2025

"Go Bus" Software Testing Project

By:

Team 1

2/20/2025

TABLE OF CONTENTS

Go Bus Software Testing Project

Project Overview:

The goal of this project is to ensure the **“Go Bus”(https://go-bus.com/)**online platform provide a smooth, secure, and reliable booking experience for users. The focus will be on testing functionalities like ticket booking, payment, user registration, and customer support.

The Test Plan is designed to prescribe the scope, approach, resources, and schedule of all testing activities of the project **“Go B”**. The plan identify the **features to be tested**, the **types of testing** to be performed and **the risks** associated with the plan.

Objectives:

TEST OBJECTIVE: are to verify the Functionality of website Go Bus, the project should focus on **Booking a Ticket**, **Searching for Buses**, **Viewing Booking Details**, **Tracking Bus Status**, **Creating an Account** to guarantee all these operation can work normally in real business environment:

- Ensure system reliability and stability.
- Validate user-friendly interface.
- Identify and report functional bugs.
- Improve compatibility across devices and browsers.

TEST PLANNING & ANALYSIS

❖ Features to be tested

Feature to be tested; all the feature of website Go Bus will be testing As an Exploratory Testing.

Module Name	Description
Booking	The user can search for trips, select trips, remove them if desired, confirm trips, Viewing Booking Detail and book them.
login process	User can login The user can login to complete the booking process.
Registeration	User can register with E-mail
Seat selection	User can Seat selection
Search for Buses	User cansearch for buses by entering my departure city, destination, and travel date so that I can find available buses.
Header& Footer (Home page)	Contains :compant,FAQ,Terms,Policy,contact us and language. In footer
Go Bus Routes(Home page)	User can check the routes; there are more than 20 Bus Routes in Egypt
Go-Bus-Classes (Home page)	User can choose bus classes, Explore Egypt with the most booked classes:Super Go D&Business class DD& Economy.
Travel Destination	User can choose travel destination and book through it.
Go Bus Stations	The user can check if a Go bus Stations at their

	location before making a reservation.

In Scope:

Functional Testing:

- ✓ User registration & login process.
- ✓ Ticket booking flow.
- ✓ Payment gateway integration (Fawry, Aman, Visa, etc.).
- ✓ Seat selection and cancellation process.
- ✓ Email and SMS notifications.

Compatibility Testing:

- ✓ Cross-browser testing (Chrome,firefox, microsoft Edge)

Usability Testing:

- ✓ UI/UX evaluation
- ✓ Localization and translation testing.

Regression Testing:

- ✓ Regression testing between features.

○ Out of Scope:

- ✕ API Development
- ✕ Mobile Application
- ✕ Security Testing and Performance Testing

❖ Project Timeline

Task	Start Date	End Date	Duration	Responsible Team Member
Planning & Analysis	20, February 2025	24, February 2025	4 Day	TEAM 1
Test Case Design	26, February 2025	29, February 2025	3 Days	TEAM 1
Functional Testing	15, March 2025	22, March 2025	7 Days	TEAM 1
Regression Testing	1,April 2025	2,April 2025	Day	TEAM 1
Reporting & Documentation	3,April 2025	5,April 2025	2 Days	TEAM 1

Final Sign-off	6,April 2025	7,April 2025	Day	TEAM 1

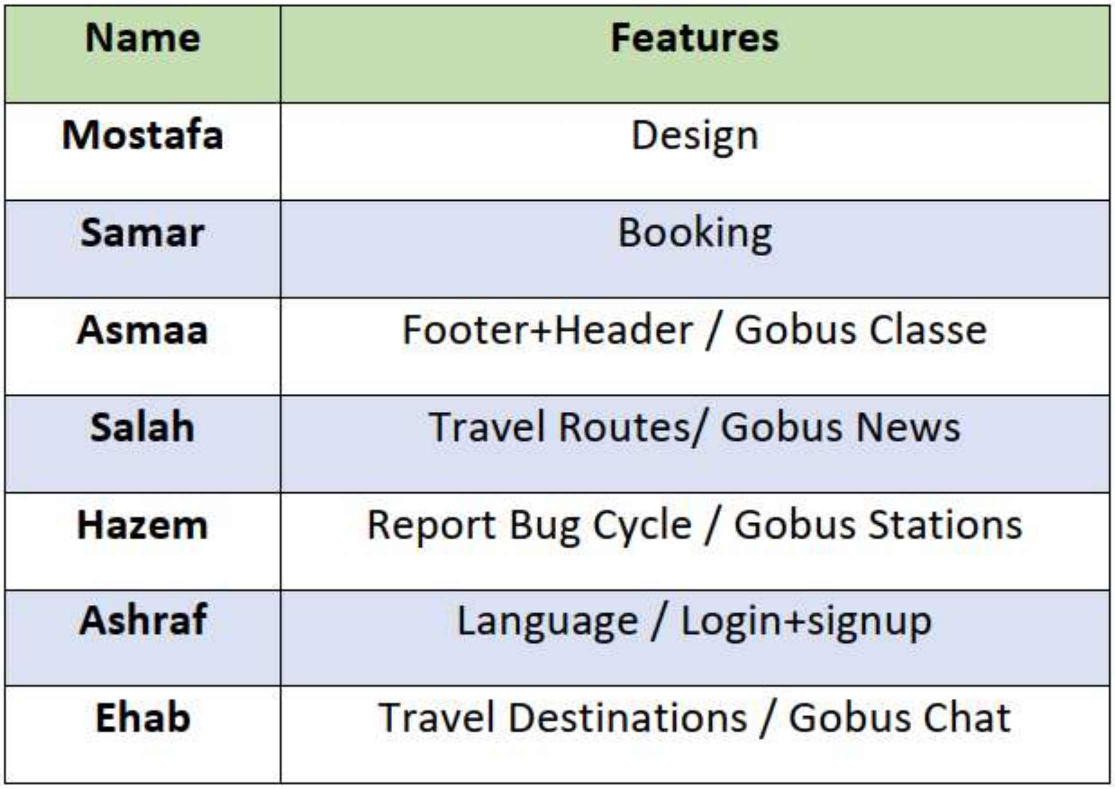
Milestones

1. **Project Kick-off** (February20, 2025)
2. **Test Plan** (February22, 2025)
3. **Test Case Completion** (February26, 2025)
4. **Functional Testing Completion** (March22, 2025)
5. **Regression Testing Completion** (April 2, 2025)
6. **QA Sign-off and Reporting** (April 7, 2025)

Risk

Risk Category	Potential Risk	Impact	Mitigation Strategy
Functional Bugs	Critical bugs in ticket booking,payment processing	High	Perform thorough functional & regression.testing before deployment
Compatibility Issues	Website or app not working on some devices/browsers	Medium	Perform cross-browser and cross-device testing
Third-Party Failures	Payment gateway (Fawry,Visa) downtime	High	Implement retry mechanisms and error handling
User.Experience Issues	Confusing navigation or booking process	Medium	Conduct usability testing and gather user feedback

Task Assignment & Roles



Name	Features
Mostafa	Design
Samar	Booking
Asmaa	Footer+Header / Gobus Classe
Salah	Travel Routes/ Gobus News
Hazem	Report Bug Cycle / Gobus Stations
Ashraf	Language / Login+signup
Ehab	Travel Destinations / Gobus Chat

Problem Statement:

1. Travelers often face challenges in finding and booking bus tickets efficiently due to:
2. Lack of real-time information on bus availability and schedules.
3. Inconvenient payment options and unreliable booking systems.
4. Difficulty in managing bookings (e.g., cancellations, refunds).
5. Poor user experience on mobile devices or during peak traffic.

Deliverables

- **Test Plan Document**
- **Test Cases and Test Scenarios**
- **Functional Test Reports**
- **Defect Report**
- **Final Test Execution Report**

Test Environment

-Windows 10 (Chrome).

Entry Criteria

1. Basic Functional Availability:The website should be accessible, and key functionalities (ticket booking, schedule display, payment process) should be ready for testing.
2. Defined Test Scope and Objectives:
   - Required tools (browser developer tools, bug-tracking software).

Exit Criteria

1. Sufficient Coverage of Key Functional Areas**:**
   - Major functionalities have been explored, and cases have been identified.
2. No Critical Defects Blocking Testing:
   - Any critical issues discovered have been documented and addressed, allowing further testing.
3. Documented Findings and Insights:
   - Observations, issues, and potential usability improvements have been recorded in a report or bug-tracking tool.

REQUIREMENTS GATHERING

User Stories

- **Search for Buses:**

As a traveler, I want to search for buses by entering my departure city, destination, and travel date so that I can find available buses.

- **Book a Ticket:**

As a traveler, I want to select a bus, choose my seat, and book a ticket so that I can secure my travel plans.

