Experiment No. 8

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	To 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 engineering 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	**Black-box testing technique for test case design:** **Black box testing** is a software testing technique where the internal workings of the application are not known to the tester. Instead, the tester focuses on the functionality and the expected behavior of the software based on input and output. The goal is to ensure that the system behaves as expected without any knowledge of the code structure, implementation, or internal paths. **Key Aspects of Black Box Testing:** • **Focus on functionality:** It tests what the software does, rather than how it does it. • **Input and output-driven:** Testers provide inputs, and check if the outputs are as expected. • **No knowledge of internal code:** The tester is not concerned with the internal structure or logic of the application. • **Used in all testing levels:** It can be applied in unit, integration, system, and acceptance testing. • **Test cases based on requirements:** Test cases are designed based on the software requirements and specifications. **The types of black box testing are:** 1. **Functional Testing** • **Explanation:** This is the most common form of black box testing, where the functionality of the software is tested by providing inputs and comparing the outputs with expected results. It ensures that the software meets its specified requirements. • **Example:** In an e-commerce application, functional testing may include testing the login feature by entering valid

	credentials and verifying that the user is logged in successfully. 2. **Boundary Value Analysis (BVA)** • **Explanation:** This type of testing focuses on the boundary values of input domains. Testers check for errors at the boundary of input ranges (just inside, on the boundary, and just outside the boundaries). This helps in identifying edge cases. • **Example:** If a field accepts numbers between 1 and 100, the test cases would focus on values like 0, 1, 100, and 101. 3. **Equivalence Partitioning (EP)** • **Explanation:** Equivalence partitioning divides input data into partitions (equivalence classes) where the system is expected to behave similarly. Instead of testing all possible inputs, testers can choose one representative from each partition to reduce the number of test cases. • **Example:** For an age field accepting values from 18 to 60, you can divide inputs into three partitions: below 18, between 18-60, and above 60. Testing one value from each partition is sufficient to assess the system's behavior for that range. 4. **Decision Table Testing** • **Explanation:** Decision table testing is used when there are multiple combinations of inputs that result in different outputs. The decision table lists all possible conditions and actions, allowing for thorough testing of complex decision-making logic. • **Example:** A loan approval system may have multiple conditions like credit score, income, and age. A decision table is used to test all possible combinations of these conditions and ensure the system behaves correctly. 5. **State Transition Testing** • **Explanation:** State transition testing is used when a system changes its state based on certain events or conditions. The testing involves validating the transitions from one state to another and ensuring the system behaves correctly in each state. • **Example:** A vending machine may move between different states like idle, accepting money, and dispensing items. This technique tests whether the machine transitions correctly

	between these states based on user inputs. 6. **Error Guessing** • **Explanation:** Error guessing is based on the tester's experience and intuition to identify potential error-prone areas in the system. The tester uses domain knowledge and experience to guess the possible inputs that may cause failures. • **Example:** In a form where users input their email address, a tester might guess that entering an invalid email format (like missing an "@" symbol) could cause an error. 7. **Use Case Testing** • **Explanation:** Use case testing ensures that the application behaves as expected based on its use cases or scenarios. Each use case defines a sequence of interactions between the user and the system, and the test verifies that the system functions correctly in each scenario. • **Example:** In a banking application, a use case may describe the process of transferring funds between accounts. Testers will verify if this sequence of steps is executed properly according to the defined use case.
LAB EXERCISE	1. Test Case: Detecting Flood Conditions Objective: Ensure that the system detects flood conditions accurately. • Input: Location = Flood-prone area, Weather = Heavy rain, River level = Critical. • Expected Result: Flood conditions should be detected, triggering further actions. • Black Box Test Type: Functional test. 2. Test Case: Holiday Announcement for Flood-Prone Institutions Objective: Ensure that a holiday announcement is made when a flood condition is detected. • Input: Flood-prone area identified, Water level = High. • Expected Result: Holiday announcement should be issued for institutions in that area. • Black Box Test Type: Functional test. 3. Test Case: No Announcement for Non-Flood-Prone Areas

	Objective: Verify that no holiday announcement is issued for educational institutions in unaffected areas. • Input: Location = Not flood-prone, Weather = Clear. • Expected Result: No holiday announcement should be made. • Black Box Test Type: Negative test. 4. Test Case: Correctness of Holiday Announcement Objective: Verify that the content of the holiday announcement is correct. • Input: Institution name, Date = 15th July, Reason = Flood, Holiday = 1 day. • Expected Result: The announcement should contain the correct institution name, date, and reason. • Black Box Test Type: Validation test. 5. Test Case: Timeliness of Holiday Announcement Objective: Ensure that the holiday announcement is sent out in a timely manner. • Input: Flood detected at 10:00 AM. • Expected Result: Holiday announcement should be sent out within 30 minutes, i.e., by 10:30 AM. • Black Box Test Type: Performance test. 6. Test Case: Handling Multiple Institutions in Flood-Prone Area Objective: Ensure that all institutions in the affected flood-prone area receive the holiday announcement. • Input: Flood-prone area = Zone A, Institutions = School 1, School 2, School 3. • Expected Result: All three institutions should receive the holiday announcement. • Black Box Test Type: Functional test. 7. Test Case: Incorrect Location Data Objective: Test how the system handles incorrect or invalid location data. • Input: Location = Invalid location coordinates. • Expected Result: System should return an error or not issue an announcement. • Black Box Test Type: Error-handling test. 8. Test Case: Announcement Repetition Prevention

Objective: Ensure that duplicate announcements are not sent to the same institution within a short period.

- Input: Flood detected at 10:00 AM, another flood detection at 10:05 AM for the same institution.
- Expected Result: Only one announcement should be sent for the 10:00 AM detection.
- Black Box Test Type: Validation test.

9. Test Case: Graceful Handling of System Overload

Objective: Test the system's behavior when there is a high volume of flood alerts.

- Input: Multiple flood alerts from different zones within a short time.
- Expected Result: The system should continue functioning without crashes and send announcements for all zones.
- Black Box Test Type: Stress test.

10. Test Case: Resumption of Announcements After System Outage

Objective: Ensure that announcements pending due to a system outage are dispatched when the system comes back online.

- Input: Flood detected during system downtime.
- Expected Result: Once the system is back online, the pending holiday announcements should be sent out.
- Black Box Test Type: Recovery test.

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