# Project Report

|  |  |
| --- | --- |
| Project Title | Plan, Schedule, Test Community Portal |
| Qualification Name (NICF) | NICF-Advanced Certificate in Infocomm Technology (Software and Applications) |
| Product Name | NICF-Capstone Project using Java |
| Module Name (NICF) | NICF-Capstone Project using Java |

|  |  |  |  |
| --- | --- | --- | --- |
| Student name | | Assessor name | |
| Muhammad Kemal | |  | |
| Date issued | Completion date | | Submitted on |
|  |  | |  |
|  | |  | |
| Project title | Plan, Schedule, Test Community Portal | | |

|  |
| --- |
| Learner declaration |
| I certify that the work submitted for this assignment is my own and research sources are fully acknowledged.  Student signature: Date: August 2023 |

Contents

[1. Project Report 0](#_Toc142999835)

[1. Project Background 2](#_Toc142999836)

[2. Project Objective 3](#_Toc142999837)

[**2.1 Test Scope:** 3](#_Toc142999838)

[**2.2 Preparation of test scripts:** 3](#_Toc142999839)

[**2.3 Test Execution:** 3](#_Toc142999840)

[3. Project Requirements Specifications 4](#_Toc142999841)

[4. Task 1 6](#_Toc142999842)

[**4.1.1.** **Functional requirements:** 6](#_Toc142999843)

[**4.1.2.** **Non-Functional Requirements:** 9](#_Toc142999844)

[**4.1.3.** **Technical Environment Requirements** 9](#_Toc142999845)

[**4.1.4.** **Testing Requirements** 10](#_Toc142999846)

[5. Task 2 11](#_Toc142999847)

[6. Task 3 12](#_Toc142999848)

[7. Task 4 13](#_Toc142999849)

[**7.4.1. Scheduling** 14](#_Toc142999850)

# Project Background

This project involves developing a community portal for Software Developers, similar to LinkedIn, for ABC Jobs Pte Ltd. This portal aims to connect programmers and software administrators, enabling them to register, find other users, view profiles, send messages, post job opportunities, and more. This project is part of the NICF-Capstone Project using Java, and students must have completed previous modules, including UI Framework (Module 3), Database Design and Implementation (Module 4), and Web Development Foundation (Module 5).

This project will be a Summative Assessment for students in the "NICF-Capstone Project using Java" course. Students must design, plan, and test the community portal they developed in Module 5. In this project, Students must develop features required in Capstone Project and do the testing phase.

Community Portal Main Page:

The community portal consists of several main pages, including:

1. Community Portal Home
2. Registration Page
3. Registration Confirmation Page
4. Update Profile Page
5. Find User Pages
6. List Search Results Page
7. Public Profile Page
8. Registration Confirmation Email
9. Login page
10. Forgot Password page
11. Forgot Password Confirmation Page

The following functions are part of the Database Design & Will be improved as features are required in Capstone Project, Depending on time availability.

1. Send messages
2. Read Messages
3. Post on Message Boards
4. Message Board List
5. Read Threads
6. Post Job Opportunities
7. List Job Opportunities & Feedback

# Project Objective

The project objective of this project is to define the test scope and establish a plan for the tests, preparation of test scripts, and execution of the tests. As such, we aim to ensure that the community portal for Software Developers functions effectively, meets specified requirements, and provides a seamless user experience.

## **2.1 Test Scope:**

**In Scope:**

**Post and Apply for Job:**

Test coverage includes functionality related to posting job opportunities and applying for those jobs within the community portal.

**Thread:**

Testing will include functionality to create threads where users can start discussions and post replies to those threads within the community portal.

**Out of scope:**

**Registration Functionality:** Tested the user registration process, including displaying a thank you page and sending a confirmation email.

**User Profile Search:** Functionality validation allows users to search for other users based on parameters such as First Name, Last Name, Company Name, City, and Country.

