VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science Engineering and Information Systems

Fall Semester (2023-24)

PROJECT REPORT

Title:

- PharmEase System -

Course: Software Testing

Course Code: SWE2005

Faculty: Prof. Prasanna M

Slot: B2+TB2

Submitted by:
Eshal Shanoj
22MIS0087

SRS DOCUMENT

Objective

The primary objective of the "PharmEase System" project is to develop a user-friendly, Java-based web application with a MySQL database backend, aimed at revolutionizing the pharmaceutical industry. This system aims to provide patients with a seamless online platform for searching, ordering, and managing medications.

Introduction

Our Online Pharmacy System or PharmEase System is envisioned as a sophisticated and user- centric software solution that streamlines the process of ordering medicines for patients while providing essential tools for pharmacy administrators to efficiently manage orders and the medicine database. This system is intended to enhance the accessibility, efficiency, and accuracy of pharmaceutical services, promoting a seamless experience for both end-users and administrators.

Purpose

- **Convenient Medication Search**: Patients can search for medicines by name, condition, or category from the comfort of their homes. This eliminates the need for physically visiting multiple pharmacies to find specific medications.
- **Order Medications Online**: Users can select the required medicines, add them to their virtual shopping cart, and place orders online. This process allows patients to order medications 24/7, even outside of pharmacy business hours.
- **Reduced Waiting Times**: Patients can place orders online and have medications prepared in advance, reducing wait times at the pharmacy. This is particularly beneficial for patients with chronic conditions who require regular prescriptions.
- **Streamlined Order Processing**: Admins can view and manage orders efficiently.
- **Cost Savings**: By automating many tasks, reducing errors, and improving inventory management, pharmacies can realize cost savings in terms of reduced labor and improved resource allocation.

- **Competitive Advantage**: Having a user-friendly online platform can set a pharmacy apart from competitors and attract more customers looking for convenient pharmaceutical services.
- **Promotion of Telehealth:** The platform can be integrated with telehealth services, enabling users to consult with healthcare professionals online and receive prescriptions digitally.

Scope

The use of a computer-based management system for improving the efficiency of a pharmacy is needed and it is an essential part of any modern continuously evolving society. The scope of this project is limited to the activities of a pharmaceutical store which includes improving health outcomes, reduce hospital and long-term care admissions, enhance access and care in the Estate and surrounding communities and ensuring best use of resources.

Key Features and Functionalities:

For Users (/Patients)

1. **User Login**: The login page is the first page that users encounter when accessing the online pharmacy software. Users are required to enter their username and password to log into their account. The purpose of this page is to authenticate users and ensure that only authorized individuals can access the system

2. **User Registration**: The registration page is where new users can create an account for the online pharmacy software. Users need to provide their personal information, such as name, email address, and contact details, as well as create a username and password. The register page helps in creating a unique account for each customer, ensuring that their personal information is securely stored.

3. **Medicine Search**: The search medicine feature allows users to search for specific medicines or medical products available in the online pharmacy's inventory. Users can enter the name or description of the medicine they are looking for, and the software will display relevant results. This feature helps users quickly find the medicines they need, saving time and effort

4. **Add Items**: The add items feature enables users to add medicines or medical products to their shopping cart. Users can select the desired quantity of each item and add them to their cart for purchase. This feature allows users to easily keep track of the items they want to buy and helps in organizing their shopping experience.

5. **Orders Total**: The place orders total feature calculates the total cost of the items in the user's shopping cart. It considers factors such as the quantity of each item and any applicable discounts or promotions. Users can review the total cost before proceeding to checkout. This feature helps users understand the cost of their purchase and ensures transparency in the ordering process.

6. **Place Order**: The purchase feature allows users to complete their order by providing the delivery information with a cash on delivery option. They need to provide their shipping address for delivery. This feature facilitates the secure and convenient purchase of medicines from the online pharmacy.

For Admins (Pharmacy Administrators):

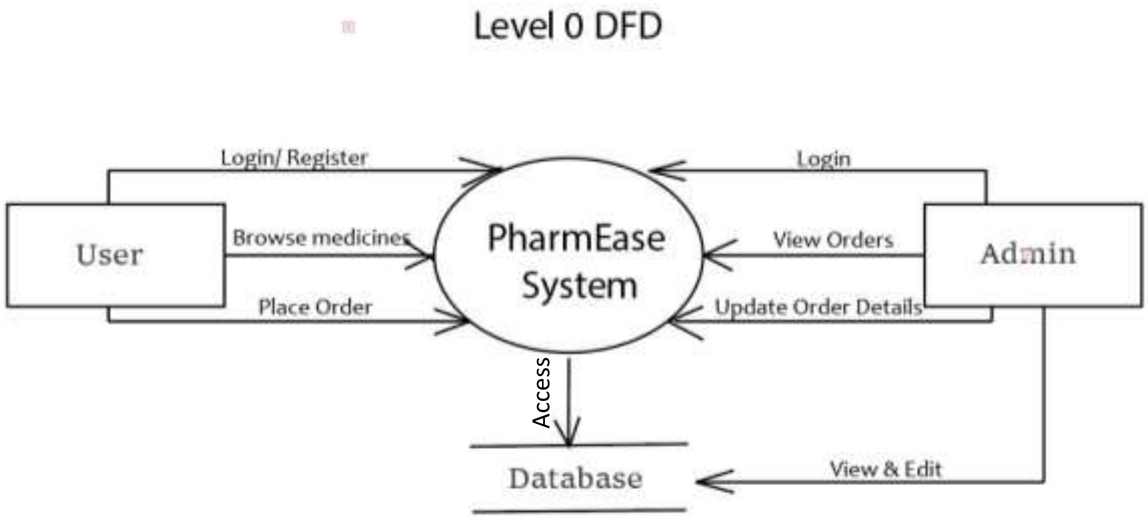
1. **Admin Login:** Administrators can securely access the system using authorized credentials.
2. **View Received Orders:** The system allows administrators to view and manage incoming orders efficiently.
3. **Medicine Database Management:** Administrators can manage the medicines database by adding, updating, or removing entries as needed, ensuring an up-to-date and accurate inventory.
4. **Update Order Details:** The admin can let the customer know the delivery updates.

Document Audience

This document is intended for various stakeholders involved in the "PharmEase System" project, including:

- Project Team: Developers, designers, and testers responsible for building and deploying the system.
- Project Managers: Individuals overseeing the project's execution and delivery.
- End-Users: Patients/ buyers who will interact with the system to order medicines.
- Administrators: Pharmacists and healthcare personnel responsible for managing the system

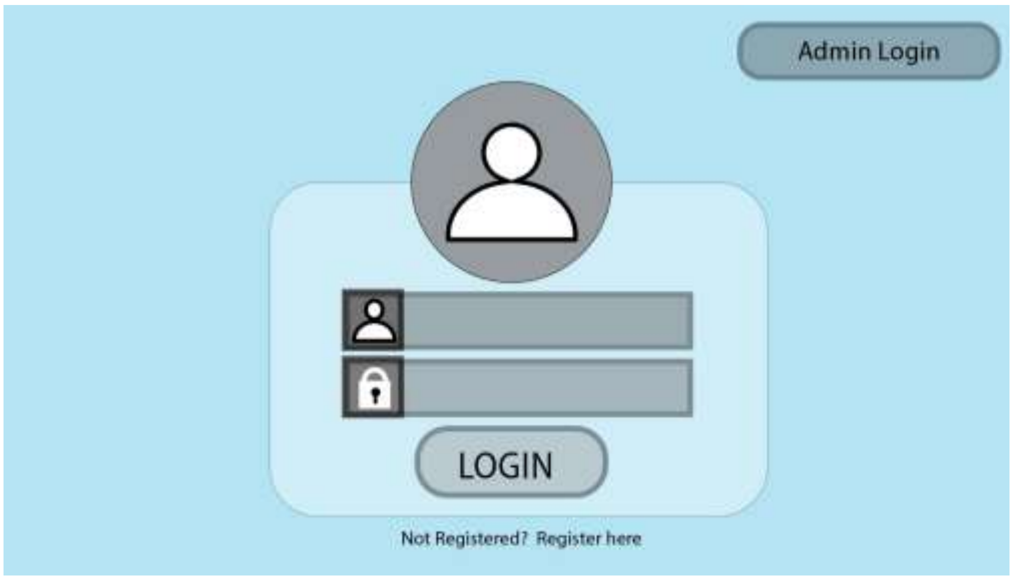
Level 0 DFD

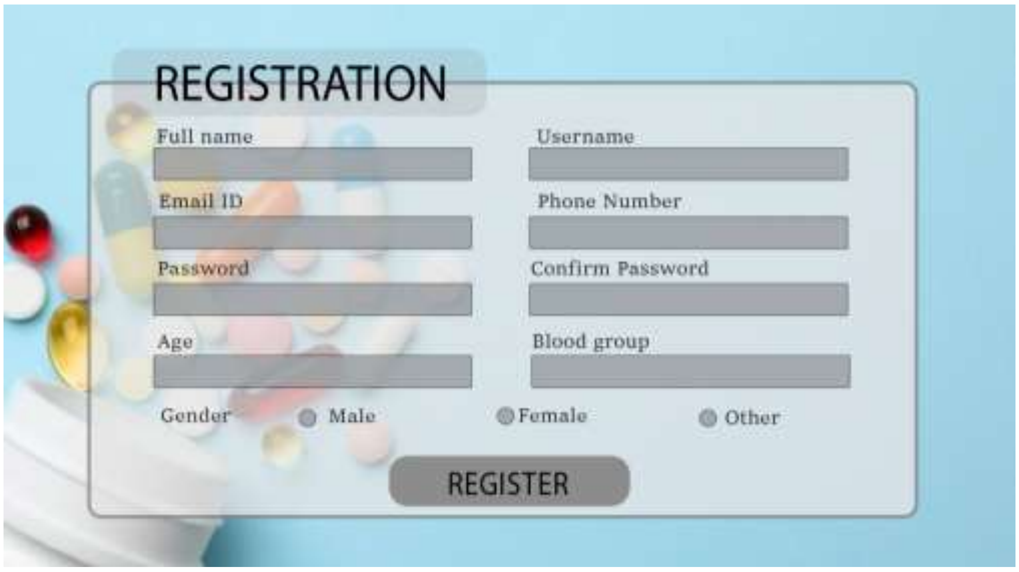


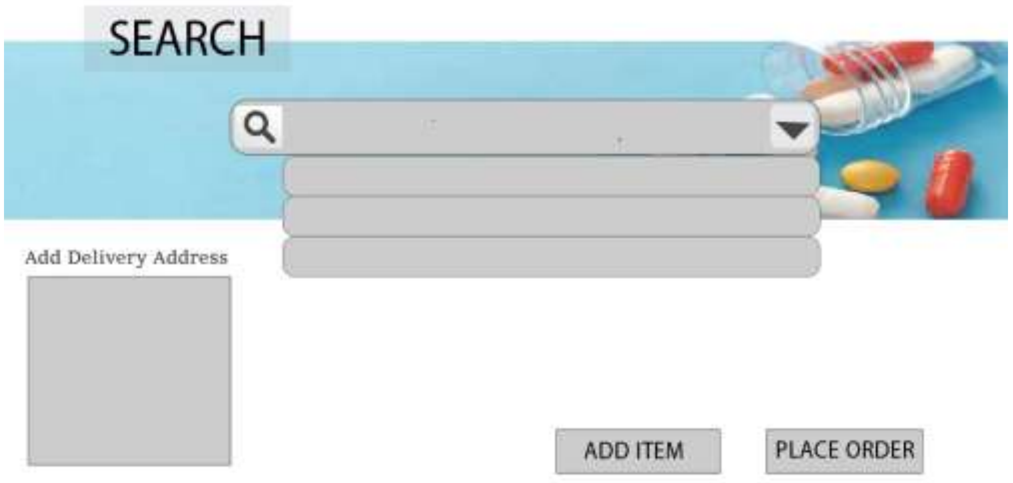
Conclusion

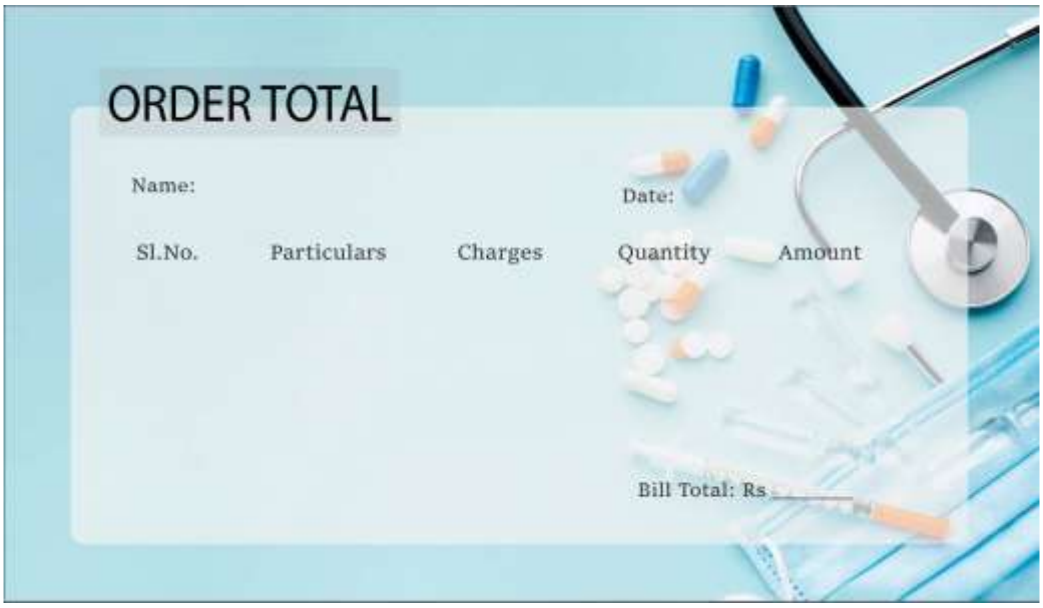
Our ‘PharmEase System’ strives to enhance medication accessibility, adherence, and the overall patient experience while optimizing pharmacy operations. We envision a healthcare ecosystem where patients can seamlessly access, order, and manage medications while enabling pharmacy administrators to efficiently oversee operations.