- **View Booking Details:**

As a traveler, I want to view my booking details (e.g., seat number, departure time) so that I can confirm my travel arrangements.

- **Cancel a Booking:**

As a traveler, I want to cancel my booking so that I can get a refund if my plans change.

- **Payment Options:**

As a traveler, I want to pay for my ticket using multiple payment methods (credit card, debit card, etc.) so that I can choose the most convenient option.

- **Track Bus Status:**

As a traveler, I want to track the real-time status of my bus so that I can plan my journey accordingly.

- **Create an Account:**

As a traveler, I want to create an account so that I can save my personal details and view my booking history.

- **Contact Customer Support:**

As a traveler, I want to contact customer support so that I can resolve any issues with my booking.

Use Cases

Use Cases	Actor	Preconditions	Main Flow:	Postconditions
Use Case 1: Search for Buses	**Traveler**	The user is on the homepage.	1.The user enters the departure city, destination, and travel date. 2.The system validates the input and displays a list of available buses. 3.The user applies filters (bus type, departure	The user views the details of the selected bus.

			time) to narrow down the results. 4.The system updates the list of buses based on the applied filters. 5.The user selects a bus to view its details.	
Use Case 2: Book a Ticket	**Traveler**	The user has selected a bus and is logged in	1.The user selects a seat from the available options. 2.The system displays the total fare and prompts the user to enter passenger details. 3.The user enters	The ticket is booked, and the user receives a confirmation.

			passenger details (name, age) and proceeds to payment. 4.The system displays available payment options. 5.The user selects a payment method and completes the payment. 6.The system generates a ticket and sends a confirmation email/SMS.	
Use Case 3: Cancel a Booking	**Traveler**	The user has a booked ticket and is logged in.	1.The user navigates to the "My Bookings" page.	The booking is canceled, and the user receives a refund (if applicable).

			2.The system displays a list of the user’s bookings. 3.The user selects the booking they want to cancel. 4.The system displays the cancellation policy and refund amount. 5.The user confirms the cancellation. 6.The system processes the cancellation. 7.The user receives a cancellation confirmation email/SMS.	

Use Case 4: Track Bus Status	**Traveler**	The user has a booked ticket.	1.The user navigates to the "My Bookings" page. 2.The system displays a list of the user's bookings. 3.The user selects the booking they want to track. 4.The system displays the real-time status of the bus (On Time,Delayed).	The user views the current status of their bus.
Contact Customer Support	**Traveler**	The user is on the website.	1.The user navigates to the "Contact Us" page. 2.The system displays a form for the user to	The user's query is submitted, and they receive a confirmation.

			enter their query. 3.The user enters their name, email, and query. 4.The user submits the form. 5.The system sends the query to customer support and displays a confirmation message.	

Scenarios

Scenario 1: Successful Booking

The user searches for buses, selects a bus, chooses a seat, completes the payment, and receives a confirmation.

Scenario 2: No Buses Available

The user searches for buses but finds no available buses for the selected route and date.

Scenario 3: Payment Failure

The user attempts to pay for a ticket, but the payment fails due to insufficient funds or network issues.

Scenario 4: Partial Refund on Cancellation

The user cancels a booking, and the system refunds a partial amount based on the cancellation policy.

Scenario 5: Bus Delayed

The user tracks their bus and finds that it is delayed by 30 minutes.

Functional Requirements

1. User Management

- Users can create an account by providing personal details (name, email, phone number, password).
- Users can log in using their email and password.
- Users can reset their password by email
- Users can update their profile information (name, contact details).

2. Bus Search and Booking

- Users can select for buses by entering departure city, destination, and travel date.
- Users can filter search results by bus type , departure time, and price.
- Users can view detailed information about a bus (amenities(images) , seat map).
- Users can select seats and book tickets for one or more passengers.
- Users can apply discount codes (promo).

3. Payment Processing

- Users can pay for tickets using multiple payment methods (credit card, Aman,pasata and fawry).

4. Booking Management

- Users can view their booking history and current bookings.
- Users can cancel a booking.

5. Notifications

- Users should receive email/SMS notifications for booking confirmation, payment success and cancellation.

Non-Functional Requirements

1. Usability

- The system should have a responsive design that works seamlessly on desktop.
- The user interface should be easy to navigate, with clear instructions.
- The system should support multiple languages (EN) for international users.

2. Reliability

- The system should automatically recover from failures (server crashes) within 5 minutes.

4. **Compatibility** The system should be compatible with major browsers (Chrome,fire fox and Microsoft Edge).

Dependency

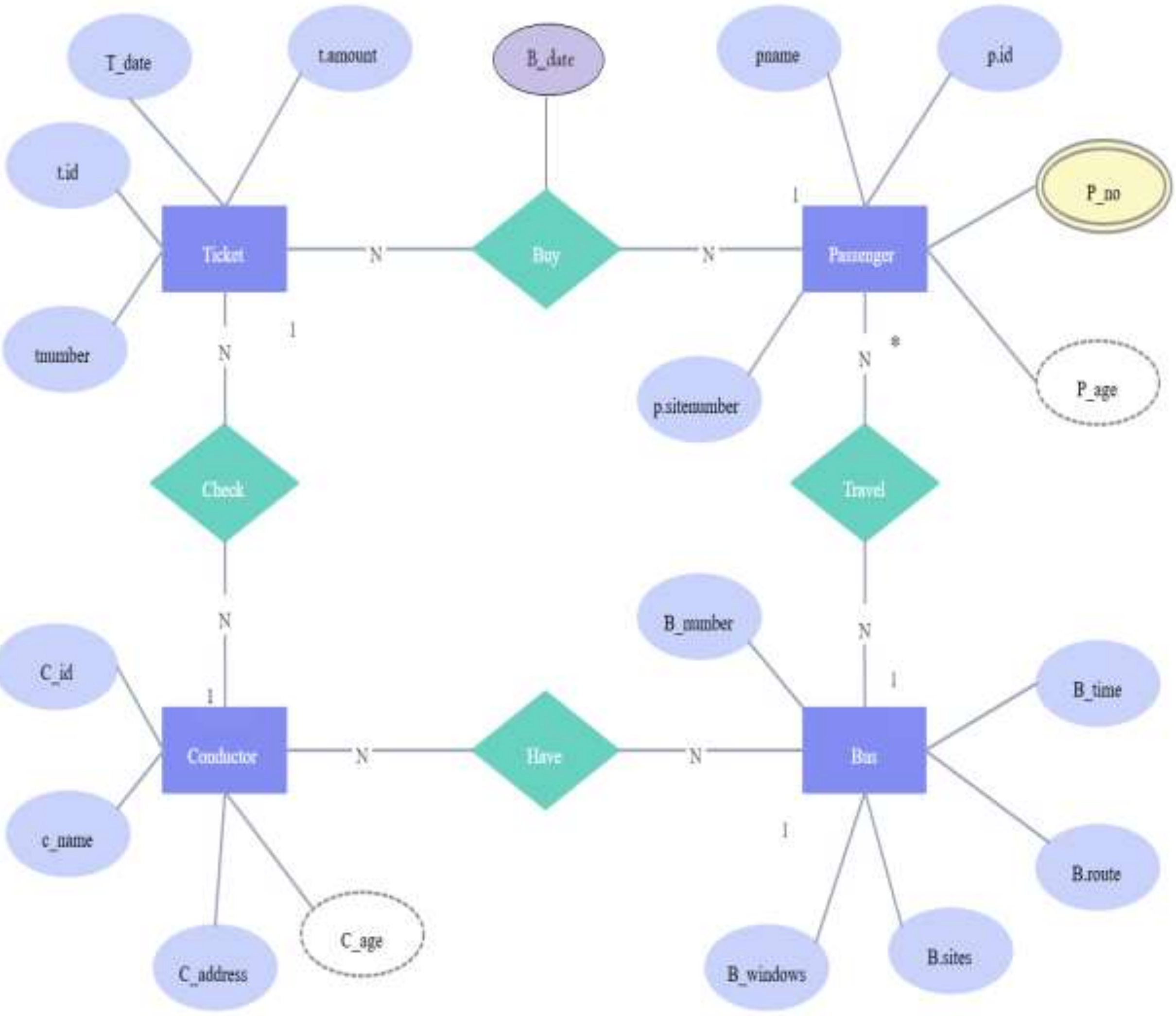
Features ID	Features Name	Priority	Dependent on	Execution Order	Notes
F_001	Login(User Authentication)	High	NA	1	Essential for secure access to services
F_002	Booking Cycle	High	F_001	2	Allows user to book tickets online;requires user Authentication
F_003	Seat Selection	High	F_002	3	Enabies users to choose their

					perferred seats durind booking
F_004	Multi language support	Medium	none	**4**	provide a diverse user base by offering English language option
F_005	User Profile Management	Medium	F_001	5	Allowes users to manage personal information and view booking history
F_006	Go Bus Routes	High	none	6	Provides users with information on bus rotes .
F_007	Bus situation	Medium	None	7	user can check if a Go bus Stations at their location before making a reservation.
F_008	Header &Footer	Medium	All test cases in this feature don’t depend on any other feature but “contact **us” denpends on NA&F_002**	8	Contains : • (Company • FAQ • Terms • Policy • Contact us
F_009	Travel Destination	Medium	**None**	9	User can choose travel destination and book through it.