**View User Profile:** Ensure users can view other users' public profiles after performing a search.

**Login and Forgot Password:** Verify the login process and ability to retrieve forgotten passwords.

**User Profile Update:** Test functionality that allows users to update their profile information.

## **2.2 Preparation of test scripts:**

Test scripts will be prepared to ensure comprehensive test coverage for all functions within the specified test scope. Each test script will include step-by-step instructions, test data, and expected results to guide testers through the execution phase.

## **2.3 Test Execution:**

The test execution phase will involve executing the test scripts prepared on the community portal. The tester will follow the specified steps and enter test data to assess the behavior of the portal and identify any issues or defects. The results will be documented, and any deficiencies identified will be reported for further resolution.

By defining the test scope, preparing test scripts, and running tests, we aim to validate the functionality and reliability of the community portal for Software Developers, ensuring its readiness for deployment and subsequent enhancements during the Capstone Project.

# Project Requirements Specifications

High-Level Community Portal Functional Features:

For Software Programmers:

**Login:** Software Programmers can log in to their accounts using their credentials.

**Register**: New Software Programmers can create an account and register on the community portal.

**Logout:** Software Programmers can log out of their accounts to end their session.

**Update Profile:**Software Programmers can edit and update their profile information.

**Search User**: The Software programmer can search for other users by parameters like First Name, Last Name, Company Name, and City.

**View Public Profiles:** Programmers can view the public profiles of other users after searching for them.

**Send Messages:** Programmers can send messages to other users on the portal.

**Create Threads:** Programmers can create threads to start discussions.

**Post Replies:** Programmers can post replies to existing threads.

**Post Job Opportunities:** Programmers can post job opportunities on the portal.

**List of Job Opportunities & Feedback:** Programmers can view the job opportunities and responses received.

For Administrator:

**Login:** Administrators can access their accounts with their login information.

**Logout:**To end their session, administrators can log out of their accounts.

**Manage User Data:** Admins can manage user data, such as user information and access rights.

**Send Bulk Emails:** Admins can send bulk emails to invite programmers to register on the community portal.

**Test method:**

**Functional Testing:**

**User Acceptance Testing (UAT):**

* UAT involves testing the community portal from an end-user perspective.
* UAT ensures that the portal meets the requirements and expectations of the users.
* Real users participate in UAT to validate portal usability, functionality, and overall user experience.

**Unit Tests:**

* Unit testing focuses on testing individual components or portal units separately.
* It verifies that each unit functions properly according to its design and requirements.
* Unit tests improve code quality and reliability by detecting and fixing defects early.

**Non-Functional Testing:**

**Compatibility - Cross-Browser Testing:**

* This test ensures the community portal works smoothly on various web browsers (e.g., Chrome, Firefox, Edge.).
* Compatibility ensures the user interface and functionality are consistent and work as expected across browsers.

**Performance Efficiency - Load Testing:**

* Load testing evaluates the portal's performance under expected user loads.
* It identifies the maximum number of concurrent users that the portal can handle without performance degradation.
* Load testing helps optimize portal performance, ensuring responsiveness and scalability.

**Portability (Different Devices):**

* This test evaluates the functionality and usability of the portal across multiple devices (e.g., desktops, laptops, tablets, smartphones).

We can thoroughly evaluate community portal features by categorizing testing techniques into functional and non-functional categories. Functional testing validates the core functionality of the portal, ensuring it meets user requirements. In contrast, non-functional testing addresses critical aspects such as performance, compatibility, and portability, contributing to a robust and user-friendly portal.

Using these selected test methods, we can systematically assess the functional features of the community portal for Software Programmers and Administrators. Thorough testing will help identify and fix any issues, ensuring that the portal meets high-quality standards and provides a smooth user experience.

# Task 1

Task Statement: Write a brief requirement specification for the Community Portal.

