MINI PROJECT REPORT

BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)

SUBMITTED BY

Shantanu Mangle	41242
Gaurav Fugat	41219
Chinmay Nikam	41248
Pratik Patil	41251

DEPARTMENT OF COMPUTER ENGINEERING
P.E.S MODERN COLLEGE OF ENGINEERING
PUNE - 411005.

SAVITRIBAI PHULE PUNE UNIVERSITY

[2021 - 22]

Progressive Education Society's Modern College of Engineering

Department of Computer Engineering Shivajinagar, Pune - 411005.

CERTIFICATE

This is to certify that the following students of Final Year Computer Engineering of PES's, Modern College of Engineering have successfully completed the Laboratory Practice II (Mini Project)under the guidance of Mr. Anand Deshmukh.

The Group Members are :

Shantanu Mangle	41242
Gaurav Fugat	41219
Chinmay Nikam	41248
Pratik Patil	41251

This is in partial fulfillment of the award of the degree Bachelor of Computer Engineering of Savitribai Phule Pune University.

Date:

Internal Guide	(Prof. Dr. Mrs. S. A. Itkar)	External Examiner
Mr. Anand Deshmukh	Head Department of Computer Engineering	

Title:

Create a small application by selecting relevant system environment/ platform and programming languages. Narrate concise Test Plan consisting features to be tested and bug taxonomy. Prepare Test Cases inclusive of Test Procedures for identified Test Scenarios. Perform selective Black-box and White-box testing covering Unit and Integration test by using suitable Testing tools. Prepare Test Reports based on Test Pass/Fail Criteria and judge the acceptance of application developed.

Problem Definition:

Perform Desktop Application testing using Automation Tool like JUnit generate Test Report by Using tool like Apache Maven

Objective:

We are going to learn how to Prepare Test Cases inclusive of Test Procedures for identified Test Scenarios. Perform selective Black-box and White-box testing covering Unit and Integration test by using suitable Testing tools. also Prepare Test Reports based on Test Pass/Fail Criteria

Prerequisite:

Knowledge of Core Java, Basic Concepts of Unit Testing, Test Cases Writing using Junit etc.

Software Requirements:

JDK 1.8
Eclipse java photon-R version
TestNG

Hardware requirements:

PIV, 2GB RAM, 500 GB HDD, Lenovo A13-4089Model.

Outcome:

You are able to understand Unit and Integration testing with Tool with Test Report.

Theory:

Unit Testing:

Unit Testing of software applications is done during the development (coding) of an application. The objective of Unit Testing is to isolate a section of code and verify its correctness. In procedural programming a unit may be an individual function or procedure.

The goal of Unit Testing is to isolate each part of the program and show that the individual parts are correct. Unit Testing is usually performed by the developer.

Unit Testing Tools:

Few examples of Unit testing tools :

1. Jtest
2. Junit
3. Nunit
4. JMockit
5. EMMA
6. PHPUnit

Junit:

JUnit is a framework for implementing testing in Java. It provides a simple way to explicitly test specific areas of a Java program, it is extensible and can be employed to test a hierarchy of program code either singularly or as multiple units.

Using a testing framework is beneficial because it forces you to explicitly declare the expected results of specific program execution routes. When debugging it is possible to write a test which expresses the result you are trying to achieve and then debug until the test comes out positive. By having a set of tests that test all the core components of the project it is possible to modify specific areas of the project and immediately see the effect the modifications have on the other areas by the results of the test, hence, side-effects can be quickly realized.

JUnit promotes the idea of first testing then coding, in that it is possible to setup test data for a unit which defines what the expected output is and then code until the tests pass. It is believed by some that this practice of "test a little, code a little, test a little, code a little..." increases programmer productivity and stability of program code whilst reducing programmer stress and the time spent debugging.

JUnit is a simple open-source Java testing framework used to write and run repeatable automated tests. It is an instance of the xUnit architecture for unit testing framework. Eclipse supports creating test cases and running test suites, so it is easy to use for your Java applications.

JUnit features include:

- Assertions for testing expected results
- Test fixtures for sharing common test data
- Test suites for easily organizing and running tests
- Graphical and textual test runners

Application Description:

Tools used in construction of application

1. Java Script
2. MySQL
3. PHP
4. Swing
5. Apache

Our application name is Car Rental Management System. The application can be used to rent or mange cars in inventory.
Application has four main functions:

1. Add car
2. Customer
3. Rent Car
4. Return car

1. Add car:
   In add car function the rental store owners can add the cars. These cars can be rented out to customers. While adding car the store owner has to fill all detail of their car.