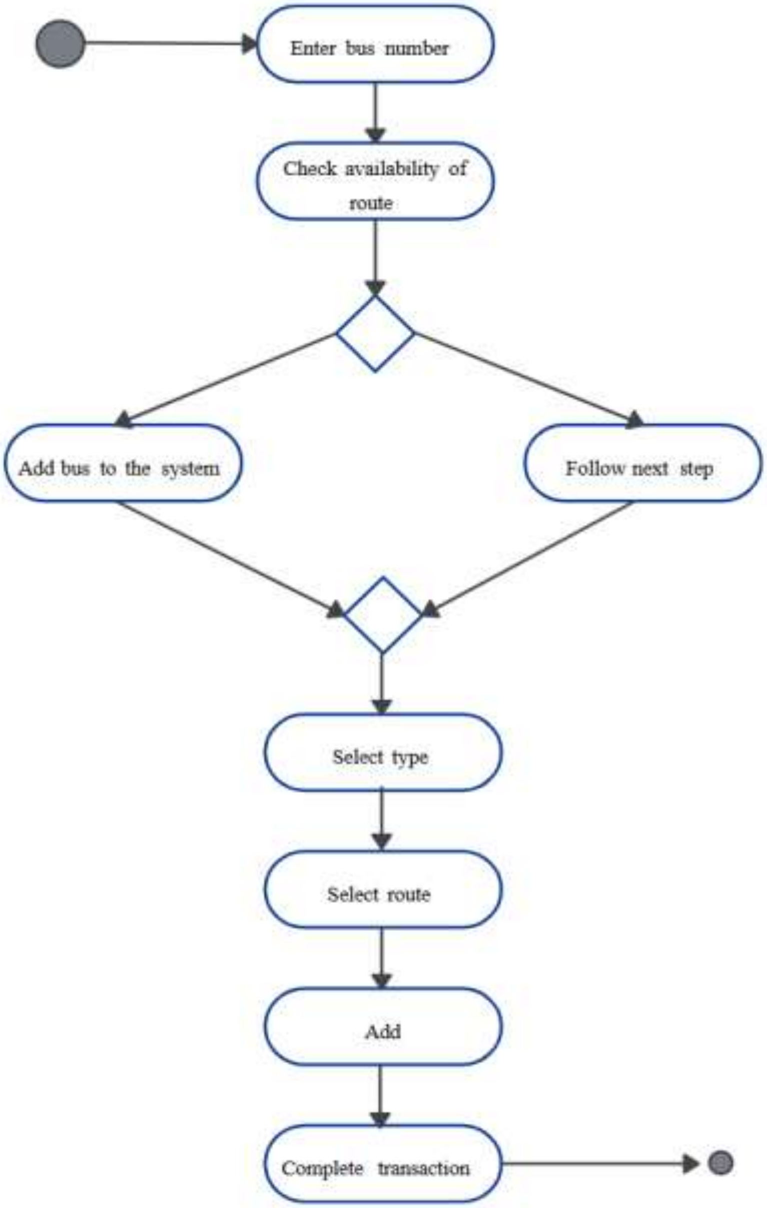
F_010	Customer Support chat	Low	None	10	Provides real_assistance to users; enhence user experience
F_011	Go Bus News	Low	None	11	All news about gobus
F_012	Report a bug	Low	None	12	If you discover a bug on the site, contact us to improve the user experience.

DESIGN

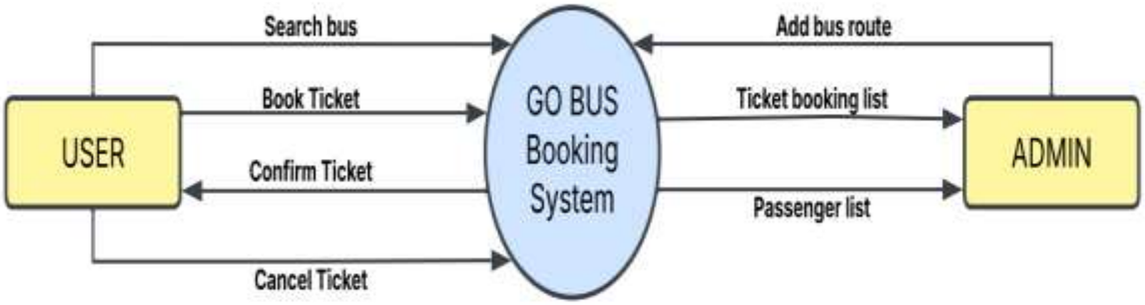
EDR


T_date
t.amount
B_date
pname
p.id
t.id
1
P_no
Ticket
N
Buy
N
Passenger
tnumber
1
N
*
N
p.sitenumber
P_age
Check
Travel
N
B_number
N
C_id
1
B_time
1
Conductor
N
Have
N
Bus
c_name
1
B.route
C_age
C_address
B_windows
B.sites

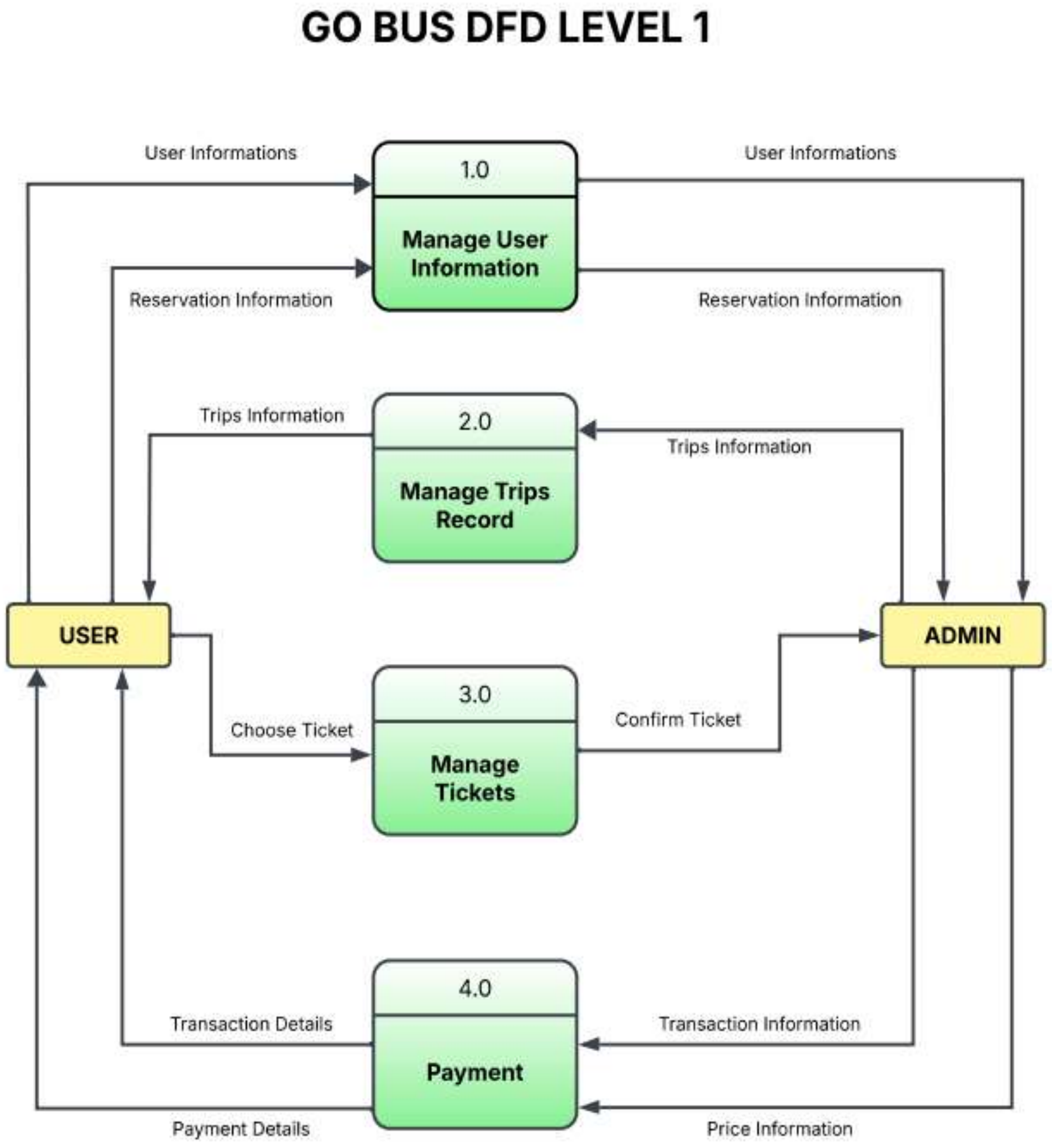
- **Activity**


Enter bus number
Check availability of route
Add bus to the system
Follow next step
Select type
Select route
Add
Complete transaction