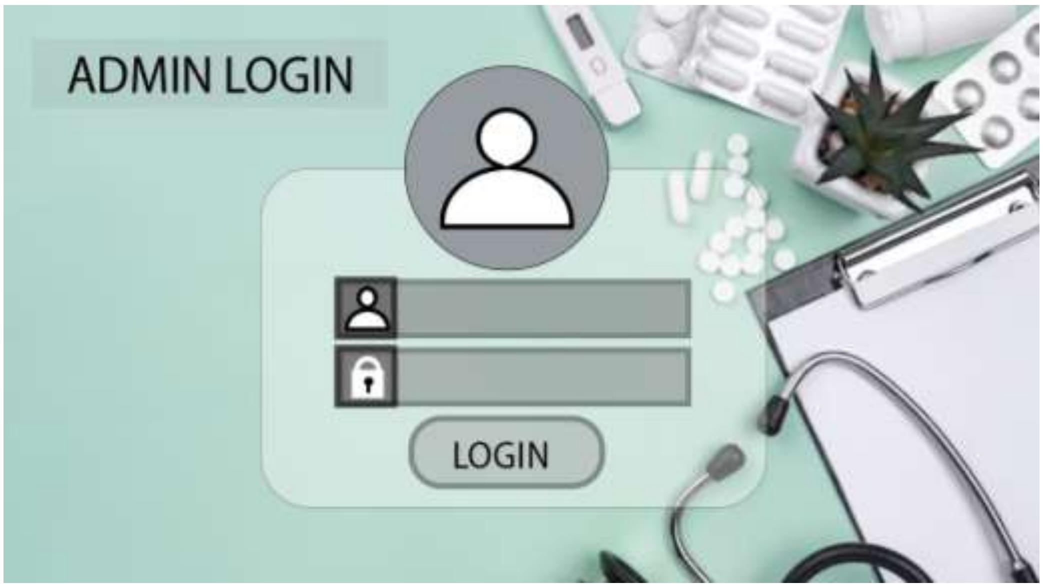
UI Design of Screens

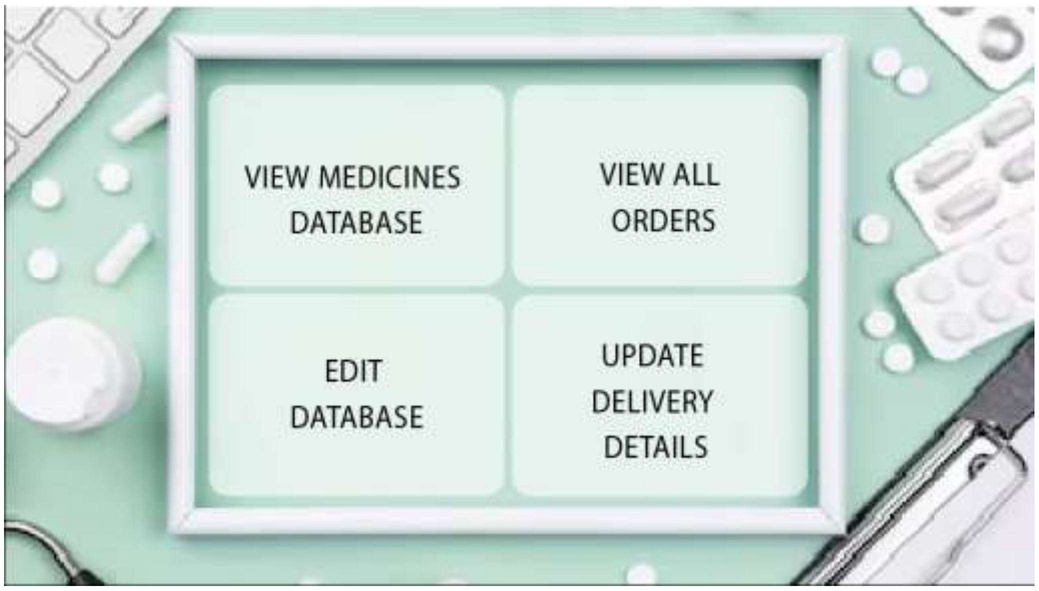







ORDER TOTAL
Name:
Date:
Sl.No.
Particulars
Charges
Quantity
Amount
Bill Total: Rs ________


ADMIN LOGIN
LOGIN


VIEW MEDICINES DATABASE
VIEW ALL ORDERS
EDIT DATABASE
UPDATE DELIVERY DETAILS

TEST SUMMARY REPORT

Introduction

In the realm of software development, the end-user experience is paramount. For the 'PharmEase System', we'll be mainly testing the user end of the application. This crucial phase of testing is dedicated to validating the functionality, usability, and overall satisfaction of the pharmaceutical solution from the perspective of our end-users.

The user end testing process is designed to scrutinize every facet of the system that directly interacts with our valued patients and buyers.

Purpose

The purpose of this report is to present the outcomes of the testing conducted on the PharmEase System, emphasizing user-centric functionalities. From login and registration procedures to the seamless search and order functionalities, our testing strategy is centred on ensuring that the "PharmEase System" meets or even exceeds the expectations outlined in our Software Requirements Specification (SRS) document.

Application Overview

The PharmEase System is a Java-based application designed to provide an intuitive platform for users to access and manage pharmaceutical services, including login, registration, search, order, and checkout functionalities.

Testing Scope

The testing primarily targeted end-user interactions, validating functionality, usability, and system responsiveness.

Our testing approach encompasses various black box testing techniques, automated testing using tools like JUnit, and non-functional testing to evaluate aspects such as user interface responsiveness, volume handling, and stress tolerance. Through meticulous testing, we endeavour to fortify the user end of the application, providing a secure and efficient platform for patients to access and manage their medications seamlessly.

Testing Performed

Types of Tests Performed:

- Black Box Testing Methods for Manual functional testing
- JUnit for Unit Testing: Ensuring code reliability.
- JMeter for Performance Testing: Comparative analysis with an external platform.
- TestComplete for UI Testing

Black Box Testing

For the identified modules/ screens, we've chosen appropriate black box testing methods as follows. These black-box testing methodologies were chosen to address specific aspects of each screen's functionality, allowing us to identify and rectify potential issues, optimize user experience, and enhance the overall reliability of the PharmEase System.

1. **User Login:**

DECISION TABLE

	R1	R2	R3	R4	R5
C1: Username left empty	**F**	**F**	**F**	**T**	**F**
C2: Password left empty	**F**	**F**	**F**	**I**	**T**
C3: Valid username	**T**	**T**	**F**	**I**	
C4: Valid password	**T**	**F**	**I**		**I**
A1: Invalid username			✓		
A2: Invalid password		✓			
A3: Username/ Password should not be empty				✓	✓
A4: Successful Login	✓				

TEST MATRIX TABLE

Test Case ID	**Username**	**Password**	**Expected Output**	**Actual Output**	**Test Fail(F)/ Pass(P)**
T1	eshxl (valid)	shell (valid)	Logs user in and moves to search screen (Rule 1)	Logged in and search screen opened up	P
T2	eshxl (valid)	!Sh (invalid)	Invalid password (Rule 2)	'Invalid password' dialog box	P
T3	Lipiiiiiii (invalid)	#Pikha04	Invalid username (Rule 3)	'Invalid username' dialog box	P
T4		05##snip	Username/ Password should not be empty (Rule 4)	'Username/ Password should not be empty' dialog box	P
T5	Lipikha._.08 (valid)		Username/ Password should not be empty (Rule 5)	'Username/ Password should not be empty' dialog box	P
T6	itzz_logaPriya a (valid)	321@Abcd (valid)	Logs user in and moves to search screen (Rule 1)	Logged in and search screen opened up	P
T7	itzz_logapriya (valid)	-------	Invalid password (Rule 2)	'Invalid Password' dialog box	P
T8	***** (invalid)	321@Abcd (valid)	Invalid Username (Rule 3)	'Invalid username' dialog box	P
T9		#Pikha04	Username/ Password should not be empty (Rule 4)	Username/ Password should not be empty' dialog box	P
T10	eshxl		Username/ Password should not be empty (Rule 5)	'Username/ Password should not be empty' dialog box	P

2. User Registration:

DECISION TABLE

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14
C1: Full Name Valid	**T**	**F**	**T**	**T**	**T**	**T**	**T**	**T**	**-**	**I**	**I**	**I**	**I**	**I**
C2: Age Valid (Integer between 18-100)	**T**	**I**	**F**	**T**	**T**	**T**	**T**	**T**	**I**	**-**	**I**	**I**	**I**	**I**
C3: Email Valid	**T**	**I**	**I**	**F**	**T**	**T**	**T**	**T**	**I**	**I**	**-**	**I**	**I**	**I**
C4: Phone No Valid	**T**	**I**	**I**	**I**	**F**	**T**	**T**	**T**	**I**	**I**	**I**	**-**	**I**	**I**
C5: Username Valid	**T**	**I**	**I**	**I**	**I**	**F**	**T**	**T**	**I**	**I**	**I**	**I**	**-**	**I**
C6: Username non-existing priorly in db	**T**	**I**	**I**	**I**	**I**	**I**	**F**	**T**	**I**	**I**	**I**	**I**	**-**	
C7: Strong password	**T**	**I**	**I**	**I**	**I**	**I**	**I**	**F**	**I**	**I**	**I**	**I**	**I**	**-**
C8: Full Name Not Empty	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**F**	**T**	**T**	**T**	**T**	**T**
C9: Age Not Empty	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**I**	**F**	**T**	**T**	**T**	**T**
C10: Email Not Empty	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**I**	**I**	**F**	**T**	**T**	**T**
C11: Phone No Not Empty	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**I**	**I**	**I**	**F**	**T**	**T**
C12: Username Not Empty	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**I**	**I**	**I**	**I**	**F**	**T**
C13: Password Not Empty	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**T**	**I**	**I**	**I**	**I**	**I**	**F**
A1: Successful Registration	✓													
A2: Invalid Name		✓												
A3: Invalid Age			✓											
A4: Invalid Email Format				✓										
A5: Invalid Phone No					✓									
A6: Invalid Username						✓								
A7: Existing Username							✓							
A8: Password weak								✓						
A9: Missing Required Fields									✓	✓	✓	✓	✓	✓

TEST MATRIX TABLE

	Full Name	Age	Email	Phone No	Blood Group	Gender	Username	Password	Expected Output	Actual Output	Test Fail(F)/Pass (P)
T1	Eshal Shanoj (valid)	18 (valid)	eshalshanoj04@gmail.com (valid)	8590827959 (valid)	O+ (selected)	Female (selected)	eshxl (valid)	shell (not strong, but valid)	Successful Registration (Rule 1)	Registration successful. You can now login with your credentials.	P
T2	##	50 (valid)	nishan@yahoo.com (valid)	8978932165 (valid)	A+ (default)	Male (default)	_nish__ (valid)	Nis0610#### (Strong)	Invalid Name (Rule 2)	Registration successful. You can now login with your credentials.	**F**
T3	Eshal Shanoj (valid)	16 (Out of range)	eshalshanoj04@gmail.com (valid)	8590827959 (valid)	O+ (selected)	Female (selected)	eshxl (valid)	shell (not strong, but valid)	Age out of range (Rule 3)	Please enter a valid age (between 18 and 100)	P
T4	Eshal Shanoj (valid)	18 (valid)	eshalshanoj0@com (valid)	8590827959 (valid)	O+ (selected)	Female (selected)	eshxl (valid)	shell (not strong, but valid)	Invalid Email Format (Rule 4)	Please enter a valid email address	P
T5	Lipikha	18 (Valid)	Lipikha@gmail.com (valid)	922328902911	O+ (selected)	Female (selected)	Lipikha._.08 (valid)	#Pikha04 (Strong)	Invalid Phone No (Rule 5)	Please enter a valid 10-digit phone number	P
T6	Loga Priya (valid)	19 (valid)	Logapriya@gmail.com (valid)	9944412546 (valid)	B+ (Selected)	Female (Selected)	Itzz.****	321@Abcd (Strong)	Invalid Username (Rule 6)	Username should contain only letters, numbers, and underscores	P
T7	Esh Xavier (Valid)	55 (valid)	eshxav@gmail.com	8899663322 (valid)	AB- (Selected)	Male (selected)	eshxl (Existing username)	XYZ#6789 (Strong)	Existing username used (Rule 7)	Username already exists. Please choose another username.	P
T8	Eshal Shanoj (Valid)	18 (Valid)	eshalshanoj04@yahoo.com (valid)	8590827959 (valid)	O+ (Selected)	Female (Selected)	eshxl (valid)	Sh***	Weak Password (Rule 8)	Your password is weak. Do you want to continue?	P
T9		18 (Valid)		9030300452 (valid)	O+ (Selected)	Female (Selected)	eshxl (valid)	Es0504#L# (Strong)	Missing Required Fields (Rule 9)	Please fill in all the fields	P
T10	Loga Priya (valid)		Logapriya@gmail.com	9944412546 (valid)	B+ (Selected)	Female (Selected)	itzz_logapriya	321@Abcd (Strong)	Missing Required Fields (Rule 10)	Please fill in all the fields	P
T11	Eshal Shanoj (Valid)	18 (Valid)		8590827959 (valid)	O+ (Selected)	Female (Selected)	eshxl (valid)	Es0504#L# (Strong)	Missing Required Fields (Rule 11)	Please fill in all the fields	P

T12	Eshal Shanoj (Valid)	18 (Valid)	eshalshanoj04@gmail.com (valid)		O+ (Selected)	Female (Selected)	eshxl (valid)	Es0504#L# (Strong)	Missing Required Fields (Rule 12)	Please fill in all the fields	P
T13	Eshal Shanoj (Valid)	18 (Valid)	eshalshanoj04@gmail.com (valid)	8590827959 (valid)	AB- (Selected)	Female (Selected)		Es0504#L# (Strong)	Missing Required Fields (Rule 13)	Please fill in all the fields	P
T14	L. Lipikha (Valid)	20 (valid)	lipi@gmail.com	9990557951 (valid)	B+ (Selected)	Female (Selected)	Lipikha._.08 (Strong)		Missing Required Fields (Rule 14)	Please fill in all the fields	P
T15	Eshal Shanoj (valid)	18 (Valid)	eshalshanoj04@gmail.com (valid)	8590827959 (valid)	O+ (Selected)	Female (Selected)	eshxl (valid)	Es0504#L# (Strong)	Successful Registration (Rule 1)	Registration successful. You can now login with your credentials.	P
T16	321 45	18 (Valid)	eshalshanoj04@gmail.com (valid)	8590827959 (valid)	O+ (Selected	Female (Selected)	eshxl (valid)	Es0504#L# (Strong)	Invalid Name (Rule 2)	Registration successful. You can now login with your credentials.	F
T17	Loga priya	--a (Invalid)	logapriya@gmail.com	3456737890	B+ (selected)	Female (selected)	itzz_logapriya (valid)	321@Abcd (Strong)	Age out of range (Rule 3)	Please enter a valid age	P
T18	Nishan L (valid)	50 (valid)	nis@ycom (invalid)	8978932165 (valid)	A+ (default)	Male (default)	_nish__ (valid)	Nis0610#### (Strong)	Invalid Email Format (Rule 4)	Please enter a valid email address	P
T19	L. Lipikha (Valid)	20 (valid)	lipi@gmail.com	+91-4254-4254 (invalid)	B+ (Selected)	Female (Selected)	Lipikha._.08 (Strong)	#Pikha04 (Strong)	Invalid Phone No (Rule 5)	Please enter a valid 10-digit phone number	P
T20	Lipikha Sourav (Valid)	38 (valid)	lipikhaa@gmail.com	8899663322 (valid)	AB+ (Selected)	Female (Selected)	Lipikha._.08 (Existing username)	KHA*kha*123 (Strong)	Invalid Username (Rule 6)	Username already exists	P
T21	Lipikha Sourav (Valid)	38 (valid)	lipikhaa@gmail.com	8899663322 (valid)	AB+ (Selected)	Female (Selected)	Lipikha._.08 (Existing username)	KHA*kha*123 (Strong)	Existing username used (Rule 7)	Username already exists. Please choose another username.	P
T22	Loga Priya (valid)	19 (Valid)	logapriya@gmail.com (valid)	3456737890 (valid)	B+ (selected)	Female (selected)	itzz_logapriya (valid)	Hiiiiii?	Weak Password (Rule 8)	Your password is weak. Do you want to continue?	P
T23		19 (Valid)	logapriya@gmail.com (valid)	3456737890 (valid)	B+ (selected)	Female (selected)	itzz_logapriya (valid)	#Pikha04 (Strong)	Missing Required Fields (Rule 9)	Please fill in all the fields	P

T24	Loga Priya (valid)		logapriya@gmail.com (valid)	3456737890 (valid)	B+ (selected)	Female (selected)	itzz_logapriya (valid)	#Pikha04 (Strong)	Missing Required Fields (Rule 10)	Please fill in all the fields	P
T25	Eshal Shanoj (Valid)	18 (Valid)		8590827959 (valid)	A+ (default)	Female (Selected)	eshxl (valid)	Es0504#L# (Strong)	Missing Required Fields (Rule 11)	Please fill in all the fields	P
T26	Eshal Shanoj (Valid)	18 (Valid)	eshalshanoj04@com (valid)		O+ (Selected)	Female (Selected)	eshxl (valid)	Es0504#L# (Strong)	Missing Required Fields (Rule 12)	Please fill in all the fields	P
T27	Loga Priya (valid)	19 (Valid)	logapriya@gmail.com (valid)	3456737890 (valid)	B+ (selected)	Female (selected)		#Pikha04 (Strong)	Missing Required Fields (Rule 13)	Please fill in all the fields	P
T28	Eshal Shanoj (Valid)	18 (Valid)	eshalshanoj04@com (valid)	8590827959 (valid)	O+ (Selected)	Female (Selected)	eshxl (valid)		Missing Required Fields (Rule 14)	Please fill in all the fields	P