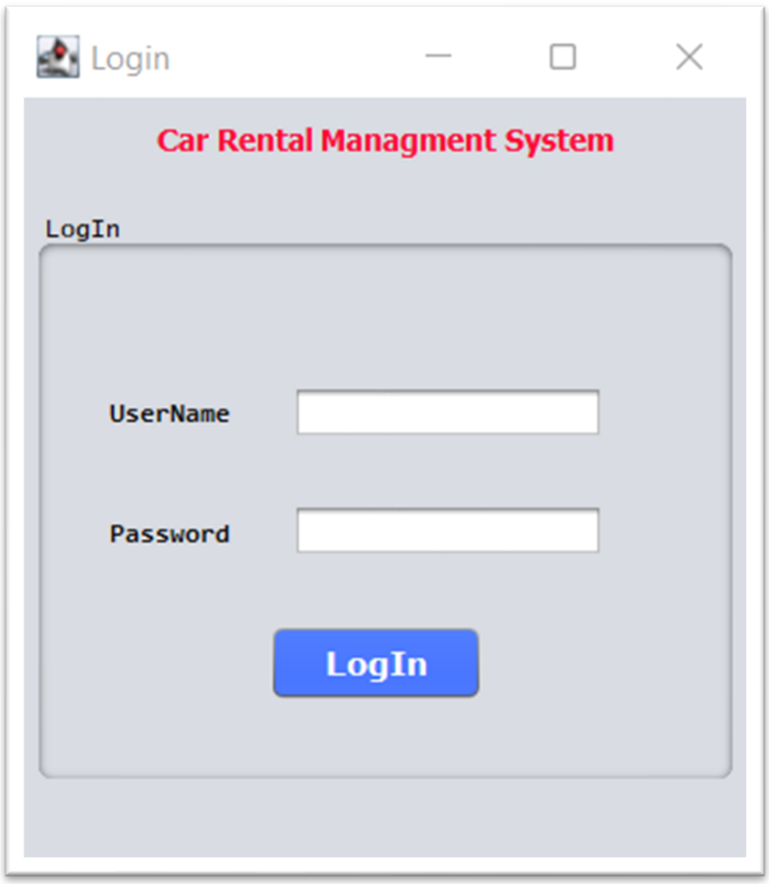
2. Customer:
   For renting a car one needs the account. In customer function new customer can crate their accounts. They must have to fill their all-needed information.

3. Rent car:
   In rent car function store owner can rent car to a customer. For renting the car store owner must create customer's account. While renting a car customer must have to specify renting date and turning back date of car.

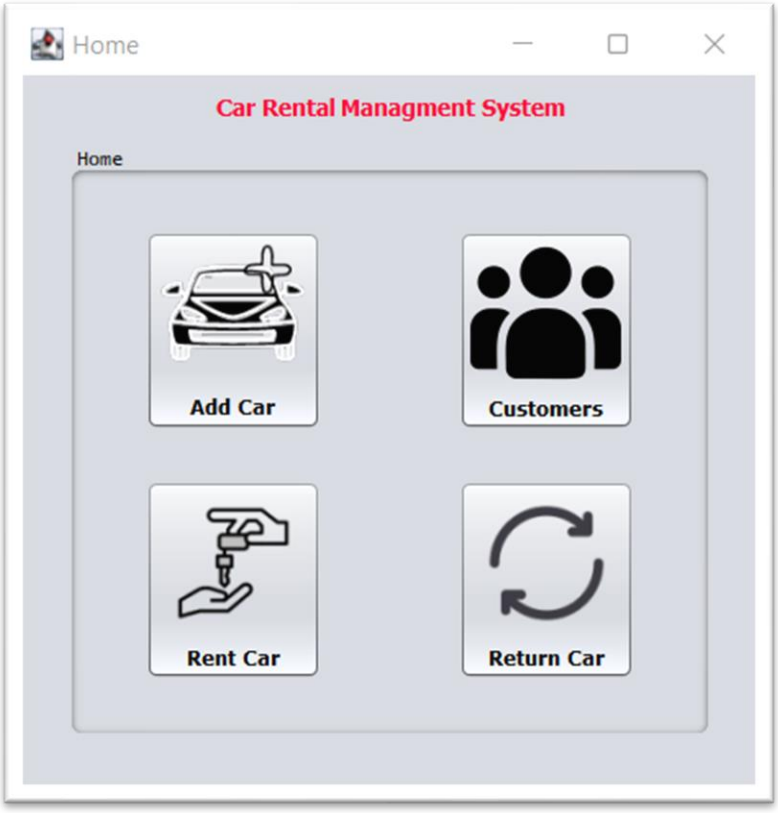
4. Return car:
   After renting the car when customer needs to return the car, store owner can use this function. At the time of returning, store owner has to provide car id. If customer is late for returning the car also must have to pay late fee.