- **CONTEXT(DFD_0)**

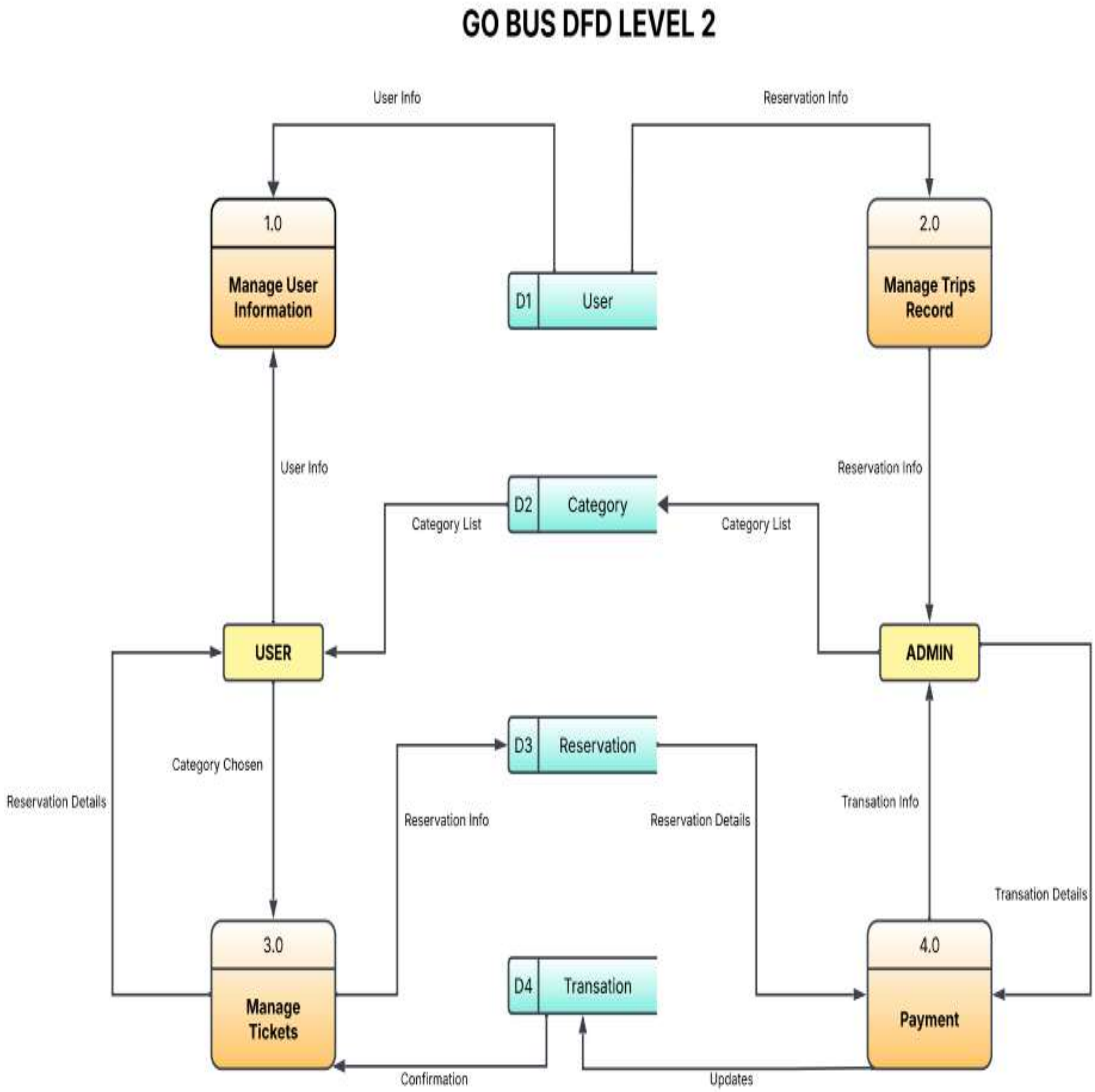


- ## DFD_1

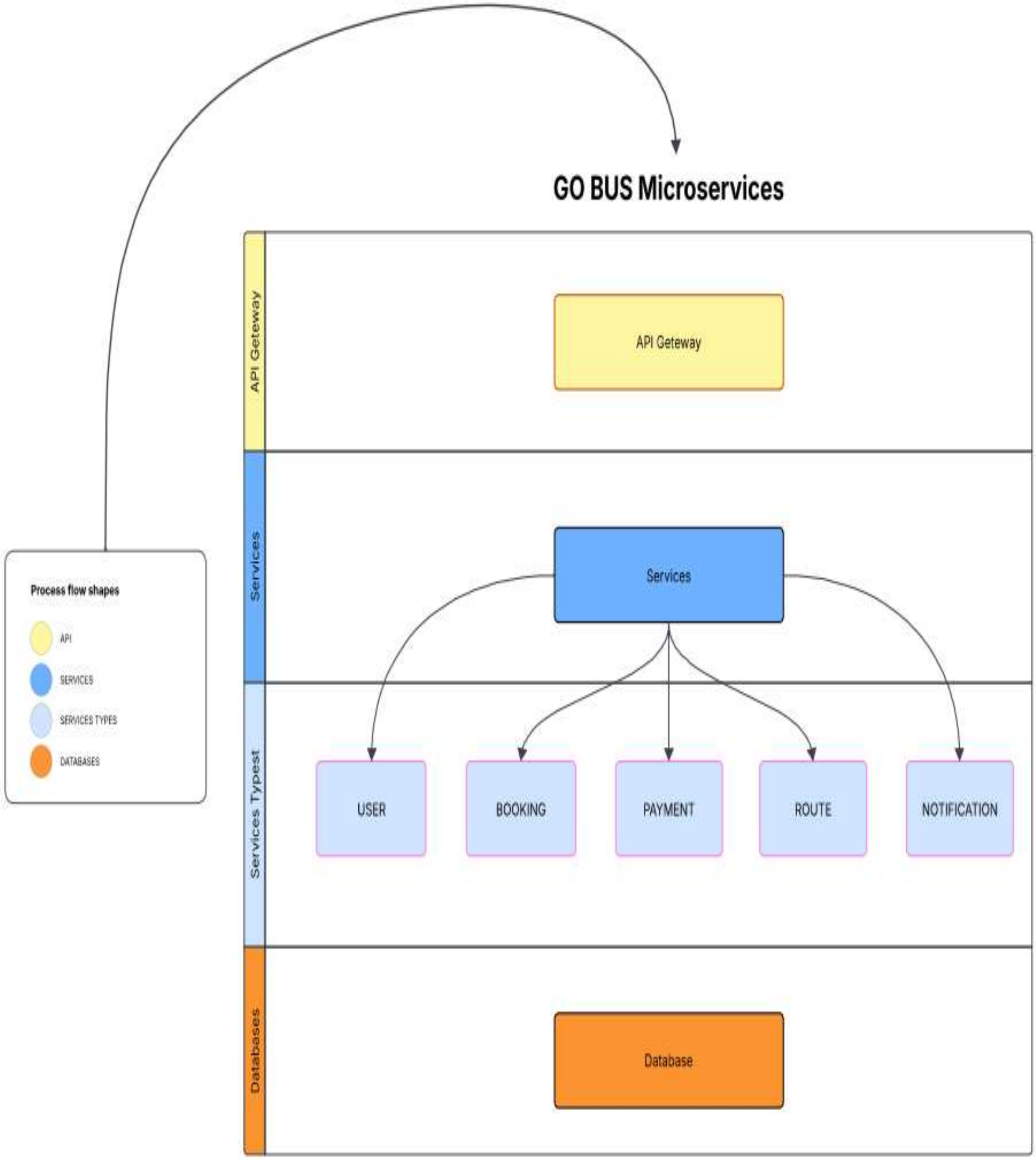


GO BUS DFD LEVEL 1

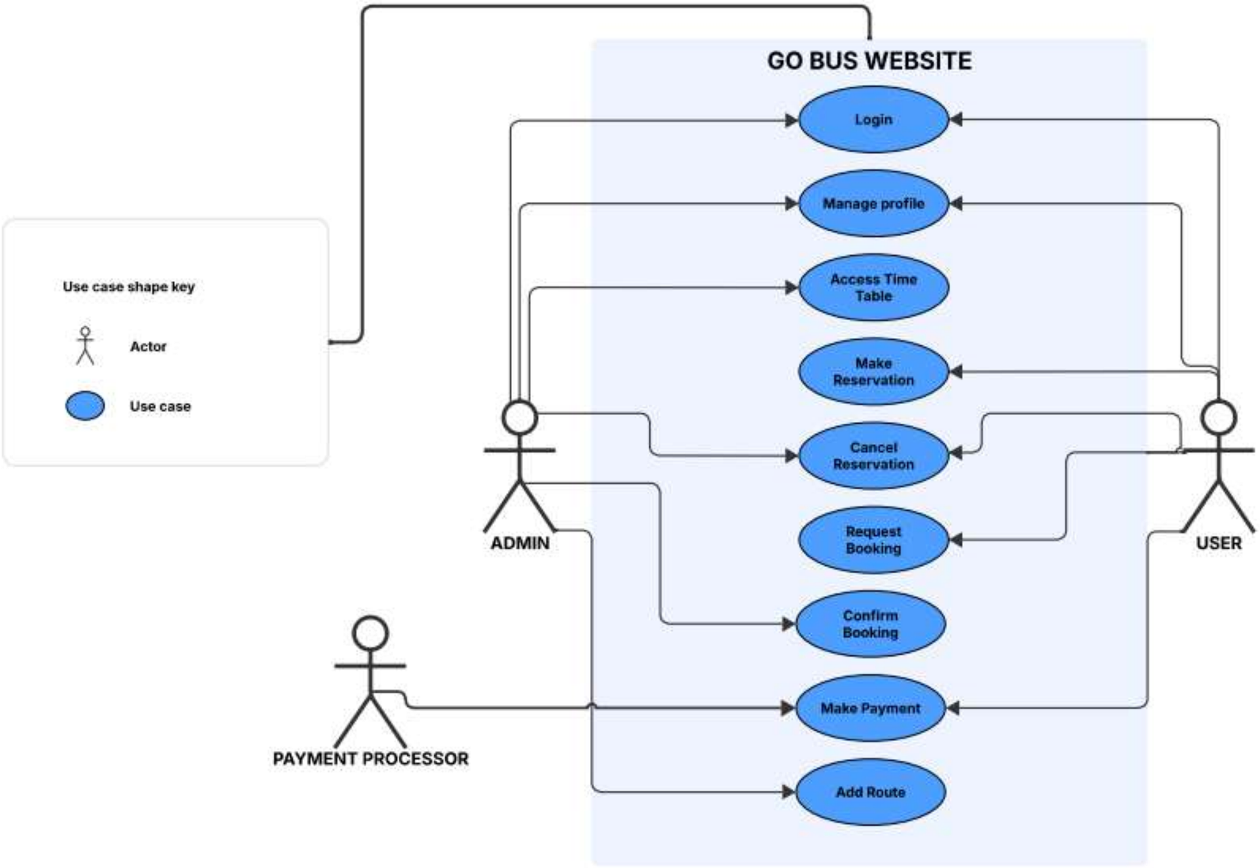
- ## DFD_2


GO BUS DFD LEVEL 2
User Info
Reservation Info
1.0
Manage User Information
D1
User
2.0
Manage Trips Record
User Info
Reservation Info
D2
Category
Category List
Category List
USER
ADMIN
D3
Reservation
Category Chosen
Reservation Details
Reservation Info
Reservation Details
Transation Info
Transation Details
3.0
Manage Tickets
D4
Transation
4.0
Payment
Confirmation
Updates