## **Functional requirements:**

System requirements

a. List of all modules to be developed：

**Public Module - Home:**

**User Registration:** The Community Portal must be able to register users with data such as First Name, Last Name, Email, Password, and City.

**Login:** Registered users must be able to log in using their email and password.

**Forgotten Password:** Users should have the option to request forgotten passwords if they forget them. The system must verify their email and then be given access to change the password.

**Update Profile:**Users can Update Profiles when successfully logged into the system.

**Public User Profile:** The Portal will provide a public profile for each user, which other registered users can view. The profile will contain important information such as Full Name, work experience, education, and City.

**Other User Search:** The user should be able to look up other users using criteria like First Name, Last Name, and City.

**Private Module - Administrator/User:**

**Administrator Role:** The portal will have an Administrator role with special rights to manage user data, send bulk emails, and post jobs.

**User Management:** Administrators can view, edit, or delete user profiles.

**Bulk Email:** Administrators must be able to send bulk emails to software programmers.

**Thread:** Users will see the thread on their homepage, and they can apply for jobs in the thread

**Job Posting:** Users must be able to post job opportunities within the portal.

b. All pages in each module with a brief description and their purpose.

|  |  |
| --- | --- |
| Web page | Description |
| Registration Page | To register as a member of the community portal, one must fill in personal information. |
| Registration Confirm | Users can get a confirmation message after registration. |
| Login | Users can log in with data already in the database. |
| Forget Password | Users can forget passwords to change passwords so users can re-enter the system. |
| Update Profile | Users can Update Profiles when successfully logged into the system. Users can add Education and Experience. |
| Search | Users can search for other users. |
| Search Result | Users can see the search result. |
| Others Profile | Users can view other users’ profiles. |
| Homepage | Users can Read and react to a thread. |

c. All pages in each module which are created, edited and viewed pages

|  |  |
| --- | --- |
| Landing Page | View Page |
| Sign In Page | Create Page |
| Sign Up Page | Create Page |
| Forgot Password Page | Edit Page |
| Homepage | View Page |
| Update Profile Page | Create and Edit Page |
| Search Others Profile Page | Create Page |
| Others Profile Page | View Page |

d. Sitemap of Public & Private Sites

Public

Private

## **Non-Functional Requirements:**

|  |  |  |
| --- | --- | --- |
| No | Requirements | Purpose |
| 1 | Performance | Ensure the Community Portal operates efficiently and responds quickly to user actions. |
| 2 | Availability | Ensure the Community Portal is accessible to users with minimal downtime. |
| 3 | Security aspects | Protect user data, prevent unauthorized access, and ensure data privacy. |
| 5 | Database backup | Regularly backup the database to prevent data loss and enable recovery in case of failures. |

## **Technical Environment Requirements**

Hardware

* Memory: 4 GB or more.
* Graphics Card: Intel HD / Radeon Graphics or more.
* CPU: Dual Core Cpu or more.
* HDD: 60 GB or more.
* OS: Windows 10 / 11
  + - * Database server – MySQL server
      * Firewall settings - Port 9091 -- Http

- Port 3306 – Database

Software

* Eclipse IDE: Download and install Eclipse IDE for Java EE Developers. Ensure a version that supports Java EE development, including the necessary tools and plugins for Spring MVC development. (Eclipse Version: 4.26.0)
* Spring Framework: Download and include the Spring framework libraries in your project. (Spring framework 5.2.10)
* Apache Tomcat: Install a servlet container like Apache Tomcat on your machine. We need it to deploy and run the Spring MVC application. Ensure that the servlet container is compatible with the version of the Spring framework you're using. (Apache Tomcat 9.0)
* Maven or Gradle: A build automation tool like Maven or Gradle can simplify the project setup and manage dependencies effectively.
* Java Development Kit (JDK): Install the latest version of JDK on your system. Eclipse requires a compatible JDK to run. (JDK 18.0.2)

Database

* MySQL WorkBench 8.0. (MySQL Server 8.0.31)

## **Testing Requirements**

**Functional Testing Requirements:**

**User registration:**

* Verify successful registration with all required fields.
* Confirm receipt of the confirmation email after registration.

