Experiment No. 7

SEMESTER: V (2024-2025) **DATE OF DECLARATION:** 23/09/24

SUBJECT: SE **DATE OF SUBMISSION:** 30/09/24

NAME OF THE STUDENT: Shaun Menezes **ROLL NO:** 40

AIM	Design test cases for "**Inefficiency in Timely Announcement of Holidays for Educational Institutions in Flood-Prone Areas**"
LEARNING OBJECTIVE	The student will understand the need and purpose of designing test cases for "Inefficiency in Timely Announcement of Holidays for Educational Institutions in Flood-Prone Areas"
LEARNING OUTCOME	The student will create test cases for "Inefficiency in Timely Announcement of Holidays for Educational Institutions in Flood-Prone Areas"
COURSE OUTCOME	**CSL501.6:** Students will be able to implement and present a case study based on software engineering concept.
PROGRAM OUTCOME	**PO1:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems. **PO2**: **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences. **PO3:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations. **PO4: Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions. **PO5: Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations. **PO8: Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice. **PO9:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings. **PO10: Communication**: Communicate effectively on complex engineer-

	ing activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions. **PO11: Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
BLOOM'S TAXONOMY LEVEL	Apply Create
THEORY	**White-box testing technique for test case design:** White-box testing, sometimes called glass-box testing, is a test-case design philosophy that uses the control structure described as part of component-level design to derive test cases. Using white-box testing methods, you can derive test cases that (1) guarantee that all independent paths within a module have been exercised at least once, (2) exercise all logical decisions on their true and false sides, (3) execute all loops at their boundaries and within their operational bounds, and (4) exercise internal data structures to ensure their validity. The types of white box testing are: 1. Basis Path Testing 2. Control structure Testing **1. Basis Path Testing:** Basis path testing is a white-box testing technique first proposed by Tom McCabe. The basis path method enables the test-case designer to derive a logical complexity measure of a procedural design and use this measure as a guide for defining a basis set of execution paths. Test cases derived to exercise the basis set are guaranteed to execute every statement in the program at least one time during testing. **2. Control Structure Testing:** The basis path testing technique is one of a number of techniques for control structure testing. Although basis path testing is simple and highly effective, it is not sufficient in itself. Other variations on control structure testing are discussed. These broaden testing coverage and improve the quality of white-box testing. The types of control structure testing are condition testing, data flow testing and loop testing.

LAB EXERCISE	The Test Cases for the problem statement “Inefficiency in Timely Announcement of Holidays for Educational Institutions in Flood-Prone Areas” are: 1. Test Case: Detection of Flood-Prone Areas Objective: Verify that the system correctly identifies a flood-prone area based on predefined geographical data. • Input: Location data and weather conditions indicating a flood-prone area. • Expected Result: System should flag the area as flood-prone. • White Box Coverage: Control flow and conditional logic in the flood detection algorithm. 2. Test Case: Flood Severity Calculation Objective: Ensure the correct calculation of flood severity based on sensor data (e.g., rainfall, water level). • Input: Sensor data showing different levels of water. • Expected Result: System should classify the severity level correctly (e.g., low, medium, high). • White Box Coverage: Path testing in the flood severity classification function. 3. Test Case: Announcement Trigger on High Flood Severity Objective: Verify that a holiday announcement is triggered when the flood severity exceeds a critical threshold. • Input: Flood severity = High. • Expected Result: System should trigger an announcement for affected educational institutions. • White Box Coverage: Branch coverage of conditional statements that determine whether to issue an announcement. 4. Test Case: No Announcement on Low Flood Severity Objective: Ensure that no holiday announcement is triggered when flood severity is low. • Input: Flood severity = Low. • Expected Result: System should not issue any holiday announcement. • White Box Coverage: Negative path testing for the announcement trigger condition.

5. Test Case: Holiday Announcement Message Formatting

Objective: Test the proper construction and formatting of holiday announcement messages.

- Input: Institution name, date of holiday, reason (flood).
- Expected Result: Correctly formatted announcement message with all details.
- White Box Coverage: String concatenation and message formatting logic.

6. Test Case: Announcement Targeting Affected Institutions Only

Objective: Ensure that only educational institutions within the affected flood-prone areas receive the holiday announcement.

- Input: List of educational institutions within and outside the affected area.
- Expected Result: Only the relevant institutions receive the holiday announcement.
- White Box Coverage: Conditional branching to check location match between flood-prone area and institution addresses.

7. Test Case: Multiple Flood Zones

Objective: Test the system's ability to handle multiple flood-prone zones and correctly issue announcements for each zone.

- Input: Multiple flood-prone zones detected, with corresponding institutions.
- Expected Result: Announcements should be correctly triggered for institutions in each respective zone.
- White Box Coverage: Loop and iteration over multiple flood zones and institutions.

8. Test Case: Timely Announcement Dispatch

Objective: Verify that the holiday announcement is dispatched within a specified time after flood detection.

- Input: Flood detection time = T.
- Expected Result: Announcement should be dispatched within the specified time frame, e.g., T + 30 minutes.
- White Box Coverage: Time-based logic and delay functions in the announcement system.

9. Test Case: Redundant Announcements

Objective: Ensure that multiple announcements for the same institution within a short period are avoided.

- Input: Multiple flood triggers for the same institution within 1 hour.
- Expected Result: System should issue only one holiday announcement.
- White Box Coverage: Logic that handles duplicate detection and prevention.

10. Test Case: System Recovery After Failure

Objective: Ensure the system can recover and continue to issue announcements after a failure (e.g., server crash).

- Input: Simulate a system failure after flood detection, followed by a system restart.
- Expected Result: System should still issue pending holiday announcements after recovery.
- White Box Coverage: Exception handling and recovery mechanisms.

Group Details

Sr. No.	Roll No.	Name of the Batch	Name of the Student
1.	34	B	Aibal Biju
2.	35	B	Ramya Kulkarni
3.	40	B	Shaun Menezes

REFERENCES

1. Roger Pressman, Software Engineering: A Practitioners Approach, (7th Edition), McGraw Hill
2. https://www.softed.com/assets/Uploads/Resources/Software Testing/White-box-testing-paper-Sharon-Robson.pdf
3. http://www.softwaretestingtimes.com/2010/04/white-box-testing-simplified.html