		1.1s
Click	getByRole('button', { name: 'ลงทะเบียน' })	186ms
Expect "toContainText"	locator('form')	70ms
Expect "toContainText"	locator('form')	24ms
Expect "toContainText"	locator('form')	37ms
Expect "toContainText"	locator('form')	24ms
Expect "toContainText"	getByRole('main')	26ms
Expect "toBe"		0ms

Below the table, a screenshot of the application shows the signup form with validation errors: "Firstname is required", "Lastname is required", "Email is required", and "Password must be at least 6 characters".

Pre condition : All other information except email is correct.

Test Case ID	TC 1-3
Input	ชื่อจริง : Love นามสกุล : SE อีเมล : ILoveSe.com รหัสผ่าน : 123456 ยืนยันรหัสผ่าน : 123456
Expected Output	<ul style="list-style-type: none"> <li>- Still on the signup page</li> <li>- Show text error “Invalid email” under the email field</li> </ul>
Actual Output	<ul style="list-style-type: none"> <li>- Still on the signup page</li> <li>- Show text error “Invalid email” under the email field</li> </ul>
Equivalence Classes Covered	3

### Playwright Result

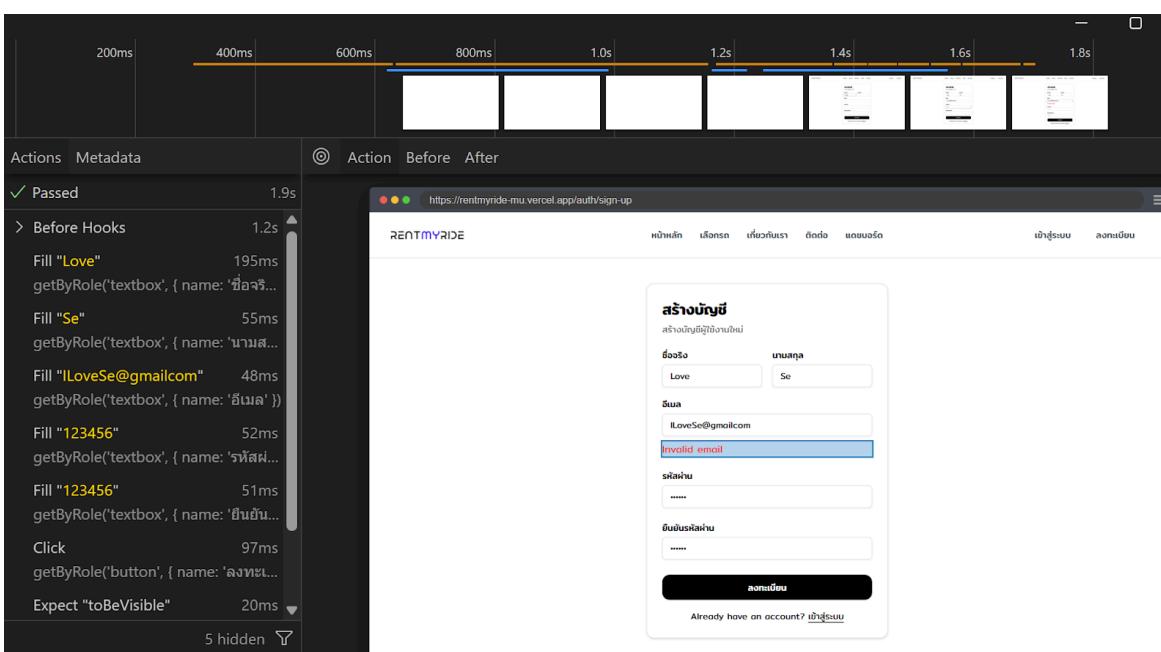


The screenshot displays the Playwright UI test results interface. At the top, a timeline shows the sequence of actions from 200ms to 2.0s. Below the timeline, a table lists the actions taken:

Actions	Metadata	Time
✓ Passed		1.9s
> Before Hooks		1.2s
Fill "Love"	getByRole('textbox', { name: 'ชื่อจริ...' })	209ms
Fill "Se"	getByRole('textbox', { name: 'นามสกุ...' })	38ms
Fill "ILoveSe.com"	getByRole('textbox', { name: 'อีเมล...' })	61ms
Fill "123456"	getByRole('textbox', { name: 'รหัสผ่าน...' })	48ms
Fill "123456"	getByRole('button')	54ms
Click		101ms
Expect "toBeVisible"		21ms
5 hidden		

The browser screenshot shows the 'RENTMYRIDE' sign-up page. The user has entered 'Love' in the first name field, 'Se' in the last name field, and 'ILoveSe.com' in the email field. An 'invalid\_email' error message is displayed below the email input field. The password fields are empty. A 'remember me' checkbox is checked. At the bottom, there is a link for existing users.

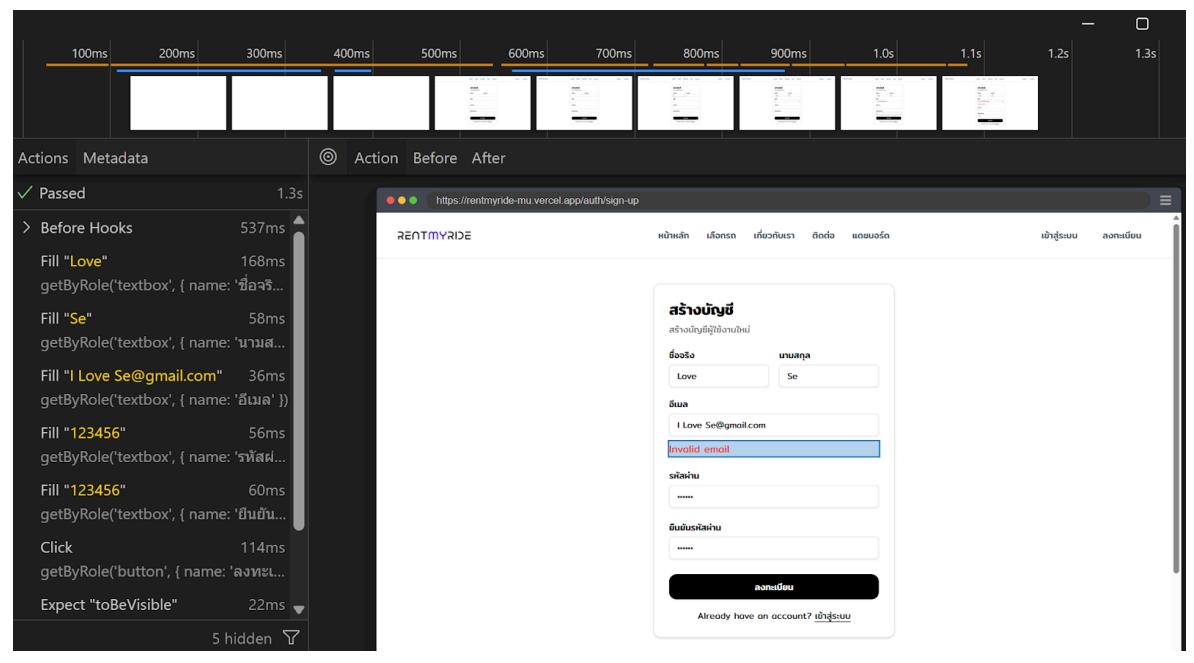
Pre condition : All other information except email is correct.

Test Case ID	TC 1-4
Input	ชื่อจริง : Love นามสกุล : SE อีเมล : ILoveSe@gmailcom รหัสผ่าน : 123456 ยืนยันรหัสผ่าน : 123456
Expected Output	<ul style="list-style-type: none"> <li>- Still on the signup page</li> <li>- Show text error “Invalid email” under the email field</li> </ul>
Actual Output	<ul style="list-style-type: none"> <li>- Still on the signup page</li> <li>- Show text error “Invalid email” under the email field</li> </ul>
Equivalence Classes Covered	4
Playwright Result	
 <p>The screenshot shows the RentMyRide sign-up page. The user has entered "ILoveSe@gmailcom" into the email field. Below the email field, the text "Invalid email" is displayed in red, indicating an error. The rest of the form fields (name, phone number, password, and confirmation) appear to be empty or have placeholder text.</p>	

Pre condition : All other information except email is correct.

Test Case ID	TC 1-5
Input	ชื่อจริง : Love นามสกุล : SE อีเมล : I Love Se@gmail.com รหัสผ่าน : 123456 ยืนยันรหัสผ่าน : 123456
Expected Output	<ul style="list-style-type: none"> <li>- Still on the signup page</li> <li>- Show text error “Invalid email” under the email field</li> </ul>
Actual Output	<ul style="list-style-type: none"> <li>- Still on the signup page</li> <li>- Show text error “Invalid email” under the email field</li> </ul>
Equivalence Classes Covered	5

### Playwright Result



The screenshot displays the Playwright test results for the sign-up process. The timeline at the top shows the sequence of actions over 1.3 seconds. The browser window on the right shows the 'RENTMYRIDE' sign-up page with the following details:

- ชื่อจริง: Love
- นามสกุล: SE
- อีเมล: I Love Se@gmail.com (highlighted in blue with an 'invalid\_email' error message)
- รหัสผ่าน: 123456
- ยืนยันรหัสผ่าน: 123456