3. Medicine Search & Add to Cart:

EQUIVALENCE CLASSES

I1 = {< Not Selected Medicines, Quantity>: Add to cart without selecting medicines, Quantity doesn't have to be entered}

I2={< Selected Medicine(s), Quantity>: Medicine(s) selected and added to cart, Quantity not entered}

I3 = {< Selected Medicines, Quantity>: Medicine(s) selected and added to cart, Alphabets/ special characters entered as Quantity}

I4 = {< Selected Medicines, Quantity>: Medicine(s) selected and added to cart, Quantity is out of range (zero or negative)}

I5 = {< Selected Medicines, Quantity>: Medicine(s) selected and added to cart, Quantity is > the maximum valid quantity (Out of range- >2147483647)}

I6 = {< Selected Medicines, Quantity>: Medicine(s) selected and added to cart, Quantity is a valid quantity}

TEST MATRIX TABLE

Test Case ID	Search Term	Quantity	Expected Output	Actual Output	Test Status
T1	Not selected	N/A	Error: Please select medicines (I1)	Please select a medicine to add to cart	P
T2	Aspirin (Selected)		Error: Please enter quantity (I2)	Please enter a valid quantity	P
T3	Paracetamol (Selected)	a$	Error: Invalid quantity (I3)	Please enter a valid no for quantity	P
T4	Amoxicillin (Selected)	0	Error: Invalid quantity (I4)	Medicine added to cart successfully!	F
T5	Aspirin (Selected)	2147483648	Error: Quantity exceeds limit (I5)	Please enter a valid no for quantity	P
T6	Paracetamol (Selected)	25	Medicines added to cart! (I6)	Medicine added to cart successfully!	P
T7	Not selected	N/A	Error: Please select medicines (I1)	Please select a medicine to add to cart	P
T8	Amoxicillin (Selected)		Error: Please enter quantity (I2)	Please enter a valid quantity	P
T9	Omeprazole (Selected)	D_	Error: Invalid quantity (I3)	Please enter a valid no for quantity	P
T10	Losatron (Selected)	-3	Error: Invalid quantity (I4)	Quantity should be a positive number	P
T11	Simvastatin (Selected)	555555566969	Error: Quantity exceeds limit (I5)	Please enter a valid no for quantity	P
T12	Prednisone (Selected)	19	Medicines added to cart! (I6)	Medicine added to cart successfully!	P

4. Orders Total & Confirm Order:

DECISION TABLE

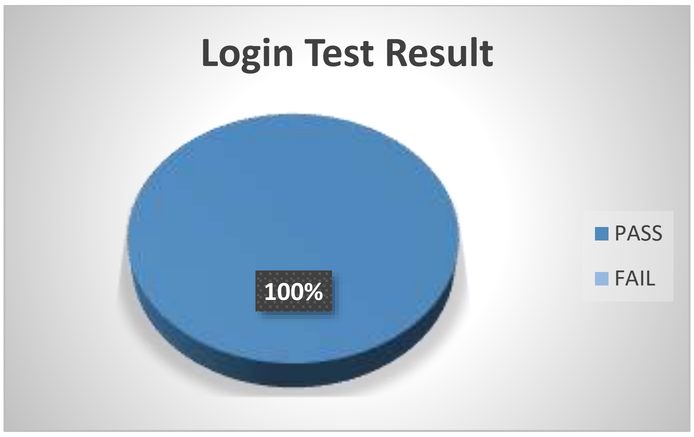
	Rule 1	Rule2	Rule 3	Rule 4	Rule 5
C1: Cart not empty	**F**	**T**	**T**	**T**	**T**
C2: COD Selected (payment method)	**T**	**F**	**T**	**T**	**T**
C3: Address not empty	**T**	**I**	**F**	**T**	**T**
C4: Address Valid	**T**	**I**	**-**	**F**	**T**
A1: Please add items to cart first	✓				
A2: Enter valid payment method		✓			
A3: Address field empty			✓		
A4: Please enter a valid address				✓	
A5: Order placed successfully					✓

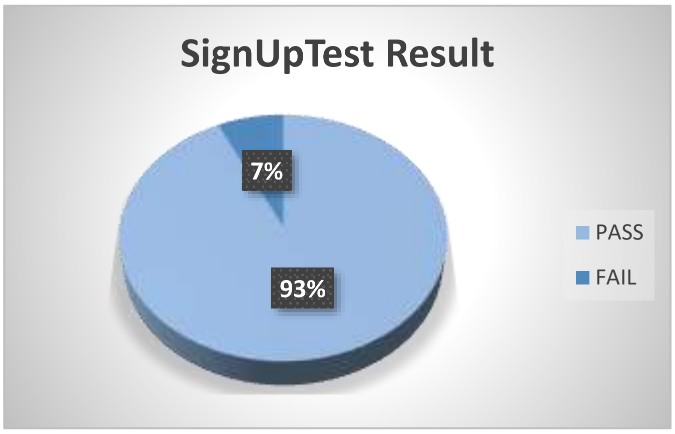
TEST MATRIX TABLE

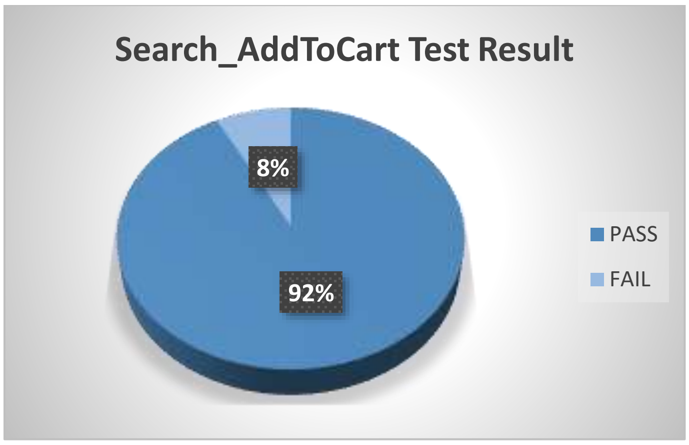
Test Case Id	**Cart**	**Payment method**	**Address**	**Expected output**	**Actual output**	**Test Status**
T1	(Empty)	COD	Bangalore	Please add items to cart first (Rule 1)	Confirm placing order? (Yes/ No)	F
T2	(2 items added; Not empty)	GPAY	Chennai	Enter Valid payment method (Rule 2)	Currently only 'Cash on Delivery' is available as a payment method.	P
T3	(1 item added; Not empty)	COD		Address field Empty (Rule 3)	Please provide an address.	P
T4	(10 items added; Not empty)	COD	****	Please enter a Valid Address (Rule 4)	Confirm placing order? (Yes/ No)	F

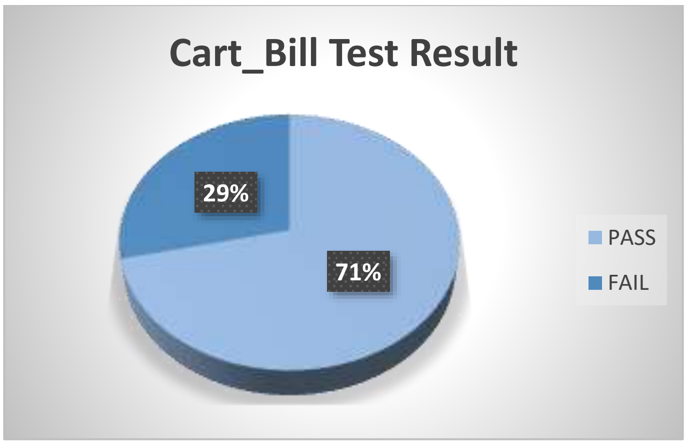
T5	(3 items added; Not empty)	COD	Bangalore	Order placed Successfully (Rule 5)	Order placed Successfully!	P
T6	(Empty)	COD	Ernakulam	Please add items to cart first (Rule 1)	Confirm placing order? (Yes/ No)	F
T7	(10 items added; Not empty)	PAYTM	Bangalore	Enter Valid payment method (Rule 2)	Currently only 'Cash on Delivery' is available as a payment method.	P
T8	(2 items added; Not empty)	COD		Address field Empty (Rule 3)	Please provide an address.	P
T9	(1 items added; Not empty)	COD	Che---	Please enter a Valid Address (Rule4)	Confirm placing order? (Yes/ No)	F
T10	(20 items added; Not empty)	COD	Chennai	Order placed Successfully (Rule 5)	Order placed Successfully!	P

Pie Charts showing Results









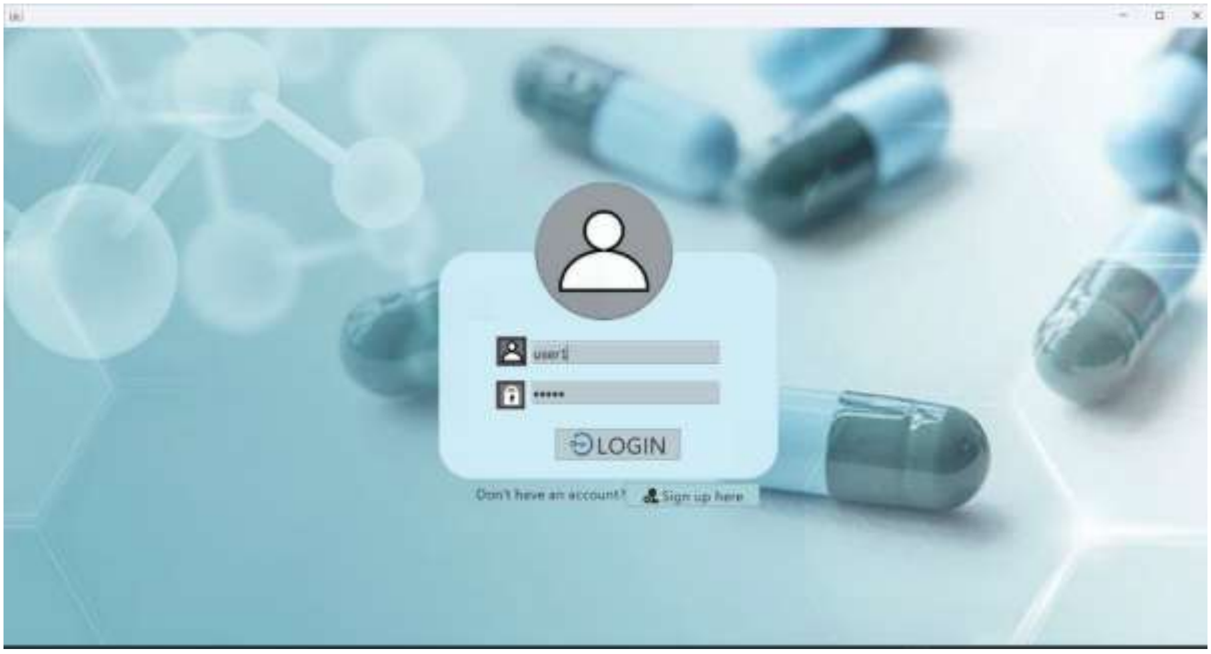
JUnit for Unit Testing: Enhancing Code Reliability

Unit testing is a type of white-box testing, where the internal structure and code implementation details are known to the tester.

JUnit is primarily used for unit testing, which involves testing individual units or components of a software application in isolation. It focuses on verifying that each unit of the software performs as designed. This approach of testing small units helps detect bugs early, enhancing code reliability. Developers focus more on code readability, leading to more robust and error-free software.

JUnit's features include annotations for test identification, assertions for expected results, and automated test runners, streamlining the testing process. It contributes to efficient and quality code development by providing automatic test execution and easily interpretable results.

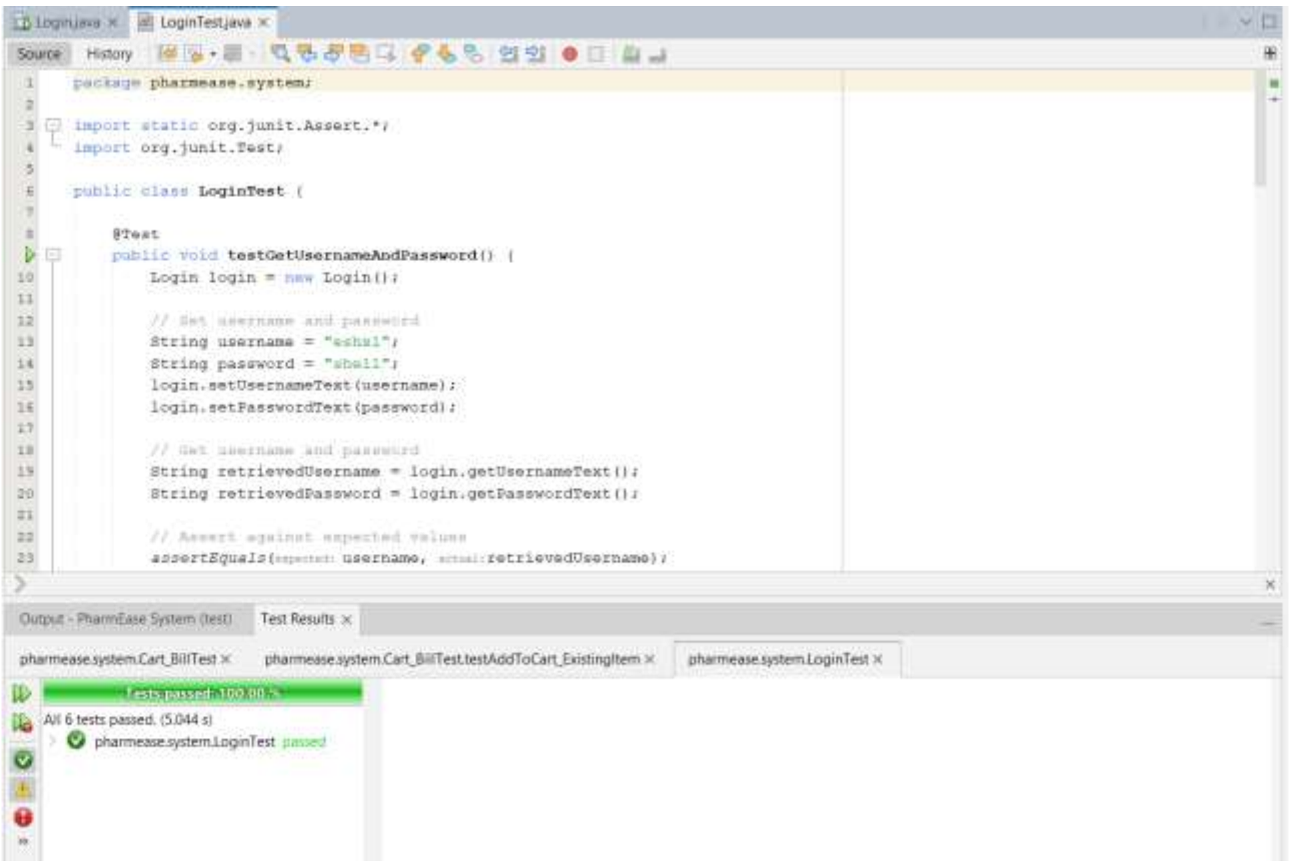
1. **Login Module**



Junit Testing Code:

```
package pharmease.system;
import static org.junit.Assert.*;
import org.junit.Test;

public class LoginTest {

    @Test
    public void testGetUsernameAndPassword() {
        Login login = new Login();
        // Set username and password
        String username = "eshxl";
        String password = "shell";
        login.setUsernameText(username);
        login.setPasswordText(password);
        // Get username and password
        String retrievedUsername = login.getUsernameText();
        String retrievedPassword = login.getPasswordText();

        // Assert against expected values
        assertEquals(username, retrievedUsername);
        assertEquals(password, retrievedPassword);
    }

    @Test
    public void testEmptyUsername() {
        Login login = new Login();

        // Set empty username and valid password
        String emptyUsername = "";
        String validPassword = "password123";
        login.setUsernameText(emptyUsername);
        login.setPasswordText(validPassword);

        assertFalse(login.validateLogin());
    }

    @Test
    public void testEmptyPassword() {
        Login login = new Login();

        // Set valid username and empty password
        String validUsername = "user123";
        String emptyPassword = "";
        login.setUsernameText(validUsername);
        login.setPasswordText(emptyPassword);

        assertFalse(login.validateLogin());
    }

    @Test
    public void testInvalidUsername() {
        Login login = new Login();
        // Set invalid username and valid password
        String invalidUsername = "invalid_user";
        String validPassword = "password123";
        login.setUsernameText(invalidUsername);
        login.setPasswordText(validPassword);

        assertFalse(login.validateLogin());
    }
```

```java
    @Test
    public void testInvalidPassword() {
        Login login = new Login();

        // Set valid username and invalid password
        String validUsername = "eshxl";
        String invalidPassword = "pass";
        login.setUsernameText(validUsername);
        login.setPasswordText(invalidPassword);

        assertFalse(login.validateLogin());
    }

    @Test
    public void testValidCredentials() {
        Login login = new Login();

        // Set valid username and password
        String validUsername = "eshxl";
        String validPassword = "shell";
        login.setUsernameText(validUsername);
        login.setPasswordText(validPassword);

        assertTrue(login.validateLogin());
    }
}
```