**Login and Authentication:**

* Test successful login with valid credentials.
* System verification rejected invalid login attempts.

**Forgotten Password:**

* Test the password reset function via the email link provided.

**User Search:**

* User verification can search for others using different parameters.
* Make sure search results are accurate and relevant.

**Public User Profile:**

* Test that the public user profile displays the correct information.
* Make sure the profile is accessible to other registered users.

**User Management (Admin):**

* Test Administrator's ability to manage user profiles.
* Ensure that the system can edit user information and delete user accounts.

**Bulk Email (Admin):**

* Confirm that Administrators can send bulk emails to Users.

**Post Job(Admin):**

* Ensure the admin can post the jobs.

**Non-Functional Testing Requirements:**

**Compatibility Testing / Cross Browser:**

* Test the portal on different web browsers (Chrome, Firefox, Edge) to ensure compatibility.
* Verify that the portal is working correctly in each supported browser.

**Performance Efficiency - Load Testing:**

* Assess portal performance under various load conditions.
* Determine response time and system stability under heavy user traffic.

**Portability Testing (Different Devices):**

* Verify portal responsiveness across devices, including desktops, tablets, and mobile phones.
* Make sure the layout and functionality of the portal adapt to different screen sizes.

# Task 2

Task Statement: Briefly explain different types of testing, and how they will help to create a better system

Functional Testing:

Unit Tests:

* Unit testing focuses on testing individual components or portal units separately.
* It verifies that each unit functions properly according to its design and requirements.
* By detecting and fixing defects early, unit tests contribute to overall code quality and reliability, resulting in a more robust and stable system.

User Acceptance Testing (UAT):

* UAT involves testing the community portal from an end-user perspective.
* Real users participate in UAT to validate portal usability, functionality and overall user experience.
* By incorporating feedback from real users, UAT helps identify user-centric issues and ensures that the system meets user expectations, resulting in a better user experience.

System Testing:

* System testing evaluates the complete community portal as an integrated system.
* It tests the interactions between different modules and components to ensure they work together seamlessly.
* System Testing validates that the portal meets specified requirements and properly performs its intended function by assessing the system.

Non-Functional Testing:

Cross-Browser Testing:

* Cross-Browser Testing verifies that the community portal functions properly in many web browsers (Chrome, Firefox, and Edge).
* This guarantees that the user interface and functionality are uniform and function properly across browsers.
* The portal's accessibility improves, reaches a bigger audience, and gives a better user experience through testing on multiple browsers.

Load Testing:

* Load Testing evaluates the portal’s performance under expected user loads.
* It identifies the maximum number of concurrent users that the portal can handle without performance degradation.
* Load Testing ensures the portal remains responsive and scalable by optimizing portal performance, providing a smooth and reliable user experience even during peak usage.

Usability Testing:

* Usability Testing assesses the community portal's friendliness and ease of use.
* This involves real users performing certain tasks on the portal to identify usability issues or points of confusion.
* By improving the user interface and navigation, Usability Testing improves overall user experience and satisfaction.

# Task 3

Task Statement: Propose relevant tests suitable for the Project Scenario

Functional testing:

1. Unit testing
2. User Acceptance testing

Non-Functional testing:

1. Cross-browser testing
2. Load Testing
3. Usability testing

|  |  |  |
| --- | --- | --- |
| **Test Scenario ID** | **Test Scenario** | **Number of test cases** |
| TS001-Unit Testing | Test each component in the application | 4 |
| TS002- UAT testing | Ensure the portal’s features align with course requirements and meet users’ expectations. | 2 |
| TS003-Cross Browser testing | Ensure the application works across multiple major browsers | 4 |
| TS004- Load Testing | Testing the performance of the application | 2 |
| TS005 – Usability Testing | Assess the ease of use for tasks | 2 |