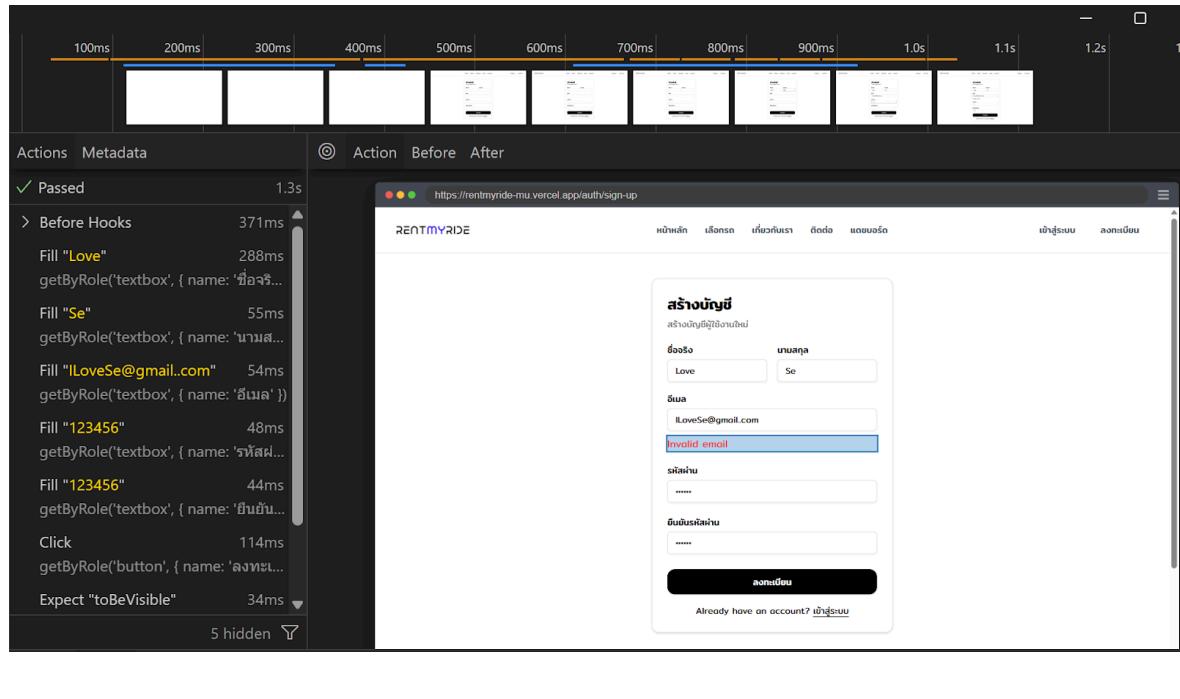
The test log on the left lists the actions taken:

- Before Hooks: 537ms
- Fill "Love": 168ms
- Fill "Se": 58ms
- Fill "I Love Se@gmail.com": 36ms
- Fill "123456": 56ms
- Fill "123456": 60ms
- Click: 114ms
- Expect "toBeVisible": 22ms

Pre condition : All other information except email is correct.

Test Case ID	TC 1-6
Input	ชื่อจริง : Love นามสกุล : SE อีเมล : ILoveSe@gmail..com รหัสผ่าน : 123456 ยืนยันรหัสผ่าน : 123456
Expected Output	<ul style="list-style-type: none"> <li>- Still on the signup page</li> <li>- Show text error “Invalid email” under the email field</li> </ul>
Actual Output	<ul style="list-style-type: none"> <li>- Still on the signup page</li> <li>- Show text error “Invalid email” under the email field</li> </ul>
Equivalence Classes Covered	6

### Playwright Result



The screenshot displays the Playwright UI test results for the signup page. On the left, a timeline shows the sequence of actions from 100ms to 1.2s. The actions listed in the Actions panel include:

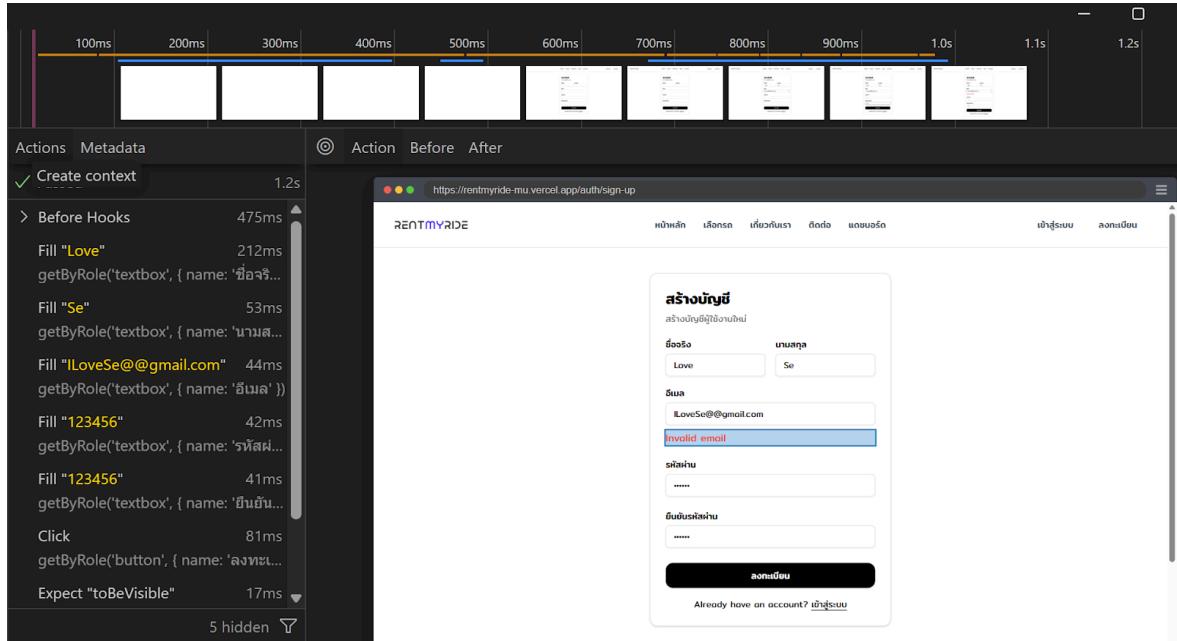
- Before Hooks: 371ms
- Fill "Love": 288ms
- Fill "Se": 55ms
- Fill "ILoveSe@gmail.com": 54ms
- Fill "123456": 48ms
- Click: 114ms
- Expect "toBeVisible": 34ms

The Metadata panel shows the total duration of 1.3s. The screenshot on the right shows the RentMyRide sign-up page. The user input fields for name, surname, email, and password have been filled. The email field contains "ILoveSe@gmail.com" and has a red border, indicating it is invalid. A text error "Invalid email" is displayed below the email input field. The password field contains "123456". The "Sign Up" button is at the bottom of the form.

Pre condition : All other information except email is correct.

Test Case ID	TC 1-7
Input	ชื่อจริง : Love นามสกุล : SE อีเมล : ILoveSe@@gmail.com รหัสผ่าน : 123456 ยืนยันรหัสผ่าน : 123456
Expected Output	<ul style="list-style-type: none"> <li>- Still on the signup page</li> <li>- Show text error “Invalid email” under the email field</li> </ul>
Actual Output	<ul style="list-style-type: none"> <li>- Still on the signup page</li> <li>- Show text error “Invalid email” under the email field</li> </ul>
Equivalence Classes Covered	7

**Playwright Result**



The screenshot shows the Playwright UI test interface. At the top, there's a timeline from 100ms to 1.2s. Below it, a table lists actions with their duration and a screenshot of the browser. The browser window displays a sign-up form for 'RENTMYRIDE' with fields for name, surname, email, and password. The email field contains 'ILoveSe@gmail.com' with a red underline and the text 'invalid\_email' below it. The password field contains '123456'. A 'Sign Up' button is at the bottom. The browser title bar shows the URL: https://rentmyride-mu.vercel.app/auth/sign-up.

Actions	Metadata	Action	Before	After
✓ Create context	1.2s			
> Before Hooks	475ms			
Fill "Love"	212ms	getByRole('textbox', { name: 'ชื่อจริ...' })		
Fill "Se"	53ms	getByRole('textbox', { name: 'นามส...' })		
Fill "ILoveSe@gmail.com"	44ms	getByRole('textbox', { name: 'อีเมล' })		
Fill "123456"	42ms	getByRole('textbox', { name: 'รหัสผ...' })		
Fill "123456"	41ms	getByRole('textbox', { name: 'ยืนยัน...' })		
Click	81ms	getByRole('button', { name: 'ลงทะเบ...' })		
Expect "toBeVisible"	17ms			
				5 hidden

Pre condition :

- All other information except email is correct.
- email [ILoveSe@gmail.com](mailto:ILoveSe@gmail.com) is already in the system

Test Case ID	TC 1-8
Input	ชื่อจริง : Love นามสกุล : SE อีเมล : ILoveSe@gmail.com รหัสผ่าน : 123456 ยืนยันรหัสผ่าน : 123456
Expected Output	<ul style="list-style-type: none"> <li>- Still on the signup page</li> <li>- Show text error “Email already exists”</li> </ul>
Actual Output	<ul style="list-style-type: none"> <li>- Still on the signup page</li> <li>- Show text error “Email already exists”</li> </ul>
Equivalence Classes Covered	8