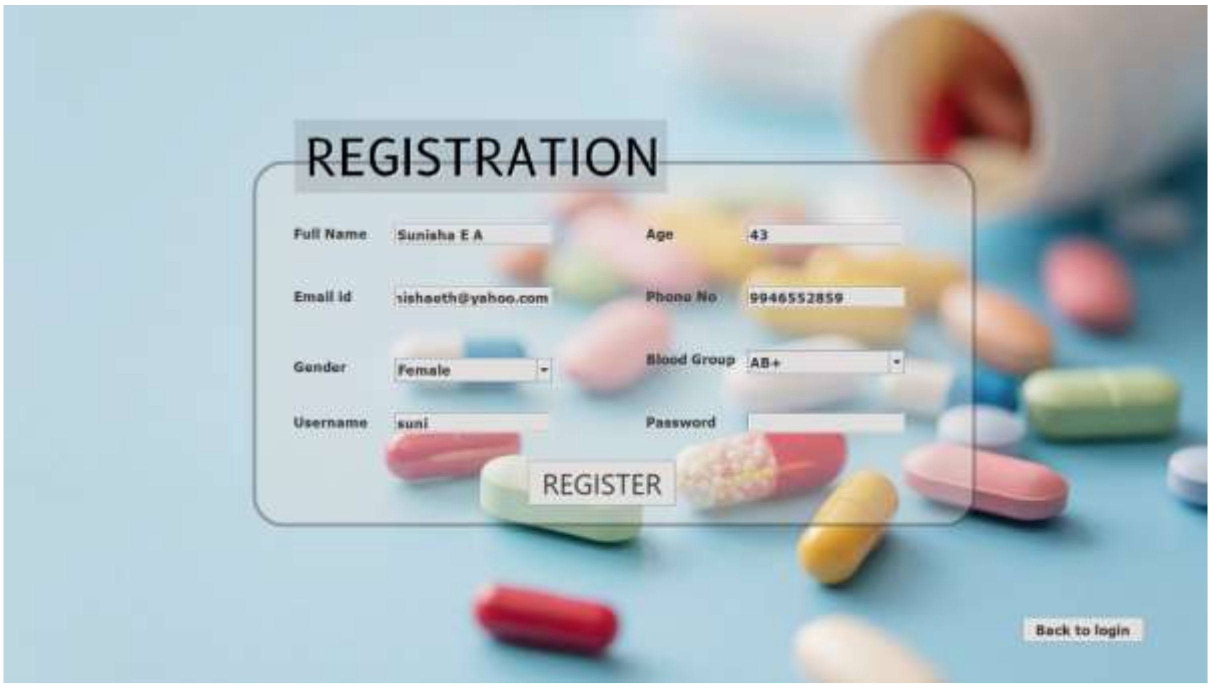


```java
package pharmease.system;

import static org.junit.Assert.*;
import org.junit.Test;

public class LoginTest {

    @Test
    public void testGetUsernameAndPassword() {
        Login login = new Login();

        // Set username and password
        String username = "eshxl";
        String password = "shell";
        login.setUsernameText(username);
        login.setPasswordText(password);

        // Get username and password
        String retrievedUsername = login.getUsernameText();
        String retrievedPassword = login.getPasswordText();

        // Assert against expected values
        assertEquals(expected: username, actual:retrievedUsername);
```

Output - PharmEase System (test)
Test Results
pharmease.system.Cart_BillTest
pharmease.system.Cart_BillTest.testAddToCart_ExistingItem
pharmease.system.LoginTest

```
Tests passed: 100.00 %
All 6 tests passed. (5.044 s)
pharmease.system.LoginTest passed
```

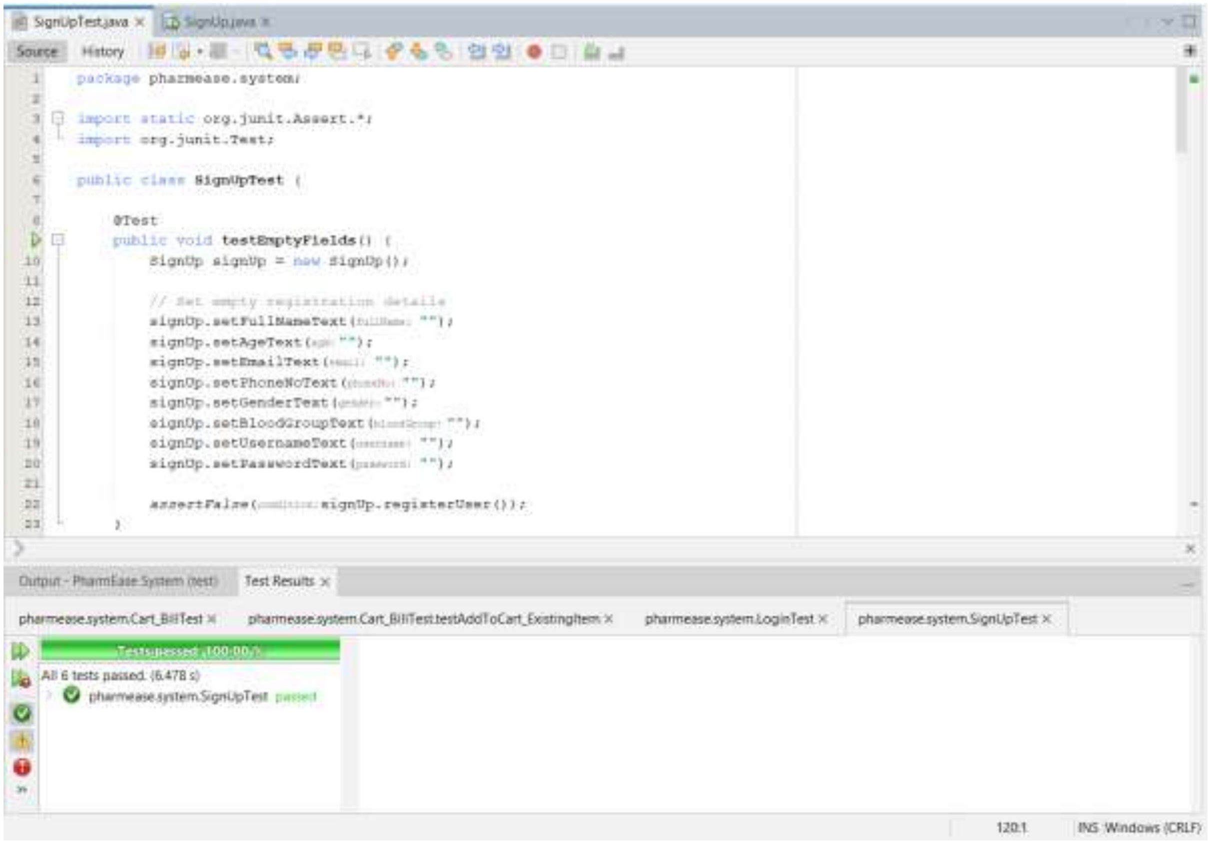
2. Registration/ Sign Up Module


REGISTRATION

Full Name	Sunisha E A	Age	43
Email id	nishaeth@yahoo.com	Phone No	9946552859
Gender	Female	Blood Group	AB+
Username	suni	Password	

<u>Junit Testing Code:</u>

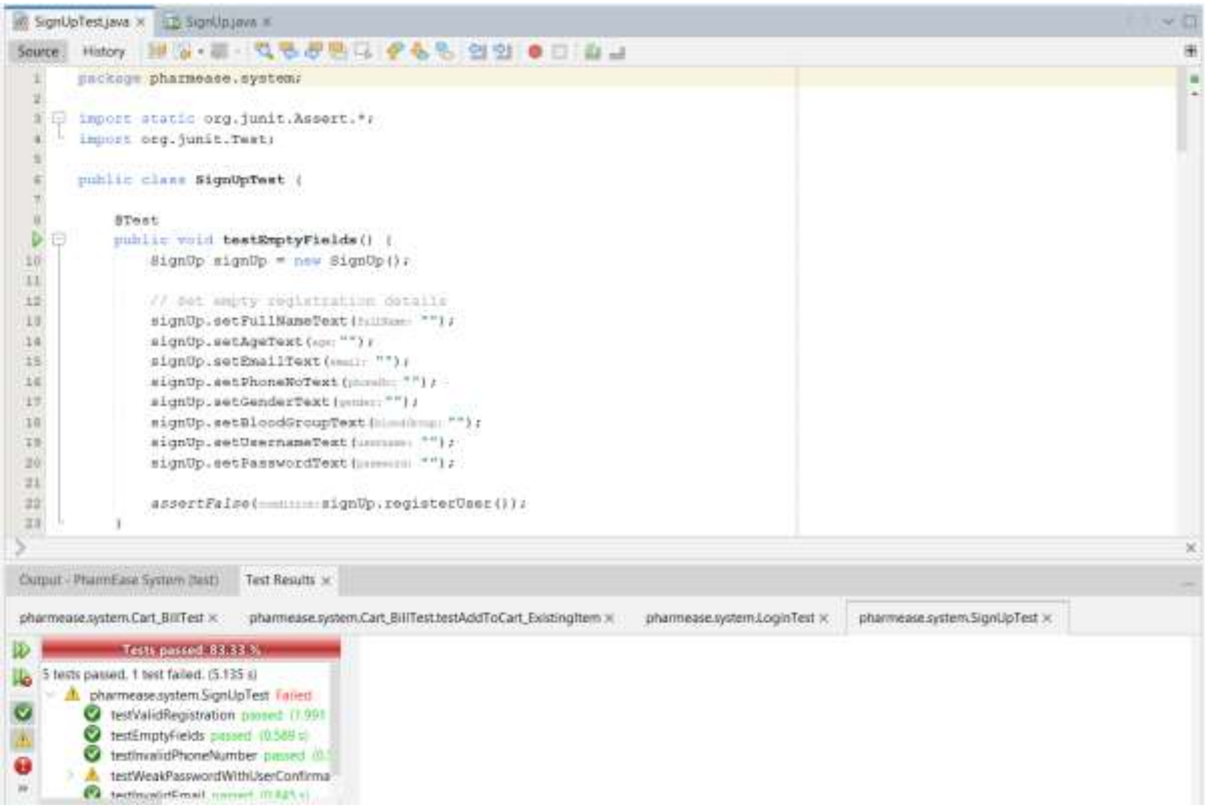
```
package pharmease.system;

import static org.junit.Assert.*;
import org.junit.Test;

public class SignUpTest {

    @Test
    public void testEmptyFields() {
        SignUp signUp = new SignUp();

        // Set empty registration details
        signUp.setFullNameText("");
        signUp.setAgeText("");
        signUp.setEmailText("");
        signUp.setPhoneNoText("");
        signUp.setGenderText("");
        signUp.setBloodGroupText("");
        signUp.setUsernameText("");
        signUp.setPasswordText("");

        assertFalse(signUp.registerUser());
    }

    @Test
    public void testInvalidEmail() {
        SignUp signUp = new SignUp();

        // Set registration details with an invalid email
        signUp.setFullNameText("Alice Smith");
        signUp.setAgeText("25");
        signUp.setEmailText("invalid_email");
        signUp.setPhoneNoText("1234567890");
        signUp.setGenderText("Female");
        signUp.setBloodGroupText("AB+");
        signUp.setUsernameText("alice123");
        signUp.setPasswordText("StrongPassword123");

        assertFalse(signUp.registerUser());
    }

    @Test
    public void testInvalidAge() {
        SignUp signUp = new SignUp();

        // Set registration details with invalid age (below 18)
        signUp.setFullNameText("David Miller");
        signUp.setAgeText("16");
        signUp.setEmailText("david@example.com");
        signUp.setPhoneNoText("1234567890");
        signUp.setGenderText("Male");
        signUp.setBloodGroupText("B+");
        signUp.setUsernameText("davidmiller");
        signUp.setPasswordText("StrongPass123");

        assertFalse(signUp.registerUser());
    }

    @Test
    public void testInvalidPhoneNumber() {
        SignUp signUp = new SignUp();

        // Set registration details with an invalid phone number
        signUp.setFullNameText("Emma Watson");
        signUp.setAgeText("28");
        signUp.setEmailText("emma@example.com");
        signUp.setPhoneNoText("12345"); // Invalid phone number format
        signUp.setGenderText("Female");
        signUp.setBloodGroupText("O+");
        signUp.setUsernameText("emmawatson");
        signUp.setPasswordText("StrongPass123");

        assertFalse("Phone number format validation failed",
signUp.registerUser());
    }
    @Test
    public void testWeakPasswordWithUserConfirmation() {
        SignUp signUp = new SignUp();

        // Set registration details with a weak password
        signUp.setFullNameText("Robert Johnson");
        signUp.setAgeText("35");
        signUp.setEmailText("robert@example.com");
        signUp.setPhoneNoText("9876543210");
        signUp.setGenderText("Male");
        signUp.setBloodGroupText("A-");
        signUp.setUsernameText("robertjohnson");
        signUp.setPasswordText("weakpass"); // Password doesn't meet
strength requirements

        boolean isRegistered = signUp.registerUser();

        if (isRegistered) {
            // Simulate user confirmation or rejection here (true for
confirmation, false for rejection)
            boolean userConfirmation = true; // Change this value to
simulate user action
            assertTrue("Registration successful and confirmed by the
user", userConfirmation);
        } else {
            assertFalse("Registration failed as expected due to weak
password", true);
        }
    }


    @Test
    public void testValidRegistration() {
        SignUp signUp = new SignUp();

        // Set valid registration details
        signUp.setFullNameText("John Doe");
        signUp.setAgeText("30");
        signUp.setEmailText("john@example.com");
        signUp.setPhoneNoText("1234567890");
        signUp.setGenderText("Male");
        signUp.setBloodGroupText("A+");
        signUp.setUsernameText("johndoe");
        signUp.setPasswordText("StrongPass123");

        assertTrue(signUp.registerUser());
    }
}
```



```
package pharmease.system;

import static org.junit.Assert.*;
import org.junit.Test;

public class SignUpTest {

    @Test
    public void testEmptyFields() {
        SignUp signUp = new SignUp();

        // Set empty registration details
        signUp.setFullNameText(fullName: "");
        signUp.setAgeText(age: "");
        signUp.setEmailText(email: "");
        signUp.setPhoneNoText(phoneNo: "");
        signUp.setGenderText(gender: "");
        signUp.setBloodGroupText(bloodGroup: "");
        signUp.setUsernameText(username: "");
        signUp.setPasswordText(password: "");

        assertFalse(condition: signUp.registerUser());
    }
```

```
Tests passed: 100.00 %
All 6 tests passed. (6.478 s)
  pharmease.system.SignUpTest passed
```