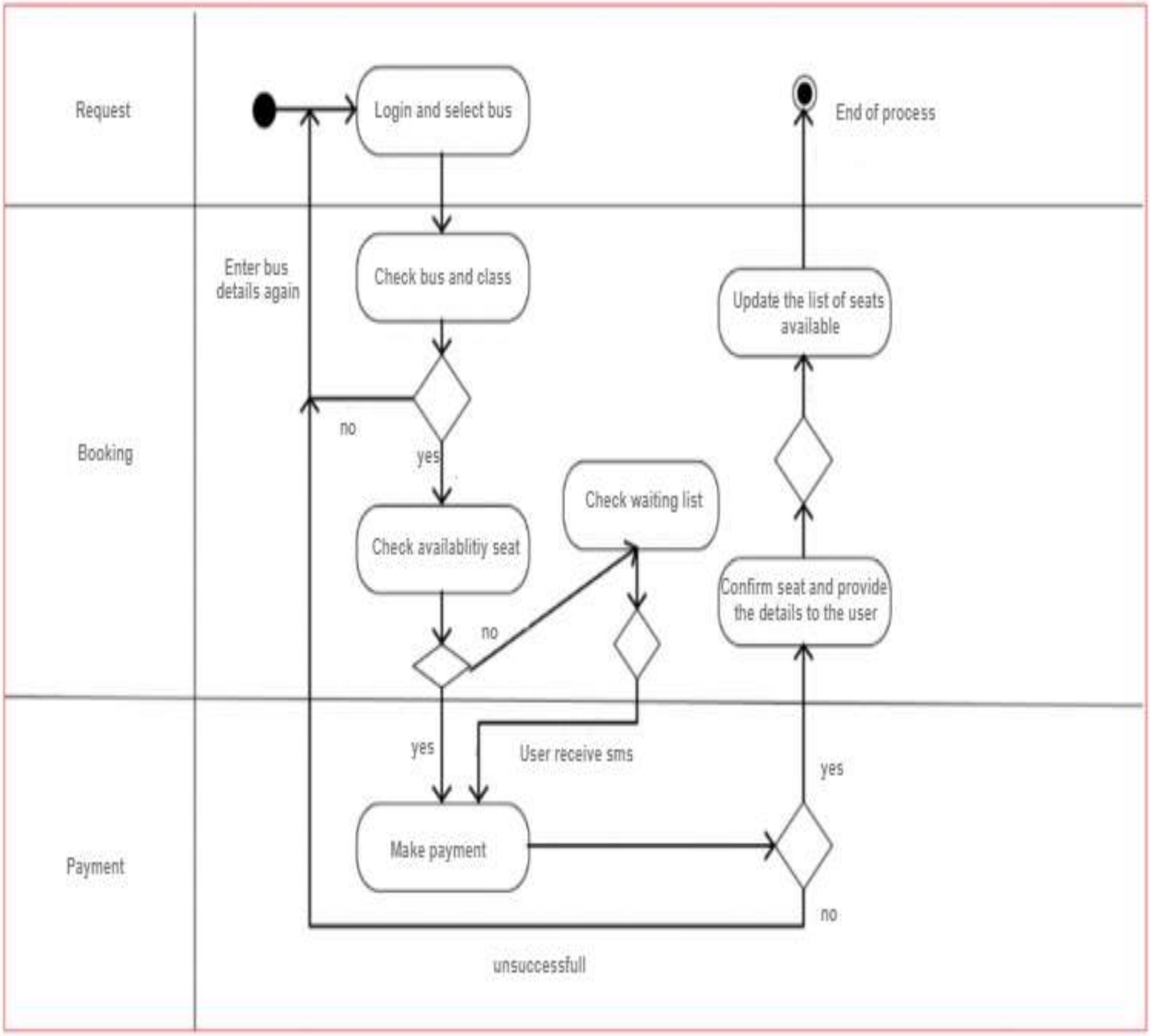
- ## MICROSERVICES



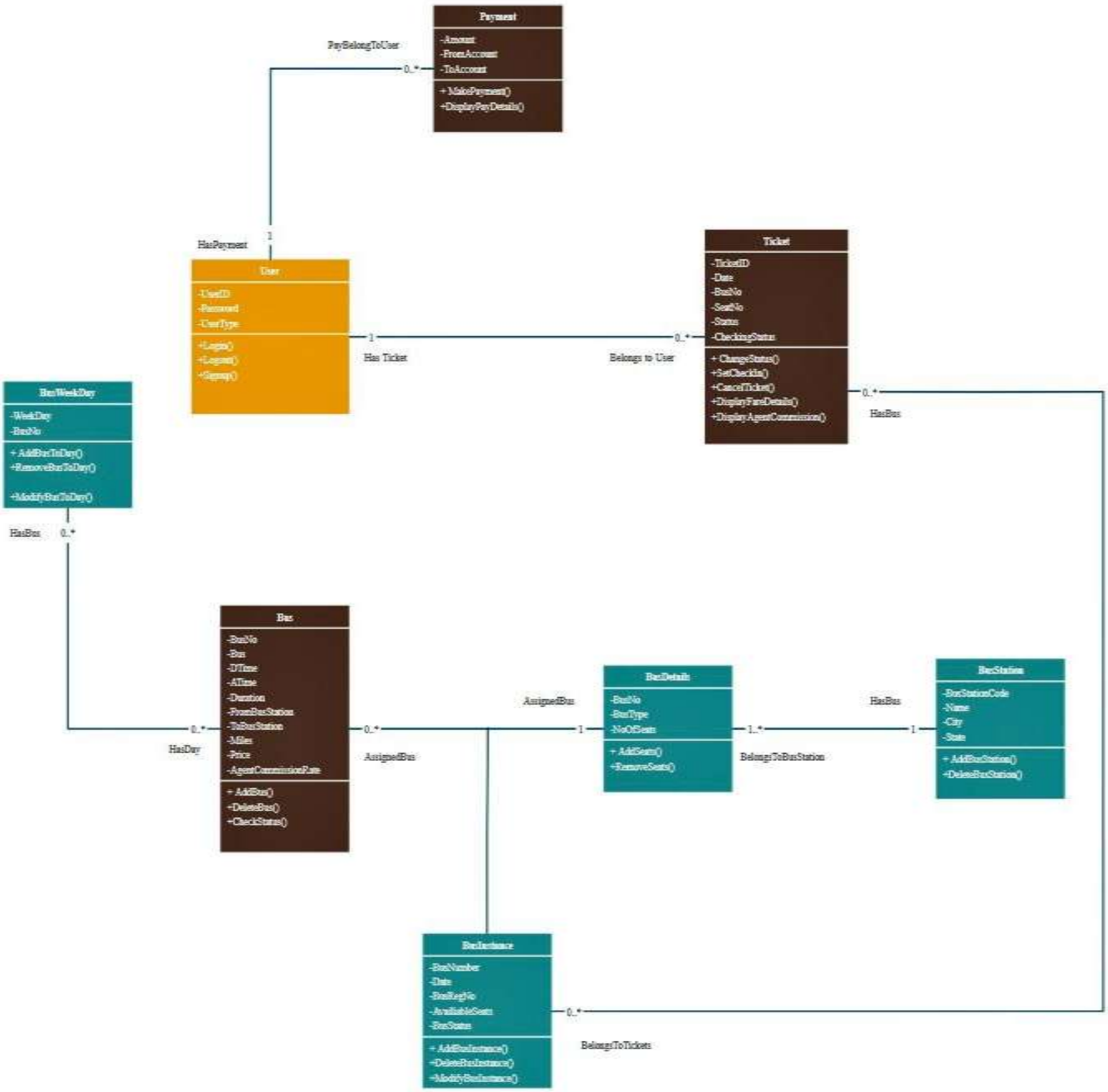
- **USE_CASE**


GO BUS WEBSITE
Login
Manage profile
Access Time Table
Make Reservation
Cancel Reservation
Request Booking
Confirm Booking
Make Payment
Add Route
ADMIN
USER
PAYMENT PROCESSOR
Use case shape key
Actor
Use case

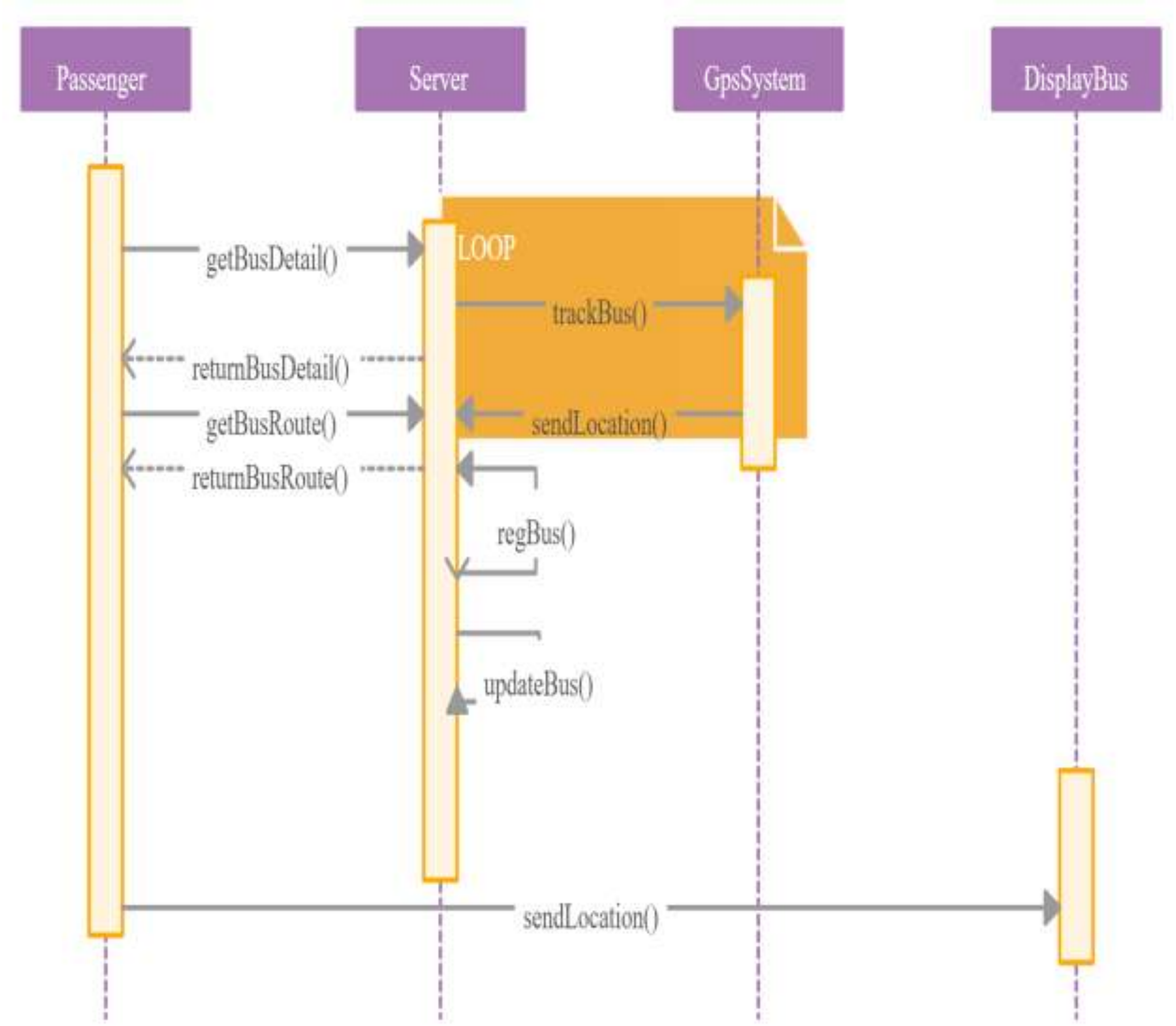
- **State**


Request
Login and select bus
End of process
Booking
Enter bus details again
Check bus and class
Update the list of seats available
no
yes
Check waiting list
Check availablitiy seat
Confirm seat and provide the details to the user
no
Payment
yes
User receive sms
yes
Make payment
no
unsuccessfull

- CLASS


Payment
-Amount
-FromAccount
-ToAccount
+ MakePayment()
+DisplayPayDetails()
PayBelongToUser
0..*
1
HasPayment
User
-UserID
-Password
-UserType
+Login()
+Logout()
+Signup()
1
Has Ticket
0..*
Belongs to User
Ticket
-TicketID
-Date
-BusNo
-SeatNo
-Status
-CheckingStatus
+ ChangeStatus()
+SetCheckIn()
+CancelTicket()
+DisplayFareDetails()
+DisplayAgentCommission()
0..*
HasBus
