EXPERIMENT NO. 9

Department of Computer Engineering

Academic Term: First Term 2023-24

Class: T.E /Computer Sem – V / Software Engineering

Practical No:	10
Title:	**Design test cases for performing white box testing**
Date of Performance:	
Roll No:	9567, 9552,9623
Team Members:	Shruti Patil, Mrunal Kotambkar,Dhruv Mayekar

Rubrics for Evaluation:

Sr. No	Performance Indicator	Excellent	Good	Below Average	Total Score
1	On time Completion & Submission (01)	01 (On Time)	NA	00 (Not on Time)	
2	Theory Understanding(02)	02(Correct)	NA	01 (Tried)	
3	Content Quality (03)	03(All used)	02 (Partial)	01(rarely followed)	
4	Post Lab Questions (04)	04(done well)	3 (Partially Correct)	2(submitted)	

Signature of the Teacher:

Department of Computer Engineering

Academic Term: First Term 2022-23

RISK INFORMATION SHEET

Name	**Roll No**
Shruti Patil	9567
Mrunal Kotambkar	9552
Dhruv Mayekar	9623

Risk Information Sheet

Risk id: P02-8-25	Date: 15/09/23 Prob: 70% Impact: High

Description:
As the volume of e-waste increases, the e-waste management software may face scalability challenges, resulting in decreased performance and operational disruptions. This risk may occur due to an influx of data, devices, and users that the software was not originally designed to handle.

Refinement/Context:
This issue is especially pertinent in areas or businesses where e-waste levels are quickly increasing. It may be aggravated further if the software has not been updated or optimized to handle increasing loads.

Mitigation/monitoring:

1. Regular Capacity Assessment: Continuously monitor the software's performance metrics, such as response times, server loads, and database query times, to detect early signs of scalability issues.
2. Load Testing: Perform load testing at regular intervals to assess how the software behaves under heavy loads and ensure it can handle growing volumes.
3. Scalability-Ready Design: Ensure that the software architecture is designed to be scalable, making use of modern technologies and frameworks that can expand as needed.

Management/contingency plan/trigger:

If scalability issues are detected, trigger the following actions:

1. Identify the specific performance bottlenecks.
2. Allocate additional hardware resources, such as servers and storage, to accommodate the increased load.
3. Optimize database queries and code to improve software efficiency.
4. Evaluate and, if necessary, upgrade the software to a more scalable version or design.
5. Regular Updates: Implement a proactive strategy for software updates and optimizations to accommodate growth and evolving technology.
6. Load Threshold Triggers: Define load threshold triggers that will automatically alert the IT team to take action before scalability issues affect operations.
7. Continuous Monitoring: Continuously monitor the software's performance, especially during periods of increased e-waste intake, to identify scalability issues before they become critical.

Current Status:

18/09/23: Mitigation steps initiated.

Originator: Aaron Pereira Assigned: Alroy Pereira