Practical NO : 8

Aim: Designing Test Suites – Software Testing

Introduction

Software testing is an essential activity in the software development lifecycle that ensures the reliability, accuracy, and efficiency of the system. It involves executing a program with the intent of finding errors and verifying that the system meets specified requirements.

A **test suite** is a collection of test cases that are designed to validate the functionality, performance, and usability of a software application. By designing effective test suites, software engineers can systematically detect defects, reduce risks, and deliver high-quality products.

In this practical, we focus on **designing test suites for the Agriculture Product Management System (APMS)**, which manages entities such as farmers, companies, products, orders, and loans.

Objective

- To understand the concept of software testing and test suite design.
- To create well-structured test cases for the Agriculture Product Management System.
- To ensure correctness, completeness, and reliability of the system.
- To identify defects early and validate system requirements.

Theory

- **Software Testing:** The process of evaluating software to detect the difference between expected and actual outcomes.
- **Test Case:** A set of inputs, execution conditions, and expected results developed for a particular objective.
- **Test Suite:** A collection of test cases intended to test a program to ensure that it behaves as expected.
- **Types of Testing:**
    - *Unit Testing* – Testing individual modules.
    - *Integration Testing* – Testing combined modules as a group.
    - *System Testing* – Testing the complete application.
    - *Acceptance Testing* – Ensuring the system meets user needs.

Composition

A test suite generally includes:

1. **Test Case ID** – Unique identifier for each test.
2. **Test Case Description** – Purpose of the test.
3. **Input Data** – Test data values to be used.
4. **Execution Steps** – Steps to perform the test.
5. **Expected Result** – The correct behavior/output.
6. **Actual Result** – The observed system response.
7. **Status (Pass/Fail)** – Outcome of test execution.

Case Study: Agriculture Product Management System

📌 Sample Test Suite

Test Case ID	Description	Input Data	Execution Steps	Expected Result	Status
TC_01	Login with valid credentials	Username: farmer1, Password: 12345	1. Open login page 2. Enter username/password 3. Click login button	Dashboard should be displayed	Pass
TC_02	Login with invalid password	Username: farmer1, Password: wrongPass	1. Open login page 2. Enter invalid credentials 3. Click login button	Error message "Invalid credentials" appears	Pass
TC_03	Add new product	Product Name: Wheat, Type: Crop	1. Go to Add Product page 2. Fill form 3. Submit	Product stored in database	Pass
TC_04	Place an order	Farmer ID: F01, Product ID: P02, Qty: 5	1. Select product 2. Enter quantity 3. Confirm order	Order created and stored in database	Pass
TC_05	Request loan	Loan Amount: 50,000, Loan Date: 01-01-25	1. Go to Loan page 2. Fill loan details 3. Submit	Loan record added successfully	Pass
TC_06	Search product	Keyword: "Rice"	1. Enter keyword in search bar 2. Click search	List of products with name "Rice" displayed	Pass

References

1. Myers, Glenford J. *The Art of Software Testing*. Wiley.
2. Sommerville, Ian. *Software Engineering (10th Edition)*. Pearson.
3. Pressman, Roger S. *Software Engineering: A Practitioner's Approach*. McGraw-Hill.
4. IEEE Standard for Software Test Documentation (IEEE 829).