BusWeekDay
-WeekDay
-BusNo
+ AddBusToDay()
+RemoveBusToDay()
+ModifyBusToDay()
HasBus
0..*
Bus
-BusNo
-Bus
-DTime
-ATime
-Duration
-FromBusStation
-ToBusStation
-Miles
-Price
-AgentCommissionRate
+ AddBus()
+DeleteBus()
+CheckStatus()
0..*
HasDay
0..*
AssignedBus
AssignedBus
1
BusDetails
-BusNo
-BusType
-NoOfSeats
+ AddSeats()
+RemoveSeats()
1..*
BelongsToBusStation
HasBus
1
BusStation
-BusStationCode
-Name
-City
-State
+ AddBusStation()
+DeleteBusStation()
BusInstance
-BusNumber
-Date
-BusRegNo
-AvailableSeats
-BusStatus
+ AddBusInstance()
+DeleteBusInstance()
+ModifyBusInstance()
0..*
BelongsToTickets

- ## Sequence


Passenger
Server
GpsSystem
DisplayBus
LOOP
getBusDetail()
trackBus()
returnBusDetail()
getBusRoute()
sendLocation()
returnBusRoute()
regBus()
updateBus()
sendLocation()

IMPLEMENTATION

As part of the long-term testing strategy, we have initiated automation testing using **Selenium WebDriver**, **TestNG**, and **Java**. This approach ensures improved efficiency, reusability, and early detection of regression issues across multiple releases.

