

PES UNIVERSITY

AUGUST – DECEMBER 2022 SEMESTER 5

SOFTWARE ENGINEERING LAB TASKS

Professor-in-charge: Dr. Jayashree R

Teaching Assistant: Aanchal Narendran (Sem VII)

We hope at this point in your semester you have achieved considerable progress in your Software Engineering Project. All of the tasks in this lab manual are geared towards improving and testing your project prior to your submissions.

SRN:

PES2UG20CS352 - Srushti S N

PES2UG20CS385 - Vinti Agrawal

PES2UG20CS389 - Vishwa Mehta

PES2UG20CS391 - Vismaya R

Problem Statement – 1: Unit Testing

A unit is the smallest block of code that functions individually. The first level of testing is Unit testing and this problem statement is geared towards the same.

- Discuss with your teammates and demarcate units in your code base
 - Note: discuss why the code snippet you have chosen can be classified as a unit

The following can be classified into a unit for unit Testing:

- User Login
- Booking
- Payment

Reason: It's a block of code which has to be tested for various invalid inputs.

- Develop test cases for both valid and invalid data

Valid cases:

1. Email id has a "@" symbol, a valid domain name abc@gmail.com
2. Phone number is 10 digits long 3482082315

Invalid cases:

1. Incorrect username format Adf_sdfDfeS2.
 2. Invalid Email id xyz@gmail.com (assuming it doesn't belong to the user or the Email id doesn't exist)
- Ideate how you could further modularize larger blocks of code into compact units with your teammates

Booking based on organization or individuals

Payment options for multiple types of methods like debit card, credit card, UPI etc.

Problem Statement – 2.a: Boundary Value Analysis

When it comes to finding errors in your code base, they are often found at locations where a condition is being tested. Due to this, developers often use Boundary Value tests to reduce defect density.

- How would you define a boundary test?
 - Note: Simple relational conditions are a basic example
- Number of rooms booked:
 - Boundary values: 0, max(available) + 1, 1, max(available)
- Date of booking and event:
 - Boundary values: Current_date, current_date - 1
- Phone number format (only ten digits)
 - Boundary values: 10,11 (Upper limit only)
- No. of payment authentication attempts allowed:
 - Boundary values: 0,1,2,5
- Build your boundary test cases and execute them

Problem Statement – 2.b: Mutation Testing

- Using your isolated units from the first problem statement, ideate with your teammates on how to mutate the code

Decrease the availability whenever a customer books a new room, so the total bill is negative.

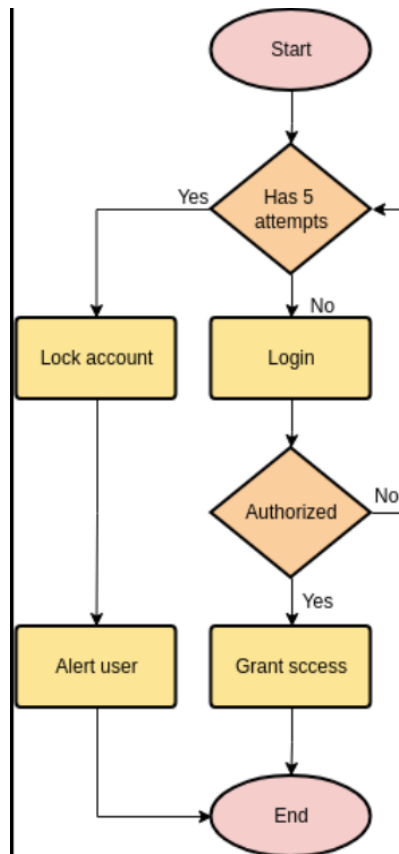
- Develop at least 3 mutants of the functioning code and test all 4 code bases using the test case from the first problem statement

Problem Statement – 3: Static Testing

Static testing involves validating your code without any execution. Under this problem statement, you will be expected to analyse and calculate the cyclomatic complexity of your code.

- Using the unit you selected in the first problem statement as an example, develop the control flow graph of your problem statement.

[Login page:](#)



- Using the Control flow graph, calculate the cyclomatic complexity of your code.

$$N = 8, E = 9, P = 1$$

$$\text{Cyclomatic Complexity} = E - N + 2P = 9 - 8 + 2 * 1 = 3$$

- Using the cyclomatic complexity as an indicator, ideate and code your unit again to reduce complexity

Problem Statement – 4: Acceptance Testing

Assume your neighboring team is the client for your code. Give them an idea of what your product is and the software requirements for the product.

- Exchange your code base and test each others projects to see if it meets user requirements

- If you identify a bug in the project you are testing, inform the opposing team of the bug
- As a team, based in clients experience, ideate modifications to the existing project that could improve client experience

Conference room booking system

Room ID 6.72 assigned on 11/17/2022 for 15 people and time frame 3:00h - 8:00h.

Rooms booked on 11/17/2022:

Room ID: 7.21	Capacity: 5
Room ID: 9.15	Capacity: 7
Room ID: 3.53	Capacity: 12
Room ID: 3.15	Capacity: 12
Room ID: 6.72	Capacity: 15

3:00-8:00

Booking form:

Date: Start time: End time: Capacity:

November 2022						
Su	Mo	Tu	We	Th	Fr	Sa
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

Script created by [Fernando](#).

Problem Statement – 5: Maintenance Activities

Once a product is completed, it is handed off to a service based company to ensure all maintenance activities are performed without the added expenditure of skilled developers. However, a few tasks are performed by the maintenance team to gauge the product better. In this problem statement, you will be asked to experiment with your code.

- Exchange code bases with your neighboring teams and reverse engineer a block of code in order to understand it's functionality.

Booking:

- Date entered should be valid
- Rooms booked should not exceed maximum capacity.
- After understanding the code block, Re-Engineer the code
- Ideate how to refactor the code and the portion of the code base you would have to change

Simplifying methods:

- Optimization of code(avoiding use of multiple nested for loops)
- Avoiding code duplication
- Use guard clauses to avoid nested if statements.
- Discuss how the new changes would impact the time and space complexity of the project during execution

Time complexity decreases and space complexity increases.

After Reverse Engineering and Re-Engineering the code, perform acceptance testing between the teams