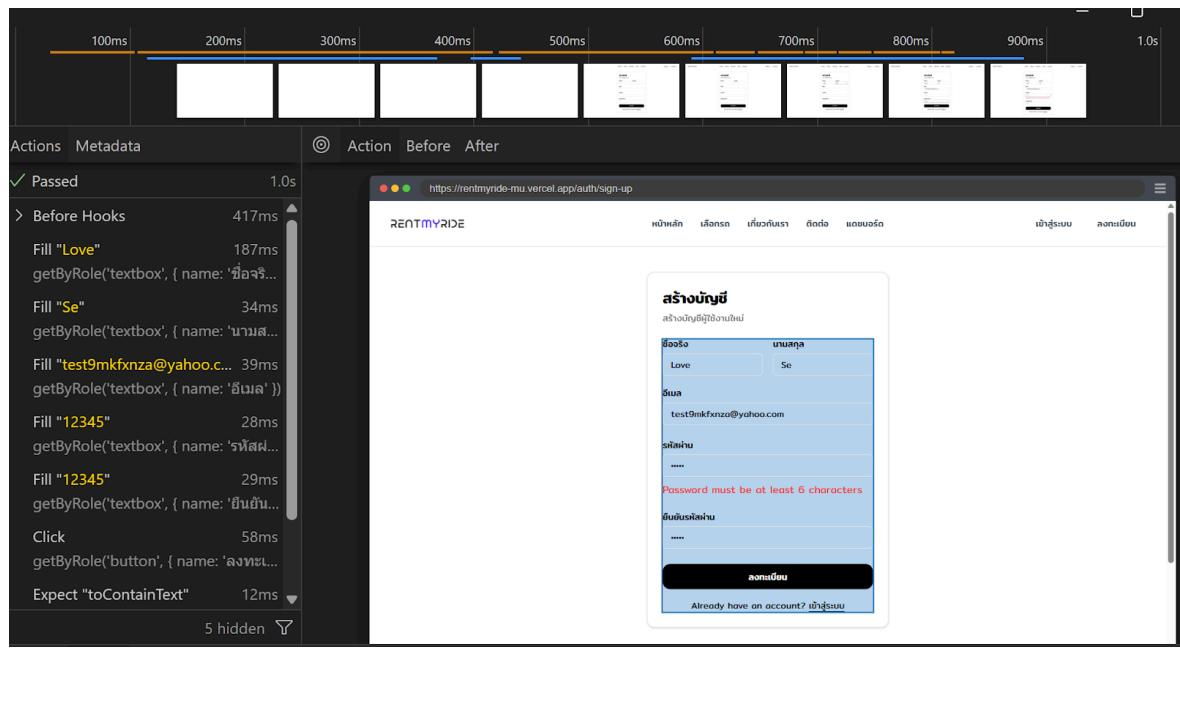
Playwright Result

Action	Metadata
Passed	2.8s
> Before Hooks	448ms
Fill "Love"	133ms
getByRole('textbox', { name: 'ชื่อจริ...' })	
Fill "Se"	20ms
getByRole('textbox', { name: 'นามส...' })	
Fill "test3ak79byt@outlook...."	18ms
getByRole('textbox', { name: 'อีเมล' })	
Fill "123456"	16ms
getByRole('textbox', { name: 'รหัสผ...' })	
Fill "123456"	20ms
getByRole('textbox', { name: 'ยืนยัน...' })	
Click	62ms
getByRole('button', { name: 'ลงทะเบ...' })	
Wait for navigation	1.2s
5 hidden	

Pre Condition : All other information except password is correct.

Test Case ID	TC 1-9
Input	ชื่อจริง : Love นามสกุล : SE อีเมล : ILoveSe@gmail.com รหัสผ่าน : 12345 ยืนยันรหัสผ่าน : 12345
Expected Output	<ul style="list-style-type: none"> <li>- Still on the signup page</li> <li>- Show text error “Password must be at least 6 characters”</li> </ul>
Actual Output	<ul style="list-style-type: none"> <li>- Still on the signup page</li> <li>- Show text error “Password must be at least 6 characters”</li> </ul>
Equivalence Classes Covered	9

### Playwright Result



The screenshot displays the Playwright UI test results for the sign-up page. On the left, a timeline shows the sequence of actions from 100ms to 1.0s. The actions listed in the Actions column include:

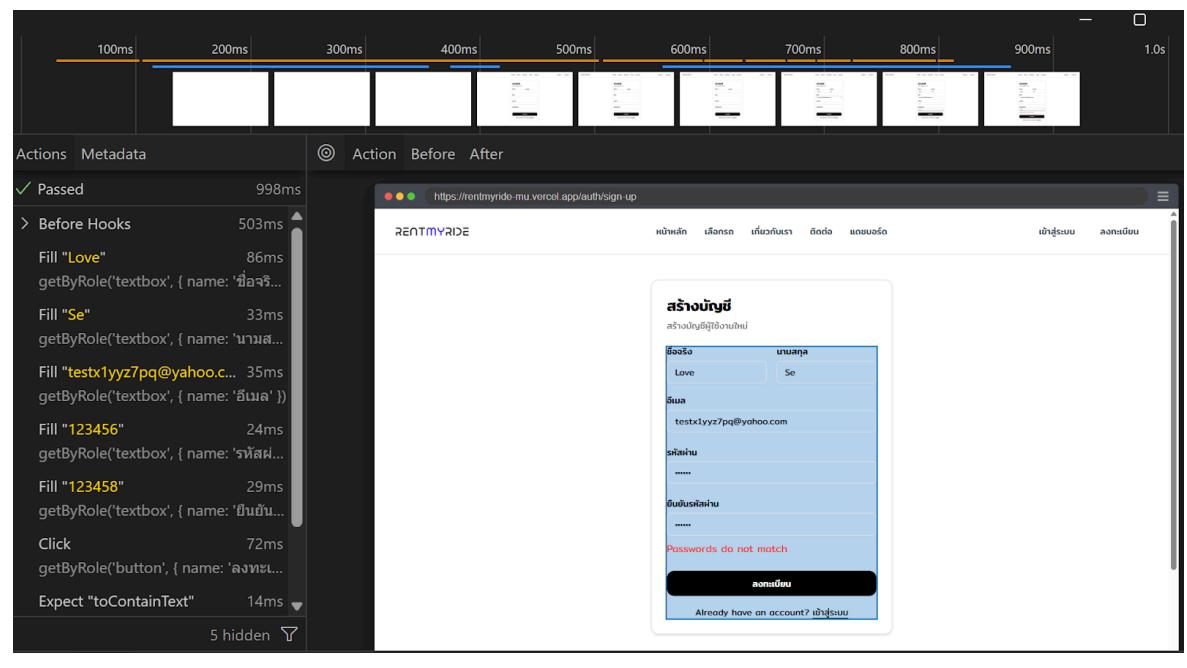
- Before Hooks (417ms)
- Fill "Love" (187ms)
- Fill "Se" (34ms)
- Fill "test9mkfxnza@yahoo.c..." (39ms)
- Fill "12345" (28ms)
- Fill "12345" (29ms)
- Click (58ms)
- Expect "toContainText" (12ms)

The Metadata column shows the duration for each action. The Before and After columns show the state of the application at those points. On the right, a screenshot of the RENTMYRIDE sign-up page is shown. The form fields contain "Love", "Se", "test9mkfxnza@yahoo.com", "12345", and "12345". A red error message "Password must be at least 6 characters" is visible below the password field. The page has a light blue background with some text in Thai.

Pre Condition : All other information except confirm password is correct.

Test Case ID	TC 1-10
Input	ชื่อจริง : Love นามสกุล : SE อีเมล : ILoveSe@gmail.com รหัสผ่าน : 123456 ยืนยันรหัสผ่าน : 123458
Expected Output	- Still on the signup page - Show text error “Passwords do not match”
Actual Output	- Still on the signup page - Show text error “Passwords do not match”
Equivalence Classes Covered	10

### Playwright Result



The screenshot shows the Playwright test results interface. At the top, a timeline bar spans from 100ms to 1.0s, with several blue horizontal bars indicating the duration of different actions. Below the timeline is a table of actions:

Actions	Metadata
✓ Passed	998ms
> Before Hooks	503ms
Fill "Love"	86ms
getByRole('textbox', { name: 'ชื่อจริ...' })	
Fill "Se"	33ms
getByRole('textbox', { name: 'นามส...' })	
Fill "testx1yyz7pq@yahoo.c..."	35ms
getByRole('textbox', { name: 'อีเมล' })	
Fill "123456"	24ms
getByRole('textbox', { name: 'รหัสผ...' })	
Fill "123458"	29ms
getByRole('textbox', { name: 'ยืนยัน...' })	
Click	72ms
getByRole('button', { name: 'ลงทะเบ...' })	
Expect "toContainText"	14ms
5 hidden	

To the right of the table, a browser window displays the 'RENTMYRIDE' sign-up page. The page has input fields for First Name, Last Name, Email, Password, and Confirm Password. The Confirm Password field contains '123458'. A red error message 'Passwords do not match' is visible below the buttons. The browser's address bar shows the URL: <https://rentmyride-mu.vercel.app/auth/sign-up>.

## User Story 2-2 (Login)

- Input Equivalence Class

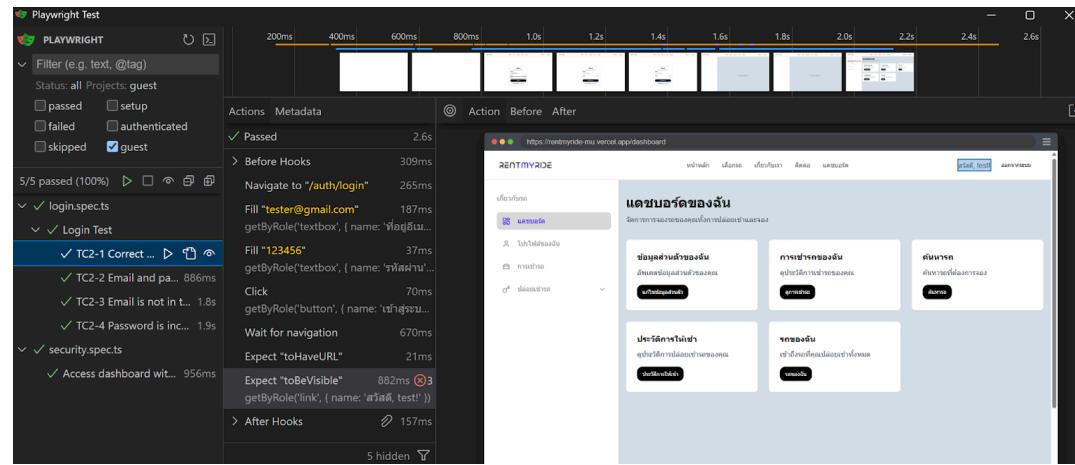
Equivalent Class ID	Input	Validation
1	Email and password correct	valid
2	Email and password is empty	invalid
3	Email is not in the system	invalid
4	Password is incorrect	invalid