Sample Screenshots of Application:

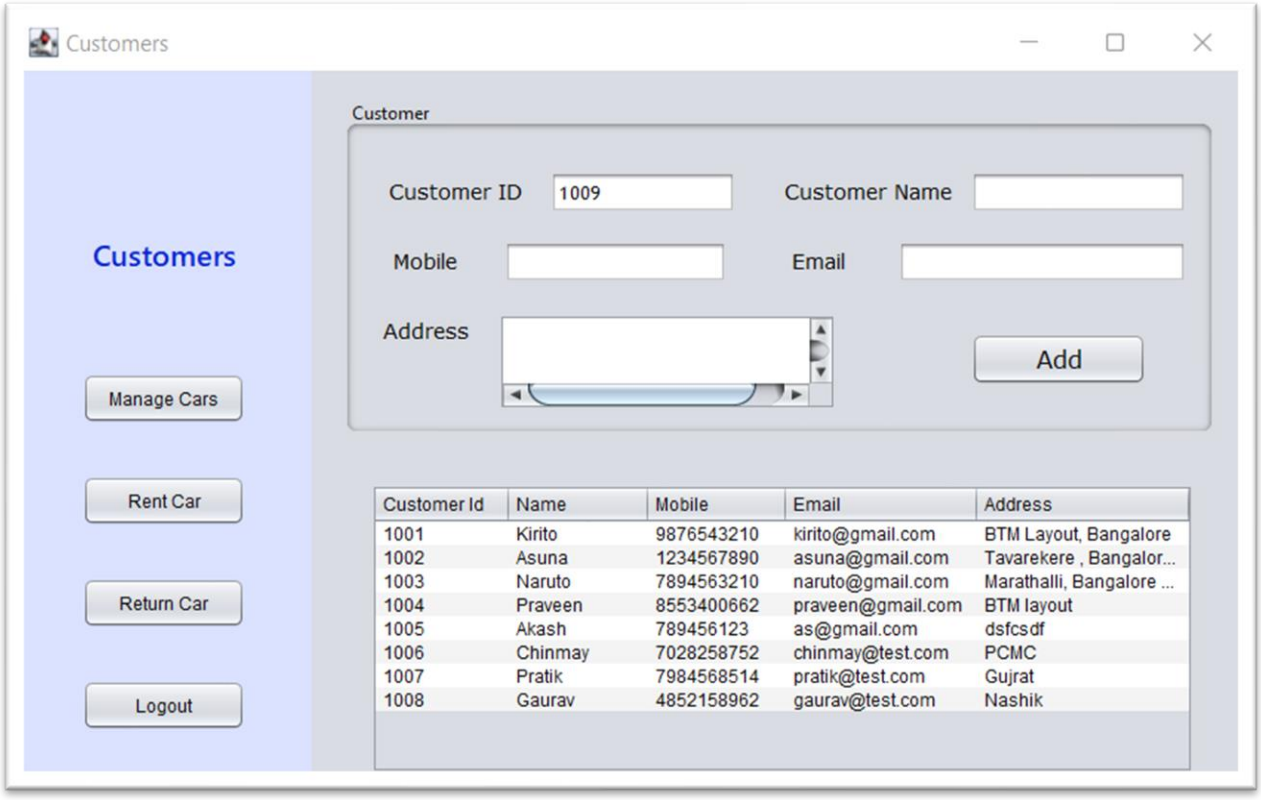
1. **Login Page:**



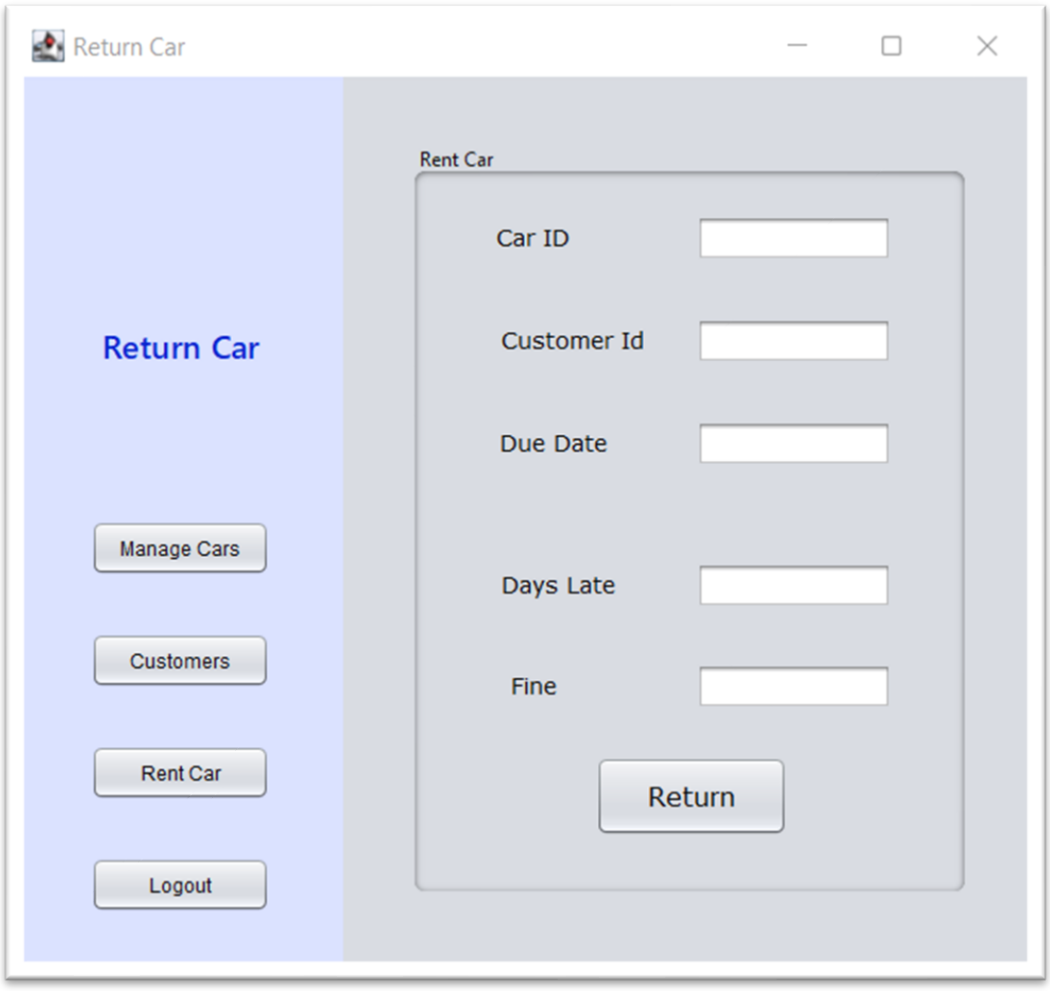