# Task 4

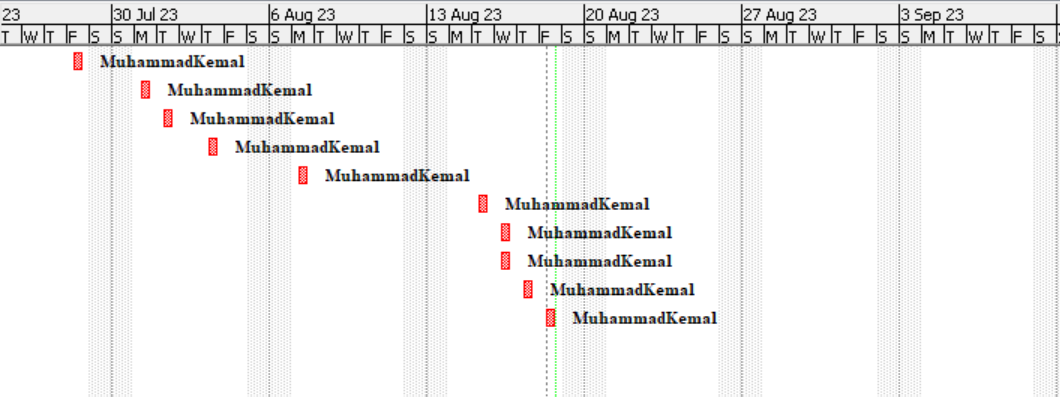
Task Statement:

Write briefly on optimal scheduling times for different tests 2. Prepare a schedule for the execution of tests. Include it as part of the Project Presentation.

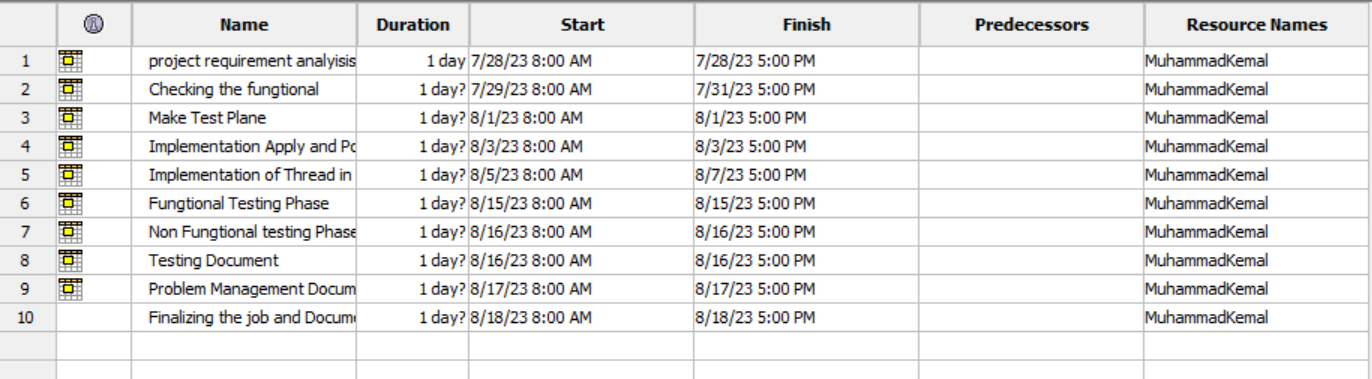
## **7.4.1. Scheduling**

|  |  |  |  |
| --- | --- | --- | --- |
| Web page | Weighting | Functional Points (FP) | Total |
| Apply for Job | 3 | 5 | 15 |
| Post Job | 3 | 5 | 15 |
| Thread | 3 | 5 | 15 |
|  |  | Total functional points | 45 |
|  |  | Estimated person hour per FP | 1.0 |
|  |  | Total hours needed | 45 |
|  |  | Total days needed | 8.5 days |