Objectives of Automation

Automate the repetitive test scenarios (e.g., login, booking, payment).

Tools & Technologies Used:

□ **Language:** Java

□ **Automation Framework:** Selenium WebDriver

□ **Test Framework:** TestNG

□ **Build Tool:** Maven

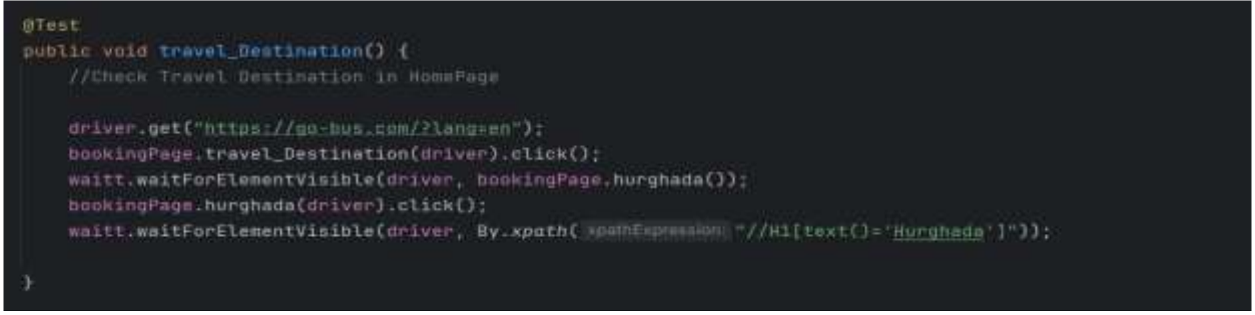
□ **IDE:** IntelliJ IDEA

Scope of Automation:

Test cases were considered for automation:

- Ticket Booking flow (Search → Select → Book).
- Login & logout & Change Password & Manage Profile.
- Travel Destination.
- Go Bus Station

Sample Code (Travel Destination Test)

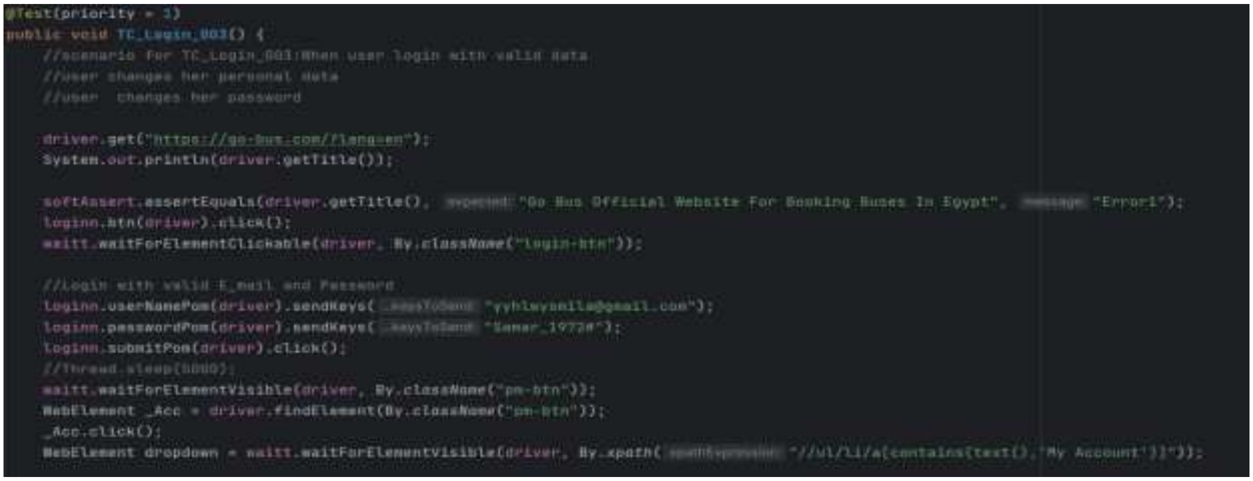


```java
@Test
public void travel_Destination() {
    //Check Travel Destination in HomePage

    driver.get("https://go-bus.com/?lang=en");
    bookingPage.travel_Destination(driver).click();
    waitt.waitForElementVisible(driver, bookingPage.hurghada());
    bookingPage.hurghada(driver).click();
    waitt.waitForElementVisible(driver, By.xpath( xpathExpression: "//H1[text()='Hurghada']"));

}
```

Sample Code (Login Test)



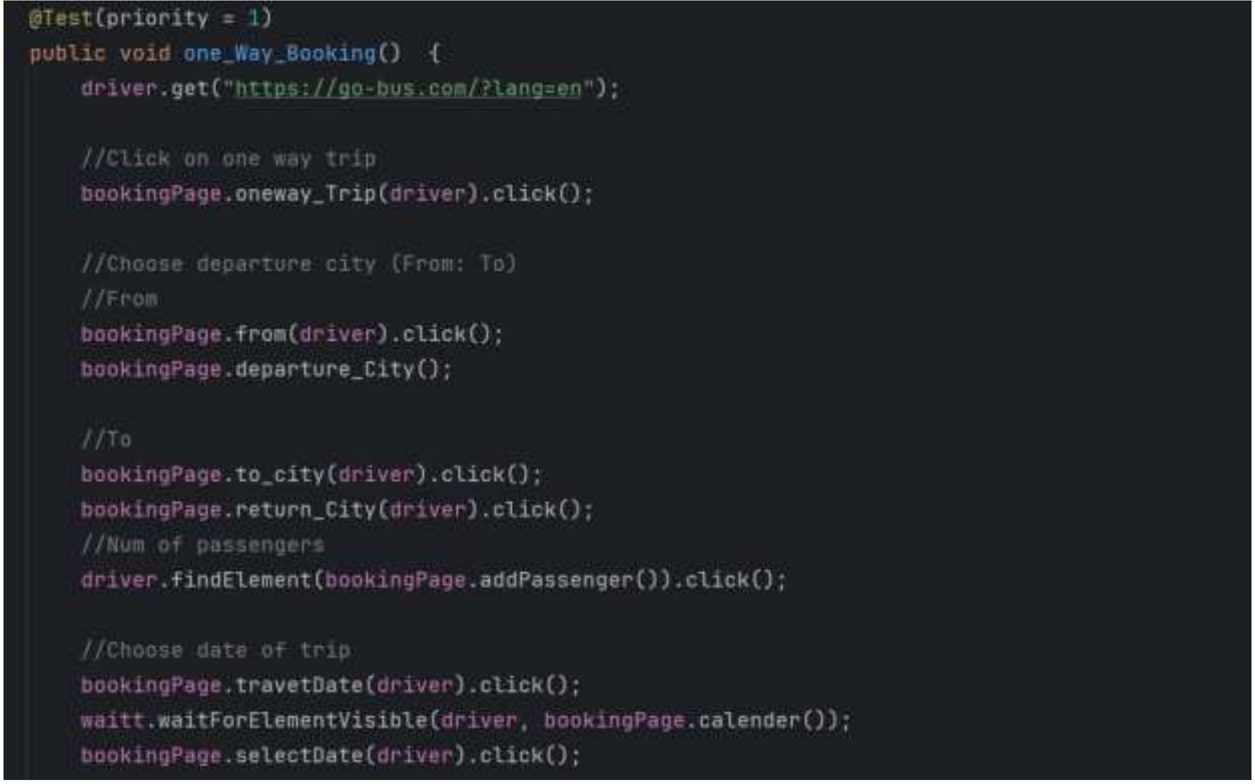
```java
@Test(priority = 3)
public void TC_Login_003() {
    //scenario for TC_Login_003:When user login with valid data
    //user changes her personal data
    //user  changes her password

    driver.get("https://go-bus.com/?lang=en");
    System.out.println(driver.getTitle());

    softAssert.assertEquals(driver.getTitle(),  expected: "Go Bus Official Website For Booking Buses In Egypt",  message: "Error1");
    loginn.btn(driver).click();
    waitt.waitForElementClickable(driver, By.className("login-btn"));

    //Login with valid E_mail and Password
    loginn.userNamePom(driver).sendKeys( ...keysToSend: "yyhlwysmila@gmail.com");
    loginn.passwordPom(driver).sendKeys( ...keysToSend: "Samar_1972#");
    loginn.submitPom(driver).click();
    //Thread.sleep(5000);
    waitt.waitForElementVisible(driver, By.className("pm-btn"));
    WebElement _Acc = driver.findElement(By.className("pm-btn"));
    _Acc.click();
    WebElement dropdown = waitt.waitForElementVisible(driver, By.xpath( xpathExpression: "//ul/li/a[contains(text(),'My Account')]"));
```

Sample Code (Booking Test)



```
@Test(priority = 1)
public void one_Way_Booking()  {
    driver.get("https://go-bus.com/?lang=en");

    //Click on one way trip
    bookingPage.oneway_Trip(driver).click();

    //Choose departure city (From: To)
    //From
    bookingPage.from(driver).click();
    bookingPage.departure_City();

    //To
    bookingPage.to_city(driver).click();
    bookingPage.return_City(driver).click();
    //Num of passengers
    driver.findElement(bookingPage.addPassenger()).click();

    //Choose date of trip
    bookingPage.travetDate(driver).click();
    waitt.waitForElementVisible(driver, bookingPage.calender());
    bookingPage.selectDate(driver).click();
```

Final Report

1. Project Summary

The objective of this testing cycle was to ensure the Go Bus online platform (https://gobus.com/) offers a smooth, secure, and reliable booking experience. Testing covered key modules including ticket booking, seat selection, payments, registration, customer support, and homepage components. All tests aimed to replicate real-world business scenarios to validate functionality.