- Test case

Pre Condition : tester@gmail.com is registered and password is match

Test Case ID	TC 2-1
Input	อีเมล : tester@gmail.com รหัสผ่าน : 123456
Expected Output	- when click “เข้าสู่ระบบ” button, you will be taken to the dashboard page - show username in header
Actual Output	- when click “เข้าสู่ระบบ” button, dashboard page is displayed - username test! shown
Equivalence Classes Covered	1

### Playwright Result



Test Case ID	TC 2-2
Input	อีเมล : <empty> รหัสผ่าน : <empty>
Expected Output	- when click “เข้าสู่ระบบ” button, error message “Email is required” and “Password is required” will show under a field - Still on the login page
Actual Output	- when click “เข้าสู่ระบบ” button, error message “Email is required” and “Password is required” are shown under a field - Still on the login page
Equivalence Classes Covered	2

### Playwright Result

The screenshot displays the Playwright UI test results for a login page. On the left, a timeline shows the execution flow from 100ms to 1.0s. The main area shows the browser window with the RENTMYRIDE login page. The browser window title is "RENTMYRIDE" and the URL is "https://rentmyride-mu.vercel.app/auth/login". The login form has fields for Email and Password. Below the form, an error message box is visible with the text "เข้าสู่ระบบ" and "ไม่ได้ระบุอีเมลหรือรหัสผ่านที่ถูกต้อง". The browser also shows navigation buttons and a status bar indicating the page is loading.

Actions	Metadata	Action	Before	After
Passed	Launch browser	1.0s		
>	Before Hooks	298ms		
	Navigate to "/auth/login"	279ms		
	Click	235ms		
	getByRole('button', { name: 'เข้าสู่ระบบ' })			
	Expect "toBeVisible"	24ms		
	getByText('Email is required')			
	Expect "toBeVisible"	11ms		
	getByText('Password is required')			
	Expect "toBe"	0ms		
>	After Hooks	147ms		
	5 hidden			

Below the table, there are links for Locator, Source, Call, Log, Errors, Console, Network, Attachments, Annotations, and a search bar.

Pre Condition : Email [ILoveSe@gmail.com](mailto:ILoveSe@gmail.com) has never been registered.

Test Case ID	TC 2-3
Input	อีเมล : <a href="mailto:ILoveSe@gmail.com">ILoveSe@gmail.com</a> รหัสผ่าน : 123456
Expected Output	<ul style="list-style-type: none"> <li>- when click “เข้าสู่ระบบ” button, error message “อีเมลหรือรหัสผ่านไม่ถูกต้อง” will show under a field</li> <li>- a login failed toast will display</li> <li>- still on the login page</li> </ul>
Actual Output	<ul style="list-style-type: none"> <li>- when click “เข้าสู่ระบบ” button, error message “อีเมลหรือรหัสผ่านไม่ถูกต้อง” are shown under a field</li> <li>- Still on the login page</li> <li>- Login failed toast appears.</li> </ul>
Equivalence Classes Covered	3

### Playwright Result

The screenshot displays the Playwright test results for a login scenario. On the left, a timeline shows the sequence of actions and their execution times. On the right, a browser window shows the 'RENTMYRIDE' login page. The user has entered 'ILoveSe@gmail.com' and '123456'. An error message 'อีเมลหรือรหัสผ่านไม่ถูกต้อง' is visible below the password field, and a red toast notification at the bottom of the screen also indicates an error.

Action	Metadata	Before	After
Passed	2.1s		
> Before Hooks	260ms		
Navigate to "/auth/login"	424ms		
Fill "ILoveSe@gmail.com"	186ms		
getByRole('textbox', { name: 'ที่อยู่อิมเมจ' })			
Fill "123456"	34ms		
getByRole('textbox', { name: 'รหัสผ่าน' })			
Click	65ms		
getByRole('button', { name: 'เข้าสู่ระบบ' })			
Expect "toContainText"	898ms <span style="color: red;">✖1</span>		
locator('form')			
Expect "toBe"	0ms		
> After Hooks	163ms		
5 hidden			

Pre Condition : tester@gmail.com is registered but password “abcdef” is incorrect

Test Case ID	TC 2-4
Input	อีเมล : tester@gmail.com รหัสผ่าน : abcdef
Expected Output	- when click “เข้าสู่ระบบ” button, error message “ อีเมลหรือรหัสผ่านไม่ถูกต้อง” will show under a field - a login failed toast will display - still on the login page
Actual Output	- when click “เข้าสู่ระบบ” button, error message “ อีเมลหรือรหัสผ่านไม่ถูกต้อง” are shown under a field - Still on the login page - Login failed toast appears.
Equivalence Classes Covered	4

### Playwright Result

The screenshot displays the Playwright test results for a login scenario. On the left, a timeline shows the sequence of actions from 100ms to 1.5s. The actions listed are:

- > Before Hooks: 328ms
- Navigate to "/auth/login": 271ms
- Fill "tester@gmail.com": 178ms
- Fill "abcdef": 37ms
- Click: 67ms
- Expect "toContainText": 381ms (1 failure)
- Expect "toBe": 0ms
- > After Hooks: 0ms

The total duration of the test is 1.5s. A browser screenshot on the right shows the RENTMYRIDE login page. The user has entered "tester@gmail.com" into the email field and "abcdef" into the password field. The "Sign-in" button is being clicked. An error message "อีเมลหรือรหัสผ่านไม่ถูกต้อง" is visible below the email input field. A red toast message "Login failed" is displayed at the bottom of the screen.

## User Story 2-3 (Sign-out)

- Input Equivalence Class

Equivalent Class ID	Input	Validation
1	Click on logout button	valid
2	Go to dashboard after logout	valid

Pre Condition : Sign in before

Test Case ID	TC 3-1
Input	-
Expected Output	- when click “ออกจากระบบ” button, - go to login page
Actual Output	- when click “ออกจากระบบ” button - go to login page
Equivalence Classes Covered	1

### Playwright Result

The screenshot shows the Playwright Test interface with the following details:

- Test Suite:** PLAYWRIGHT
- Status:** all Projects: setup authentication...
- Actions:** 0/2 passed (0%)
- Test Cases:**
  - auth.setup.ts**: Passed (6.2s)
  - authed**: Passed (6.2s)
    - logout.spec.ts**: Passed (4.2s)
      - Signout from a...**: Clicked on "Logout" button (1.0s)
      - Expect "toBeVisible" (322ms)
      - Expect "toHaveURL" (22ms)
  - not-authed** (2 tests):
    - register.spec.ts**
    - security.spec.ts**:
      - Access dashboard without auth (479ms)
- Metadata:** 6 hidden
- Timeline:** Shows a timeline from 0.0s to 4.0s with several screenshot markers.
- Logs:** Shows a log entry for the "Logout" click action.
- Snapshot:** A screenshot of the application showing the login page with fields for "Email" (john@example.com) and "Password".

## Pre Condition : Logout before

Test Case ID	TC 3-2
Input	-
Expected Output	<ul style="list-style-type: none"> <li>- when go to dashboard page</li> <li>- redirect to login page</li> </ul>
Actual Output	<ul style="list-style-type: none"> <li>- when go to dashboard page</li> <li>- redirect to login page</li> </ul>
Equivalence Classes Covered	2

## Playwright Result

The screenshot shows the Playwright Test interface with the following details:

- Test Status:** 1/1 passed (100%)
- Test Suite:** PLAYWRIGHT
- Test Filter:** Filter (e.g. text, @tag)
- Timeline:** A horizontal timeline from 200ms to 1.6s showing the sequence of actions.
- Actions Tab:**
  - auth.setup.ts: Passed (1.7s)
  - authed: Navigate to "/dashboard" (733ms)
  - Expect "toHaveURL" (86ms)
  - not-authed: register.spec.ts (4.0s)
  - security.spec.ts: Access dashboard (373ms)
- Metadata Tab:** Shows the same actions with their respective execution times.
- Browser Screenshot:** A screenshot of a browser window titled "เรียกทรัพย์" (Rent My Ride) showing a login form. The URL is https://rentmyride-mu.vercel.app/auth/login. The form fields are "ชื่อผู้ใช้" (john@example.com) and "รหัสผ่าน". Below the form is a button labeled "เข้าสู่ระบบ" (Log In).
- Bottom Navigation:** Locator, Source, Call, Log, Errors, Console, Network (41), Attachments (1), Annotations.

