

# Group Project – Activity 5

Software Testing + Coding

**Sprint 3**



**Software Testing**

**TDresearchteam**  
Technical Debt Research Team



**VCU**

Computer Science  
College of Engineering

## **Sprint 3: Test Planning**

# Remember

A **test case** is a set of actions performed on a system to determine if it satisfies software requirements and functions correctly. It describes a particular scenario to be tested.

## TC02 - Login Page - Authenticate Successfully on gmail.com

Last updated on: 29th Nov 2021, Last Saved by: Jake Bartlett  
A registered user should be able to successfully login at gmail.com

**PRECONDITION:** the user must already be registered with an email address and password.  
**ASSUMPTION:** a supported browser being used.

### TEST STEPS:

1. Navigate to gmail.com
2. In the 'email' field, enter the email address of the registered user
3. Click the 'Next' button
4. Enter the password of the registered user
5. Click 'Sign in'

### EXPECTED RESULTS:

A page displaying the gmail user's inbox should load, showing any new messages at the top of the page



# Remember

- Equivalence Class Partitioning

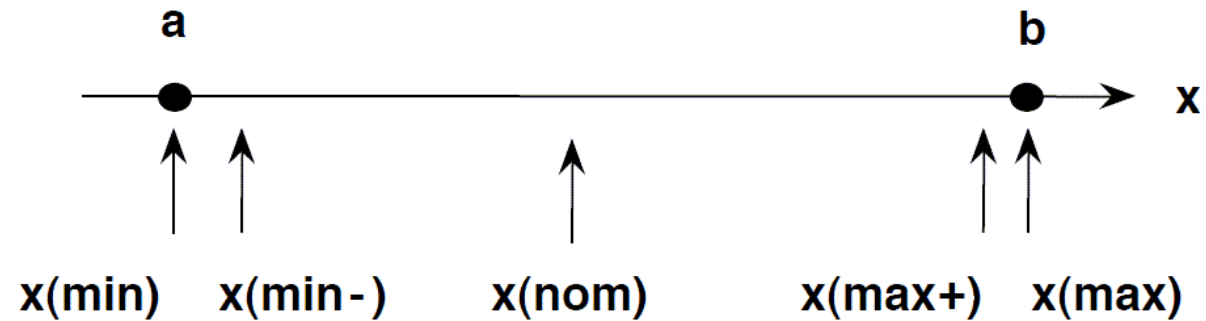
You divide a set of test condition into a partition that can be considered



← Equivalent classes

- Boundary Value Analysis

You test boundaries between equivalence partitions



# Sprint 3: Test Planning

- **For this activity, your group will define test cases for the project based on the use cases specified in the previous stage.** The test cases should cover key aspects of each use case to ensure thorough testing.
- **Instructions:**
  - Define Test Cases:
    - Select **ONE use case** to be implemented in this sprint.
    - For the use case you selected, define **four** (if possible) distinct test cases that focus on different aspects of the use case (e.g., typical behavior vs. edge cases).
    - Each test case should include the following components (consider the provided template):
      - Test Case ID: A unique identifier for the test case.
      - Test Objective: A clear description of what the test case aims to verify.
      - Preconditions: Any setup or conditions that must be in place before running the test.
      - Test Steps: A detailed list of steps to execute the test.
      - Input Values: Data used to test the scenario.
      - Expected Results: What the system should do in response to each step, including any outputs or changes in the system's state.



# Sprint 3: Test Planning

- **Instructions:**

- **Ensure Coverage:**
  - Your test cases should cover both positive scenarios (where everything works as expected) and negative scenarios (where errors, exceptions, or unexpected inputs occur).
- **Consider Boundary Cases:**
  - Test cases should address not only standard usage but also boundary conditions (e.g., handling minimum or maximum values, empty inputs, invalid data).



## **Sprint 3: Coding**

# Sprint 3: Coding

- **This activity aims to apply coding practices, collaborative development, and ensure alignment between the implementation and the specified requirements.**
- **Instructions:**
  - **Select a Use Case:**
    - Each team must review their use cases from the previous stage and select one use case to focus on for this sprint.
    - Ensure that the selected use case is well-defined, with clear functional requirements and expected outcomes.
  - **Break Down the Use Case:**
    - Divide the use case into smaller, manageable tasks (e.g., UI design, back-end logic, data handling, etc.).
    - Assign tasks to team members based on their strengths and the scope of each task.
  - **Implement the Use Case:**
    - Develop the code for the use case based on the team's task distribution.
    - Ensure the implementation follows the design and functionality specified in the use case.
    - Use version control (e.g., Git) for collaborative coding, and ensure all team members contribute regularly with proper commits and code reviews.





## **Sprint 3: Test Case Execution**

# Sprint 3: Test Case Execution

- **Each group will take their planned test cases and execute them against the implemented code for this sprint. The goal is to verify the functionality of the code based on the use case specifications.**
- Each group will also create a 2-minutes video demonstrating the execution of the test cases.
- **Instructions:**
  - Review Test Cases:
    - Revisit the test cases your group has defined for the selected use case.
    - Ensure each test case has clear preconditions, steps, and expected outcomes
  - Execute Test Cases:
    - Run each test case against the implemented code.
    - Record the results, noting whether the code passes or fails the test case based on the expected outcomes.



# Sprint 3: Test Case Execution

- **Instructions:**

- Record a Video:

- Record a 2-minutes video showing the test cases execution, including:
      - Briefly describing the test cases.
      - Executing the test cases in the actual system (e.g., running the code, interacting with the application).
      - Showing the result of the test cases (pass or fail) and comparing it to the expected result.

- Video Requirements:

- Ensure that the video is concise and demonstrates a clear connection between the test case steps and the actual execution.
    - Audio narration or text overlays can be used to explain what's happening in the video.
    - Focus on showing the relevant part of the code or system functionality being tested.



# Deliverables

# Activity Description



Deadline: Dec 3.



**Deliverable:** test cases and recorded 2-minutes video

- Consider the provided Test Case Template.
- Submit test cases on Canvas.
- The 2-minutes video should be posted on your Git repo.



# How will I be evaluated?

- **(2.0 points)** All required test cases have been provided. Each test case includes all components (ID, objective, preconditions, steps, input values, and expected results).
- **(2.0 points)** Test cases are clearly written, with well-defined steps and objectives. The expected results are unambiguous and directly align with the test objective.
- **(2.0 points)** The test cases are appropriate for the corresponding use cases. They accurately reflect the functionality and behavior of the system.
- **(2.0 points)** Test cases adequately cover both normal/positive scenarios and edge/negative cases (e.g., boundary conditions, invalid inputs).
- **(2.0 points)** The video clearly demonstrates the selected test case execution, and the steps are easy to follow within the time limit.



# Group Project – Activity 5

Software Testing + Coding

**Sprint 3**



**Software Testing**

**TDresearchteam**  
Technical Debt Research Team



**VCU**

Computer Science  
College of Engineering