



আন্তর্জাতিক ইসলামী বিশ্ববিদ্যালয় চট্টগ্রাম  
الجامعة الإسلامية العالمية شيتاغونغ  
International Islamic University Chittagong

DEPARTMENT OF COMPUTER AND COMMUNICATION ENGINEERING

## Lab Report: 05

**Course Title:** System Analysis, Design and Development Sessional

**Course Code:** CCE-3508

**Report on:** Testing and Implementation of the System

**Team Name:** Byte Belles

### Submitted By:

ID: E221206  
Bilkis Akter Dalia  
E221209  
Umme Sumaiya  
E221226  
Mansura Akther Eite  
Semester: 5<sup>th</sup>  
Section: 5F

### Submitted To:

Muhammad Kamal Hossen  
Associate Professor  
Dept. of CSE, CUET  
Adjunct Faculty, CCE, IIUC

**Submission Date:** 22/05/2024

Signature

## **Introduction:**

This report outlines the testing and implementation strategy for the proposed online job management system. Effective testing ensures that the system meets all functional and non-functional requirements, while a well-planned implementation strategy facilitates a smooth transition from development to production. The key objectives are to identify and fix defects, validate system functionalities, and ensure the system is ready for deployment.

## **Test Case Design:**

1. Purpose:  
Test cases are designed to validate the system's functionality, performance, and reliability. They provide a structured approach to testing, ensuring all aspects of the system are thoroughly evaluated.
2. Test Case Components:  
Each test case includes:
  - i. Test Case ID: A unique identifier for the test case.
  - ii. Description: A brief overview of the test scenario.
  - iii. Preconditions: Conditions that must be met before executing the test.
  - iv. Test Steps: Detailed steps to execute the test.
  - v. Expected Results: The anticipated outcome of the test.
  - vi. Actual Results: The actual outcome post-execution.
  - vii. Status: Pass or fail based on the comparison of expected and actual results.
3. Example Test Cases
  - a. User Registration-
    - i. Test Case ID: TC001
    - ii. Description: Verify that a new user can register successfully.
    - iii. Preconditions: User must have a valid email address.
    - iv. Test Steps:
      - Navigate to the registration page.
      - Enter valid user details (name, email, password).
      - Submit the form.
    - v. Expected Results: User receives a confirmation email, and account is created.
    - vi. Actual Results: [To be filled post-testing]
    - vii. Status: [To be filled post-testing]
  - b. Job Posting by Employer-
    - i. Test Case ID: TC002
    - ii. Description: Verify that an employer can post a job listing.
    - iii. Preconditions: Employer must be logged in.
    - iv. Test Steps:
      - Navigate to the job posting page.
      - Enter job details (title, description, requirements)
      - Submit the job listing.
    - v. Expected Results: Job listing appears in the job search results.
    - vi. Actual Results: [To be filled post-testing]
    - vii. Status: [To be filled post-testing]

## **White-Box Testing:**

1. Overview:  
White-box testing involves testing internal structures and workings of the system. It requires knowledge of the code and is typically conducted by developers.
2. Techniques:
  - i. Unit Testing: Testing individual components or modules to ensure they function correctly in isolation.
  - ii. Integration Testing: Testing the interaction between integrated units or components to identify interface defects.
  - iii. Code Coverage Analysis: Ensuring all code paths are tested to maximize coverage and detect any unreachable code.
3. Example:
  - i. Unit Testing:
    - Function: `registerUser()`
    - Objective: Ensure the function correctly creates a new user.
    - Test: Input valid user details and verify a user record is created in the database.
  - ii. Integration Testing:
    - Modules: User Registration and Email Notification
    - Objective: Verify that a confirmation email is sent after user registration.
    - Test: Register a new user and check the email inbox for the confirmation email.

## **Black-Box Testing:**

1. Overview:  
Black-box testing focuses on testing the system's functionality without knowledge of internal code. Testers validate the system against specified requirements.
2. Techniques:
  - i. Functional Testing: Verifying that the system behaves as expected for different inputs.
  - ii. Usability Testing: Ensuring the system is user-friendly and intuitive.
  - iii. Performance Testing: Assessing the system's responsiveness, stability, and scalability under load.
3. Example:
  - a. Functional Testing:
    - i. Scenario: User logs in with valid credentials.
    - ii. Steps:
      - Navigate to the login page.
      - Enter valid username and password.
      - Click the login button.
    - iii. Expected Result: User is redirected to the dashboard.
  - b. Usability Testing:
    - i. Scenario: Job seeker searches for jobs.
    - ii. Steps:
      - Log in as a job seeker.
      - Enter a job title in the search bar.
      - Click the search button.

- iii. Expected Result: Relevant job listings are displayed clearly and are easy to navigate.

### **Documenting the System:**

1. Purpose:  
Comprehensive documentation is crucial for the maintenance, support, and future development of the system. It ensures all stakeholders have a clear understanding of the system's functionality, architecture, and usage.
2. Types of Documentation:
  - a. User Manuals: Guides for end-users detailing how to use the system's features.
  - b. Technical Documentation: In-depth documentation for developers, including system architecture, database schemas, and API references.
  - c. Test Documentation: Includes test plans, test cases, test scripts, and test results.
  - d. Release Notes: Summarize new features, enhancements, and bug fixes in each release.
3. Example Contents:
  - a. User Manual:
    - Overview of the system.
    - Step-by-step instructions for key functionalities (e.g., registration, job search, application submission).
    - Troubleshooting tips and FAQs.
  - b. Technical Documentation:
    - System architecture diagram.
    - Detailed descriptions of modules and components.
    - API endpoints and their usage.
    - Database schema with ER diagrams.

### **Conclusion:**

Thorough testing and meticulous documentation are pivotal for the successful implementation of the proposed online job management system. White-box testing ensures the internal logic is sound, while black-box testing validates the system's functionality against user requirements. Comprehensive documentation supports both users and developers, facilitating smooth operation and maintenance. By following this structured approach, we can ensure that the system is robust, user-friendly, and ready for deployment, ultimately providing a valuable service to job seekers and employers.