## Manual black-box testing

### EPIC 5 : Car search & browse

#### User Story 5-1 (Search and filter)

- Input Equivalence Class

Equivalent Class ID	Input	Validation
1	Location free text(Any string; can be from suggestions or typed)	Valid
2	Empty location & empty dates	Valid
3	Location only (no dates)	Valid
4	Only one date provided (start only or end only)	Invalid
5	Both dates, valid start $\leq$ end day (end exclusive in code)	Valid
6	reversed dates	Invalid
7	Date min attributes respected (start $\geq$ today, end $\geq$ start or today)	Valid
8	Price buckets p0–p4 + “All”	Valid
9	Seats radio 4/5/7 + “All”	Valid
10	Gear multi-select: “ออโต้”, “ธรรมดា”, “กี๊ จัง ออโต้”	Valid
11	Clear All resets	Valid
12	Inputs reflect value (controlled)	Valid
13	Popularity sort (primary)	Valid

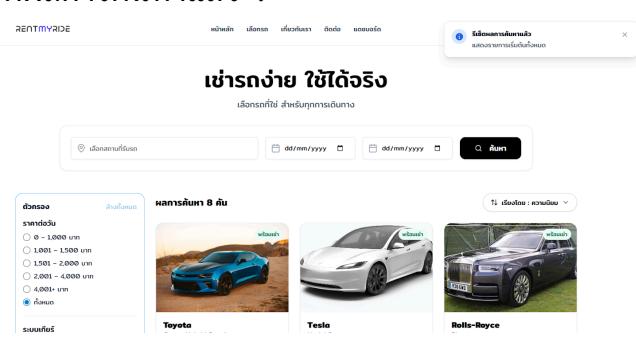
14	Price ASC sort	Valid
15	Rating DESC sort	Valid
16	Year DESC sort	Valid

- Test case

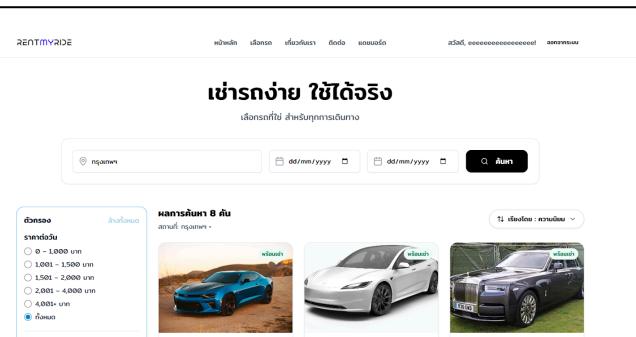
Pre Condition : User is logged in, main page with car card is loaded, and the search system is ready.

Test Case ID	TC 4-1
Action	Focus location, ArrowDownx2, Enter
Expected Output	Suggestions open , selection commits to input, list closes
Actual Output	
Equivalence Classes Covered	1

Pre Condition : User is logged in, main page with car card is loaded, and the search system is ready.

Test Case ID	TC 4-2
Action	Click page outside suggestions.
Expected Output	Suggestions close.
Actual Output	<p><b>cars:reset</b> (verify event dispatched). Toast with title “รีเซ็ตผลการค้นหาแล้ว”.</p> 
Equivalence Classes Covered	2

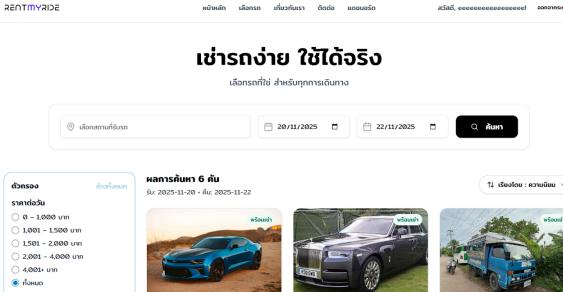
Pre Condition : User is logged in, main page with car card is loaded, and the search system is ready.

Test Case ID	TC 4-3
Action	Type location="กรุงเทพ"; leave dates empty; click <b>Search</b>
Expected Output	Call <b>searchCarsByLocation("กรุงเทพ")</b> once; dispatch <b>cars:search-results</b> with <b>unlabeled DbCar[]</b>
Actual Output	
Equivalence Classes Covered	3

Pre Condition : User is logged in, main page with car card is loaded, and the search system is ready.

Test Case ID	TC 4-4
Action	Set only start or only end
Expected Output	toast “ເລືອກວັນທີໄມ່ຄຽບ”; no request
Actual Output	
Equivalence Classes Covered	4

Pre Condition : User is logged in, main page with car card is loaded, and the search system is ready.

Test Case ID	TC 4-5
Action	Set start=2025-11-20, end=2025-11-22, click <b>Search</b>
Expected Output	Two calls: <code>searchAvailableCars(dates, maybe empty location)</code> to form <code>availSet</code> and <code>searchCarsByLocation(location)</code> to get scope; dispatch labeled cars where <b>only</b> verified+free are "ພັດມາເຊົາ"
Actual Output	
Equivalence Classes Covered	5

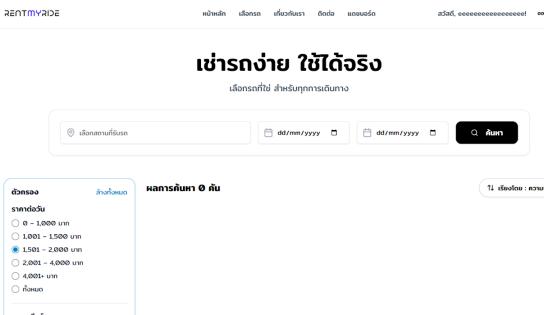
Pre Condition : User is logged in, main page with car card is loaded, and the search system is ready.

Test Case ID	TC 4-6
Action	Reversed dates (start $\geq$ end)
Expected Output	Toast “ช่วงวันไม่ถูกต้อง ...”; no calls
Actual Output	
Equivalence Classes Covered	6

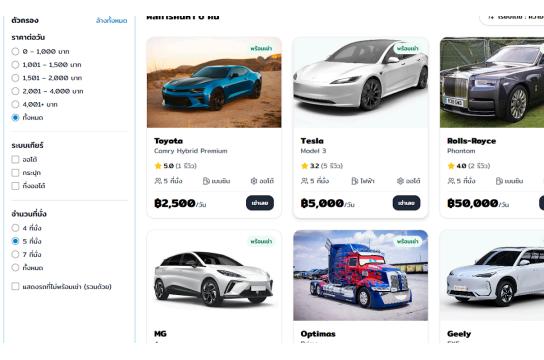
Pre Condition : User is logged in, main page with car card is loaded, and the search system is ready.

Test Case ID	TC 4-7
Action	Verify input min attributes
Expected Output	start.min=today; `end.min = start
Actual Output	
Equivalence Classes Covered	7

Pre Condition : User is logged in, main page with car card is loaded, and the search system is ready.

Test Case ID	TC 4-8
Action	Click price 1501-2000 บาท
Expected Output	onChange with {price:{min:1501,max:2000}}; chip shows; Clear All enabled
Actual Output	
Equivalence Classes Covered	8

Pre Condition : User is logged in, main page with car card is loaded, and the search system is ready.

Test Case ID	TC 4-9
Action	Click Seats 5
Expected Output	seats:{min:5,max:5}; radio reflects
Actual Output	
Equivalence Classes Covered	9

Pre Condition : User is logged in, main page with car card is loaded, and the search system is ready.

Test Case ID	TC 4-10
Action	Toggle both gears then untick one
Expected Output	gear_types updates without duplicates (Set logic)
Actual Output	
Equivalence Classes Covered	10

Pre Condition : User is logged in, main page with car card is loaded, and the search system is ready.

Test Case ID	TC 4-11
Action	Click <b>Clear All</b> with any filter active
Expected Output	Filters reset; Clear All disabled; listing re-filters accordingly
Actual Output	
Equivalence Classes Covered	11

Pre Condition : User is logged in, main page with car card is loaded, and the search system is ready.

Test Case ID	TC 4-12
Action	Prop → UI reflection
Expected Output	gear_types updates without duplicates (Set logic)
Actual Output	
Equivalence Classes Covered	12

Pre Condition : User is logged in, main page with car card is loaded, and the search system is ready.

Test Case ID	TC 4-13
Action	Click Popularity sort (primary)
Expected Output	sorted in order (Popularity)
Actual Output	
Equivalence Classes Covered	13

Pre Condition : User is logged in, main page with car card is loaded, and the search system is ready.

Test Case ID	TC 4-14
Action	Click Price ASC sort
Expected Output	sorted in order (Price ASC)
Actual Output	
Equivalence Classes Covered	14

Pre Condition : User is logged in, main page with car card is loaded, and the search system is ready.

Test Case ID	TC 4-15
Action	Click Rating DESC sort
Expected Output	sorted in order (Rating DESC)
Actual Output	
Equivalence Classes Covered	15

Pre Condition : User is logged in, main page with car card is loaded, and the search system is ready.

Test Case ID	TC 4-16
Action	Click Rating DESC sort
Expected Output	sorted in order (Year DESC sort)
Actual Output	
Equivalence Classes Covered	16

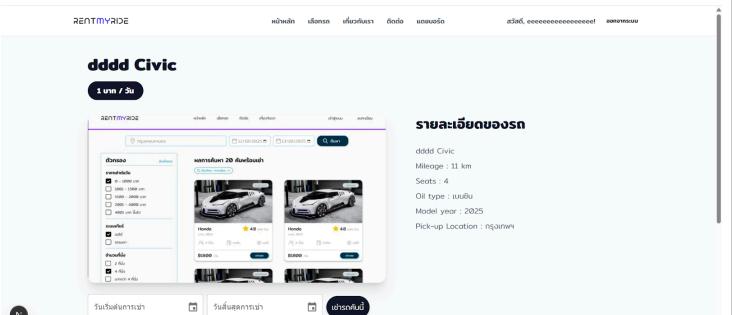
## User Story 5-2 (View car details)

- Input Equivalence Class

Equivalent Class ID	Input	Validation
1	View a car successfully after searching for the correct condition	valid
2	Unable to view vehicle details because the user is not logged in	Invalid
3	The user cannot access the car information page because the car has already been booked.	Invalid

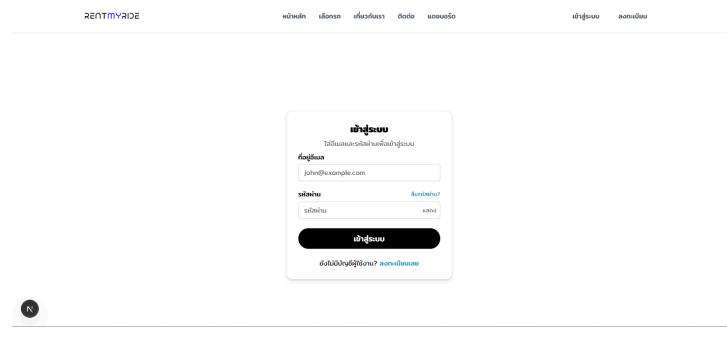
Pre Condition :

