**Practical – 08**  
**Designing Test Suites for Garage Management System**

**Software Testing** | **Need for Software Testing** | **Testing Frameworks** | **Test Cases** | **Types of Software Testing** | **Unit Testing** | **Integration Testing** | **Write Test Cases**

**Aim of the Experiment**

To design and implement test suites and test cases for the Garage Management System to ensure its functionality, correctness, and performance.

**Introduction**

Testing is a crucial phase in software development that ensures the system works as expected. In this experiment, various test cases are designed and executed on different modules of the Garage Management System such as customer management, service booking, invoice generation, and payment processing.

**Objectives**

After completing this experiment, you will be able to:

* Understand various software testing techniques and methodologies.
* Design and apply unit and integration test cases.
* Identify and fix bugs or errors in the Garage Management System.
* Ensure all components of the system work properly when integrated.

**Theory**

**Software Testing**

Software testing is the process of evaluating and verifying that a software product or application does what it is supposed to do. It helps in identifying defects or issues in the application.

**Verification and Validation**

* **Verification**: Ensures the product is built correctly according to specifications.
* **Validation**: Ensures the correct product is built that fulfills user needs.

**Standards for Software Test Documentation**

* Test Plan
* Test Case Document
* Test Data
* Bug Report

**Testing Frameworks**

Testing frameworks provide a structured way to write and execute tests. Commonly used frameworks include JUnit, PyTest, Selenium, etc.

**Need for Software Testing**

* To ensure the software works as expected
* To find and fix bugs before delivery
* To maintain software quality
* To validate customer requirements

**Test Cases and Test Suite**

* **Test Case**: A single unit of testing with specific inputs and expected outputs.
* **Test Suite**: A collection of test cases that validate the behavior of a software program.

**Types of Software Testing**

1. **Unit Testing**
   * Testing individual components or functions.
   * Example: Testing bookService() method in Service class.
2. **Integration Testing**
   * Testing combined parts of the application to check their interactions.
   * Example: Verifying how Customer and Service interact when a booking is made.
3. **System Testing**
   * Testing the complete integrated system as a whole.
   * Example: Testing the entire workflow from customer login to invoice generation and payment.

**Sample Test Cases**

| **Test Case ID** | **Module** | **Description** | **Input** | **Expected Output** | **Status** |
| --- | --- | --- | --- | --- | --- |
| TC01 | Customer Login | Valid login credentials | Username: "admin", Pass: "123" | Redirect to dashboard | Pass |
| TC02 | Booking | Book service with valid customer ID | Customer ID: 1, Type: Oil Change | Booking confirmation | Pass |
| TC03 | Invoice | Generate invoice for completed job | Service ID: 101 | Invoice with correct amount | Pass |
| TC04 | Payment | Make payment using UPI | Invoice ID: 501, UPI method | Payment success confirmation | Pass |

**Conclusion**

Through this practical, we learned how to design and execute different types of test cases for a real-world application like the Garage Management System. By using unit and integration testing techniques, we ensured that individual components and the system as a whole perform as expected, leading to a reliable and bug-free software system.