2. Test Objectives

- Validate the core functionalities: Booking, Searching Buses, Payment, Registration, Login, Seat Selection.
- Ensure critical paths such as booking flow and payment are robust.
- Verify UI elements like Header, Footer, Bus Routes, Bus Classes, Stations, and Destinations are working as expected.

3. Features is Tested

Module	Description

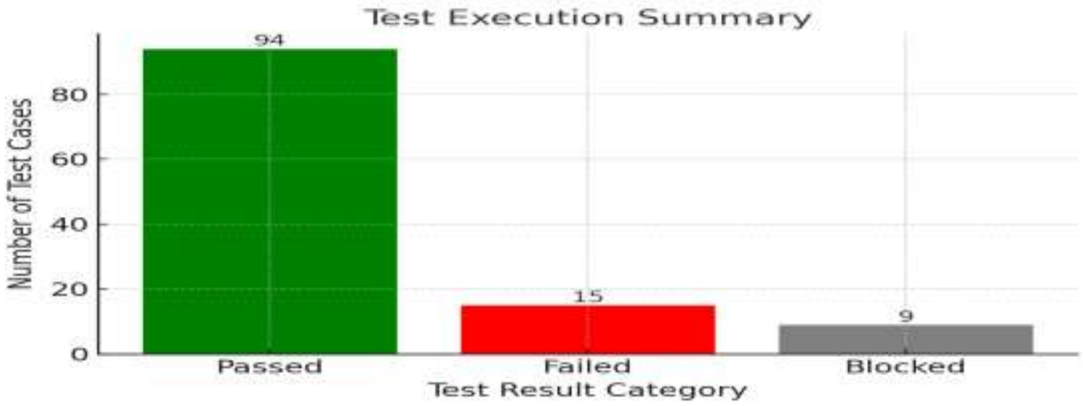
Booking	Users can search, select, remove, confirm trips, and view booking details
Login	Users can log in to complete bookings
Registration	Users can register using an email
Seat Selection & Cancellation	Users can select or cancel seats
Payment Gateway	Users can pay via Visa, MasterCard, Fawry, etc.
Search Buses	Users can find buses by city, destination, and date
Header/Footer	Contains company info, FAQ, policies, contact, and social icons
Go Bus Stations	Users can check nearby stations
Bus Routes	Availability of 20+ routes across Egypt
Travel Destination	Users can select destinations.

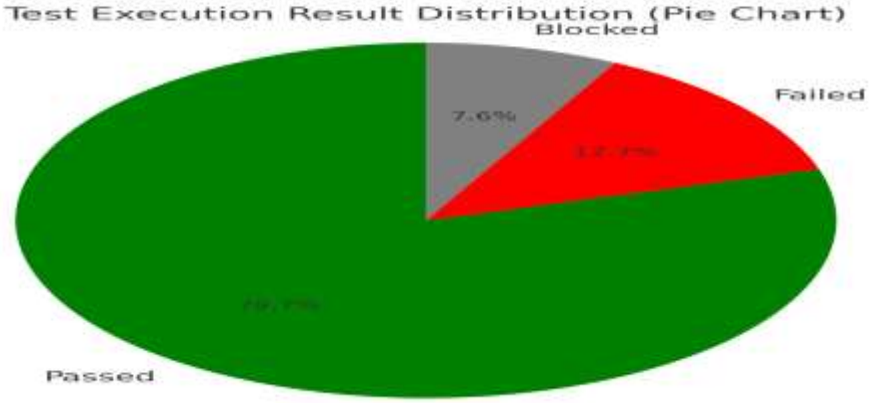
4. Out of Scope

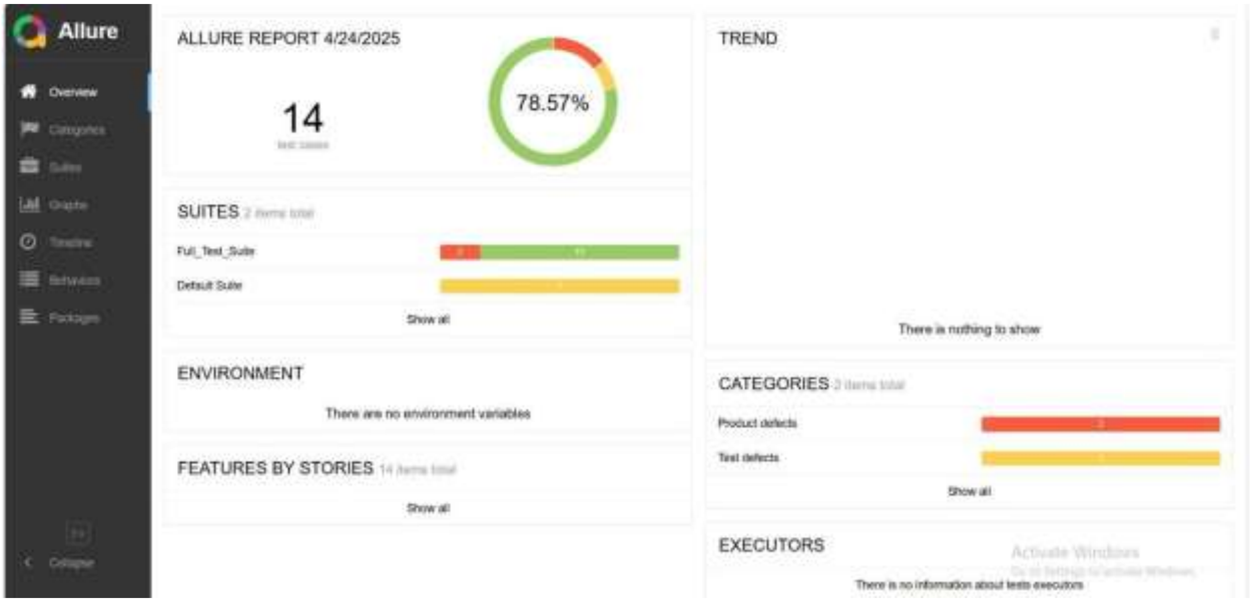
✗ API Development
✗ Mobile Application
✗ Security & Performance Testing

5. Test Execution Summary

- **Total Test Cases: 118**
- **Passed: 94**
- **Failed: 15**
- **Blocked: 9**
- **Pass Rate: 79.66%**


Test Execution Summary
Number of Test Cases
0
20
40
60
80
94
15
9
Passed
Failed
Blocked
Test Result Category


Test Execution Result Distribution (Pie Chart)
Blocked
Failed
Passed
7.6%
12.7%
79.7%



ALLURE REPORT 4/24/2025

14
test cases

78.57%

SUITES 2 items total

Full_Test_Suite

Default Suite

Show all

ENVIRONMENT

There are no environment variables

FEATURES BY STORIES 14 items total

Show all

TREND

There is nothing to show

CATEGORIES 3 items total

Product defects

Test defects

Show all

EXECUTORS

There is no information about tests executors

7. Recommendations

• The website experiences noticeable load times, especially during navigation between key pages. Performance optimization is recommended to enhance user experience.

• Certain features occasionally return temporary errors but recover after a refresh or retry. It's advised to investigate and resolve the root-causes to ensure stability.

• A general performance review is recommended, focusing on server response time, database optimization, and front-end efficiency.

8. Conclusion

Despite minor UI issues and recoverable errors, the GoBus.com platform meets the core functional requirements. All critical user flows—searching, booking, payment, and account creation—were validated successfully under typical usage scenario.