1. User logged in, valid date selected, car is available

Test Case ID	TC 5-1
Action	Press “ເຊົ່າຮຄນິ້ນ໌” Button
Expected Output	- Go to view car detail page with correct information such as start date, end date, price, day
Actual Output	
Equivalence Classes Covered	1

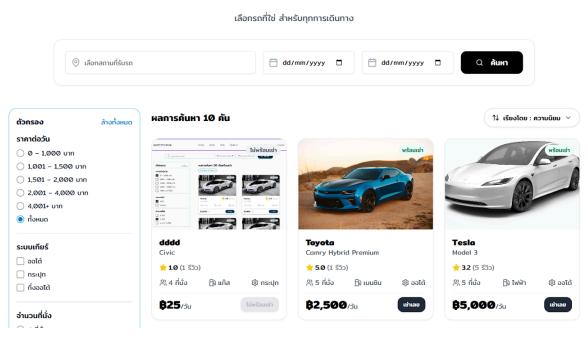
Pre Condition :

2. User is not logged in, valid date selected, car is available

Test Case ID	TC 5-2
Action	Press “เข้ารถคันนี้” Button
Expected Output	- The user cannot access the car information page; they are redirected to the login page.
Actual Output	
Equivalence Classes Covered	2

Pre Condition :

3. User logged in, valid date selected, car is not verified

Test Case ID	TC 5-3
Action	User cannot access the car detail page because the car has not been verified.
Expected Output	The “เข้าเลย” button has been changed to “ไม่พร้อมเข้า,” which causes the user to be unable to view the car detail page.
Actual Output	
Equivalence Classes Covered	3

## EPIC 6 : Car Renting Process

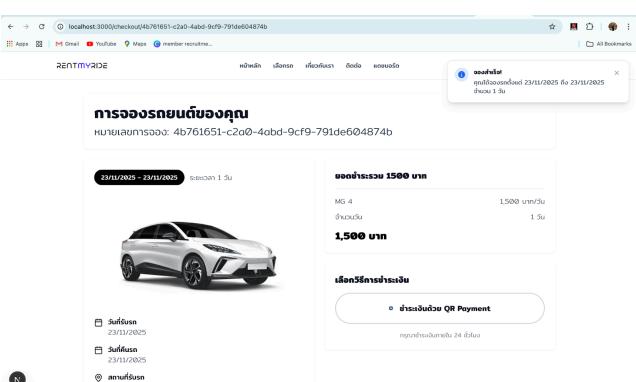
### User Story 6-1 (Select Rental Date)

- Input Equivalence Class

Equivalent Class ID	Input	Validation
1	Select valid rental dates (start date $\leq$ end date, both future dates) and car is available	valid
2	Select valid dates but the car is already booked for that period	invalid
3	Select invalid date range (start date $>$ end date or past date)	invalid
4	Missing one or both dates (incomplete date input)	invalid

- Test case

Pre Condition : User is logged in, car detail is loaded, and the system has pre-existing booking data.

Test Case ID	TC 6-1
Action	Select a start date and end date where the car is available. Then press “เช่ารถคันนี้” button.
Expected Output	<ul style="list-style-type: none"> <li>- Redirects to Checkout page showing correct rental details: start date, end date, price, total days</li> <li>- A toast notification appears confirming successful date selection.</li> </ul>
Actual Output	
Equivalence Classes Covered	1

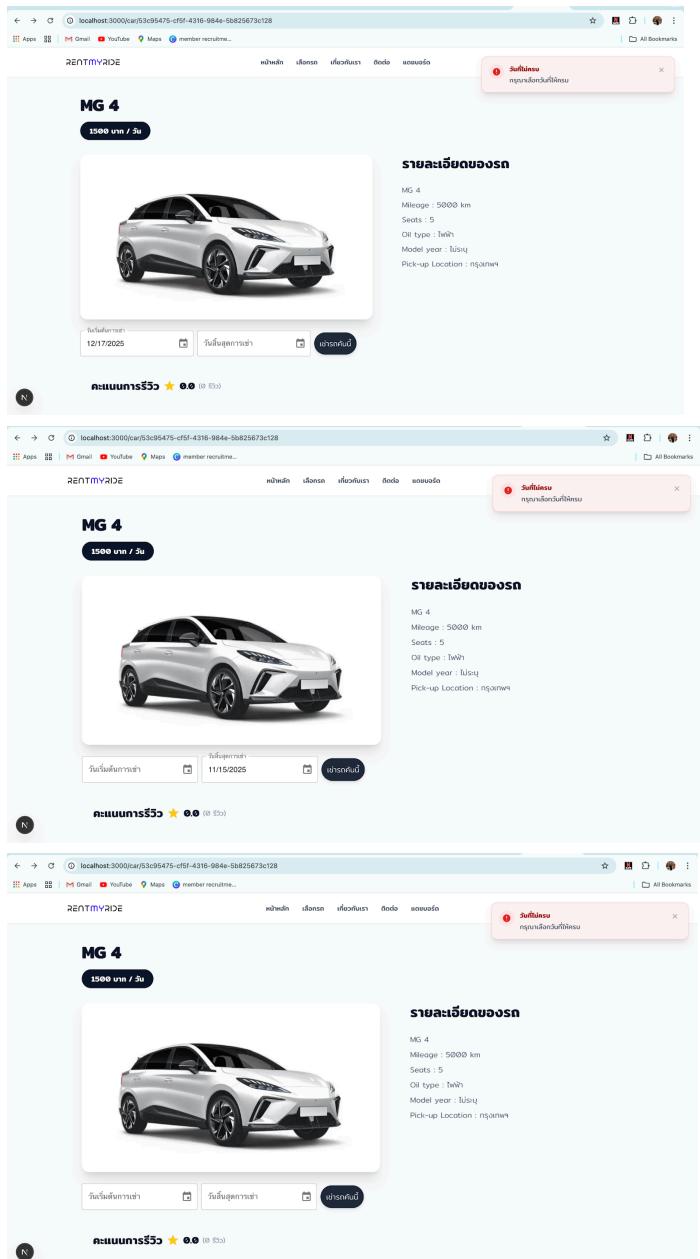
Pre Condition : User is logged in, car detail is loaded, and the system has pre-existing booking data.

Test Case ID	TC 6-2
Action	Select a date range where the car is already booked. Then press “เช่ารถคันนี้” button.
Expected Output	- A toast notification appears “การจองไม่สำเร็จ รถยนต์ไม่ว่างในช่วงเวลาที่เลือก”.
Actual Output	
Equivalence Classes Covered	2

Pre Condition : User is logged in, car detail is loaded, and the system has pre-existing booking data.

Test Case ID	TC 6-3
Action	Select an invalid date range (start date after end date or past date). Then press “ເຊົ່າຄົ້ນນີ້” button.
Expected Output	- A toast notification appears “ເກີດຂໍອິດພລາດ ວັນເຮີມ ຕ້ອງໄມ່ເກີນວັນສິນສຸດ”.
Actual Output	
Equivalence Classes Covered	3

Pre Condition : User is logged in, car detail is loaded, and the system has pre-existing booking data.

Test Case ID	TC 6-4
Action	Select only one date (either start date or end date), then click the “ເຊົ່າຄົ້ນນີ້” button.
Expected Output	- A toast notification appears “ວັນທີໄມ້ກຽບ ກຽນາເລືອກວັນໃຫ້ກຽບ”.
Actual Output	
Equivalence Classes Covered	4

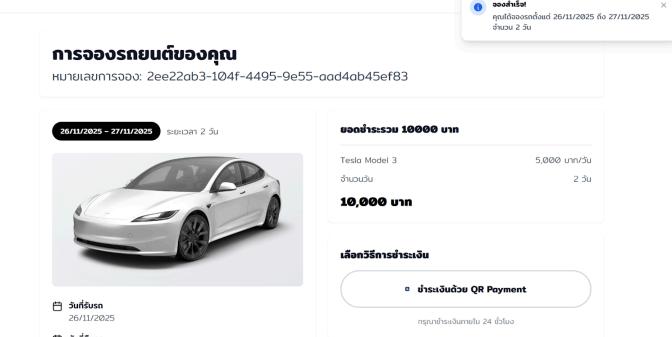
## User Story 6-2 (Book a car)

- Input Equivalence Class

Equivalent Class ID	Input	Validation
1	Book a car successfully after selecting date	valid
2	Booking a car that becomes unavailable right before booking	invalid

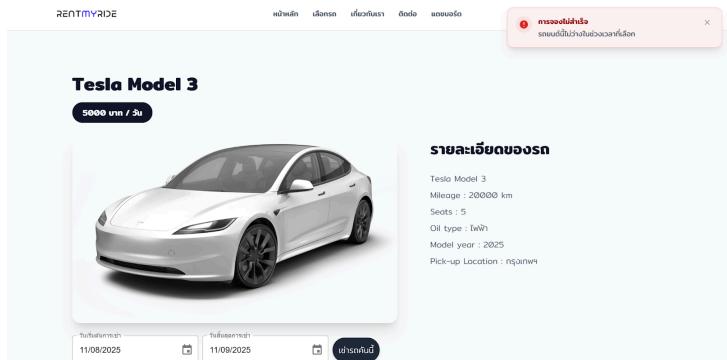
- Test case

Pre Condition : User logged in, valid date selected, car is available.