2. **Home Page:**



3. Add Customer Page:



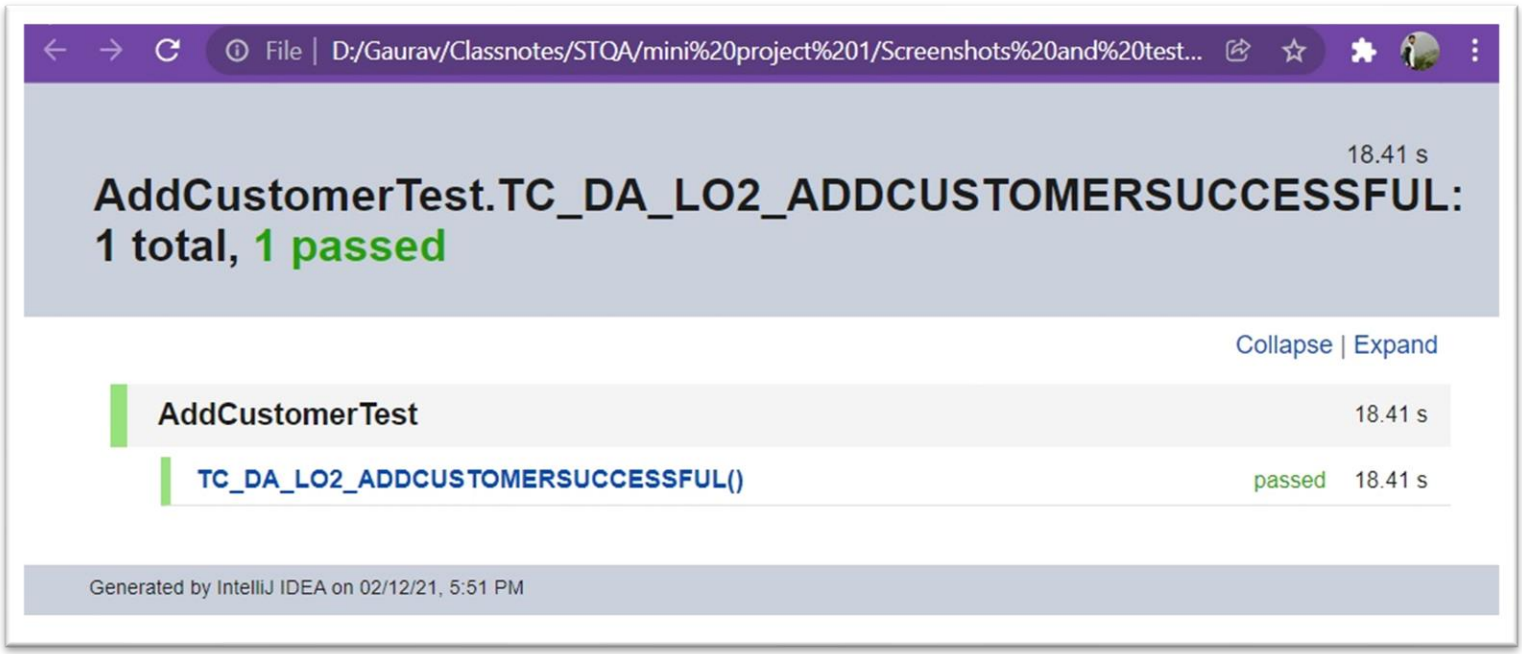
4. Return Page:



Output Logs of Sample Tests:

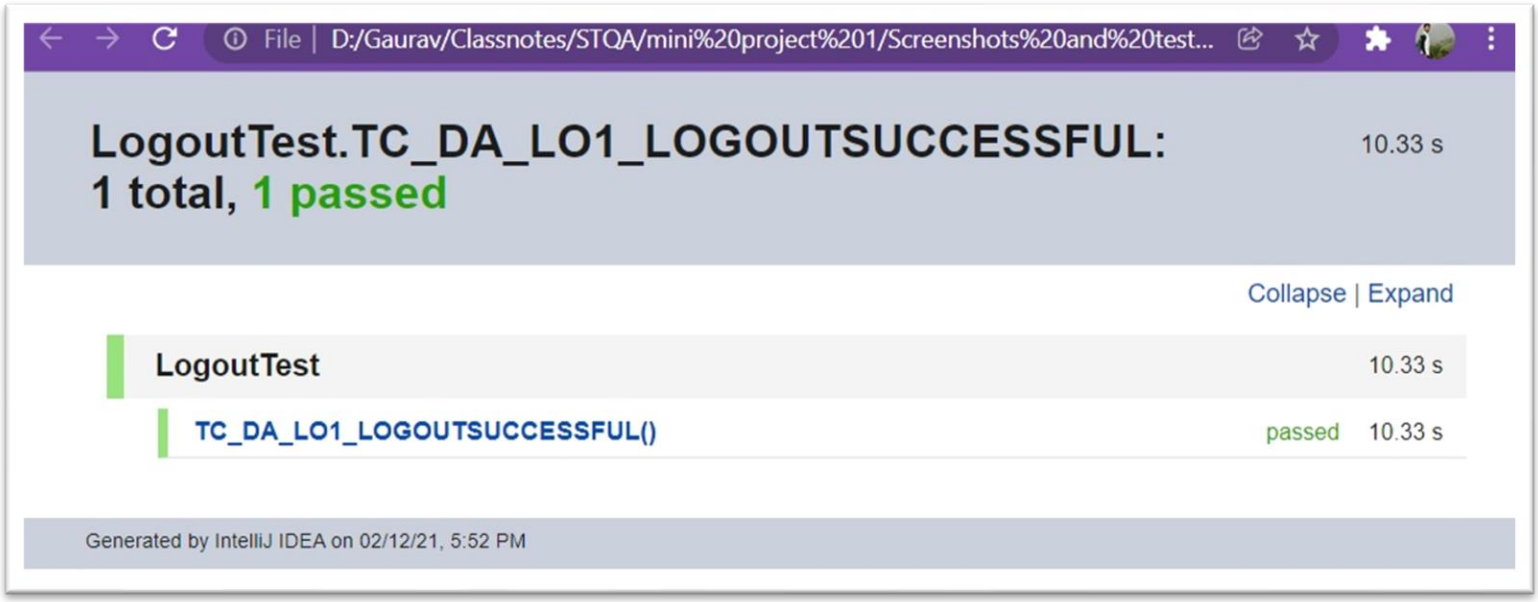
1. Login Test:

2. Add Customer :



3. Return Car Test:

4. Log Out Test:



Conclusion:

In this way we performed automation testing using Junit on a self-developed application and verified that no bugs or defects were found.