**Gantt Chart**

****

**Gantt Table**

****

**Test Schedule IU6 (Slide 15 and refer to class notes)**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Schedule ID: | TS0001 |  |  |
| Product ID / Name: | ABC Community portal website |  |  |
| Product Version or Build: | 1.0 |  |  |
| Present Owner: | Muhammad kemal | Software Engineer |  |
| Created On: | 28 July 2023 |  |  |
| Review On: | 18 August 2023 |  |  |
| Review By: | Muhammad Kemal | PM |  |
| Review Comments: | Good |  |  |
| Current Version: | 1. (Make apply/post jobs, adding thread feature) |  |  |
| Change Details: | If your version is 1.0, put “NIL”  If your version is not 1.0, insert the change you made in your Power BI report <Nil> |  |  |
| Current Status: | Approved |  |  |
| Signing Off Authority: | Arvinder Kaur  Muhammad Kemal | QA manager  PM | 18th August 2023 |

# Task 5: Phase Test Plan

Task Statement: Develop a Phase Test Plan for testing the application.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Activity | Start | End | Person Days | Resource |
| Identifying and Brainstorming requirements | 28-7-2023 | 28-7-2023 | 1 | Muhammad Kemal |
| Checking the functional of the previous project | 31-7-2023 | 31-7-2023 | 1 | Muhammad Kemal |
| Make a Test Plan | 01-8-2023 | 01-8-2023 | 1 | Muhammad Kemal |
| Implementation of Apply and Post Jobs | 03-8-2023 | 07-8-2023 | 1 | Muhammad Kemal |
| Implementation of Thread in Software Programmers | 09-8-2023 | 11-8-2023 | 1 | Muhammad Kemal |
| Functional Testing Phase | 14-8-2023 | 14-8-2023 | 1 | Muhammad Kemal |
| Non Functional Testing Phase | 15-8-2023 | 15-8-2023 | 1 | Muhammad Kemal |
| Testing Documentation | 15-8-2023 | 15-8-2023 | 1 | Muhammad Kemal |
| Problem Management Documentation | 16-8-2023 | 16-8-2023 | 1 | Muhammad Kemal |
| Finalizing the Jobs Portal and Documentation | 18-8-2023 | 18-8-2023 | 1 | Muhammad Kemal |

Goal

This test aims to identify and measure the problems/issues in the web application.

Objectives

Validate and verify that system functionality works according to requirements.

**Features not to be tested**

1. Login
2. Logout
3. Registration
4. Update Profile
5. Search for Others
6. Views Others

**Features Be Tested**

1. Post Job
2. Apply for a Job
3. Post Thread
4. Comment on Thread

**Approach**

1. Unit Testing
2. UAT Testing
3. Cross Browser Testing
4. Load Testing
5. Usability Testing

**Suspension Criteria:**

Component tests for thread and task functionality will be paused under the following conditions:

* Discovery of critical errors that prevented the successful completion of the test, such as a severe system crash or data corruption.
* Implemented significant changes to thread and job features that may affect testing.
* Other unforeseen conditions make further testing unfeasible.

**Resumption Criteria:**

Component testing for thread and task functionality will continue once the following criteria are met:

* All issues identified in the suspension criteria have been resolved or appropriately mitigated.
* New software builds containing fixes or updates related to critical errors have been implemented.
* New software build that includes necessary upgrades, addressing identified critical issues, available for testing.

**Exit Criteria (Complete Test):**

Component testing can be considered complete if the following conditions are met:

Scope of Requirements:

* All requirements categorized as High and Medium priority have undergone testing, which includes validation and functional verification.
* The scope of testing includes a thorough evaluation of the functionality, user interaction, and data handling specified by the requirements.

Defect Resolution and Verification:

* Critical and Moderate severity defects identified during testing have been treated with care.
* The resolution of each defect is strictly verified, ensuring that the repair is effective and does not cause unwanted side effects.