Test Case ID	TC 7-1
Action	Press “เข้ารถคันนี้” Button
Expected Output	<ul style="list-style-type: none"> <li>- Go to checkout page with correct information such as start date, end date, price, day</li> <li>- Toast with renting details will displayed</li> </ul>
Actual Output	
Equivalence Classes Covered	1

Pre Condition :

4. User logged in, valid date selected, car is not available

Test Case ID	TC 7-2
Action	Press “เช่ารถคันนี้” Button
Expected Output	<ul style="list-style-type: none"> <li>- Go to checkout page with correct information such as start date, end date, price, day</li> <li>- Toast with renting details will displayed</li> <li>- Toast with error will displayed</li> <li>- Still in</li> </ul>
Actual Output	
Equivalence Classes Covered	2

## EPIC 12 : Consent

### User Story 12-1 (Consent)

- Input Equivalence Class

Equivalent Class ID	Input	Validation
1	Accept Consent	valid
2	Decline Consent	invalid

- Testcase

Pre condition : login with valid email and password

Test Case ID	TC 8-1
Action	- accept consent - login
Expected Output	login success and go to dashboard page
Actual Output	login success and go to dashboard page
Equivalence Classes Covered	1

### Playwright Result

The screenshot displays the Playwright test results interface. On the left, a timeline shows the execution of actions over time, from 100ms to 1.8s. Below the timeline is a table of actions with their descriptions and execution times. On the right, a browser window shows the 'RENTMYRIDE' dashboard page with a message 'กรุณาตรวจสอบว่าคุณได้รับ...'.

Actions	Metadata	Time
✓ Passed		1.8s
> Before Hooks		184ms
Navigate to "/auth/login"		319ms
Click	getByRole("button", { name: "กดตกลงเงื่อนไข" })	162ms
Fill "tester@gmail.com"	getByRole("textbox", { name: "ที่อยู่อีเมล" })	19ms
Fill "123456"	getByRole("textbox", { name: "รหัสผ่าน" })	18ms
Click	getByRole("button", { name: "เข้าสู่ระบบ" })	55ms
Wait for navigation		885ms
Expect "toBe"		1ms
> After Hooks		172ms

- Testcase

Pre condition : login with valid email and password

Test Case ID	TC 8-2
Action	<ul style="list-style-type: none"> <li>- decline consent</li> <li>- login</li> </ul>
Expected Output	<ul style="list-style-type: none"> <li>- login not success</li> <li>- toast “ມີເອນຸຄູາຕໍ່ໃຫ້ເຂົ້າສູ່ຮະບບ” displayed</li> <li>- still on login page</li> </ul>
Actual Output	<ul style="list-style-type: none"> <li>- login not success</li> <li>- toast “ມີເອນຸຄູາຕໍ່ໃຫ້ເຂົ້າສູ່ຮະບບ” displayed</li> <li>- still on login page</li> </ul>
Equivalence Classes Covered	2

Playwright Result

The screenshot displays the Playwright UI test results. On the left, a timeline shows the sequence of actions from 50ms to 950ms. The actions listed in the Actions column include navigating to the login page, entering email and password, and clicking the login button. The Metadata column provides execution times for each step. On the right, a browser screenshot shows the login page with a red toast message at the top stating "ມີເອນຸຄູາຕໍ່ໃຫ້ເຂົ້າສູ່ຮະບບ". The browser URL is https://rentmyride-mu.vercel.app/auth/login.

Actions	Metadata
Passed	954ms
> Before Hooks	180ms
Navigate to "/auth/login"	382ms
Click	150ms
getByRole("button", { name: 'ປົກລົງເຂົ້າ' })	17ms
Fill "tester@gmail.com"	17ms
getByRole("textbox", { name: 'ທີ່ອ່ານືນດຳ' })	17ms
Fill "123456"	17ms
getByRole("textbox", { name: 'ລັດສິດ່ານ' })	17ms
Click	46ms
getByRole("button", { name: 'ປົກລົງເຂົ້າ' })	12ms
Expect "toBeVisible"	0ms
getByText("ມີເອນຸຄູາຕໍ່ໃຫ້ເຂົ້າສູ່ຮະບບ", { exact: true })	0ms
Expect "toBe"	0ms
> After Hooks	132ms
5 hidden	

## Back-end function testing

function : IsAdmin

```
+ import { isAdmin } from "@/lib/authServices";
+
+ describe('Test function isAdmin ', () => {
+     let mockSupabase: any;
+
+     beforeEach(() => {
+         mockSupabase = {
+             auth: {
+                 getUser: jest.fn(),
+                 from: jest.fn().mockReturnThis(),
+                 select: jest.fn().mockReturnThis(),
+                 eq: jest.fn().mockReturnThis(),
+                 single: jest.fn(),
+             };
+         });
+     });
+
```

This runs **before every single test**, so each test starts with a fresh `mockSupabase` object.

```
it('should return true if user exists and role is admin', async () => {
    mockSupabase.auth.getUser.mockResolvedValue({
        data: { user: { id: '123', email: 'tester@gmail.com' } },
        error: null
    });
    mockSupabase.single.mockResolvedValue({
        data: { role: 'admin' },
        error: null
    });

    const result = await isAdmin(mockSupabase);
    expect(result).toBe(true);

    //check in ieuung
    expect(mockSupabase.from).toHaveBeenCalledWith('user_info');
    expect(mockSupabase.select).toHaveBeenCalledWith('role');
    expect(mockSupabase.eq).toHaveBeenCalledWith('user_id', '123');

});
});
```

this test case create a mock data which its role is admin to ensure that `isAdmin` function works correctly by returning true

```

it('should return false if user does not exist', async () => {
    mockSupabase.auth.getUser.mockResolvedValue({
        data: { user: null },
        error: null
    });

    const result = await isAdmin(mockSupabase);
    expect(result).toBe(false);
});

```

this test case create a mock data with nothing in it to ensure that isAdmin function works correctly by returning false

```

it('should return false if getUser have error', async () => {
    mockSupabase.auth.getUser.mockResolvedValue({
        data: {user: null},
        error : {message : 'auth error'}
    });

    const result = await isAdmin(mockSupabase);
    expect(result).toBe(false);
});

```

this test case simulate the authentication error from getUser to make sure that isAdmin function works correctly by returning false

```

it('should return false if role error' , async () => {
    mockSupabase.auth.getUser.mockResolvedValue({
        data: { user: { id: '123',email: 'tester@gmail.com' } },
        error: null
    });
    mockSupabase.single.mockResolvedValue({
        data: null,
        error: {message:'role error'}
    });
}

```

this test case simulate the role error from getUser to make sure that isAdmin function works correctly by returning false

```

it('should return false if user information does not exist', async () => {
    mockSupabase.auth.getUser.mockResolvedValue({
        data: { user: { id: '123', email: 'tester@gmail.com' } },
        error: null
    });
    mockSupabase.single.mockResolvedValue({
        data: null,
        error: null
    });

    const result = await isAdmin(mockSupabase);
    expect(result).toBe(false);
});

```

this test case simulate if there is no role in the user from getUser to make sure that isAdmin function works correctly by returning false

```

it('should return false if role is not admin', async () => {
    mockSupabase.auth.getUser.mockResolvedValue({
        data: { user: { id: '123', email: 'tester@gmail.com' } },
        error: null
    });
    mockSupabase.single.mockResolvedValue({
        data: { role: 'user' },
        error: null
    });

    const result = await isAdmin(mockSupabase);
    expect(result).toBe(false);
});

```

this test case simulate if the user's role is “user” from getUser to make sure that isAdmin function works correctly by returning false

```

afterEach(() => {
    jest.clearAllMocks();
});

```

This will reset all mocks so that calls from one test **don't leak** into another test.

## result

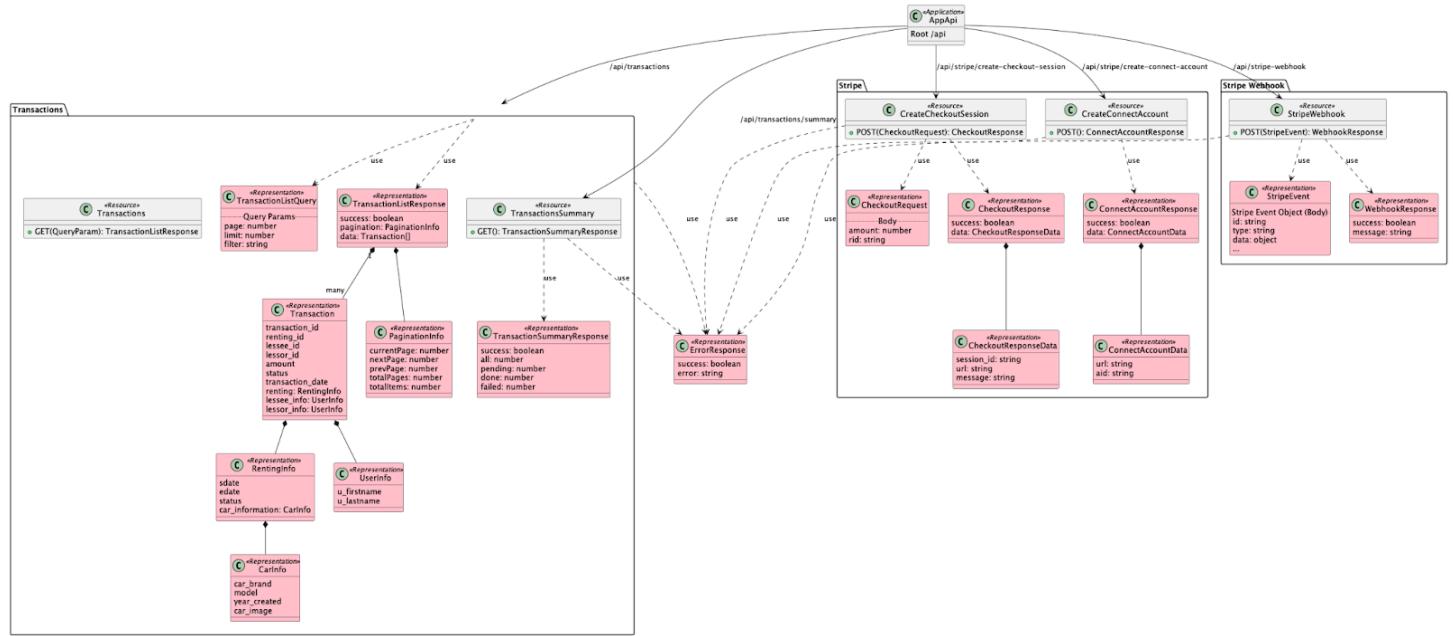
```
C:\Users\peera\OneDrive\שולחן העבודה\year 3\se\rent_my_ride>npx jest --coverage isAdmin.test.ts
PASS  test/isAdmin.test.ts
  Test function isAdmin
    ✓ should return true if user exists and role is admin (2 ms)
    ✓ should return false if user does not exist (1 ms)
    ✓ should return false if getUser have error (1 ms)
    ✓ should return false if role error
    ✓ should return false if user information does no exist
    ✓ should return false if role is not admin

-----|-----|-----|-----|-----|-----|
File      | % Stmtts | % Branch | % Funcs | % Lines | Uncovered Line #s
-----|-----|-----|-----|-----|-----|
All files |    100  |     100  |     100  |     100  |
isAdmin.ts |    100  |     100  |     100  |     100  |
-----|-----|-----|-----|-----|-----|
Test Suites: 1 passed, 1 total
Tests:       6 passed, 6 total
Snapshots:   0 total
Time:        0.644 s, estimated 1 s
Ran all test suites matching isAdmin.test.ts.
```