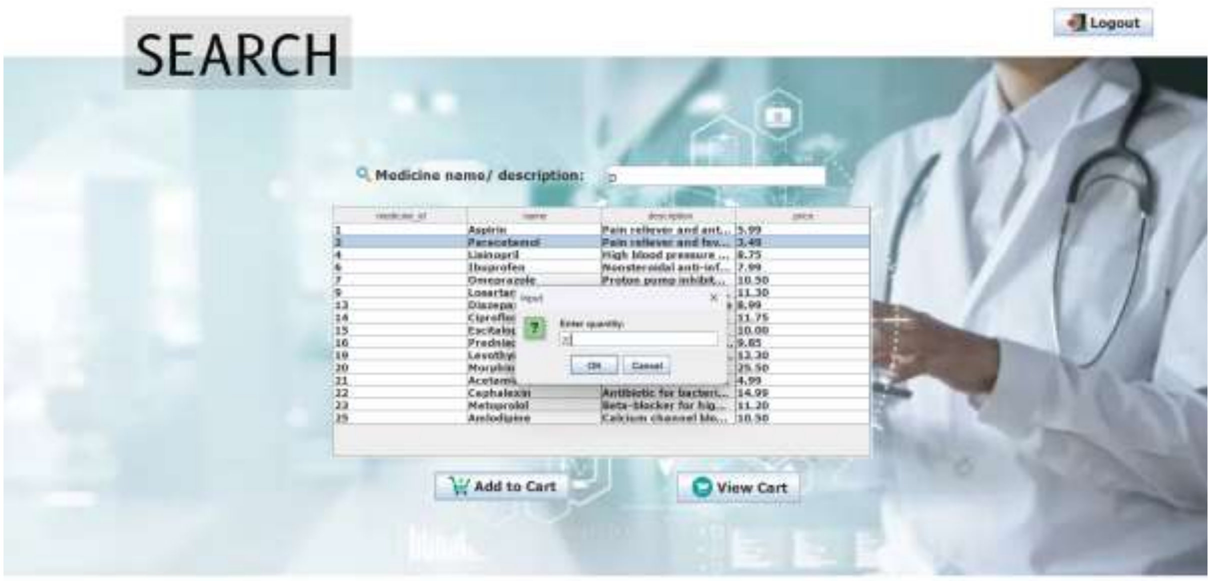
Failed test cases:



```
package pharmease.system;

import static org.junit.Assert.*;
import org.junit.Test;

public class SignUpTest {

    @Test
    public void testEmptyFields() {
        SignUp signUp = new SignUp();

        // Set empty registration details
        signUp.setFullNameText(fullName: "");
        signUp.setAgeText(age: "");
        signUp.setEmailText(email: "");
        signUp.setPhoneNoText(phoneNo: "");
        signUp.setGenderText(gender: "");
        signUp.setBloodGroupText(bloodGroup: "");
        signUp.setUsernameText(username: "");
        signUp.setPasswordText(password: "");

        assertFalse(condition: signUp.registerUser());
    }
```

```
Tests passed: 83.33 %
5 tests passed, 1 test failed. (5.135 s)
  pharmease.system.SignUpTest Failed
    testValidRegistration passed (1.991
    testEmptyFields passed (0.589 s)
    testInvalidPhoneNumber passed (0.
    testWeakPasswordWithUserConfirma
    testInvalidEmail passed (0.845 s)
```

3. Search and Add to Cart Module



Medicine name/ description: p

medicine_id	name	description	price
1	Aspirin	Pain reliever and ant...	5.99
3	Paracetamol	Pain reliever and fev...	3.49
4	Lisinopril	High blood pressure ...	8.75
6	Ibuprofen	Nonsteroidal anti-inf...	7.99
7	Omeprazole	Proton pump inhibit...	10.50
9	Losartar		11.30
13	Diazepa		8.99
14	Ciproflo		11.75
15	Escitalo		10.00
16	Prednis		9.85
19	Levothy		12.30
20	Morphi		25.50
21	Acetami		4.99
22	Cephalexin	Antibiotic for bacteri...	14.99
23	Metoprolol	Beta-blocker for hig...	11.20
25	Amlodipine	Calcium channel blo...	10.50

Input

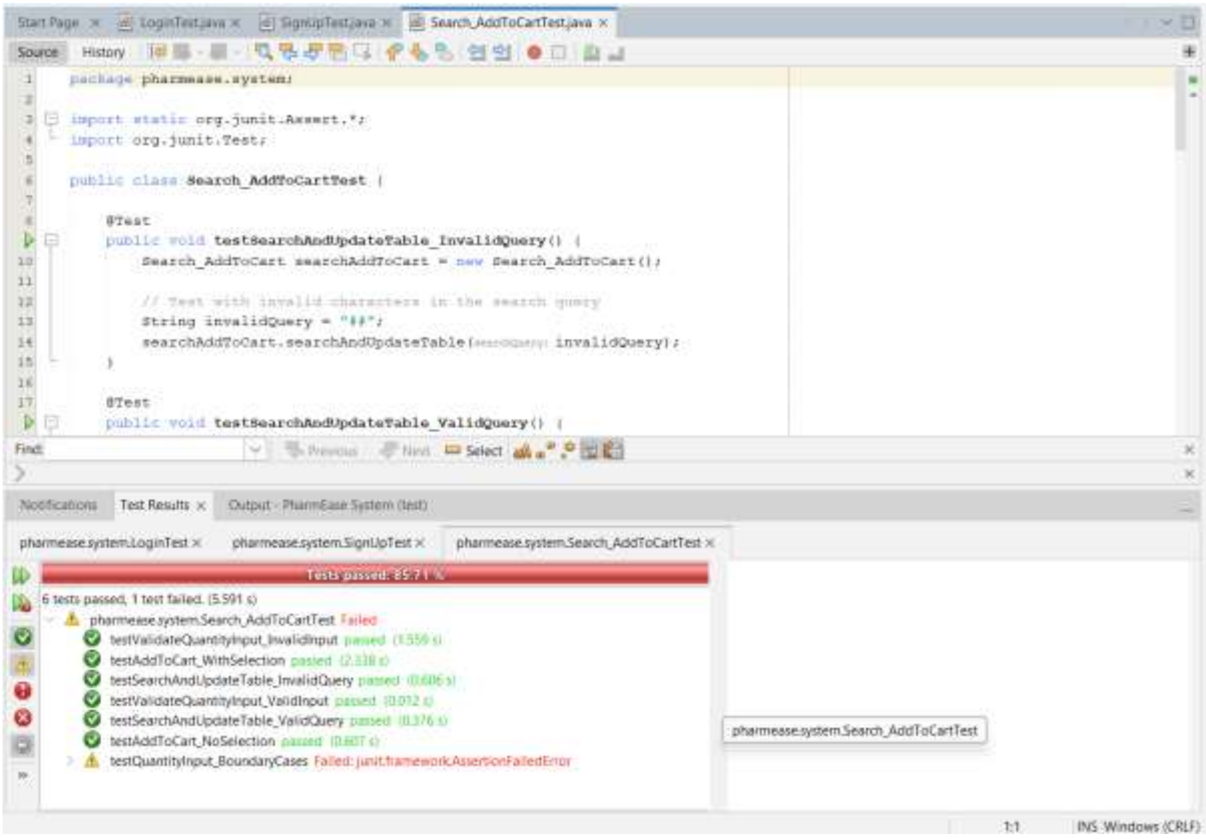
Enter quantity:

2

OK Cancel

Junit Testing Code:

```
package pharmease.system;

import static org.junit.Assert.*;
import org.junit.Test;

public class Search_AddToCartTest {

    @Test
    public void testSearchAndUpdateTable_InvalidQuery() {
        Search_AddToCart searchAddToCart = new Search_AddToCart();

        // Test with invalid characters in the search query
        String invalidQuery = "##";
        searchAddToCart.searchAndUpdateTable(invalidQuery);
    }

    @Test
    public void testSearchAndUpdateTable_ValidQuery() {
        Search_AddToCart searchAddToCart = new Search_AddToCart();

        // Test with valid characters in the search query
        String validQuery = "ValidQuery";
        searchAddToCart.searchAndUpdateTable(validQuery);
    }

    @Test
    public void testQuantityInput_BoundaryCases() {
        Search_AddToCart searchAddToCart = new Search_AddToCart();
    }
}


        // Test quantity input boundaries (0 and very large values)
        String zeroQuantity = "0";
        String largeQuantity = "1000000";

        assertFalse(searchAddToCart.validateQuantityInput(zeroQuantity));
// Expecting false for 0 quantity
        assertTrue(searchAddToCart.validateQuantityInput(largeQuantity));
// Expecting true for a large quantity
    }

    @Test
    public void testValidateQuantityInput_InvalidInput() {
        Search_AddToCart searchAddToCart = new Search_AddToCart();

        // Test validateQuantityInput method with invalid quantity input
        String invalidQuantity = "abc";

assertFalse(searchAddToCart.validateQuantityInput(invalidQuantity));
    }

        @Test
    public void testValidateQuantityInput_ValidInput() {
        Search_AddToCart searchAddToCart = new Search_AddToCart();

        // Test validateQuantityInput method with valid quantity input
        String validQuantity = "5";
        assertTrue(searchAddToCart.validateQuantityInput(validQuantity));
    }

    @Test
    public void testAddToCart_NoSelection() {
        Search_AddToCart searchAddToCart = new Search_AddToCart();

        // Test addToCart method with no selection in the table
        searchAddToCart.addToCart();
    }

    @Test
    public void testAddToCart_WithSelection() {
        Search_AddToCart searchAddToCart = new Search_AddToCart();

        // TODO: Simulate the selection in the table
        // Test addToCart method with a valid selection in the table
        searchAddToCart.addToCart();
        // TODO: Assert if the correct item is added to the cart
    }
}
```

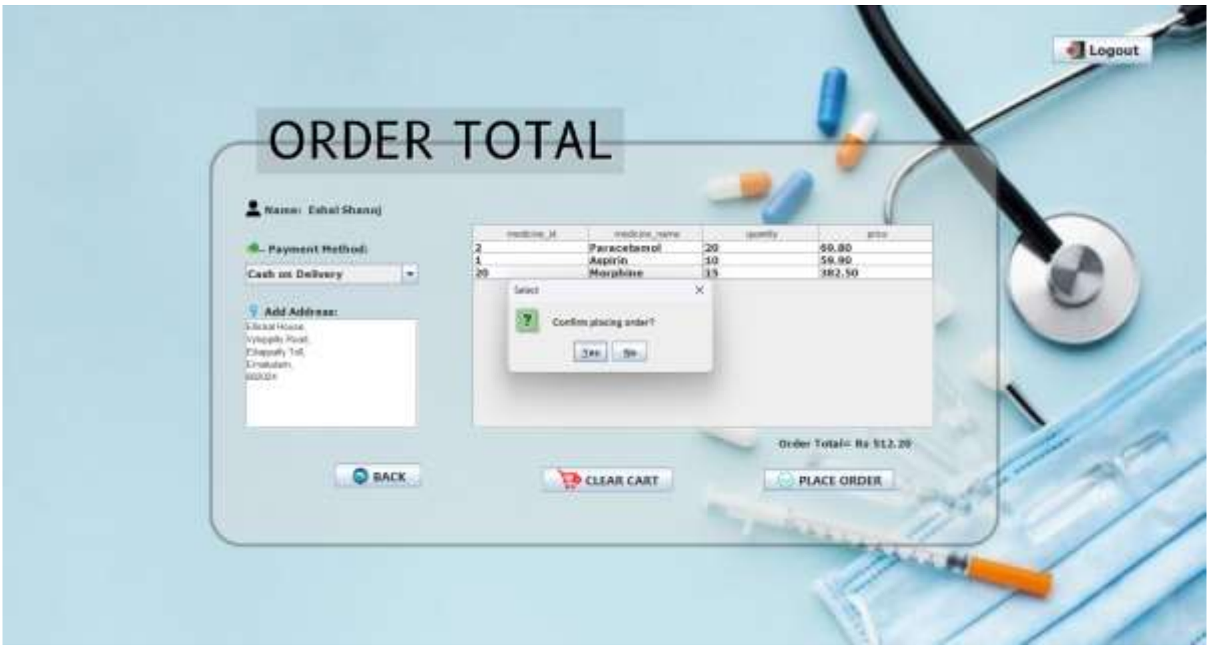


```
package pharmease.system;

import static org.junit.Assert.*;
import org.junit.Test;

public class Search_AddToCartTest {

    @Test
    public void testSearchAndUpdateTable_InvalidQuery() {
        Search_AddToCart searchAddToCart = new Search_AddToCart();

        // Test with invalid characters in the search query
        String invalidQuery = "$#";
        searchAddToCart.searchAndUpdateTable(searchQuery: invalidQuery);
    }

    @Test
    public void testSearchAndUpdateTable_ValidQuery() {
```

```
Tests passed: 85.71 %
6 tests passed, 1 test failed. (5.591 s)
  pharmease.system.Search_AddToCartTest Failed
    testValidateQuantityInput_InvalidInput passed (1.559 s)
    testAddToCart_WithSelection passed (2.338 s)
    testSearchAndUpdateTable_InvalidQuery passed (0.606 s)
    testValidateQuantityInput_ValidInput passed (0.012 s)
    testSearchAndUpdateTable_ValidQuery passed (0.376 s)
    testAddToCart_NoSelection passed (0.607 s)
  > testQuantityInput_BoundaryCases Failed: junit.framework.AssertionFailedError
```

Start Page
LoginTest.java
SignUpTest.java
Search_AddToCartTest.java
Source
History
Find:
Previous
Next
Select
Notifications
Test Results
Output - PharmEase System (test)
pharmease.system.LoginTest
pharmease.system.SignUpTest
pharmease.system.Search_AddToCartTest
1:1
INS Windows (CRLF)

4. Place Order Module



ORDER TOTAL

Name: Eshal Shanuj

Payment Method: Cash on Delivery

Add Address:
[illegible]

medicine_id	medicine_name	quantity	price
2	Paracetamol	20	69.80
1	Aspirin	10	59.90
20	Morphine	15	382.50

Select

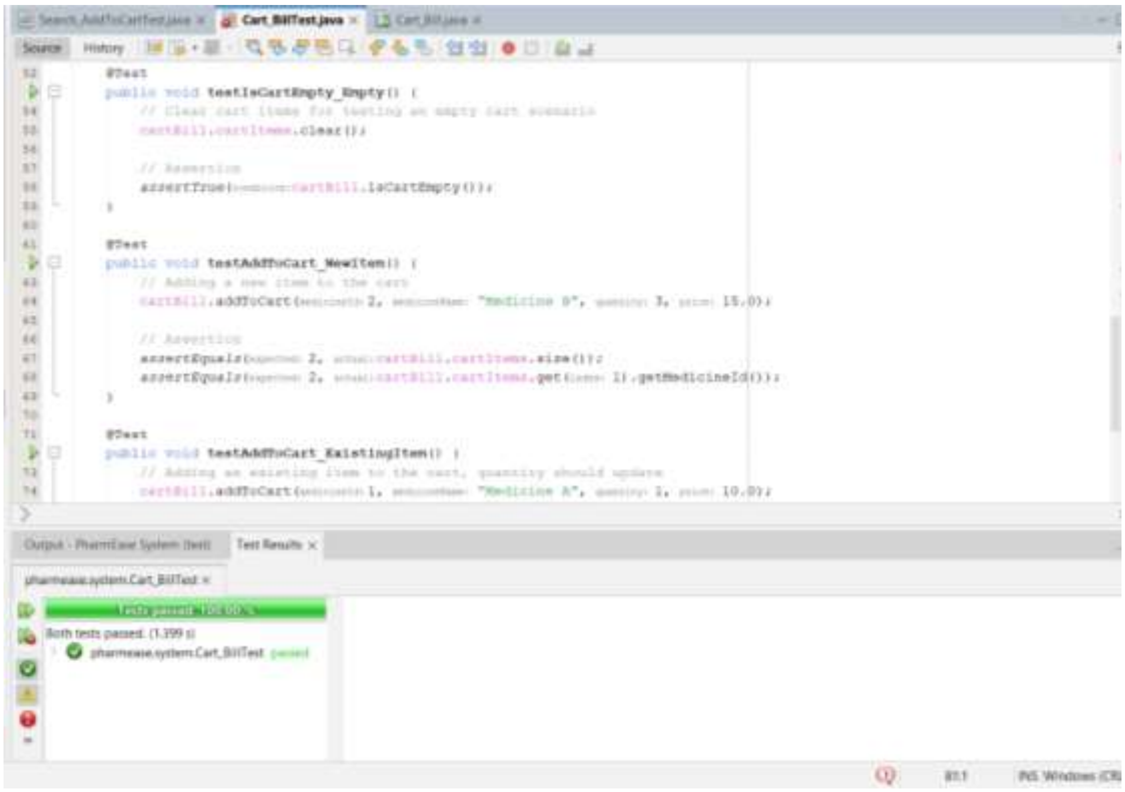
Confirm placing order?