**Test Metrics**

|  |  |
| --- | --- |
| **Reported By:** | Muhammad Kemal |
| **Report Date:** | 17/8/2023 |
|  |  |
| **Tested By:** | Muhammad Kemal |
| **Tested Date:** | 17/8/2023 |
| **Testing Name/ Test Scenario ID** | Unit Testing/ TS001 / UT001 / UT002 / UT003 |
| **Total Number of Test Cases** | 4 |
| **No. of Pass:** | 4 |
| **No. of Failures:** | 0 |
| **Elapsed Time:** | 00:07:00 |
|  |  |
| **Tested By:** | Muhammad Kemal |
| **Tested Date:** | 17/8/2023 |
| **Testing Name/ Test Scenario ID** | UAT Testing / TS002 / UAT001 / UAT002 |
| **Total Number of Test Cases** | 2 |
| **No. of Pass:** | 2 |
| **No. of Failures:** | 0 |
| **Elapsed Time:** | 00:07:00 |
|  |  |
| **Tested By:** | Muhammad Kemal |
| **Tested Date:** | 17/8/2023 |
| **Testing Name/ Test Scenario ID** | Cross Browser Testing/ TS003 / CB001 / CB002 / CB003 / CB004 |
| **Total Number of Test Cases** | 4 |
| **No. of Pass:** | 4 |
| **No. of Failures:** | 0 |
| **Elapsed Time:** | 00:06:00 |
|  |  |
| **Tested By:** | Muhammad Kemal |
| **Tested Date:** | 17/8/2023 |
| **Testing Name/ Test Scenario ID** | Load Testing/ TS004 / LT001 / LT002 |
| **Total Number of Test Cases** | 2 |
| **No. of Pass:** | 2 |
| **No. of Failures:** | 0 |
| **Elapsed Time:** | 00:10:00 |
|  |  |
| **Tested By:** | Muhammad Kemal |
| **Tested Date:** | 17/8/2023 |
| **Testing Name/ Test Scenario ID** | Usability Testing / TS005 / UST001 / UST002 |
| **Total Number of Test Cases** | 2 |
| **No. of Pass:** | 2 |
| **No. of Failures:** | 0 |
| **Elapsed Time:** | 00:04:00 |
|  |  |
| **SUMMARY LOG STATUS** | |
| **Total Number of Test Cases** | 14 |
| **Total Number of Pass** | 14 |
| **Total Number of Failures** | 0 |
| **Pass %** | 100% |
| **Total Elapsed Time** | 00.34.00 |

**Test Deliverables**

1. Phase Test plan
2. Test schedules
3. Test scripts
4. Test Results
5. Bugs write-up

# Task 6

Task Statement: Execute the tests and provide the reports as part of the Project Presentation.

**Scenario 1: Unit Testing**

|  |  |
| --- | --- |
| **Test Scenario** | Test each component in the application |
| **TS001** |
| **Test Cases** | Unit Testing |
| **UT001** | Test creating and posting a thread, ensuring data is stored correctly. |
| **UT002** | Test Comments on threads |
| **UT003** | Test posting a job opportunity, validating input fields and data storage. |
| **UT004** | Testing Apply for Job |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Scenario ID | Test Case | Risk | Preconditions | Expected Result | Test Data | Actual Result | Final Result |
| TS001 | UT001 | 5 | All of the code is already developed and ready to test. | Thread saved to the database | Image: Band.png  Body: “example body” | As Expected | Pass |
| UT002 | 5 |  |  | As Expected | Pass |
| UT003 | 5 |  |  | As Expected | Pass |
|  | UT004 | 5 |  |  |  | As Expected | Pass |

|  |  |
| --- | --- |
| **Test Case ID** | **Evidence** |
| **UT001** | **Before**    **After** |
| **UT002** |  |
| **UT003** |  |
| **UT004** |  |