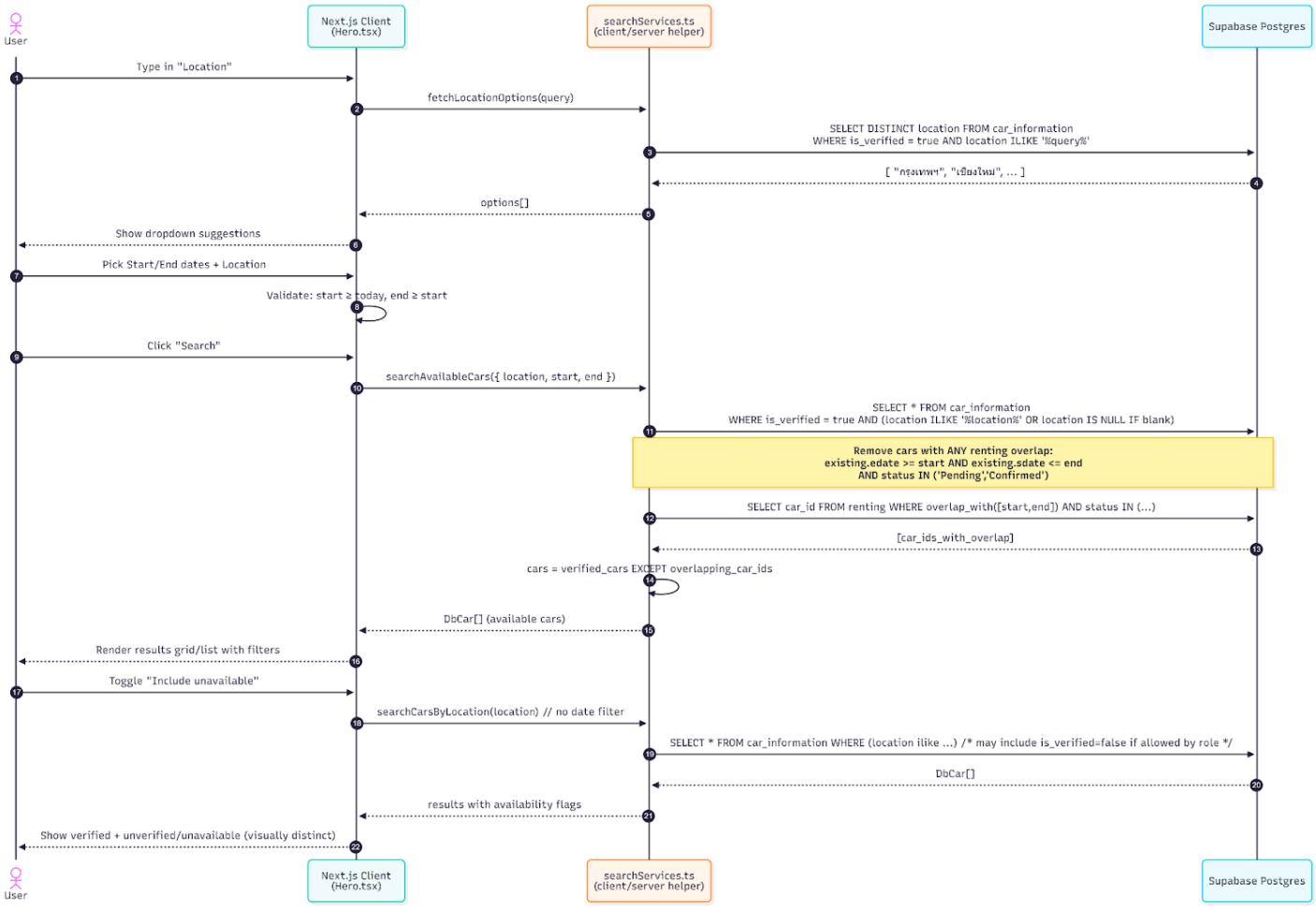
## Test Coverage Matrix

User Story ID	Test Cases ID	Status
US2-1	TC1-1 TC1-2 TC1-3 TC1-4 TC1-5 TC1-6 TC1-7 TC1-8 TC1-9 TC1-10	P,P,P,P,P,P,P,P,P
US2-2	TC2-1 TC2-2 TC2-3 TC2-4	P,P,P,P
US2-3	TC3-1 TC3-2	P,P
US5-1	TC4-1 TC4-2 TC4-3 TC4-4 TC4-5 TC4-6 TC4-7 TC4-8 TC4-9 TC4-10 TC4-11 TC4-12 TC4-13 TC4-14 TC4-15 TC4-16	P,P,P,P,P,P,P,P,P,P,P,P,P
US5-2	TC5-1 TC5-2 TC5-3	P,P,P
US6-1	TC6-1 TC6-2 TC6-3 TC6-4	P,P,P,P
US6-2	TC7-1 TC7-2	P,P
US12-1	TC8-1 TC8-2	P,P

## UML Diagram



## Seq Diagram



## Swagger

The screenshot shows the Rent My Ride API documentation page. At the top, there's a header with the title "Rent My Ride API" and "OAS 3.0". Below the header is a detailed API description for a car rental platform, including sections on authentication, endpoints, and admin privileges.

The main content area is organized into several sections:

- Cars**: Car management endpoints. Includes a **PATCH /api/car/{cid}** endpoint for updating car information.
- Stripe**: Stripe payment integration endpoints. Includes **POST /api/stripe/create-checkout-session** and **POST /api/stripe/create-connect-account** endpoints.
- Transactions**: Transaction management endpoints. Includes **GET /api/transactions** and **GET /api/transactions/summary** endpoints.
- Webhooks**: Webhook endpoints. Includes a **POST /api/stripe-webhook** endpoint for Stripe webhook handling.

Below the sections, there's a detailed view of the **PATCH /api/car/{cid}** endpoint:

- Description**: Update car information. This endpoint requires admin privileges.
- Parameters**:
  - cid** (required): Example : car\_123456
  - string** (path): car\_123456
- Request body** (required): **application/json**. The schema is defined as follows:

```
{
  "car_brand": "Toyota",
  "model": "Camry",
  "mileage": 50000,
  "year_created": 2020,
  "number_of_seats": 5,
  "gear_type": "auto",
  "oil_type": "unbranded",
  "daily_rental_price": 1500,
  "is_verified": true,
  "status": "available",
  "location": "Bangkok",
  "car_conditionrating": 4.5,
  "car_image": "https://example.com/car-image.jpg"
}
```

Responses		
Code	Description	Links
200	Car updated successfully	No links
	Media type <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">application/json</div> <span style="border: 1px solid #ccc; padding: 2px 5px; border-radius: 5px; display: inline-block;">▼</span> <small>Controls Accept header.</small>	
	Example Value   Schema <pre>{   "success": true,   "data": {     "car_id": "car_123456",     "car_brand": "Toyota",     "model": "Camry",     "mileage": 50000,     "year_created": 2020,     "number_of_seats": 5,     "gear_type": "Manual",     "oil_type": "Unbranded",     "daily_rental_price": 1500,     "is_verified": true,     "status": "available",     "location": "Bangkok",     "car_conditionrating": 4.5,     "car_image": "https://example.com/car-image.jpg",     "created_at": "2025-11-16T21:11:56.638Z",     "updated_at": "2025-11-16T21:11:56.638Z"   } }</pre>	
400	Bad request	No links
401	Unauthorized - Admin privileges required	No links
500	Internal server error	No links

Schemas	^
Car >	
Transaction >	
Pagination >	

```

Car <-
car_id           <-- string
                  example: car_123456
car_brand        <-- string
                  example: Toyota
model            <-- string
                  example: Camry
mileage          <-- number
                  example: 50000
year_created     <-- number
                  example: 2020
number_of_seats  <-- number
                  example: 5
gear_type         <-- string
                  example: Manual
oil_type          <-- string
                  example: Unbranded
daily_rental_price <-- number
                  example: 1500
is_verified       <-- boolean
                  example: true
status            <-- string
                  example: available
location          <-- string
                  example: Bangkok
car_conditionrating <-- number
                  example: 4.5
car_image         <-- string($uri)
                  example: https://example.com/car-image.jpg
created_at        <-- string($date-time)
updated_at        <-- string($date-time)
}

```