Order Total= Rs 512.20

Junit Testing Code:

```
package pharmease.system;

import org.junit.Before;
import org.junit.Test;
import static org.junit.Assert.*;
```

```
public class Cart_BillTest {

    private Cart_Bill cartBill;

    @Before
    public void setUp() {
        // You might need to mock the database connection or set up a test
database for the cart items.
        cartBill = new Cart_Bill("username");
        // Creating some mock cart items for testing addToCart() method
        cartBill.cartItems = new ArrayList<>();
        cartBill.cartItems.add(new Cart_Bill.CartItem(1, "Medicine A", 2,
10.0));
    }

    @Test
    public void testCalculateOrderTotal_WithItems() {
        // Action
        cartBill.calculateOrderTotal();

        // Assertion
        double expectedTotal = 20.0; // Replace with the expected total
based on your cart items.
        assertEquals(expectedTotal,
Double.parseDouble(cartBill.jLabel5.getText()), 0.01);
    }

    @Test
    public void testCalculateOrderTotal_EmptyCart() {
        // Clear cart items for testing an empty cart scenario
        cartBill.cartItems.clear();

        // Action
        cartBill.calculateOrderTotal();

        // Assertion
        double expectedTotal = 0.0;
        assertEquals(expectedTotal,
Double.parseDouble(cartBill.jLabel5.getText()), 0.01);
    }

    @Test
    public void testIsCartEmpty_NotEmpty() {
        // Assuming the cart has items
        cartBill.cartItems.add(new Cart_Bill.CartItem(1, "Medicine A", 2,
10.0));
        // Assertion
        assertFalse(cartBill.isCartEmpty());
    }
    @Test
    public void testIsCartEmpty_Empty() {
        // Clear cart items for testing an empty cart scenario
        cartBill.cartItems.clear();

        // Assertion
        assertTrue(cartBill.isCartEmpty());
    }

    @Test
    public void testAddToCart_NewItem() {
        // Adding a new item to the cart
        cartBill.addToCart(2, "Medicine B", 3, 15.0);

        // Assertion
        assertEquals(2, cartBill.cartItems.size());
        assertEquals(2, cartBill.cartItems.get(1).getMedicineId());
    }

    @Test
    public void testAddToCart_ExistingItem() {
        // Adding an existing item to the cart, quantity should update
        cartBill.addToCart(1, "Medicine A", 1, 10.0);

        // Assertion
        assertEquals(1, cartBill.cartItems.size());
        assertEquals(3, cartBill.cartItems.get(0).getQuantity());
    }
}
```



```
    @Test
    public void testIsCartEmpty_Empty() {
        // Clear cart items for testing an empty cart scenario
        cartBill.cartItems.clear();

        // Assertion
        assertTrue(condition: cartBill.isCartEmpty());
    }

    @Test
    public void testAddToCart_NewItem() {
        // Adding a new item to the cart
        cartBill.addToCart(medicineId: 2, medicineName: "Medicine B", quantity: 3, price: 15.0);

        // Assertion
        assertEquals(expected: 2, actual: cartBill.cartItems.size());
        assertEquals(expected: 2, actual: cartBill.cartItems.get(index: 1).getMedicineId());
    }

    @Test
    public void testAddToCart_ExistingItem() {
        // Adding an existing item to the cart, quantity should update
        cartBill.addToCart(medicineId: 1, medicineName: "Medicine A", quantity: 1, price: 10.0);
```

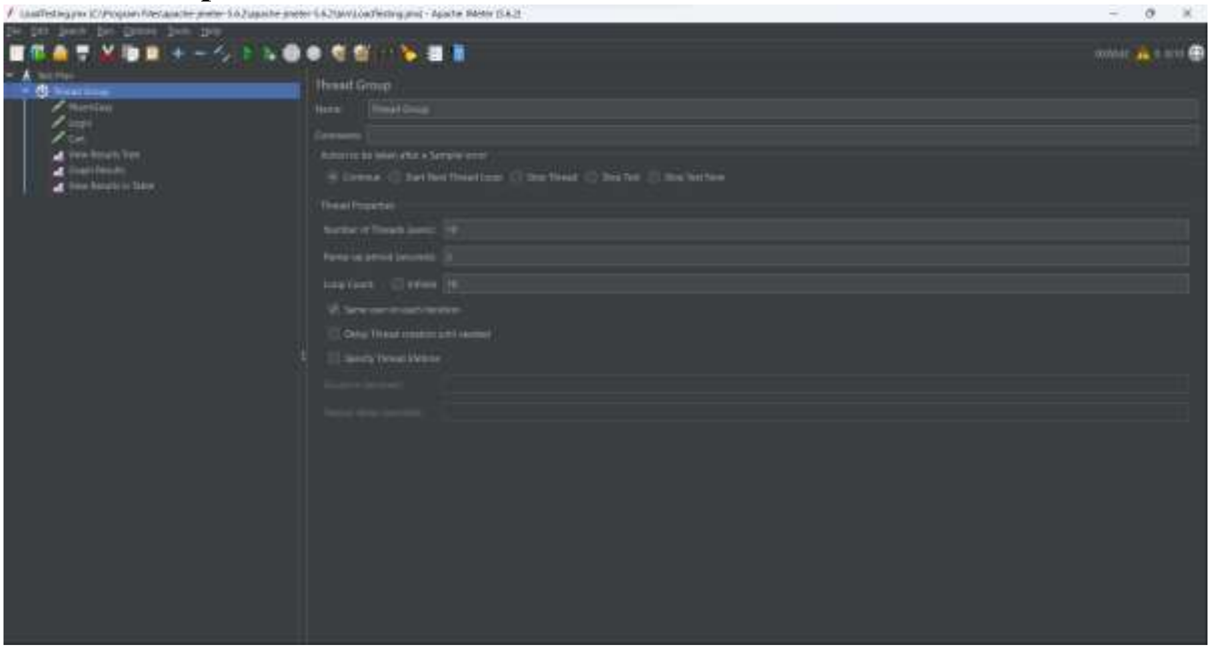
```
pharmease.system.Cart_BillTest
Tests passed: 100.00 %
Both tests passed. (1.399 s)
pharmease.system.Cart_BillTest passed
```

JMeter: Evaluating System Performance

Performance Testing is generally categorized as a type of non-functional testing. Non-functional testing encompasses aspects of a system that are not related to specific behaviours or functions but rather focus on characteristics such as performance, scalability, reliability, and usability. JMeter stands out as an open-source tool for performance testing, offering a comprehensive suite of features to evaluate an application's behaviour under different scenarios. By simulating heavy loads, concurrent users, and stress conditions, JMeter assesses system response times, throughput, and scalability.

But the testing methodology employed for our Java-based PharmEase system differs significantly from JMeter's primary focus on website load and performance testing. JMeter, a robust tool designed for assessing web applications' scalability and response under heavy loads, is predominantly oriented towards testing web-based systems. As our PharmEase system is an intricately designed Java-based application with a unique architecture, relying on JMeter for its testing may not align with its complex functionalities. However, we will utilize JMeter to assess the website 'https://pharmeasy.in/' as a comparative benchmark for performance evaluation. This external website provides an advanced platform akin to certain aspects of our PharmEase System, allowing us to glean insights and optimize our system's performance based on industry standards and best practices.

Thread Groups:



Thread Group

Name: Thread Group

Comments:

Action to be taken after a Sampler error

(●) Continue ( ) Start Next Thread Loop ( ) Stop Thread ( ) Stop Test ( ) Stop Test Now

Thread Properties

Number of Threads (users): 10

Ramp-up period (seconds): 3

Loop Count: ( ) Infinite 10

[✓] Same user on each iteration

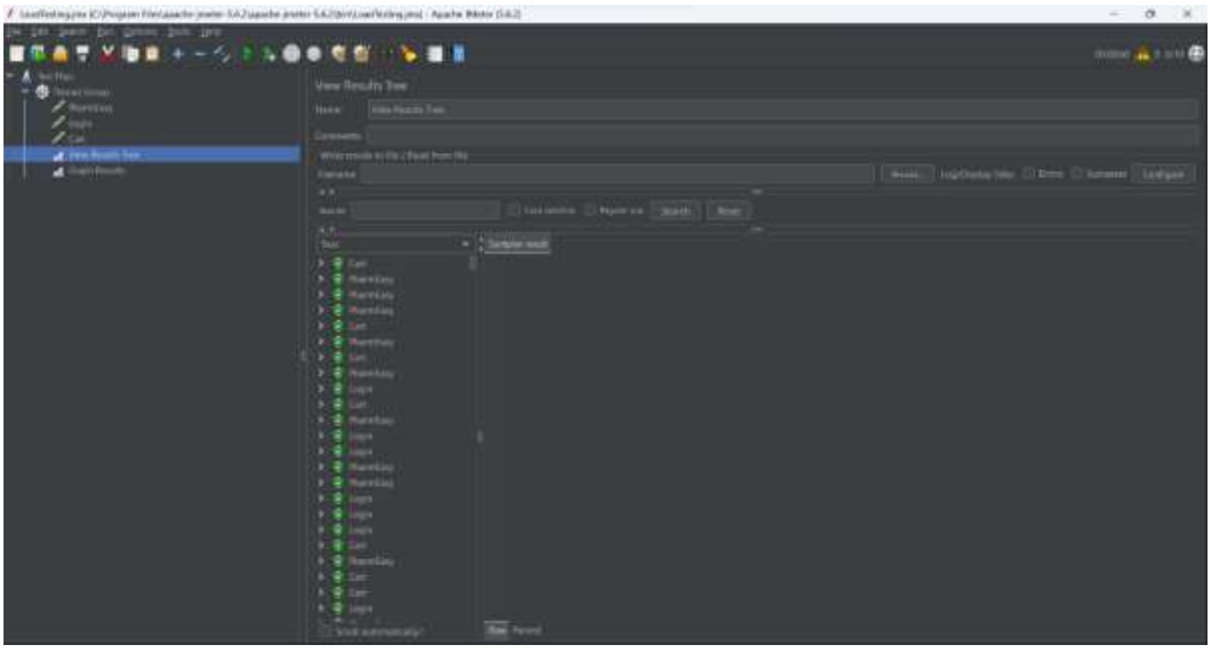
[ ] Delay Thread creation until needed

[ ] Specify Thread lifetime

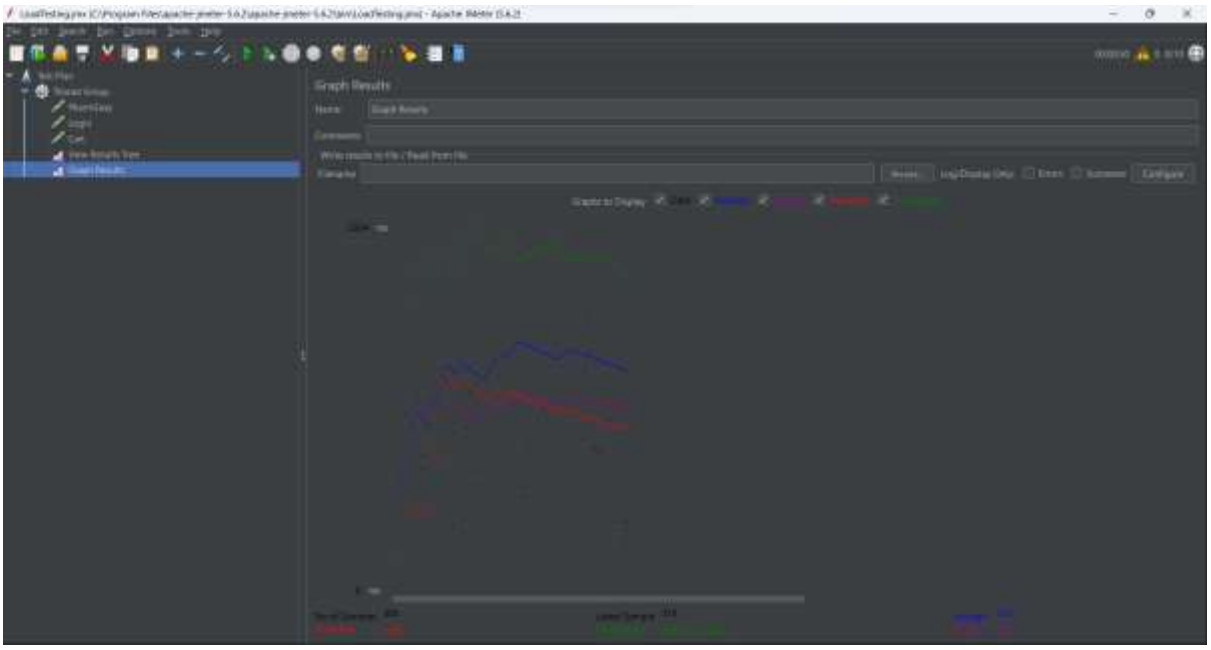
Duration (seconds):

Startup delay (seconds):

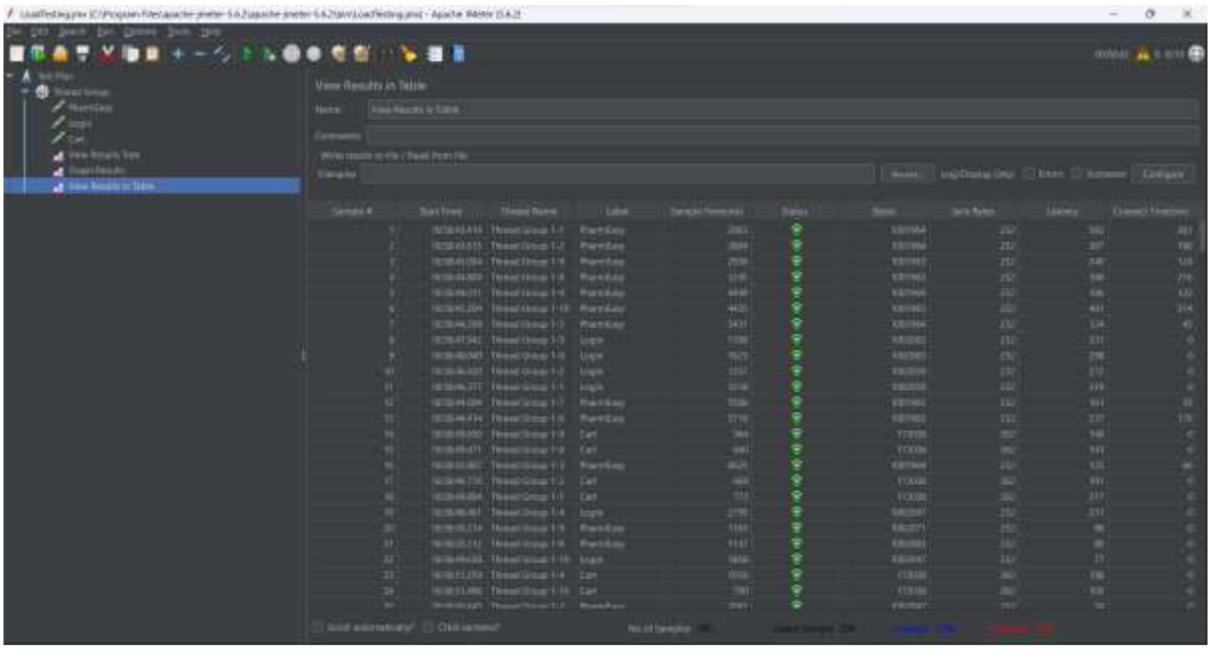
View Result Tree:



Graph Results:



View Results in table:



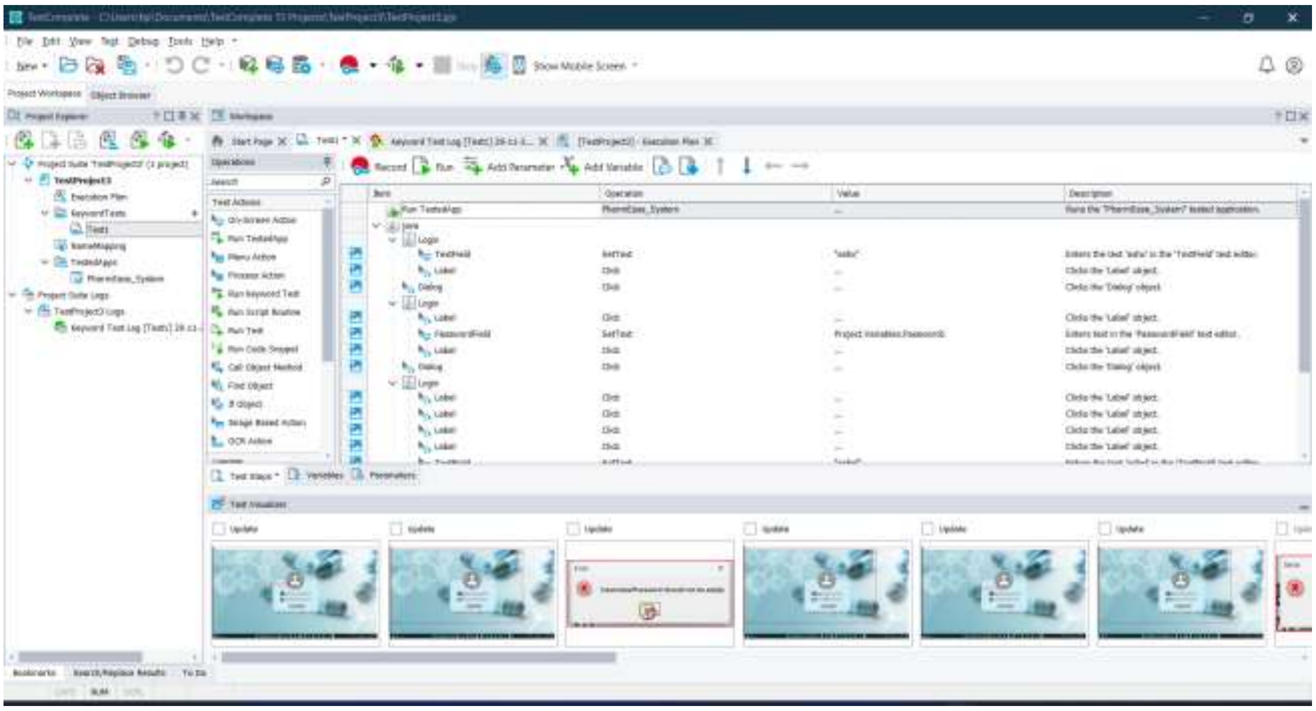
TestComplete: For Automation Testing of GUI

TestComplete is a powerful automation testing tool primarily used for functional, regression, and GUI testing across various applications, including web, desktop, and mobile.

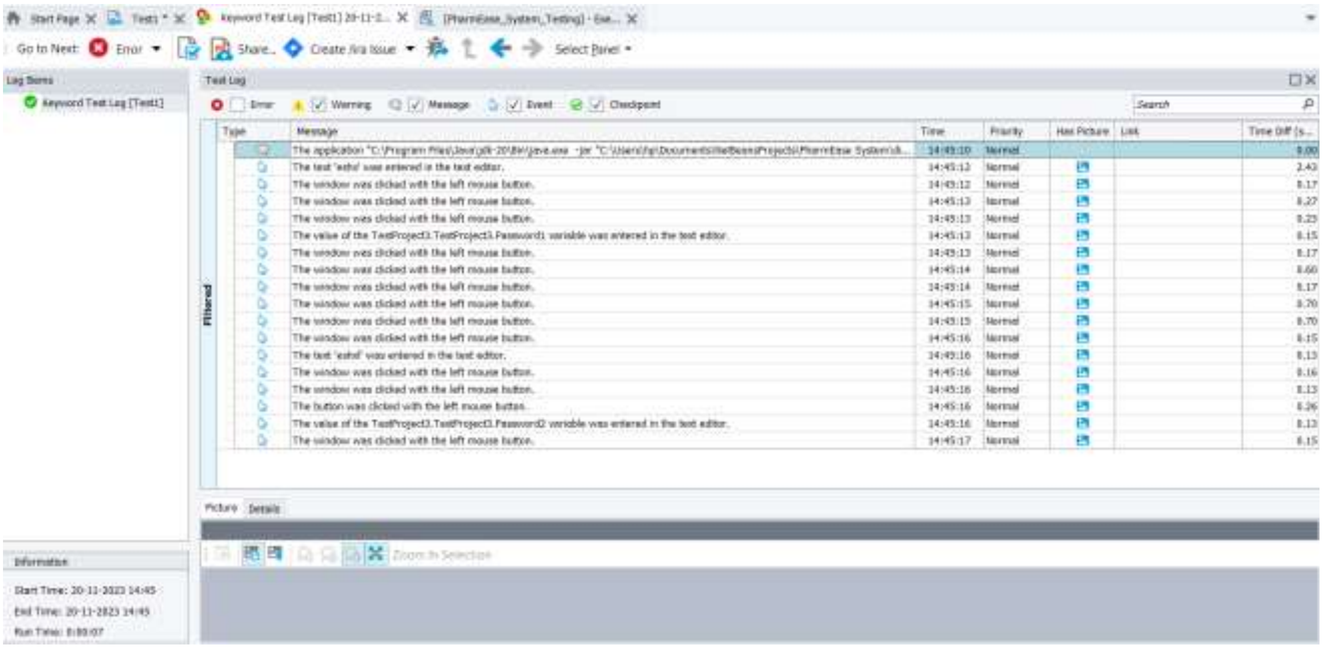
For GUI testing of a Java-based desktop application like PharmEase, TestComplete could be used effectively. It provides:

- Cross-Platform Support: TestComplete can handle GUI testing for multiple platforms, including Windows-based applications like those developed in Java Swing.
- Object Recognition: It excels in recognizing and interacting with GUI elements within the application, enabling testers to automate actions like button clicks, text input, and validations.
- Scripting Flexibility: TestComplete supports scripting languages like JavaScript, Python, and VBScript, providing flexibility in writing test scripts according to your project needs.
- Recording and Playback: Its recording feature allows users to record interactions with the application's GUI and generate corresponding test scripts, making it easier to create automated tests.

1. User Login:



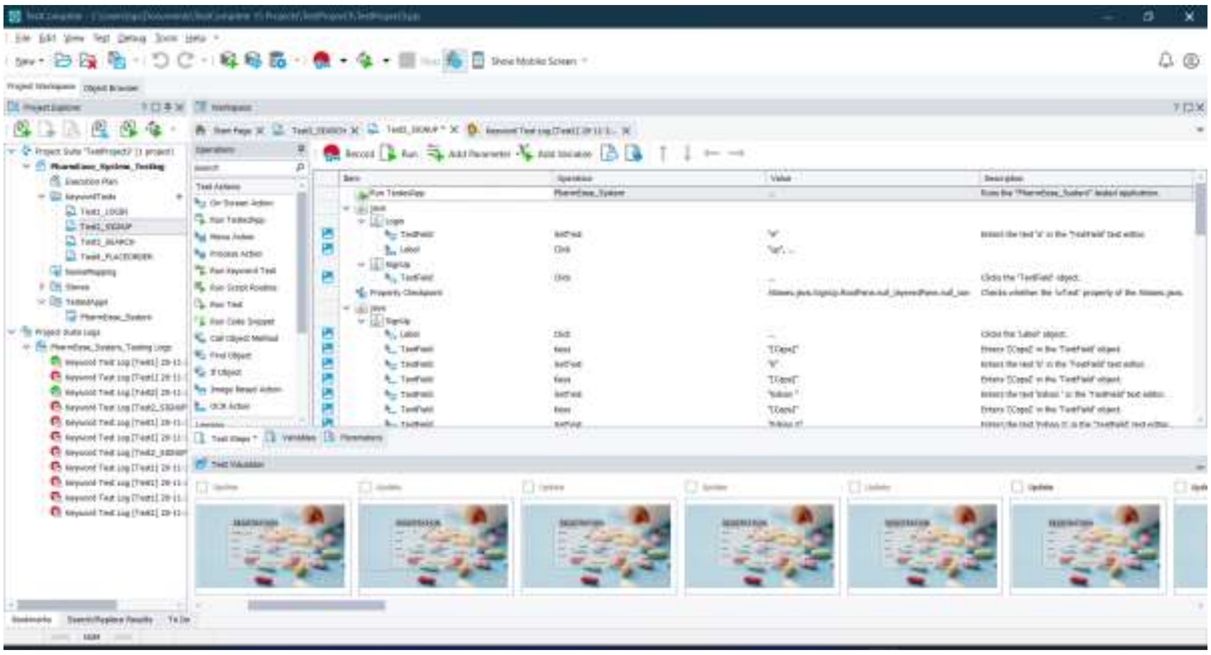
Item	Operation	Value	Description
Run TestedApp	PharmEase_System	...	Runs the "PharmEase_System" tested application.
Java			
Login			
TextField	SetText	"asha"	Enters the text 'asha' in the 'TextField' text editor.
Label	Click	...	Clicks the 'Label' object.
Dialog	Click	...	Clicks the 'Dialog' object.
Login			
Label	Click	...	Clicks the 'Label' object.
PasswordField	SetText	Project.Variables.Password1	Enters text in the 'PasswordField' text editor.
Label	Click	...	Clicks the 'Label' object.
Dialog	Click	...	Clicks the 'Dialog' object.
Login			
Label	Click	...	Clicks the 'Label' object.
Label	Click	...	Clicks the 'Label' object.
Label	Click	...	Clicks the 'Label' object.
Label	Click	...	Clicks the 'Label' object.

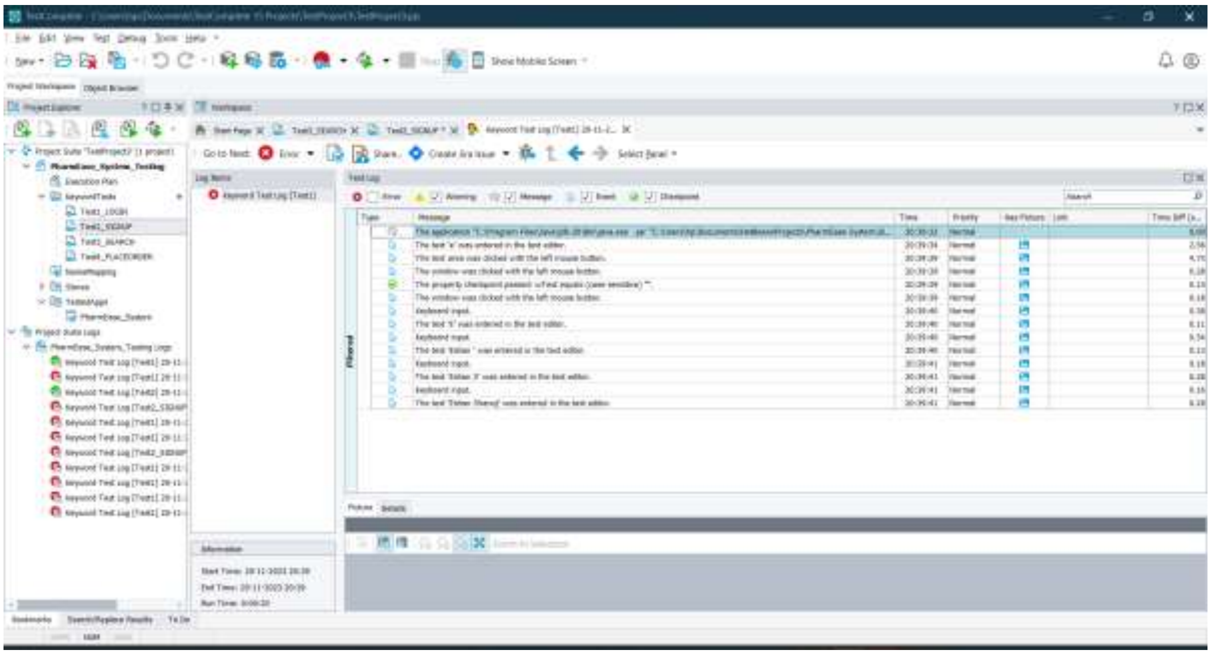


Type	Message	Time	Priority	Has Picture	Link	Time Diff (s...
	The application "C:\Program Files\Java\jdk-20\bin\java.exe -jar "C:\Users\hp\Documents\NetBeansProjects\PharmEase System\d...	14:45:10	Normal			0.00
	The text 'asha' was entered in the text editor.	14:45:12	Normal			2.43
	The window was clicked with the left mouse button.	14:45:12	Normal			0.17
	The window was clicked with the left mouse button.	14:45:13	Normal			0.27
	The window was clicked with the left mouse button.	14:45:13	Normal			0.23
	The value of the TestProject3.TestProject3.Password1 variable was entered in the text editor.	14:45:13	Normal			0.15
	The window was clicked with the left mouse button.	14:45:13	Normal			0.17
	The window was clicked with the left mouse button.	14:45:14	Normal			0.60
	The window was clicked with the left mouse button.	14:45:14	Normal			0.17
	The window was clicked with the left mouse button.	14:45:15	Normal			0.70
	The window was clicked with the left mouse button.	14:45:15	Normal			0.70
	The window was clicked with the left mouse button.	14:45:16	Normal			0.15
	The text 'ashaf' was entered in the text editor.	14:45:16	Normal			0.13
	The window was clicked with the left mouse button.	14:45:16	Normal			0.16
	The window was clicked with the left mouse button.	14:45:16	Normal			0.13
	The button was clicked with the left mouse button.	14:45:16	Normal			0.26
	The value of the TestProject3.TestProject3.Password2 variable was entered in the text editor.	14:45:16	Normal			0.13
	The window was clicked with the left mouse button.	14:45:17	Normal			0.15

Start Time: 20-11-2023 14:45
End Time: 20-11-2023 14:45
Run Time: 0:00:07

2. Registration

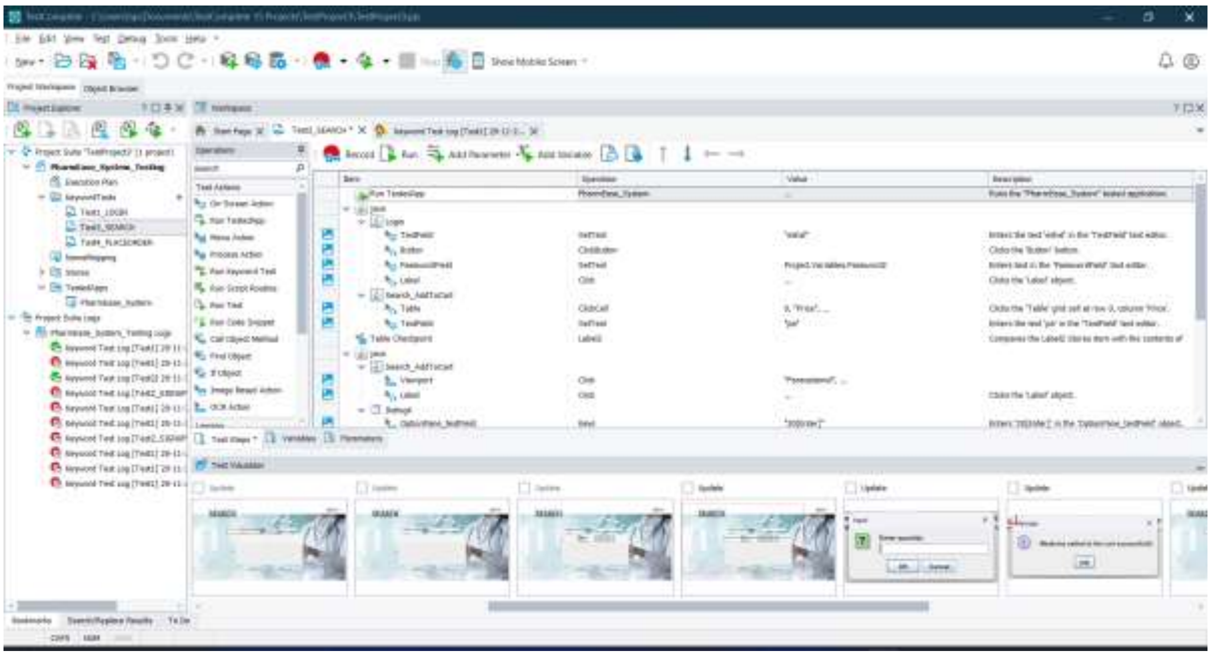




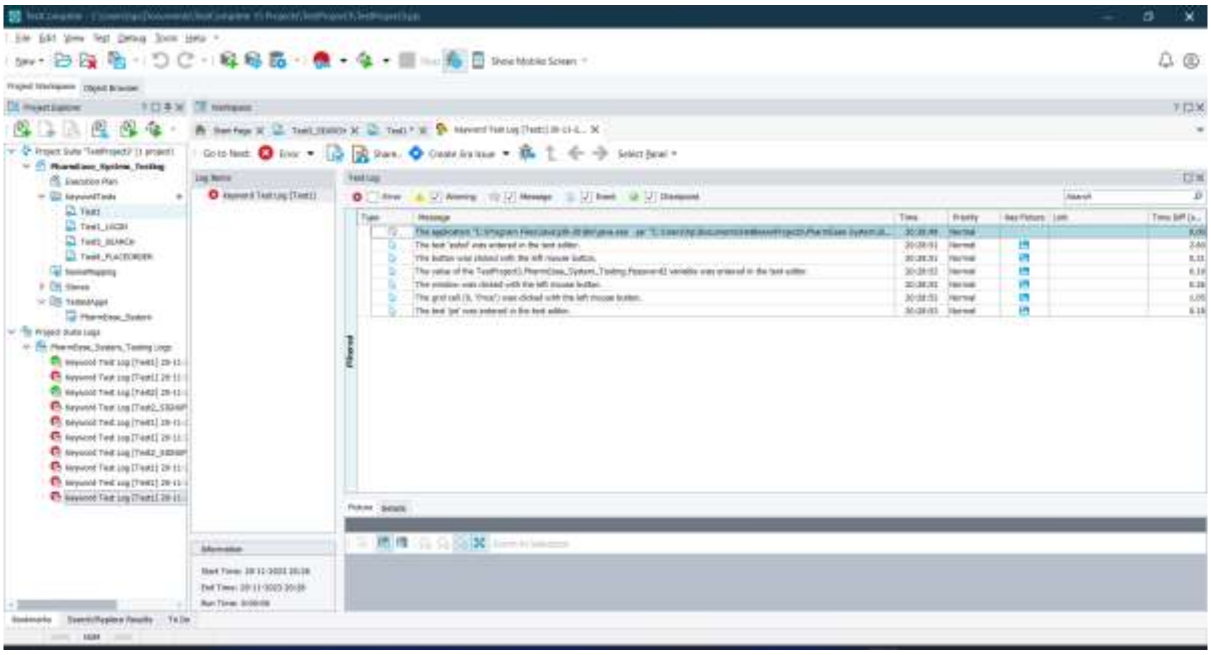
Type	Message	Time	Priority	Has Picture	Link	Time Diff (s...)
	The application "[illegible]" was started.	20:39:32	Normal			[illegible]
	The text 'a' was entered in the text editor.	20:39:34	Normal			2.56
	The text area was clicked with the left mouse button.	20:39:39	Normal			4.70
	The window was clicked with the left mouse button.	20:39:39	Normal			0.28
✓	The property checkpoint passed: [illegible] equals (case sensitive) "".	20:39:39	Normal			0.13
	The window was clicked with the left mouse button.	20:39:39	Normal			0.18
	Keyboard input.	20:39:40	Normal			0.38
	The text 'S' was entered in the text editor.	20:39:40	Normal			0.11
	Keyboard input.	20:39:40	Normal			0.54
	The text 'Sidsas ' was entered in the text editor.	20:39:40	Normal			0.13
	Keyboard input.	20:39:41	Normal			0.18
	The text 'Sidsas 3' was entered in the text editor.	20:39:41	Normal			0.28
	Keyboard input.	20:39:41	Normal			0.16
	The text 'Sidsas Sharaf' was entered in the text editor.	20:39:41	Normal			0.28

Start Time: 28-11-2023 20:39
End Time: 28-11-2023 20:39
Run Time: 0:00:20

3. Search_AddToCart



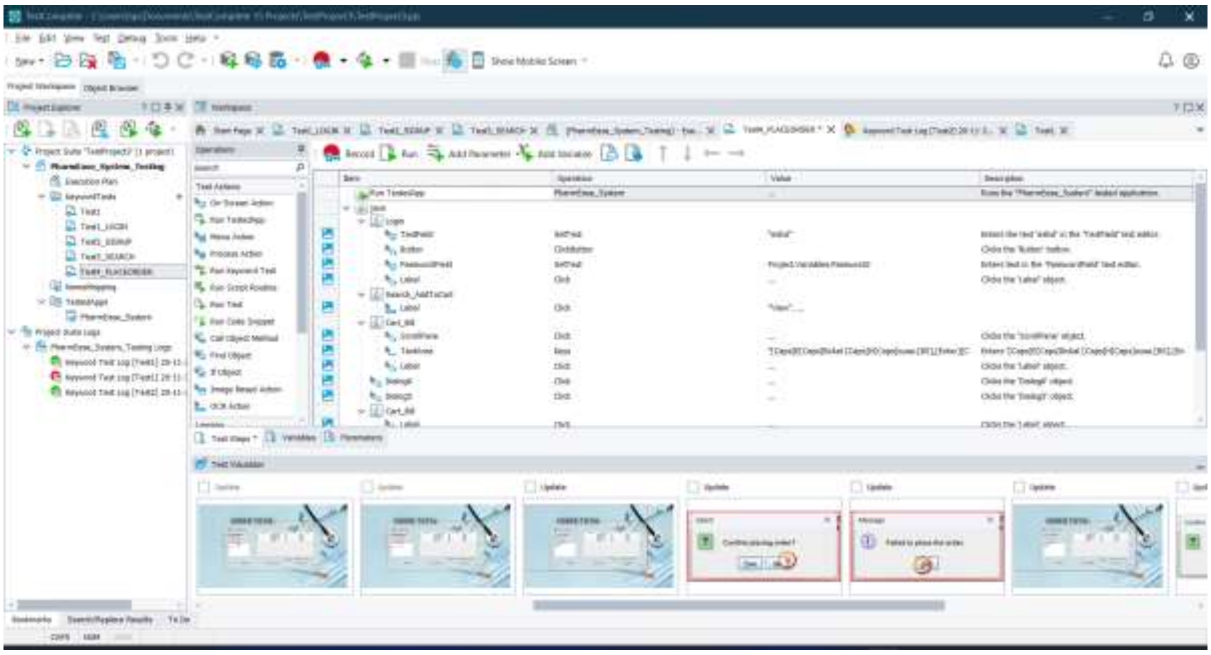
Item	Operation	Value	Description
Run TestedApp	PharmEase_System	...	Runs the "PharmEase_System" tested application.
Java			
Login			
TextField	SetText	"ashal"	Enters the text 'ashal' in the 'TextField' text editor.
Button	ClickButton		Clicks the 'Button' button.
PasswordField	SetText	Project.Variables.Password2	Enters text in the 'PasswordField' text editor.
Label	Click	...	Clicks the 'Label' object.
Search_AddToCart			
Table	ClickCell	0, "Price", ...	Clicks the 'Table' grid cell at row 0, column 'Price'.
TextField	SetText	"pa"	Enters the text 'pa' in the 'TextField' text editor.
Table Checkpoint	Label2		Compares the Label2 stores item with the contents of
Java			
Search_AddToCart			
Viewport	Click	"Paracetamol", ...	
Label	Click	...	Clicks the 'Label' object.
Swing#			
OptionPane_textField	Keys	"20[Enter]"	Enters '20[Enter]' in the 'OptionPane_textField' object.



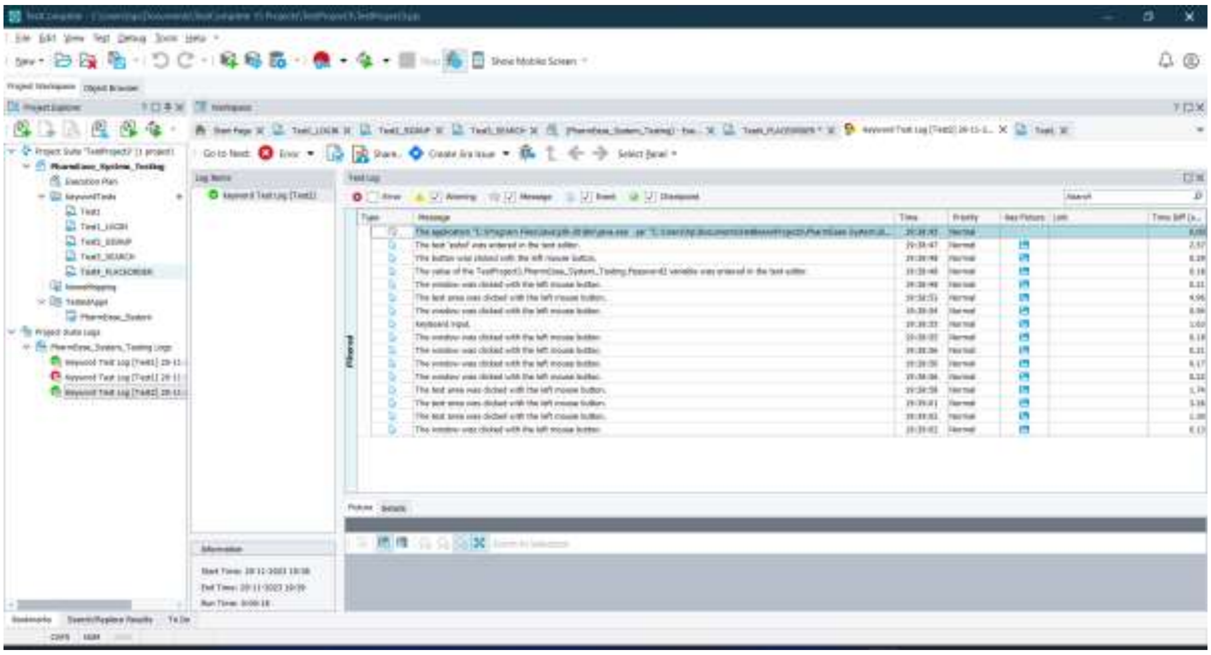
Type	Message	Time	Priority	Has Picture	Link	Time Diff (s...)
	The application "C:\Program Files\Java\jdk-20\bin\java.exe -jar "C:\Users\hp\Documents\...\PharmEase System.ja...	20:28:49	Normal			0.00
	The text 'ashal' was entered in the text editor.	20:28:51	Normal	✓		2.60
	The button was clicked with the left mouse button.	20:28:51	Normal	✓		0.31
	The value of the TestProject1.PharmEase_System_Testing.Password2 variable was entered in the text editor.	20:28:52	Normal	✓		0.19
	The window was clicked with the left mouse button.	20:28:52	Normal	✓		0.28
	The grid cell (0, 'Price') was clicked with the left mouse button.	20:28:53	Normal	✓		1.05
	The text 'pa' was entered in the text editor.	20:28:53	Normal	✓		0.18

Start Time: 29-11-2023 20:28
End Time: 29-11-2023 20:28
Run Time: 0:00:06

4. Cart_Bill



Item	Operation	Value	Description
Run TestedApp	PharmEase_System		Runs the "PharmEase_System" tested application.
Login			
TextField	SetText	"[illegible]"	Enters the text '[illegible]' in the 'TextField' text editor.
Button	ClickButton		Clicks the 'Button' button.
PasswordField	SetText	Project.Variables.Password2	Enters text in the 'PasswordField' text editor.
Label	Click	...	Clicks the 'Label' object.
Search_AddToCart			
Label	Click	"[illegible]", ...	
Cart_Bill			
ScrollPane	Click	...	Clicks the 'ScrollPane' object.
TextArea	Keys	[illegible]	Enters [illegible]
Label	Click	...	Clicks the 'Label' object.
Dialog	Click	...	Clicks the 'Dialog' object.
Dialog	Click	...	Clicks the 'Dialog' object.
Cart_Bill			
Label	Click	...	Clicks the 'Label' object.



Type	Message	Time	Priority
	The application "[illegible]" ...	[illegible]	Normal
	The text '[illegible]' was entered in the text editor.	[illegible]	Normal
	The button was clicked with the left mouse button.	[illegible]	Normal
	The value of the TestProject1.PharmEase_System_Testing.Password2 variable was entered in the text editor.	[illegible]	Normal
	The window was clicked with the left mouse button.	[illegible]	Normal
	The text area was clicked with the left mouse button.	[illegible]	Normal
	The window was clicked with the left mouse button.	[illegible]	Normal
	Keyboard input.	[illegible]	Normal
	The window was clicked with the left mouse button.	[illegible]	Normal
	The window was clicked with the left mouse button.	[illegible]	Normal
	The window was clicked with the left mouse button.	[illegible]	Normal
	The window was clicked with the left mouse button.	[illegible]	Normal
	The text area was clicked with the left mouse button.	[illegible]	Normal
	The text area was clicked with the left mouse button.	[illegible]	Normal
	The text area was clicked with the left mouse button.	[illegible]	Normal
	The window was clicked with the left mouse button.	[illegible]	Normal

Conclusion

Through comprehensive testing methodologies, including black box, JUnit for unit testing, JMeter for performance evaluation, and TestComplete for GUI automation, the testing process for the PharmEase System was exhaustive and thorough. Results obtained from various test scenarios, decision tables, equivalence classes, and automation tools revealed insights into system behavior, identified issues, and confirmed compliance with defined rules and expectations. The visual representations provided by JMeter and TestComplete aided in assessing performance and GUI responsiveness.

In conclusion, this testing phase has contributed significantly to refining and optimizing the PharmEase System, instilling confidence in its robustness and adherence to user-centric functionalities. Identified issues have been addressed, ensuring a smoother, more reliable user experience, thereby meeting the project objectives outlined in the Software Requirements Specification.